Automatización de la etapa de síntesis lógica, optimización del proceso de instalación de las aplicaciones de Synopsys y documentación de los pasos de *front-end* y *back-end* para el diseño del circuito integrado El Gran Jaguar

Kevin David Hernández Guarcas

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Automatización de la etapa de síntesis lógica, optimización del proceso de instalación de las aplicaciones de Synopsys y documentación de los pasos de *front-end* y *back-end* para el diseño del circuito integrado El Gran Jaguar

Trabajo de graduación presentado por Kevin David Hernández Guarcas
para optar al grado académico de Licenciado en Ingeniería Electrónica

Guatemala,

2025

Vo.Bo.:

(f) _____

MSc. Jonathan de los Santos

Tribunal Examinador:

(f) _____

MSc. Jonathan de los Santos

(f) _____

MSc. Carlos Esquit

(f) _____

Ing. José Alejandro Ruiz Orozco

Fecha de aprobación: Guatemala, 23 de Junio de 2025.

Prefacio

Estoy muy feliz de haber podido aportar y contribuir al desarrollo del primer circuito integrado nanométrico hecho en Guatemala. Este ha sido un proyecto multigeneracional, realizado por catedráticos y estudiantes de la Universidad del Valle de Guatemala, del cual tuve la oportunidad de formar parte. El desarrollo de este primer circuito integrado en Guatemala representa un gran avance tecnológico no solo para el país, sino también a nivel regional en Centroamérica, abriendo puertas para muchos guatemaltecos apasionados por la nanoelectrónica.

Llegar a este punto de mi carrera universitaria es algo con lo que siempre soñé. Quiero agradecer en primer lugar a Dios, porque sin su ayuda este logro no hubiera sido posible. Han sido años llenos de obstáculos y luchas, pero Él siempre estuvo a mi lado.

En segundo lugar, quiero agradecer a mis padres, quienes me apoyaron en todo momento de diversas formas. Siempre creyeron en mí y me animaron en los momentos más difíciles. También quiero agradecer a mis hermanos. A mi hermano Pablo, por su apoyo durante la pandemia, y especialmente a mi hermano Samuel, quien siempre estuvo ahí para ayudarme en todo lo que necesitara durante mis estudios universitarios. Ha sido un gran ejemplo a seguir, tanto personal como profesionalmente. Sin su apoyo, no estaría cumpliendo este sueño.

Agradezco también a mis amigos de la carrera, con quienes tuve la oportunidad de compartir gran parte de mi vida universitaria. Sin ellos, la experiencia no habría sido la misma. Juntos vivimos momentos muy alegres, y también superamos momentos de estrés, pero con su compañía y apoyo, fue posible sobrellevar esas dificultades.

Asimismo, quiero agradecer a mis catedráticos. En especial, agradezco a Pablo Mazariegos, Luis Rivera, Carlos Esquit y Otto Girón, quienes impartieron los cursos que más disfruté durante mi carrera: programación de microcontroladores, simulación de circuitos y fabricación de PCB, electrónica digital, procesamiento de señales, sistemas de control, arquitectura de computadoras, nanoelectrónica e instrumentación y automatización industrial. Gracias a ellos, he desarrollado conocimientos y habilidades que sé que me servirán en mi vida profesional, especialmente para especializarme en áreas como la automatización y la electrónica automotriz.

Un agradecimiento especial a MSc. Carlos Esquit, quien siempre demostró su entusiasmo por enseñarnos de la mejor manera. Ha sido uno de los principales responsables de introducirnos al campo de la nanoelectrónica y, sin él, no habría sido posible llevar a cabo el desarrollo del primer circuito integrado de la región. Finalmente, agradezco a mi asesor, MSc. Jonathan de los Santos, por su constante apoyo y orientación a mis compañeros y a mí para llevar a cabo este proyecto.

Les deseo a mis amigos Rodrigo Cruz, Lourdes Ruiz, Katherine Rac y Diana Alvarado los mejores éxitos en su vida personal y profesional.

Índice

Prefacio iv

Lista de figuras xiv

Lista de cuadros xix

Resumen xxii

Abstract xxiv

1. Introducción 1

2. Antecedentes 3
- 2.1. Línea de investigación . . . 3
- 2.2. Inicios del diseño del circuito integrado El Gran Jaguar . . . 4
- 2.3. Avances del circuito integrado El Gran Jaguar . . . 5
- 2.4. Avances 2020 . . . 5
- 2.5. Avances 2021 . . . 6
- 2.6. Avances 2022 . . . 6
- 2.7. Avances 2023 . . . 9

3. Justificación 11

4. Objetivos 13
- 4.1. Objetivo general . . . 13
- 4.2. Objetivos específicos . . . 13

5. Alcance 14

6. Marco teórico 15
- 6.1. *Design flow* general para ASIC . . . 15
- 6.2. Parte del flujo general *front-end* . . . 16
- 6.3. Parte del flujo general *back-end* . . . 17
- 6.4. Flujo *front-end* propuesto por Synopsys . . . 18

6.5. Aplicaciones de Synopsys . . . 18
6.5.1. Verilog Compiler Simulator (VCS) . . . 18
6.5.2. Discovery Visual Environment (DVE) . . . 20
6.5.3. Design Compiler . . . 22
6.5.4. Design Vision . . . 22
6.6. Protocolo Secure Shell (SSH) . . . 23
6.6.1. Cifrado SSH con claves públicas y privadas . . . 24
6.7. SSH File Transfer Protocol (SFTP) . . . 25
6.8. Bourne Again Shell (Bash) . . . 25
6.9. *Script* Bash . . . 25

7. Levantamiento de servidor compartido NFS **27**
7.1. Pasos para levantar servidor NFS . . . 27
7.2. Pasos para montar servidor compartido . . . 31

8. Identificación de librerías y archivos de TSMC de 180 nm **33**
8.1. Librerías de 180 nm . . . 33
8.2. Archivos Verilog . . . 34
8.3. Archivos PDF . . . 35

9. Jerarquía de directorios **36**
9.1. Jerarquía de directorios general . . . 37
9.2. Jerarquía de directorios para la síntesis lógica . . . 38

10. Flujo de diseño a seguir para la síntesis lógica **42**

11. Diseño de circuitos en HDL **45**
11.1. Diseño del módulo de un circuito en Verilog . . . 46
11.2. Diseño del *testbench* para simulación de un circuito . . . 46
11.3. Diseño del módulo de pines de entrada y salida de un circuito en Verilog . . . 47
11.3.1. Selección y configuración del pin de entrada y salida . . . 48
11.3.2. Declaración de pines de entrada y salida . . . 52

12. Configuración y utilización de herramientas para simulación y sintetización de un circuito **55**
12.1. Uso de VCS y DVE para simulación . . . 56
12.1.1. Uso de VCS . . . 56
12.1.2. Uso de DVE . . . 58
12.2. Uso de Design Vision con la interfaz gráfica . . . 61
12.2.1. Importación de librerías . . . 62
12.2.2. Importación de archivos . . . 64
12.2.3. Restricciones de diseño . . . 69
12.2.4. Verificación de diseño . . . 72
12.2.5. Compilación de diseño . . . 73
12.2.6. Generación de reportes . . . 75
12.2.7. Generación de archivos de salida . . . 76
12.3. Uso de Design Vision con comandos . . . 77
12.3.1. Análisis de los *scripts* y explicación de comandos utilizados en años anteriores . . . 78

12.3.2. Uso de los *scripts* utilizados en años anteriores . . . 83
12.3.3. Análisis de los nuevos *scripts* realizados . . . 86
12.3.4. Uso de los nuevos *scripts* . . . 91

13.Sintetización lógica de distintos circuitos descritos en Verilog utilizando el flujo de diseño A **95**
13.1. Proceso para la síntesis lógica de una compuerta *not* . . . 95
13.1.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench* . . . 95
13.1.2. Simulación del diseño original del circuito . . . 97
13.1.3. Síntesis lógica del diseño original del circuito . . . 98
13.1.4. Simulación del circuito sintetizado . . . 100
13.1.5. Unión del circuito sintetizado con el módulo de pines de I/O . . . 101
13.1.6. Síntesis lógica del circuito sintetizado con pines de I/O . . . 102
13.1.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O . . . 103
13.2. Proceso para la síntesis lógica de una ALU de 4 bits . . . 104
13.2.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench* . . . 104
13.2.2. Simulación del diseño original del circuito . . . 108
13.2.3. Síntesis lógica del diseño original del circuito . . . 108
13.2.4. Simulación del circuito sintetizado . . . 112
13.2.5. Unión del circuito sintetizado con el módulo de pines de I/O . . . 113
13.2.6. Síntesis lógica del circuito sintetizado con pines de I/O . . . 117
13.2.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O . . . 120
13.3. Proceso para la síntesis lógica de un *carry look-ahead adder* de 4 bits . . . 121
13.3.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench* . . . 121
13.3.2. Simulación del diseño original del circuito . . . 125
13.3.3. Síntesis lógica del diseño original del circuito . . . 125
13.3.4. Simulación del circuito sintetizado . . . 128
13.3.5. Unión del circuito sintetizado con el módulo de pines de I/O . . . 129
13.3.6. Síntesis lógica del circuito sintetizado con pines de I/O . . . 131
13.3.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O . . . 133
13.4. Proceso para la síntesis lógica de una *carry look-ahead adder* de 16 bits . . . 134
13.4.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench* . . . 134
13.4.2. Simulación del diseño original del circuito . . . 139
13.4.3. Síntesis lógica del diseño original del circuito . . . 139
13.4.4. Simulación del circuito sintetizado . . . 144
13.4.5. Unión del circuito sintetizado con el módulo de pines de I/O . . . 145
13.4.6. Síntesis lógica del circuito sintetizado con pines de I/O . . . 151
13.4.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O . . . 155
13.5. Proceso para la síntesis lógica de una *carry look-ahead adder* de 32 bits . . . 156
13.5.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench* . . . 156
13.5.2. Simulación del diseño original del circuito . . . 163
13.5.3. Síntesis lógica del diseño original del circuito . . . 163
13.5.4. Simulación del circuito sintetizado . . . 170

13.5.5. Unión del circuito sintetizado con el módulo de pines de I/O 171
13.5.6. Síntesis lógica del circuito sintetizado con pines de I/O 180
13.5.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O . . 188
13.6. Proceso para la síntesis lógica de una *carry look-ahead subtractor* de 16 bits . 189
13.6.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench* 189
13.6.2. Simulación del diseño original del circuito 194
13.6.3. Síntesis lógica del diseño original del circuito 195
13.6.4. Simulación del circuito sintetizado 201
13.6.5. Unión del circuito sintetizado con el módulo de pines de I/O 201
13.6.6. Síntesis lógica del circuito sintetizado con pines de I/O 208
13.6.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O . . 214
13.7. Proceso para la síntesis lógica de un multiplicador de 32 bits usando un *carry look-ahead adder* de 16 bits 215
13.7.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench* 215
13.7.2. Simulación del diseño original del circuito 222
13.7.3. Síntesis lógica del diseño original del circuito 223
13.7.4. Simulación del circuito sintetizado 241
13.7.5. Unión del circuito sintetizado con el módulo de pines de I/O 242
13.7.6. Síntesis lógica del circuito sintetizado con pines de I/O 243
13.7.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O . . 255
13.8. Proceso para la síntesis lógica de un divisor de 32 bit usando un *carry look-ahead subtractor* de 16 bits 256
13.8.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench* 256
13.8.2. Simulación del diseño original del circuito 267
13.8.3. Síntesis lógica del diseño original del circuito 268
13.8.4. Simulación del circuito sintetizado 288
13.8.5. Unión del circuito sintetizado con el módulo de pines de I/O 289
13.8.6. Síntesis lógica del circuito sintetizado con pines de I/O 289
13.8.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O . . 309
13.9. Proceso para la síntesis lógica de el nanochip El Gran Jaguar 310
13.9.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench* 310
13.9.2. Simulación del diseño original del circuito 313
13.9.3. Síntesis lógica del diseño original del circuito 313
13.9.4. Simulación del circuito sintetizado 315
13.9.5. Unión del circuito sintetizado con el módulo de pines de I/O 316
13.9.6. Síntesis lógica del circuito sintetizado con pines de I/O 317
13.9.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O . . 317

14.Mejoras realizadas en el *script* `ejecutar_sintesis_logica_2.bash` 319

15.*Script* realizado para mejora del proceso de las aplicaciones de Synopsys 325
15.1. Pasos previos a realizar antes de ejecutar el *script* 334
15.1.1. Verificación de *scripts* con permisos de ejecución 334
15.1.2. Creación de un usuario con permisos de *root* 334

15.2. Explicación general del *script* 335
15.2.1. Función `Dependencias_Iniciales` 336
15.2.2. Función `Synopsys_Installer` 336

16. Conclusiones **337**

17. Recomendaciones **339**

18. Bibliografía **341**

19. Anexos **344**
19.1. Circuito del nanochip "El gran Jaguar" 344
19.1.1. Circuito original 344
19.1.2. Circuito sintetizado junto con los pines de entradas y salidas 667

Lista de figuras

1. Flujo de diseño para síntesis lógica (*front-end*) y síntesis física (*back-end*) presentado por el Ing. Steven Rubio [2] . . . 4
2. Síntesis lógica del circuito a fabricar. (a) Esquemático del circuito sin el uso de restricciones. (b) Esquemático del circuito utilizando restricciones presentado por el Ing. Luis Nájera [3] . . . 5
3. Síntesis física obtenida a través del *software* IC Compiler II . . . 6
4. 1er escenario con 2 errores de densidad . . . 7
5. 1er escenario de síntesis física obtenida a través del *software* IC Compiler II con errores de densidad . . . 7
6. 2do escenario de síntesis física obtenida a través del *software* IC Compiler II sin errores de densidad . . . 8
7. Interfaz gráfica para generar los archivos necesarios para realizar las verificaciones físicas . . . 8
8. Síntesis física obtenida a través del *software* IC Compiler II sin errores de densidad [24] . . . 9
9. Errores en las verificaciones de *layout vs schematic* (LVS) [24] . . . 9
10. Verificaciones de *design rule check* (DRC) [24] . . . 10

11. Flujo de diseño general para ASIC . . . 16
12. Flujo *front-end* propuesto por Synopsys . . . 18
13. Interfaz gráfica DVE 1 . . . 20
14. Interfaz gráfica DVE 2 . . . 21
15. Interfaz gráfica DVE 3 . . . 21
16. Interfaz gráfica DVE 4 . . . 21
17. Visualización del *waveform* . . . 22
18. *Logical core* del circuito integrado El Gran Jaguar . . . 23
19. Interfaz gráfica de Design Vision . . . 23
20. Funcionamiento de las claves SSH . . . 24

21. Agregación del directorio compartido al archivo /etc/exports . . . 29
22. Verificación del estado del servicio NFS . . . 30
23. Verificación de permisos de firewall . . . 30
24. Configuración para montar la carpeta automáticamente . . . 32

25. Carpeta montada en las máquinas clientes . . . 32

26. Jerarquía de directorios general . . . 37
27. Diagrama de la jerarquía de directorios para la síntesis lógica . . . 39

28. Flujo de diseño para la síntesis lógica. Flujo a) flujo propuesto en años anteriores modificado. b) flujo propuesto por el estudiante Noel Prado [24] . . . 42

29. Esquemático del circuito del pin PDDW0204SCDG . . . 48
30. Tabla de verdad de pin PDDW0204SCDG . . . 49
31. Módulo del pin PDDW0204SCDG definido en el archivo tpd018nv.v . . . 49
32. Representación de pin PDDW0204SCDG configurado como entrada . . . 50
33. Representación de pin PDDW0204SCDG configurado como salida . . . 51
34. Circuito representativo de la conexión entre pines de entrada y salida con el exterior de un circuito . . . 53

35. Generación de archivo simulador . . . 57
36. Ejecución de simulador . . . 58
37. Interfaz gráfica DVE 1 . . . 59
38. Interfaz gráfica DVE 2 . . . 59
39. Interfaz gráfica DVE 3 . . . 60
40. Interfaz gráfica DVE 4 . . . 60
41. Visualización del *waveform* del circuito *not* realizado . . . 61
42. Interfaz gráfica de Design Vision . . . 62
43. Ubicación de librerías agrupadas en un mismo directorio . . . 62
44. Configuración de librerías . . . 63
45. Variable search_path . . . 63
46. Variable target_library . . . 63
47. Variable link_library . . . 64
48. Importación de archivos 1 . . . 65
49. Importación de archivos 2 . . . 65
50. Importación de archivos 3 . . . 66
51. Importación de archivos 4 . . . 66
52. Importación de archivos 5 . . . 66
53. Importación de archivos 6 . . . 67
54. Importación de archivos 7 . . . 67
55. Importación de archivos 8 . . . 67
56. Importación de archivos 9 . . . 68
57. Importación de archivos 10 . . . 68
58. Importación de archivos 11 . . . 68
59. Importación de archivos 12 . . . 68
60. Vista al esquemático . . . 69
61. Esquemático de el circuito . . . 69
62. Esquemático de el circuito 2 . . . 69
63. Enlace del diseño con las librerías . . . 70
64. Enlace del diseño con las librerías 2 . . . 70
65. Unificación del diseño . . . 71
66. Unificación del diseño 2 . . . 71
67. Creación de relojes . . . 72

68. *Warning* de creación de relojes virtuales para circuitos asíncronos 72
69. Síntesis del circuito 73
70. Síntesis del circuito 74
71. Esquemático del circuito sintetizado 74
72. Reportes de Design Vision 75
73. Reporte de área del circuito *not* sintetizado 75
74. Generación de archivos de salida 76
75. Generación de archivos de salida 2 77
76. Archivo del *not* sintetizado 77
77. Creación de carpetas del circuito 84
78. Archivos necesarios en la carpeta del circuito en *inputs* 84
79. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar 85
80. Ejecución del *script* ejecutar_sintesis_logica.bash sin errores 85
81. Archivos generados en la carpeta del circuito en *outputs* 85
82. Archivo not_sintetizado.v: circuito *not* sintetizado 86
83. Archivos necesarios en la carpeta del circuito en *inputs* 91
84. Definición de la variable PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar 92
85. Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores 93
86. Archivos generados en la carpeta del circuito en *outputs* 93
87. Archivo not_sintesis_final.v: circuito *not* con pines de I/O sintetizado 94

88. Archivos necesarios en la carpeta del circuito en *inputs* 96
89. Simulación de una *not* en su diseño original 98
90. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar 99
91. Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores 99
92. Archivos generados en la carpeta del circuito en Outputs_iniciales 100
93. Simulación de la *not* en su diseño sintetizado 101
94. Archivos generados en la carpeta del circuito en Outputs_iniciales 101
95. Archivos generados en la carpeta del circuito en Outputs_finales 102
96. Simulación del *netlist* final 104
97. Archivos necesarios en la carpeta del circuito en *inputs* 104
98. Simulación de una ALU de 4 bits en su diseño original 108
99. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar 109
100. Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores 110
101. Archivos generados en la carpeta del circuito en Outputs_iniciales 110
102. Simulación de la ALU de 4 bits en su diseño sintetizado 113
103. Archivos generados en la carpeta del circuito en Outputs_iniciales 113
104. Archivos generados en la carpeta del circuito en Outputs_finales 117
105. Simulación del *netlist* final 121
106. Archivos necesarios en la carpeta del circuito en *inputs* 121
107. Simulación del sumador de 4 bits en su diseño original 125
108. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar 126
109. Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores 127

110. Archivos generados en la carpeta del circuito en Outputs_iniciales 127
111. Simulación del sumador de 4 bits en su diseño sintetizado 129
112. Archivos generados en la carpeta del circuito en Outputs_iniciales 129
113. Archivos generados en la carpeta del circuito en Outputs_finales 131
114. Simulación del *netlist* final 134
115. Archivos necesarios en la carpeta del circuito en *inputs* 134
116. Simulación del sumador de 16 bits en su diseño original 139
117. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar 140
118. Modificación del *script* comandos_sintesis_logica_2.script 141
119. Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores 142
120. Archivos generados en la carpeta del circuito en Outputs_iniciales 142
121. Simulación del sumador de 16 bits en su diseño sintetizado 145
122. Archivos generados en la carpeta del circuito en Outputs_iniciales 146
123. Archivos generados en la carpeta del circuito en Outputs_finales 151
124. Simulación del *netlist* final 156
125. Archivos necesarios en la carpeta del circuito en *inputs* 157
126. Simulación de sumador de 32 bits en su diseño original 163
127. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar 164
128. Modificación del *script* comandos_sintesis_logica_2.script 165
129. Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores 166
130. Archivos generados en la carpeta del circuito en Outputs_iniciales 166
131. Simulación del sumador de 32 bits en su diseño sintetizado 171
132. Archivos generados en la carpeta del circuito en Outputs_iniciales 171
133. Archivos generados en la carpeta del circuito en Outputs_finales 180
134. Simulación del *netlist* final 189
135. Archivos necesarios en la carpeta del circuito en *inputs* 190
136. Simulación del sumador de 16 bits en su diseño original 195
137. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar 195
138. Modificación del *script* comandos_sintesis_logica_2.script 196
139. Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores 197
140. Archivos generados en la carpeta del circuito en Outputs_iniciales 197
141. Simulación del restador de 16 bits en su diseño sintetizado 201
142. Archivos generados en la carpeta del circuito en Outputs_iniciales 202
143. Archivos generados en la carpeta del circuito en Outputs_finales 208
144. Simulación del *netlist* final 215
145. Archivos necesarios en la carpeta del circuito en *inputs* 216
146. Simulación de sumador de 32 bits en su diseño original 223
147. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar 224
148. Modificación del *script* comandos_sintesis_logica_2.script 225
149. Advertencia #1 al terminar la síntesis 226
150. Advertencia #2 en la primera síntesis 226
151. Archivos generados en la carpeta del circuito en Outputs_iniciales 227
152. Simulación del circuito sintetizado con errores obtenido a partir del diseño inicial del circuito 227

153. Advertencia al terminar la síntesis . . . 228
154. Advertencia en la primera síntesis solucionada . . . 228
155. Simulación del multiplicador de 32 bits en su diseño sintetizado . . . 242
156. Archivos generados en la carpeta del circuito en Outputs_iniciales . . . 242
157. Archivos generados en la carpeta del circuito en Outputs_finales . . . 243
158. Simulación del *netlist* final . . . 256
159. Archivos necesarios en la carpeta del circuito en *inputs* . . . 257
160. Simulación de sumador de 32 bits en su diseño original . . . 267
161. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar . . . 268
162. Modificación del *script* comandos_sintesis_logica_2.script . . . 269
163. Advertencia al terminar la síntesis . . . 270
164. Error en la primera síntesis . . . 270
165. Advertencia al terminar la síntesis . . . 271
166. Error solucionado de la primera síntesis . . . 272
167. Archivos generados en la carpeta del circuito en Outputs_iniciales . . . 272
168. Simulación del divisor de 32 bits en su diseño sintetizado . . . 289
169. Archivos generados en la carpeta del circuito en Outputs_iniciales . . . 289
170. Archivos generados en la carpeta del circuito en Outputs_finales . . . 290
171. Simulación del *netlist* final . . . 310
172. Archivos necesarios en la carpeta del circuito en *inputs* . . . 311
173. Creación del circuito nanochip El Gran Jaguar . . . 311
174. Simulación del nanochip en su diseño original . . . 313
175. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar . . . 314
176. Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores . . . 315
177. Archivos generados en la carpeta del circuito en Outputs_iniciales . . . 315
178. nanochip en su diseño sintetizado . . . 316
179. Archivos generados en la carpeta del circuito en Outputs_iniciales . . . 316
180. Archivos generados en la carpeta del circuito en Outputs_finales . . . 317
181. Simulación del *netlist* final . . . 318

182. Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar . . . 320
183. Modificación del *script* comandos_sintesis_logica_2.script . . . 320
184. Ejecución del *script* comandos_sintesis_logica_3.script para sintetizar un sumador de 16 bits . . . 321

185. Interfaz del *script* realizado . . . 325

Lista de cuadros

Cuadro 1: Generación de simulación VCS . . . 19
Cuadro 2: Comando para compilar y simular un diseño de *hardware* descrito en Verilog utilizando VCS . . . 19
Cuadro 3: Comando para invocar DVE desde la terminal . . . 20
Cuadro 4: Comando para inicializar Design Compiler por línea de comando . . . 22
Cuadro 5: Comando para inicializar Design Compiler desde interfaz . . . 22
Cuadro 6: Ejemplo de CMD con el comando para cargar y leer archivos después de inicializar Design Compiler . . . 22
Cuadro 7: Comando para inicializar Design Vision por línea de comando . . . 23
Cuadro 8: Línea de encabezado para ejecutar *script* con Bash . . . 25
Cuadro 9: Ejemplo básico de *script* Bash . . . 26

Cuadro 10: Archivo not.v, diseño de una *not* en Verilog . . . 46
Cuadro 11: Archivo not_tb.v, diseño del *testbench* para simulación de una *not* en Verilog . . . 47
Cuadro 12: Módulo del pin PDDW0204SCDG definido en el archivo tpd018nv.v . . . 50
Cuadro 13: Configuración del pin PDDW0204SCDG como entrada . . . 51
Cuadro 14: Configuración del pin PDDW0204SCDG como salida . . . 52
Cuadro 15: Declaración de pines de entrada, salida y *wires* para una *not* . . . 53
Cuadro 16: Configuración de los pines de entrada y salida para una *not* . . . 54
Cuadro 17: Archivo not_IO.v: módulo final de pines de entrada y salida para una *not* . . . 54

Cuadro 18: Comando para compilar y simular un diseño de *hardware* descrito en Verilog utilizando VCS . . . 56
Cuadro 19: Comando para compilar y simular el circuito *not* . . . 57
Cuadro 20: Comando para ejecutar la simulación . . . 57
Cuadro 21: Comando para abrir la interfaz gráfica de DVE . . . 58
Cuadro 22: Comando para abrir la interfaz gráfica de Design Vision . . . 61
Cuadro 23: Importación de librerías . . . 64
Cuadro 24: Comando para verificación del diseño . . . 73
Cuadro 25: *Script* ejecutar_sintesis_logica.bash utilizado para ejecutar la síntesis lógica . . . 78

Cuadro 26: *Script* comandos_sintesis_logica.script con los comandos para realizar la síntesis lógica . . . 79
Cuadro 27: Comandos para importación de librerías . . . 79
Cuadro 28: Importación del archivo Verilog del circuito . . . 80
Cuadro 29: Configuración de restricciones . . . 80
Cuadro 30: Creación de reloj . . . 81
Cuadro 31: Creación de reloj virtual . . . 81
Cuadro 32: Comando para verificación del diseño . . . 81
Cuadro 33: Comando para ejecutar síntesis lógica al circuito . . . 82
Cuadro 34: Comando para la generación de reportes . . . 82
Cuadro 35: Comandos para la generación de archivos de salida . . . 82
Cuadro 36: Comando para ejecutar el *script* ejecutar_sintesis_logica.bash . . . 85
Cuadro 37: *Script* ejecutar_sintesis_logica_2.bash utilizado para ejecutar la síntesis lógica . . . 87
Cuadro 38: *Script* comandos_sintesis_logica_2.script con los comandos para realizar la síntesis lógica . . . 90
Cuadro 39: Comando para ejecutar el *script* ejecutar_sintesis_logica_2.bash . . . 92

Cuadro 40: Archivo not.v: diseño de una *not* en Verilog . . . 96
Cuadro 41: Archivo IO_not.v: diseño de módulo de entradas y salidas para una *not* en Verilog . . . 97
Cuadro 42: Archivo not_tb.v: diseño del *testbench* para simulación de una *not* en Verilog . . . 97
Cuadro 43: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 98
Cuadro 44: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica.bash . . . 99
Cuadro 45: Archivo not_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito . . . 100
Cuadro 46: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 100
Cuadro 47: Archivo not_sintetizado_and_IO.v: circuito sintetizado con pines de I/O . . . 102
Cuadro 48: Archivo not_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O . . . 103
Cuadro 49: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 104
Cuadro 50: Archivo ALU4b.v: diseño de una ALU de 4 bits en Verilog . . . 105
Cuadro 51: Archivo IO_ALU4b.v: diseño de módulo de entradas y salidas para una ALU de 4 bits en Verilog . . . 106
Cuadro 52: Archivo ALU4b_tb.v: diseño del *testbench* para simulación de una ALU de 4 bits en Verilog . . . 108
Cuadro 53: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 108
Cuadro 54: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash . . . 109
Cuadro 55: Archivo ALU4b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito . . . 112

Cuadro 56: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 113
Cuadro 57: Archivo ALU4b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O . . . 117
Cuadro 58: Archivo ALU4b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del diseño original del circuito con pines de I/O . . . 120
Cuadro 59: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 120
Cuadro 60: Archivo ADDER4b.v: diseño de un sumador de 4 bits en Verilog . . . 122
Cuadro 61: Archivo IO_ADDER4b.v: diseño de módulo de entradas y salidas para un sumador de 4 bits en Verilog . . . 124
Cuadro 62: Archivo ADDER4b_tb.v: diseño del *testbench* para simulación de un sumador de 4 bits en Verilog . . . 124
Cuadro 63: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 125
Cuadro 64: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash . . . 126
Cuadro 65: Archivo ADDER4b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito . . . 128
Cuadro 66: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 128
Cuadro 67: Archivo ADDER4b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O . . . 131
Cuadro 68: Archivo ADDER4b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O . . . 133
Cuadro 69: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 133
Cuadro 70: Archivo ADDER16b.v: diseño de un sumador de 16 bits en Verilog . . . 135
Cuadro 71: Archivo IO_ADDER16b.v: diseño de módulo de entradas y salidas para un sumador de 16 bits en Verilog . . . 138
Cuadro 72: Archivo ADDER16b_tb.v: diseño del *testbench* para simulación de un sumador de 16 bits en Verilog . . . 139
Cuadro 73: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 139
Cuadro 74: Modificación del *script* comandos_sintesis_logica_2.script . . . 140
Cuadro 75: Comando para ejecutar la síntesis lógica del circuito con el *script* `ejecutar_sintesis_logica_2.bash` . . . 141
Cuadro 76: Archivo ADDER16b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito . . . 144
Cuadro 77: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 145
Cuadro 78: Archivo ADDER16b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O . . . 150
Cuadro 79: Archivo ADDER16b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O . . . 155
Cuadro 80: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 156
Cuadro 81: Archivo ADDER32b.v: diseño de un sumador de 32 bits en Verilog . . . 157

Cuadro 82: Archivo IO_ADDER32b.v: diseño de módulo de entradas y salidas para un sumador de 32 bits en Verilog . . . 162
Cuadro 83: Archivo ADDER32b_tb.v: diseño del *testbench* para simulación de un sumador de 32 bits en Verilog . . . 162
Cuadro 84: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 163
Cuadro 85: Modificación del *script* comandos_sintesis_logica_2.script . . . 164
Cuadro 86: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash . . . 165
Cuadro 87: Archivo ADDER32b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito . . . 170
Cuadro 88: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 171
Cuadro 89: Archivo ADDER32b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O . . . 180
Cuadro 90: Archivo ADDER32b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O . . . 188
Cuadro 91: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 188
Cuadro 92: Archivo SUB16b.v: diseño de un restador de 16 bits en Verilog . . . 191
Cuadro 93: Archivo IO_SUB16b.v: diseño de módulo de entradas y salidas para un restador de 16 bits en Verilog . . . 193
Cuadro 94: Archivo SUB16b_tb.v: diseño del *testbench* para simulación de un restador de 16 bits en Verilog . . . 194
Cuadro 95: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 194
Cuadro 96: Modificación del *script* comandos_sintesis_logica_2.script . . . 196
Cuadro 97: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash . . . 196
Cuadro 98: Archivo SUB16b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito . . . 201
Cuadro 99: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 201
Cuadro 100: Archivo SUB16b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O . . . 208
Cuadro 101: Archivo SUB16b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O . . . 214
Cuadro 102: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 215
Cuadro 103: Archivo MULT32b.v: diseño inicial de un multiplicador de 32 bits en Verilog . . . 217
Cuadro 104: Archivo MULT32b.v: diseño final de un multiplicador de 32 bits en Verilog . . . 218
Cuadro 105: Archivo IO_MULT32b.v: diseño de módulo de entradas y salidas para un multiplicador de 32 bits en Verilog . . . 221
Cuadro 106: Archivo MULT32b_tb.v: diseño del *testbench* para simulación de un multiplicador de 32 bits en Verilog . . . 222

Cuadro 107: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 223
Cuadro 108: Modificación del *script* comandos_sintesis_logica_2.script . . . 224
Cuadro 109: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash . . . 225
Cuadro 110: Archivo MULT32b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito . . . 241
Cuadro 111: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 242
Cuadro 112: Archivo MULT32b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O . . . 255
Cuadro 113: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 256
Cuadro 114: Archivo DIV32b.v: diseño inicial de un divisor de 32 bits en Verilog . 259
Cuadro 115: Archivo DIV32b.v: diseño final de un divisor de 32 bits en Verilog . . 260
Cuadro 116: Archivo IO_DIV32b.v: diseño de módulo de entradas y salidas para un divisor de 32 bits en Verilog . . . 265
Cuadro 117: Archivo DIV32b_tb.v: diseño del *testbench* para simulación de un divisor de 32 bits en Verilog . . . 266
Cuadro 118: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 267
Cuadro 119: Modificación del *script* comandos_sintesis_logica_2.script . . . 268
Cuadro 120: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash . . . 269
Cuadro 121: Archivo DIV32b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito . . . 288
Cuadro 122: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 288
Cuadro 123: Archivo DIV32b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O . . . 309
Cuadro 124: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 310
Cuadro 125: Archivo IO_nanochip.v: diseño de módulo de entradas y salidas para el nanochip en Verilog . . . 312
Cuadro 126: Archivo nanochip_tb.v: diseño del *testbench* para simulación del nanochip en Verilog . . . 313
Cuadro 127: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE . . . 313
Cuadro 128: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash . . . 314
Cuadro 129: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 316
Cuadro 130: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE . . . 317

Cuadro 131: *Script* ejecutar_sintesis_logica_3.bash . . . 324

Resumen

El presente trabajo se centra en tres puntos principales. El primero consiste en la mejora del proceso de automatización en la etapa de síntesis lógica, con el objetivo de lograr la síntesis lógica de cualquier circuito descrito en Verilog. El segundo punto es la optimización del proceso de instalación automatizada de las aplicaciones de Synopsys. Finalmente, el trabajo busca documentar todo el proceso realizado en las partes de *front-end* y *back-end* del flujo de diseño de circuitos, con el fin de comprender mejor el uso de las herramientas de Synopsys, la estructura seguida y los *scripts* empleados para automatizar los procesos, haciendo posible que este proceso sea replicable para cualquier tecnología.

El trabajo se ha estructurado en secciones organizadas de manera secuencial para abordar los diferentes puntos clave. Estas secciones son:

- **Implementación de un servidor NFS:** se explica cómo se configuró un servidor NFS para facilitar el intercambio de archivos entre los miembros del equipo de trabajo.
- **Librerías de TSMC:** se detallan las librerías de TSMC utilizadas en el proceso.
- **Jerarquía de directorios:** se describen las mejoras realizadas en la jerarquía de directorios, con el objetivo de optimizar la organización de archivos y asegurar una síntesis ordenada y eficiente de cualquier circuito.
- **Flujo de diseño para la síntesis lógica:** se describe el flujo de diseño seguido para llevar a cabo la síntesis lógica de los circuitos.
- **Desarrollo en Verilog:** se explica cómo diseñar circuitos, *testbench* y módulos de pines de entrada y salida en Verilog, con el fin de entender su construcción.
- **Configuración de herramientas y *scripts*:** se aborda la configuración y el uso de las herramientas y *scripts* empleados en esta primera etapa, para entender su funcionamiento.
- **Resultados de síntesis de circuitos:** se presentan los resultados de la síntesis de varios circuitos, con el objetivo de verificar que el *script* desarrollado permite la síntesis tanto de circuitos combinacionales como secuenciales.

- **Mejoras del *script* utilizado para la síntesis lógica:** se presentan mejoras significativas en uno de los *scripts*, con el propósito de simplificar la síntesis de circuitos y permitir al usuario una interacción más directa y eficiente desde la terminal.
- ***Script* de instalación de aplicaciones Synopsys:** se explica un *script* que mejora el proceso de instalación de las aplicaciones de Synopsys, incluyendo la descarga e instalación de todas las dependencias necesarias antes de proceder con la instalación de dichas aplicaciones.

Los resultados obtenidos fueron satisfactorios, ya que se lograron cumplir con los objetivos establecidos. Se construyó una jerarquía de directorios que permite organizar de manera eficiente los archivos necesarios y generados durante el proceso de síntesis lógica de cualquier circuito. Además, se desarrollaron tres nuevos *scripts* para la síntesis lógica que permiten realizar la síntesis del circuito original y del circuito sintetizado, junto con el módulo de pines de entrada y salida, de manera automática. Esto elimina la necesidad de realizar la síntesis dos veces de forma manual, como se hacía en años anteriores. También se llevó a cabo con éxito la síntesis lógica de varios circuitos, tanto síncronos como asíncronos, comenzando con un circuito *not*, *and*, *xor* y *or*, seguido de ALU, sumadores de 4, 16 y 32 bits, contadores de 2, 4 y 8 bits, un restador de 16 bits, multiplicadores y divisores de 32 bits, y el nanochip El Gran Jaguar. Este proceso permitió identificar aspectos clave a considerar al sintetizar circuitos secuenciales dependientes del reloj, algo que no se había realizado en años anteriores.

Además, se mejoró el proceso de descarga de las aplicaciones de Synopsys mediante un nuevo *script*, que crea todas las dependencias iniciales (directorios, permisos, instalación de bibliotecas y aplicaciones, y creación de llaves de seguridad RSA para la conexión al servidor de Synopsys) y lleva a cabo la instalación del instalador de Synopsys, lo cuales son pasos necesarios antes de descargar las aplicaciones.

En este trabajo se elaboró la documentación necesaria, tanto escrita como en formato de video, para facilitar la comprensión del funcionamiento y uso de las diversas herramientas empleadas en el proceso de síntesis lógica.

Para futuras investigaciones en esta línea de trabajo, sería interesante implementar y sintetizar diseños más grandes o con arquitecturas más sofisticadas, con el fin de validar la robustez y escalabilidad del proceso. Un ejemplo de esto podría ser un microcontrolador *nibbler* de 4 bits, y luego escalarlo a uno de 8, 16 y 32 bits. También sería valioso contar con una interfaz gráfica que permita interactuar directamente con el sistema para ingresar las librerías a utilizar, los archivos de entrada a sintetizar, y para interactuar de manera directa con la aplicación de DVE o Waveview, con el objetivo de visualizar las señales tanto del circuito original como del sintetizado en una misma interfaz, evitando así la interacción a nivel de comandos.

En cuanto al *script* que permite la descarga e instalación de las aplicaciones de Synopsys (APPS.sh), se podrían realizar mejoras. Actualmente, en este *script*, la variable "apps_real" contiene directamente los nombres iniciales de las aplicaciones. Sin embargo, Synopsys puede cambiar esos nombres, lo que generaría un error si el nombre de la aplicación no coincide al intentar descargarla. Para solucionar esto, se podría mejorar el proceso para que la variable no contenga directamente los nombres de las aplicaciones, sino que haga referencia a los

nombres contenidos en el archivo myproductsrev.txt, que contiene todos los nombres de las aplicaciones. De esta manera, si los nombres que contiene el archivo cambian, también se actualizaría la variable "apps_real", evitando así errores al intentar descargar alguna aplicación.

Abstract

The present work focuses on three main points. The first consists of improving the automation process in the logic synthesis stage, with the aim of achieving the logic synthesis of any circuit described in Verilog. The second point is the optimization of the automated installation process of Synopsys applications. Finally, the work seeks to document the entire process carried out in the *front-end* and *back-end* stages of the design flow, in order to better understand the use of Synopsys tools, the structure followed, and the *scripts* used to automate the processes, making this procedure replicable for any technology.

The work has been structured into sections organized sequentially to address the key aspects. These sections are:

- **Implementation of an NFS server:** explains how an NFS server was configured to facilitate file sharing among team members.
- **TSMC libraries:** details the TSMC libraries used in the process.
- **Directory hierarchy:** describes the improvements made to the directory hierarchy, aiming to optimize file organization and ensure an orderly and efficient synthesis of any circuit.
- **Design flow for logic synthesis:** describes the design flow followed to carry out the logic synthesis of circuits.
- **Development in Verilog:** explains how to design circuits, *testbench*, and input/output pin modules in Verilog, in order to understand their construction.
- **Configuration of tools and *scripts*:** addresses the configuration and use of the tools and *scripts* employed in this first stage to understand their operation.
- **Synthesis results:** presents the results of the synthesis of several circuits, with the aim of verifying that the developed *script* allows the synthesis of both combinational and sequential circuits.

- **Improvements to the logic synthesis *script*:** presents significant improvements in one of the *scripts*, with the purpose of simplifying circuit synthesis and allowing more direct and efficient interaction from the terminal.
- **Installation *script* for Synopsys applications:** explains a *script* that improves the installation process of Synopsys applications, including downloading and installing all the necessary dependencies before proceeding with the installation of said applications.

The results obtained were satisfactory, as the objectives set were achieved. A directory hierarchy was built that allows efficient organization of the files needed and generated during the logic synthesis process of any circuit. In addition, three new *scripts* were developed for logic synthesis that automatically perform the synthesis of the original circuit and the synthesized circuit, together with the input/output pin module. This eliminates the need to perform the synthesis twice manually, as was done in previous years. The logic synthesis of several circuits was also successfully carried out, both synchronous and asynchronous, starting with *not*, *and*, *xor*, and *or* circuits, followed by ALU, adders of 4, 16, and 32 bits, counters of 2, 4, and 8 bits, a 16-bit subtractor, 32-bit multipliers and dividers, and the nanochip El Gran Jaguar. This process allowed the identification of key aspects to consider when synthesizing sequential circuits dependent on the clock, something that had not been done in previous years.

In addition, the download process of Synopsys applications was improved through a new *script*, which creates all the initial dependencies (directories, permissions, installation of libraries and applications, and creation of RSA security keys for connection to the Synopsys server) and carries out the installation of the Synopsys installer, which are necessary steps before downloading the applications.

In this work, the necessary documentation was prepared, both written and in video format, to facilitate the understanding of the operation and use of the various tools employed in the logic synthesis process.

For future research in this line of work, it would be interesting to implement and synthesize larger designs or designs with more sophisticated architectures, in order to validate the robustness and scalability of the process. An example of this could be a *nibbler* microcontroller of 4 bits, and then scale it to one of 8, 16, and 32 bits. It would also be valuable to have a graphical interface that allows direct interaction with the system to input the libraries to be used, the input files to be synthesized, and to directly interact with the DVE or Waveview application, with the objective of visualizing the signals of both the original and synthesized circuit in the same interface, thus avoiding command-level interaction.

Regarding the *script* that allows the download and installation of Synopsys applications (APPS.sh), improvements could be made. Currently, in this *script*, the variable "apps_real" directly contains the initial names of the applications. However, Synopsys may change these names, which would generate an error if the application name does not match when attempting to download it. To solve this, the process could be improved so that the variable does not directly contain the application names, but instead references the names contained in the myproductsrev.txt file, which includes all the application names. In this way, if the names contained in the file change, the "apps_real" variable would also be updated, thus avoiding errors when trying to download an application.

CAPÍTULO 1

Introducción

La nanoelectrónica es una rama de la electrónica que se especializa en el desarrollo y manipulación de dispositivos a escala nanométrica, lo que ha permitido integrar millones de transistores en un solo chip. No obstante, el diseño de circuitos integrados requiere una serie de pasos estructurados, conocido como flujo de diseño (*design flow*), que se divide en dos etapas principales: *front-end* y *back-end*. La fase de *front-end* se centra en la síntesis lógica de los circuitos, mientras que la etapa de *back-end* aborda la síntesis y verificación físicas.

Este trabajo se centra en la fase de *front-end*, cubriendo desde el diseño de circuitos en un lenguaje de descripción de *hardware* (HDL) hasta la síntesis lógica en celdas de 180 nm, proporcionadas por Taiwan Semiconductor Manufacturing Company Limited (TSMC). Para facilitar la comprensión, el documento está organizado en capítulos que detallan cada paso del proceso.

En el capítulo 8, titulado *Identificación de librerías y archivos de TSMC de 180 nm*, se examinan las librerías específicas de esta tecnología, necesarias para la síntesis lógica de circuitos. El capítulo 9, *Jerarquía de directorios*, presenta la organización de archivos empleada para una síntesis ordenada y eficiente.

El capítulo 10, *Flujo de diseño a seguir para la síntesis lógica*, describe en profundidad el proceso seguido para la síntesis lógica de circuitos en Verilog. Luego, el capítulo 11, *Diseño de circuitos en HDL*, aborda el diseño de circuitos en Verilog, incluyendo el desarrollo del módulo lógico del circuito, el *testbench* para simulación y el módulo de pines de entrada y salida que debe usarse para la parte de síntesis física, el cual permite que cualquier circuito pueda interactuar con el exterior.

En el capítulo 12, *Configuración y uso de herramientas para simulación y síntesis de circuitos*, se detallan herramientas de Synopsys como VCS y DVE, utilizadas para simular circuitos y verificar su comportamiento lógico, así como Design Compiler y Design Vision para la síntesis lógica. El capítulo 13, *Sintetización lógica de distintos circuitos descritos en Verilog utilizando el Flujo de Diseño A*, documenta el proceso de diseño, simulación y síntesis de diversos circuitos, validando la capacidad del flujo de diseño para sintetizar

cualquier circuito en Verilog, incluyendo diseños síncronos y asíncronos, como el nanochip El Gran Jaguar, sumadores, restadores, contadores, ALU y módulos de multiplicación y división de 32 bits.

Luego, en el capítulo 14, *Mejoras realizadas en el script ejecutar_ sintesis_ logica_ 2.bash*, se presentan nuevas mejoras en uno de los *scripts* utilizados para la síntesis lógica. Este nuevo *script* tiene como objetivo simplificar el proceso de síntesis de cualquier circuito, permitiendo al usuario interactuar directamente desde la terminal, evitando la necesidad de realizar ciertas configuraciones manuales directamente en los *scripts* realizados.

Finalmente, otro enfoque clave de este trabajo fue la optimización del proceso de descarga e instalación de herramientas de Synopsys mediante un *script* en Bash, que automatiza la instalación de las dependencias iniciales antes de la obtención de las aplicaciones, garantizando una descarga confiable y minimizando los errores comunes en el proceso manual.

Este documento ofrece una guía detallada de la etapa de *front-end*, permitiendo que el proceso sea replicable para diferentes tecnologías, así como una guía clara sobre cómo instalar las herramientas de Synopsys en una máquina con sistema operativo Linux.

CAPÍTULO 2

Antecedentes

2.1. Línea de investigación

Uno de los primeros avances que se dio en el estudio de circuitos nanométricos por parte de la Universidad del Valle de Guatemala fue en el año 2013, donde se realizó uno de los cambios en el mapa curricular de la carrera de Ingeniería Electrónica para impartir por primera el curso Introducción a sistemas de diseño VLSI. En este curso básicamente se impartió la teoría básica sobre el diseño de VLSI, donde fue algo totalmente nuevo, y que los egresados del Departamento de Ingeniería Electrónica y Mecatrónica no tenían conocimiento a profundidad sobre el diseño de circuitos a nivel nanométrico.

En el 2014, el Departamento de Ingeniería Electrónica y Mecatrónica logró un acuerdo académico con la compañía Synopsys, para lograr tener acceso a sus herramientas de *software* y librerías para realizar la elaboración y diseño de circuitos nanoelectrónicos, lo que permitió que estudiantes de la carrera de Ingeniería Electrónica pudieran estudiar con mayor profundidad la electrónica a nivel nanométrico. Esto permitió el primer trabajo de graduación orientado en el campo de VLSI, en donde se realizó el diseño de un sumador/restador completo de 32 bits con tecnología CMOS de 28 nanómetros usando aplicaciones de diseño de la empresa Synopsys, realizado por el Ing. Jonathan de los Santos [1] en el mismo año.

En base a los avances de estos últimos años, en el 2015 se realizaron cambios en el mapa curricular de Ingeniería Electrónica, sustituyendo el curso de Introducción a sistemas de diseño VLSI por los cursos de Nanoelectrónica 1 y 2, con el fin de profundizar temas de la teoría sobre el diseño de VLSI y enseñar a estudiantes el uso de herramientas que ofrece el *software* Synopsys, y que son utilizados por los líderes de la industria de semiconductores. De esta forma, se impulsó la iniciativa de crear el primer nanochip de Centroamérica, logrando el apoyo para realizar la fabricación de un nanochip por la empresa TSMC (empresa líder en la fabricación de semiconductores) a través del Centro Interuniversitario de Microelectrónica IMEC.

2.2. Inicios del diseño del circuito integrado El Gran Jaguar

De esta manera es como se logra el primer acercamiento a un proyecto multigeneracional que inició en 2019, con la implementación y documentación de un flujo de diseño para la fabricación del primer circuito integrado en la historia de Guatemala y con la elaboración de la primera etapa del circuito integrado, el cual fue realizado como trabajo de graduación por los Ingenieros Steven Rubio [2] y Luis Nájera[3], con la supervisión de M.Sc.Carlos Esquit.

En el proyecto presentado por el Ing. Steven Rubio se presentó el flujo de diseño (*design flow*), en donde se planteó el proceso de diseño o los pasos a seguir para la fabricación de un circuito integrado (CI) que van desde el diseño de su funcionalidad hasta los planos utilizados para su fabricación. Según este primer avance, se plantearon dos esquemas generales mínimos que se tenían que seguir para lograr los resultados esperados, tomando en cuenta las herramientas con las que se contaban, en donde el proceso de diseño se divide en dos categorías: síntesis lógica (*front-end*) y síntesis física (*back-end*). Cabe mencionar que el flujo de diseño inicial se realizó con las herramientas del *software* Synopsys y aplicaron el proceso de diseño con librerías de TSMC.

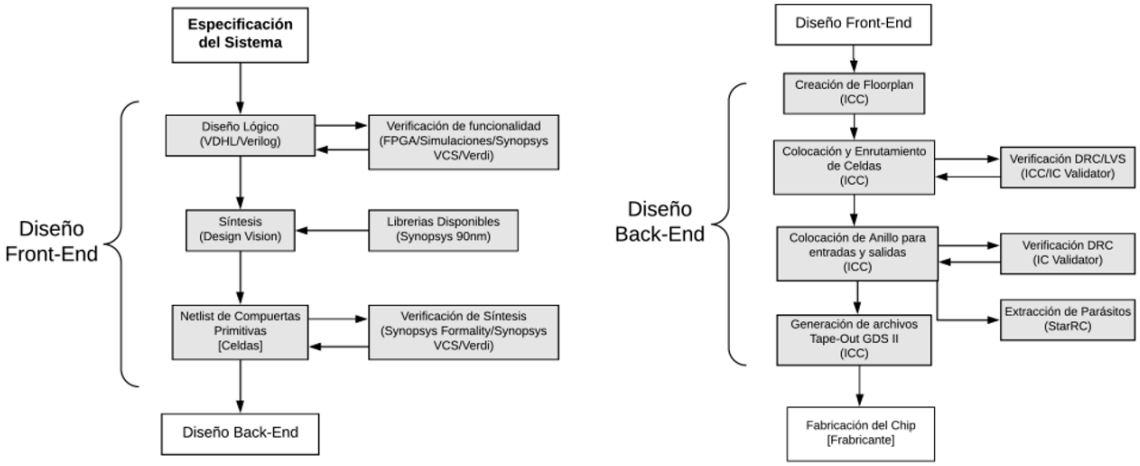


Figura 1: Flujo de diseño para síntesis lógica (*front-end*) y síntesis física (*back-end*) presentado por el Ing. Steven Rubio [2]

El otro primer avance presentado por el Ing. Luis Nájera, se enfocó en los primeros pasos del flujo de diseño para la elaboración de un circuito nanométrico con tecnología CMOS. El objetivo principal de este trabajo fue la elaboración de circuitos en lenguaje descriptivo de *hardware* para su posterior mapeo a *netlists* de compuertas lógicas por medio de una síntesis, y su verificación de funcionamiento tanto antes como después de la síntesis. Para realizar la síntesis lógica se utilizó la herramienta Design Vision, desarrollada por Synopsys, la cual fue de gran importancia para realizar la conversión de circuitos elaborados en lenguaje descriptivo Verilog a un *netlists* de compuertas lógicas. De la misma manera, se mostró la utilización de restricciones para la optimización del circuito acorde a las especificaciones colocadas. También se presentó la funcionalidad del circuito integrado, el cual es lograr un circuito que contará con 12 pines. 2 pines destinados a Vdd y Vss, 1 pin destinado a una entrada de reloj, 1 pin destinado a una entrada de *reset* y 8 pines de salida. Básicamente al

recibir un pulso de reloj los ocho pines de salida mostrarán una representación de 8 bits de una letra. Con cada pulso de reloj la letra mostrada en los pines de salida cambiará, con el objetivo de imprimir un mensaje.

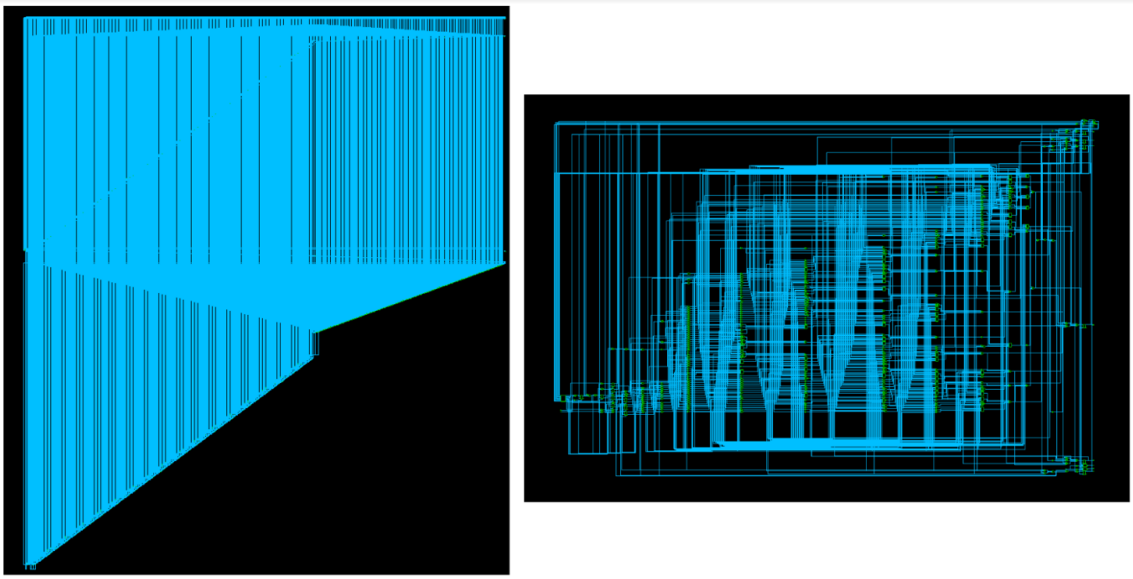


Figura 2: Síntesis lógica del circuito a fabricar. (a) Esquemático del circuito sin el uso de restricciones. (b) Esquemático del circuito utilizando restricciones presentado por el Ing. Luis Nájera [3]

2.3. Avances del circuito integrado El Gran Jaguar

Habiendo asentado las bases para este proyecto, desde 2020 se han realizado avances para lograr el objetivo de fabricar el primer circuito integrado de Guatemala. Se han presentado trabajos de graduación por parte de estudiantes de Ingeniería Electrónica de la Universidad del Valle de Guatemala con distintos enfoques, pero que se complementan para lograr el diseño y fabricación del nanochip.

2.4. Avances 2020

En 2020, los trabajos de graduación se centraron en la extracción de parásitos y simulación en HSPICE. Esta etapa es crucial para lograr la realización de extracciones para la simulación del chip y así comprobar si el diseño cumple con las funciones planteadas [4].

De la misma manera, se llevó a cabo el proceso de implementación del anillo de entradas/salidas, en donde se escogen los pads de entradas/salidas adecuados [5]. Se realizaron las verificaciones ERC (*electrical rule check*), ARC (*antenna rule check*) y verificación DRC (*design rule check*), las cuales básicamente son verificaciones físicas que un circuito integrado debe tener para poder ser fabricado [6] [7].

También se realizaron pruebas específicas para estudiar partes del flujo de diseño y las

herramientas que se utilizan para ello [7] [8]. Sin embargo, es importante recalcar que en el 2019 se presentaron errores del anillo de entradas/salidas, tampoco se obtuvo lo esperado en la corrida de las pruebas de verificaciones ERC, ARC y DRC, por lo que se trabajó en ello durante el 2020.

2.5. Avances 2021

En 2021, algunos estudiantes se encargaron de desarrollar la etapa completa de la síntesis lógica para lograr el diseño lógico del nanochip [9]. Otros trabajaron en la parte del *back-end* del flujo de diseño, lo cual como se mencionó anteriormente, abarca lo que es el diseño y desarrollo de la síntesis física. También se realizó la ejecución de la fase de verificación física *layout vs schematic* (LVS) [10]. En la Figura 3 se muestra la síntesis física obtenida a través del *software* IC Compiler II. [11]

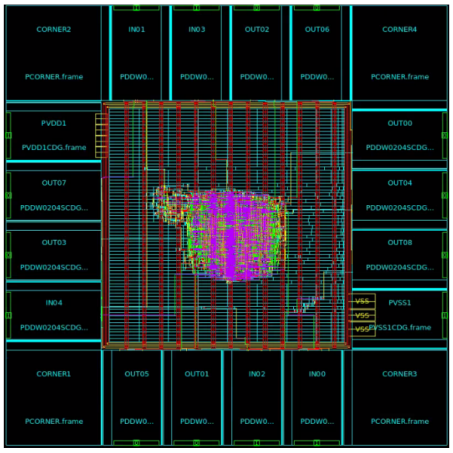


Figura 3: Síntesis física obtenida a través del *software* IC Compiler II

Es importante mencionar que en cada año que se ha trabajado en el proyecto del nanochip ha sido necesario replicar todo lo realizado en años anteriormente, incluyendo las verificaciones de reglas eléctricas, de antena y de diseño. Además, a pesar de que en ese año se logró pasar las pruebas de verificación de antena, ERC y LVS, durante las reglas de diseño (DRC) se obtuvieron errores de densidad, los cuales no pudieron ser solucionados en ese año [11] [12] [13].

2.6. Avances 2022

En el 2022, nuevamente se replicó lo realizado en los años anteriores. Se siguió corrigiendo errores de diseño, ya que a pesar de que se logró realizar de forma exitosa las verificaciones físicas de ARC, LVS y ERC, aún quedaba pendiente la solución de los seis errores de densidad provenientes de la verificación DRC. Algunos estudiantes con el fin de reducir los errores de DRC y densidad realizaron pruebas con distintos escenarios, en donde se destacaron dos

escenarios:

- donde solo hay 2 errores de densidad y ninguno de reglas [14].
- donde no tiene errores de densidad, pero tiene errores de reglas. Dichos errores son demasiados por lo que no se pueden solucionar [14].

```
Density Report Summary:
-----------------------------------------------------------------------
Window = {{0 0 534.48 532.24}}
---------------------
Layer     Density(%)
---------------------
METAL1    0
METAL2    0
METAL3    0
METAL4    0
METAL5    21.1881
METAL6    22.9467
```

Figura 4: 1er escenario con 2 errores de densidad

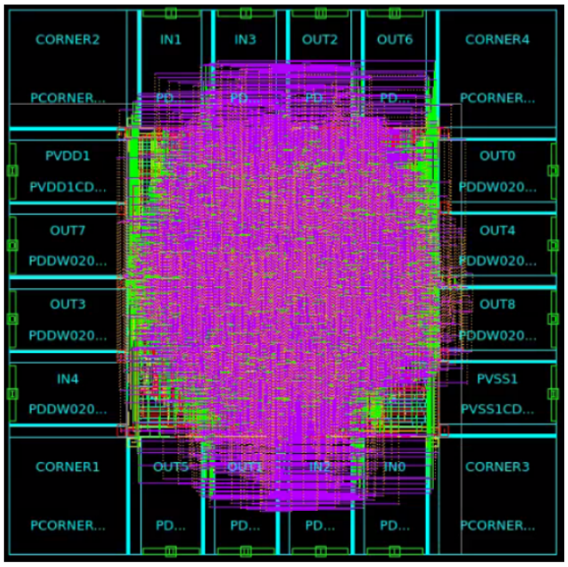


Figura 5: 1er escenario de síntesis física obtenida a través del *software* IC Compiler II con errores de densidad

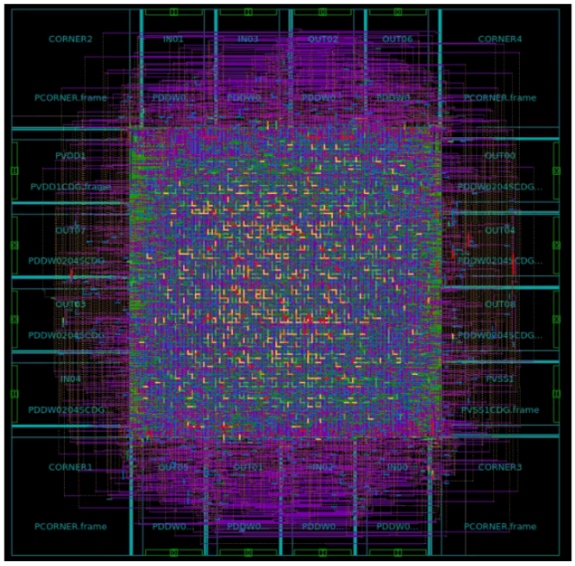


Figura 6: 2do escenario de síntesis física obtenida a través del *software* IC Compiler II sin errores de densidad

También se trabajó en la solución para los errores en la generación del *deck* resultante de la extracción de parásitos. Además de ello, se trabajó en una interfaz gráfica para generar los archivos necesarios para realizar las verificaciones físicas de DRC, LVS, ERC, antena y extracción de parásitos [14].

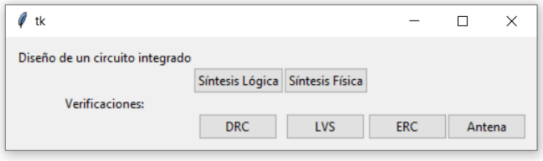


Figura 7: Interfaz gráfica para generar los archivos necesarios para realizar las verificaciones físicas

Por otro lado, se trabajó en la automatización de etapas de flujo de diseño (específicamente de los procesos de síntesis lógica, física y verificación de extracción de parásitos) [15] [16]. Asimismo, se automatizó el proceso de las verificaciones físicas (DRC, ERC, LVS y las verificaciones de antena) y también se realizó un *script* para automatizar todo el proceso de instalación de *software* de Synopsys [17]. Se realizó un estudio profundo de nuevas herramientas útiles para el diseño del nanochip. Algunas de estas fueron StarRC, IC Compiler II, PrimeSim, entre otras [18] [19] [20] [21]. Además, se estudió e implementó un nuevo flujo de diseño proporcionado por Synopsys, y se realizaron pruebas físicas implementando y verificando la funcionalidad del código obtenido durante la síntesis lógica utilizando un FPGA Xilinx Virtex-5 Genesys [22].

Como complemento también se trabajó en una aplicación y sintetizador digital, utilizando un microcontrolador Tiva C el cual tuvo la función de recibir un bus de datos de 8 bits (proveniente de las salidas de la FPGA) para así transmitirlo mediante comunicación serial (UART) a una computadora. Luego por medio de una aplicación realizada en Python, se

realizó una conversión de una cadena de texto a audio para así lograr el sintetizador digital, y lograr probar el funcionamiento de la síntesis lógica [23]. Para ello también se trabajó en la creación de archivos Verilog para lograr realizar las pruebas físicas con la plataforma de desarrollo antes mencionada [15].

2.7. Avances 2023

En el 2023, se realizaron mejoras al flujo de diseño que se ha propuesto en años anteriores, simplificando la jerarquía de directorios que es utilizado para realizar la síntesis lógica y física. Además, se siguió trabajando en la corrección de errores de densidad, ya que sin la corrección de los errores de densidad y polisilicio no se logrará el requisito mínimo para la fabricación del nanochip [24].

Para corregir los errores de densidad se empleó el *runset* para realizar el relleno de metal proporcionado por TSMC. Habiendo corregido los errores de densidad, fue necesario verificar que no existieran cortos o *nets* desconectados. Sin embargo, al realizar las verificaciones de *layout vs schematic* (LVS) se presentaron errores utilizando el *runset* antes mencionado, a pesar de que se genera una síntesis física sin errores de densidad. Dichos errores se originaron debido a una falta de conexión entre los pines de alimentación y el anillo, así también la falta de conexión entre el anillo y la red de alimentación interna del circuito. Por lo que no se resolvió por completo los errores de densidad, pero sí se redujo su porcentaje [24].

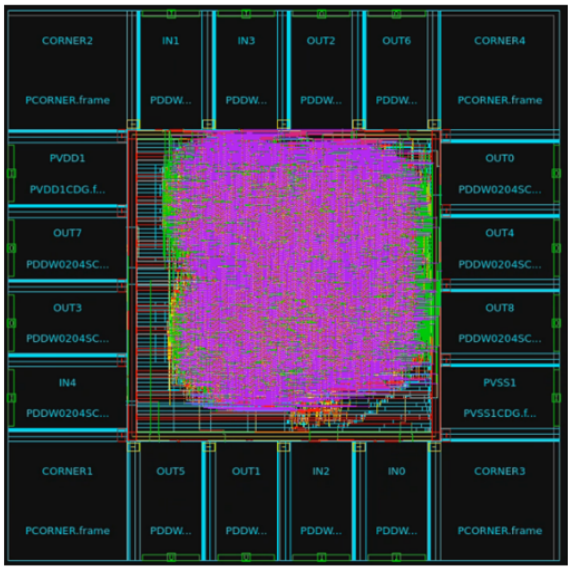


Figura 8: Síntesis física obtenida a través del *software* IC Compiler II sin errores de densidad [24]

```
Information: Detected floating route violation for Net VDD.
BBox: (125.0000 367.0000)(409.4800 373.0000). (RT-587)
Information: Detected floating route violation for Net VSS.
BBox: (261.3900 418.2400)(273.0900 426.2400). (RT-587)
```

Figura 9: Errores en las verificaciones de *layout vs schematic* (LVS) [24]

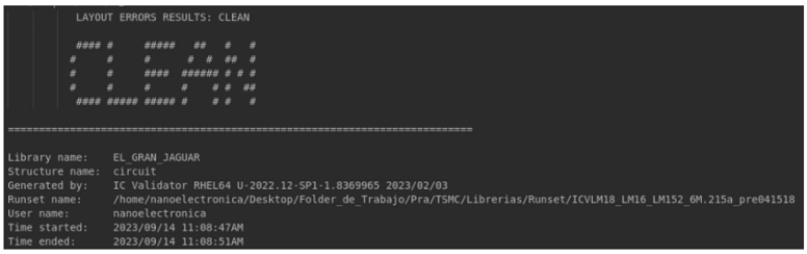


Figura 10: Verificaciones de *design rule check* (DRC) [24]

CAPÍTULO 3

Justificación

El proceso de diseño, estudio y fabricación de circuitos integrados es un gran reto, ya que es un proceso complejo que conlleva aprender conceptos que son de gran importancia en la nanoelectrónica, asimismo conlleva utilizar herramientas de software avanzadas que utilizan empresas como Intel, AMD, Qualcomm, TSMC, entre otras, que no están disponibles para todo el público. Pocos países, como lo son Estados Unidos, Taiwán, Corea del Sur, China, entre otros, se destacan y son de los pocos que contribuyen significativamente en el diseño o fabricación de nanochips, ya que tienen disponible los conocimientos y las herramientas para ello. Por esa misma razón, muy pocas personas que viven en países de Latinoamérica y que tienen interés en el tema de nanoelectrónica han tenido poco acceso a este tipo de conocimiento, incluyendo tanto la teoría, así como los recursos tecnológicos necesarios para llevar a cabo distintas prácticas de diseños de circuitos integrados.

Sin embargo, Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica de la Universidad del Valle de Guatemala cuenta con las herramientas de la empresa Synopsys, las cuales no cualquier persona o institución educativa tiene acceso. Estas herramientas son de gran importancia para lograr diseñar y fabricar circuitos integrados. Por esa razón, los estudiantes de Ingeniería Electrónica de la Universidad del Valle de Guatemala tienen la ventaja de tener disponible tanto el conocimiento como las herramientas, ya que en su plan de estudios incluyen dos cursos (Nanoelectrónica 1 y 2) en donde se aborda la teoría del diseño de circuitos integrados y se utilizan las herramientas de Synopsys para realizar diferentes prácticas. Esta integración de herramientas avanzadas es parte de la estrategia de mejora continua en el ámbito académico y de innovación del Departamento de Ingeniería Electrónica.

Este trabajo de graduación se enfoca en actualizar el diseño del circuito El Gran Jaguar descrito en Verilog, además se realizará la síntesis lógica del circuito modificado y se mejorará la automatización de dicha etapa, esto con el fin de que se logre realizar la síntesis lógica para cualquier circuito descrito en HDL. Además de ello, este trabajo se centra en documentar todo el proceso necesario para llevar a cabo la síntesis lógica y toda la parte de *back-end* para el diseño de un circuito integrado (incluyendo texto, fotos y videos). Dicha

documentación detallará el uso de cada herramienta que se utiliza en cada paso del flujo de diseño, además de los comandos que se emplean, ya que esta información será de gran importancia para el encargado de automatizar todo el flujo de diseño para la elaboración de un circuito integrado. Finalmente, se mejorará el proceso de la instalación automatizada de las aplicaciones de Synopsys, ya que en este año (2024) durante el proceso de la instalación de las nuevas versiones de las aplicaciones en una instalación en limpio de una máquina virtual Rocky Linux, surgieron errores al ejecutar el *script* que hace el trabajo de la instalación automática de las aplicaciones de Synopsys, ya que hay ciertas dependencias que hay que instalar y configurar previamente. Esto se realizará con el fin de que al ejecutar un primer *script* se realicen las configuraciones e instalaciones necesarias, de modo que al ejecutar posteriormente el *script* de instalación de aplicaciones de Synopsys, el proceso se complete sin inconvenientes.

CAPÍTULO 4

Objetivos

4.1. Objetivo general

Mejorar el proceso de automatización de la etapa de síntesis lógica, para que se logre la sintetización lógica para cualquier circuito descrito en Verilog. Además, se busca documentar todo el proceso que se realiza en la parte de *front-end* y *back-end* del flujo de diseño para el diseño del circuito integrado, así también mejorar el proceso de la instalación automatizada de las aplicaciones de Synopsys.

4.2. Objetivos específicos

- Replicar el avance del proyecto del circuito integrado realizado en años anteriores.
- Realizar un *script* para sintetizar de forma lógica cualquier circuito descrito en Verilog.
- Colaborar con el equipo de trabajo encargado de realizar las pruebas físicas ERC, LVS, antena, extracción de parásitos y pruebas finales, proporcionándoles los circuitos sintetizados lógicamente.
- Generar una documentación detallada de cada una de las fases del proceso de diseño de un circuito integrado.
- Realizar las mejoras del proceso de la instalación automatizada de las aplicaciones de Synopsys, para evitar errores en una instalación limpia de Rocky Linux al usar el *script* existente de descarga e instalación de aplicaciones.

CAPÍTULO 5

Alcance

El alcance de este trabajo es poder mejorar el proceso de automatización de la etapa de la síntesis lógica, para que se logre realizar la sintetización lógica para cualquier circuito descrito en Verilog, utilizando las librerías de TSMC de tecnología de 180 nm. Se utilizarán algunas herramientas de software, como Python y las herramientas de Synopsys, específicamente VCS, DVE y Design Vision. Además, se espera poder documentar y detallar todo el proceso que se realiza en la parte de *front-end y back-end* del flujo de diseño para el diseño del circuito integrado El Gran Jaguar y otros circuitos. Para ello se estará en constante comunicación con las personas encargadas de la parte de *back-end*, para documentar todo el proceso, desde los comandos y las herramientas empleadas. Por último se espera mejorar el proceso de la instalación automatizada de las aplicaciones de Synopsys, ya que el *script* desarrollado para automatizar la descarga de las aplicaciones no contemplaba un entorno de instalación desde cero. Esto requería la descarga de ciertas dependencias iniciales y la realización de configuraciones previas antes de ejecutar el *script*. Por esa razón se espera mejorar el proceso de las instalaciones de las aplicaciones de Synopsys, para lograr una instalación limpia de Rocky Linux al usar el *script* existente de descarga e instalación de aplicaciones sin ningún problema.

Se espera que este proyecto pueda desarrollarse de forma exitosa desde enero de 2024 hasta noviembre de 2024. Sin embargo, es importante considerar que la migración a la nueva tecnología de 65 nm podría verse afectada por demoras en la recepción de las nuevas librerías de 65 nm de TSMC.

CAPÍTULO 6

Marco teórico

6.1. *Design flow* general para ASIC

Para lograr diseñar un ASIC (circuito integrado de aplicación específica) es necesario llevar a cabo una serie de pasos, lo cual es conocido como *design flow* o flujo de diseño. El flujo de diseño sirve como guía para los diseñadores de circuitos integrados, lo que permite el progreso tanto del diseño como de la fabricación de estos, para lograr las especificaciones deseadas sin errores. Esta serie de pasos básicamente se separa en dos partes principales, *front-end* y *back-end*, en donde la parte de *front-end* se enfoca en la parte de síntesis lógica y la de *back-end* en la síntesis física. En la Figura 11 se muestra un flujo de diseño general para diseñar un circuito integrado de aplicación específica [25].

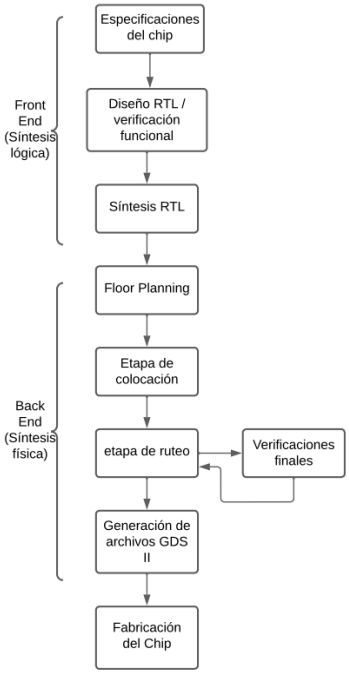


Figura 11: Flujo de diseño general para ASIC

6.2. Parte del flujo general *front-end*

La primera parte del flujo de diseño general presentado en la Figura 11 contiene los detalles y pasos necesarios para diseñar el funcionamiento específico del chip en lenguaje descriptivo de *hardware*. La explicación de estos pasos se muestra a continuación [25].

1. Especificaciones del chip:

   en este primer paso se definen las características, microarquitectura, funcionalidades (qué funciones debe cumplir) y especificaciones (potencia, área, tiempo, entre otras) del circuito integrado.

2. Diseño RTL y verificación funcional:

   se elabora el RTL *(register transfer level)*, lo cual consiste en la elaboración abstracta del circuito en un HDL *(hardware description language)*, el cual suele realizarse en Verilog, ya sea de forma *behavioral* o *structural*. Además, una vez que se haya realizado el código se procede a realizar simulaciones del comportamiento lógico del circuito.

3. Síntesis RTL:

   habiendo realizado simulaciones del código generado, se trabaja en la síntesis lógica, lo cual consiste en traducir el código RTL en una lista de compuertas lógicas, empleando herramientas de Synopsys. Esto se conoce como *netlist*.

6.3. Parte del flujo general *back-end*

Como se mencionó anteriormente, la parte del flujo *back-end* hace referencia específicamente a los pasos para llevar a cabo la etapa de la síntesis física, es decir, a la implementación física del circuito, en donde se transforma el *netlist* obtenido en el último paso de la síntesis lógica en un diseño físico *layout* compuesto por *frames* o celdas. La explicación de estos pasos se muestra a continuación [25].

1. *Floor planning*:

   este paso consiste en la conversión del *netlist* de compuertas lógicas obtenidas en el paso anterior a una representación a nivel silicio de los componentes que conforman el circuito *(layout)*. Además, se organiza el área rectangular que ocupa el chip con los elementos para obtener la menor cantidad de retardos por la interconexión entre módulos.

2. Etapa de colocación y ruteo:

   la etapa de colocación y ruteo consiste en establecer las celdas en una ubicación tomando en cuenta varios factores, como los requisitos de sincronización, las longitudes de las conexiones de las celdas, y la disipación de energía, para así interconectarlas.

3. Verificaciones finales:

   luego del enrutamiento el diseño de ASIC se somete a verificaciones físicas para comprobar si el circuito funciona como fue diseñado y así cumplir con la funcionalidad tanto lógica como eléctrica. Algunas de estas verificaciones son:

   - ***Design rule check* (DRC):** verifica que el diseño del layout cumpla con las restricciones impuestas por la tecnología, para que este no sea irrealizable y disfuncional.
   - ***Layout vs schematic* (LVS):** se verifica la funcionalidad del diseño comparando un *netlist* del *layout* con el *netlist* del esquemático obtenido en la síntesis lógica.
   - ***Parasitic extraction*:** se utiliza para modelar y analizar efectos electromagnéticos que suceden en presencia de elementos parásitos como capacitancias, inductancias y resistencias. Con esta herramienta se obtienen los parámetros eléctricos de los elementos del *layout* para verificar las características eléctricas del circuito.
   - ***Antenna rule check*:** previene efectos de antena, los cuales pueden dañar a los transistores durante los pasos de manufactura por la acumulación de excesos de carga en los alambres de metal que no están conectados a los canales PN de los transistores.
   - ***Electrical rule checking* (ERC):** verifica que las conexiones de VCC (voltaje) y VSS (tierra) sean correctas.

6.4. Flujo *front-end* propuesto por Synopsys

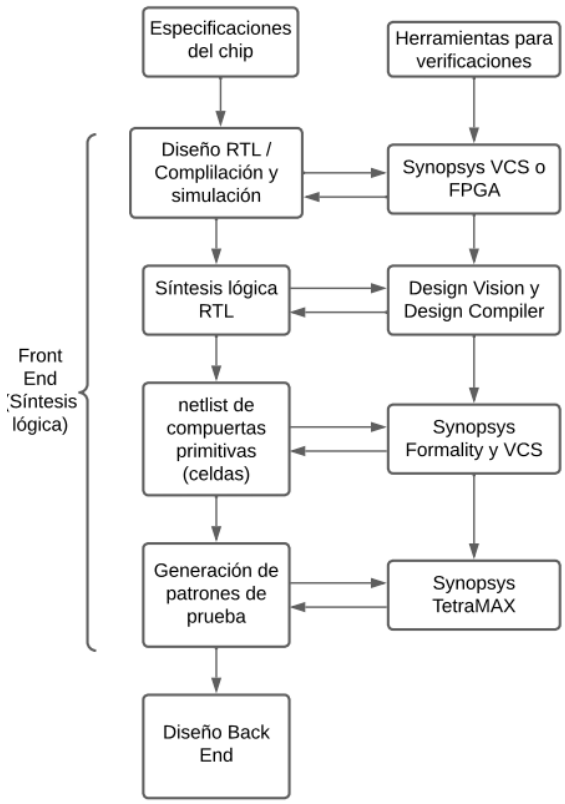


Figura 12: Flujo *front-end* propuesto por Synopsys

En la Figura 12 se muestra la etapa de *front-end* de un flujo alternativo propuesto por la empresa Synopsys para llevar a cabo el diseño de un ASIC (circuito integrado de aplicación específica). Además de los pasos, se muestran algunos nombres de herramientas de Synopsys que se explicarán posteriormente y que serán de utilidad para llevar a cabo cada uno de estos pasos.

6.5. Aplicaciones de Synopsys

6.5.1. Verilog Compiler Simulator (VCS)

Verilog Compiler Simulator es una herramienta de Synopsys desarrollada para analizar, compilar y simular circuitos elaborados en HDL *(hardware description language)* como Verilog, VHDL, HDL mixto, SystemVerilog, OpenVera y SystemC. Por esa razón es de gran importancia para acelerar la verificación completa del sistema al ofrecer la simulación Verilog, VHDL y HDL, ya que permite comprobar su funcionamiento antes y después de la elaboración de la síntesis física. Además, por medio de una interfaz gráfica permite observar el comportamiento del circuito para determinar si este funciona correctamente [26].

Para poder realizar simulaciones con VCS se debe contar con un archivo de *testbench* en formato de Verilog, en donde se describa el estado de las entradas del sistema en diferentes tiempos. Además, debe contener las líneas de código presentadas en el Cuadro 1 [15].

```
$fsdbDumpfile("file_name.fsdb");
$fsdbDumpvars(0, file_name_tb);
$dumpfile("file_name.vcd");
$dumpvars(0, file_name_tb);
```

Cuadro 1: Generación de simulación VCS

Con la primera línea de código presentado en el Cuadro 1 se genera un archivo .fsdb con el nombre file_name.fsdb, el cual almacenará información sobre la simulación. La siguiente línea indica al simulador que incluya las variables de la instancia 'file_name_tb' en el archivo .fsdb. Luego se genera un archivo file_name.vcd que va a contener el *waveform* de la simulación y por último se indica al simulador que incluya las variables de la instancia 'file_name_tb' en el archivo .vcd. Tanto el archivo .fsdb y.vcd son de gran importancia para el diseño del circuito integrado.

```
vcs -v file_name.v file_name_tb.v -o name_simulacion -full64 -debug_access+all
```

Cuadro 2: Comando para compilar y simular un diseño de *hardware* descrito en Verilog utilizando VCS

Por otro lado, en el Cuadro 2 se muestra el comando para generar un ejecutable de la simulación del archivo. Luego de haber ejecutado la línea de código mencionada anteriormente, se escribe en la línea de comandos ./name_simulacion para ejecutar la simulación y generar los archivos .fsdb y .vcd. A continuación se explican las partes del comando que se muestran en el Cuadro 2.

- **vcs:** este comando de la herramienta VCS se utiliza para compilar y simular diseños de *hardware* descritos en lenguajes como Verilog.
- **file_name.v:** este es el nombre del archivo Verilog que se desea compilar y simular. Es importante mencionar que se pueden colocar más archivos Verilog si forman parte del diseño.
- **file_name_tb.v:** este es el archivo de *testbench* asociado con el diseño, el cual contiene las pruebas que se ejecutarán en el diseño durante la simulación.
- **-o nombre_simulacion:** con este comando se especifica el nombre del archivo de salida de la simulación.
- **-full64:** esta opción indica que la compilación y la simulación se realizarán en un entorno de 64 bits.
- **-debug_access+all:** esta opción habilita el modo de depuración y acceso a todas las señales en el diseño durante la simulación.

6.5.2. Discovery Visual Environment (DVE)

DVE es una interfaz gráfica de usuario (GUI) interactiva que se utiliza para analizar y depurar diseños de SystemVerilog, VHDL, Verilog y SystemC. Por medio de esta interfaz gráfica se puede arrastrar y soltar señales en varias vistas o usar las opciones del menú para ver y comparar las señales que se desean analizar. Es importante mencionar que se debe configurar VCS para que funcione en DVE. Asimismo, deben utilizarse la misma versión de VCS y DVE para garantizar una depuración sin problemas de la simulación.

```
dve -full64
```

Cuadro 3: Comando para invocar DVE desde la terminal

en el Cuadro 3 se muestra el comando para invocar DVE desde la terminal. En la Figura 13 se muestra la interfaz gráfica que se genera al ejecutar el comando del Cuadro 3. Para visualizar el *waveform* de la simulación (archivo.vcd) que se produce al ejecutar el archivo de simulación creado con el comando vcs (véase Cuadro 2) se deben seguir los siguientes pasos:

1. Ir a *File/Open Database* (véase Figura 13).
2. Seleccionar el archivo ".vcd" que se desea visualizar de la carpeta del proyecto (véase Figura 14).
3. Luego se encontrará el nombre del modelo *testbench* en el *Hierarchy box* (véase Figura 15).
4. Dar clic derecho al modelo de *testbench* → clic izquierdo en *Add To Waves* → clic izquierdo en *New Wave View* (véase Figura 16).
5. Se generará una nueva ventana con el resultado del *waveform* de las *inputs* y *outputs* (véase Figura 17).

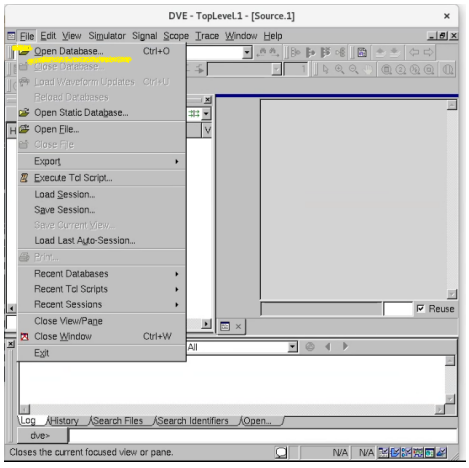


Figura 13: Interfaz gráfica DVE 1

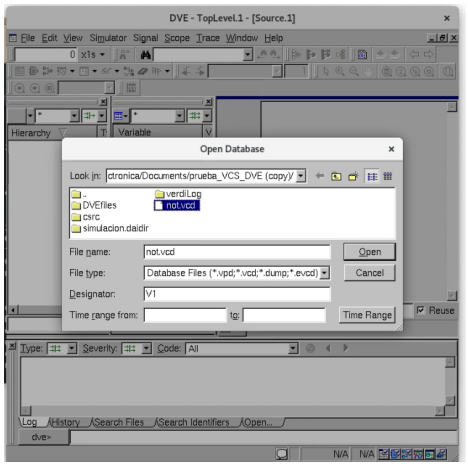


Figura 14: Interfaz gráfica DVE 2

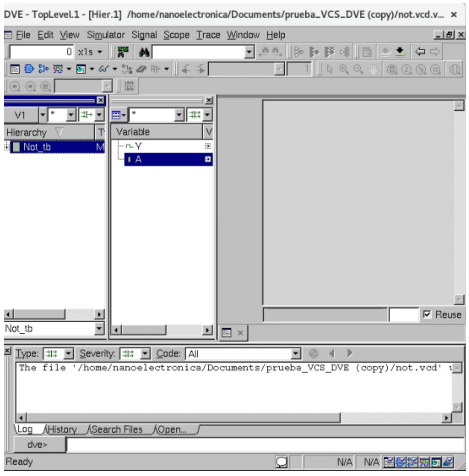


Figura 15: Interfaz gráfica DVE 3

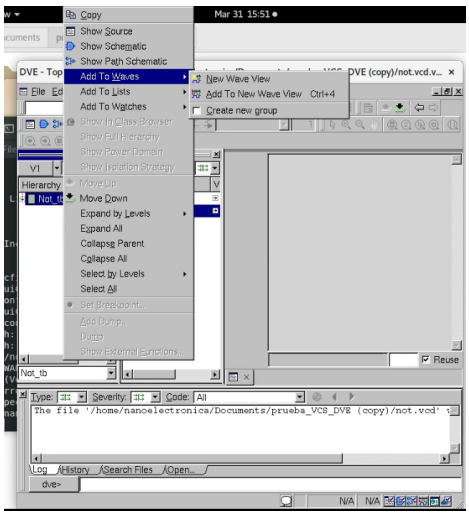


Figura 16: Interfaz gráfica DVE 4

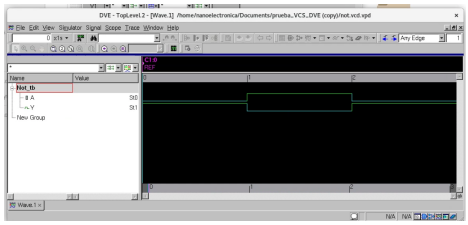


Figura 17: Visualización del *waveform*

6.5.3. Design Compiler

Design Compiler es otro *software* de Synopsys, el cual se encarga de optimizar los diseños para proporcionar la representación lógica de una función determinada. Lo que hace es tomar una descripción RTL *(register-transfer level)* de un diseño digital y lo traduce en una representación a nivel de compuertas.

Por medio de esta herramienta se logra realizar la síntesis lógica (lo que implica la traducción de la descripción del diseño a un conjunto de compuertas) y conexiones lógicas que implementan la misma funcionalidad, además admite una amplia gama de estilos de diseño, por lo que es altamente optimizable y puede aplicar diversas técnicas de optimización para mejorar aspectos como la velocidad, el área y el consumo de energía del diseño. Por esa razón es una de las herramientas más importantes para lograr llevar a cabo el flujo de diseño, principalmente la parte de *front-end*, ya que se utiliza en todos los diseños de circuitos integrados para generar la *netlist* lógica, la cual luego se utilizará en la etapa de *back-end* [27].

```
dc_shell
```

Cuadro 4: Comando para inicializar Design Compiler por línea de comando

```
dc_shell -gui
```

Cuadro 5: Comando para inicializar Design Compiler desde interfaz

```
dc_shell > read_file miarchivo.v
```

Cuadro 6: Ejemplo de CMD con el comando para cargar y leer archivos después de inicializar Design Compiler

6.5.4. Design Vision

Design Vision es una herramienta de visualización y análisis de *netlist* utilizada en el diseño lógico de circuitos integrados, por lo que es de gran utilidad para analizar un diseño a nivel de compuertas, multiplexores y *flip-flops*, por esa razón se utiliza para realizar tareas como la verificación de la *netlist* generada y la exploración de la topología del diseño [28].

Se puede utilizar por medio de una interfaz de texto, en donde se ingresan líneas de comando en la CMD, y también cuenta con una interfaz gráfica, con el cual se puede acceder

a un menú con los comandos de síntesis para el análisis visual de la *netlist*. Con esta herramienta se obtiene un esquemático con el circuito sintetizado, lo que es de gran utilidad para analizar factores como el tiempo y la lógica generada. En la Figura 18 se puede observar un ejemplo del esquemático obtenido del *logical core* sintetizado del circuito integrado El Gran Jaguar con la herramienta Design Vision [18].

```
design_vision
```

Cuadro 7: Comando para inicializar Design Vision por línea de comando

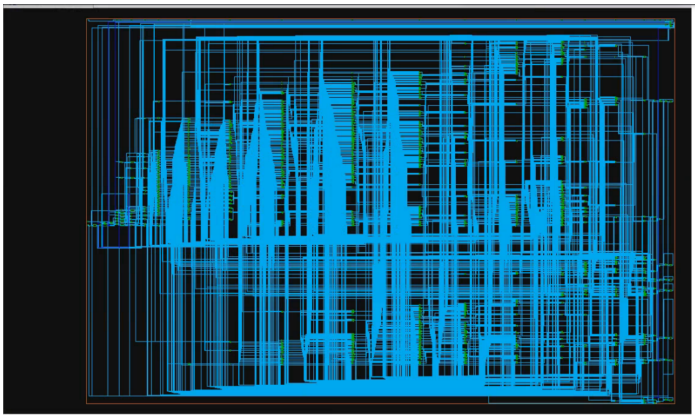


Figura 18: *Logical core* del circuito integrado El Gran Jaguar

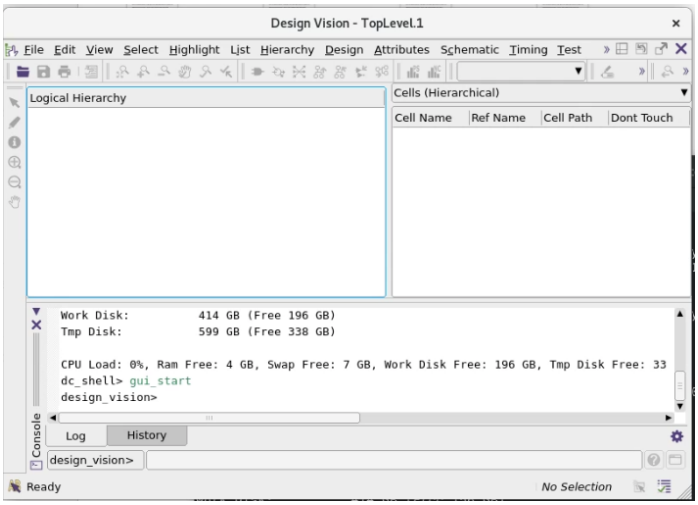


Figura 19: Interfaz gráfica de Design Vision

6.6. Protocolo Secure Shell (SSH)

SSH es un protocolo de red que permite a los usuarios acceder y controlar de manera segura una computadora o un servidor de forma remota. Este protocolo facilita la configuración de una autenticación segura mediante contraseñas y claves públicas, ofreciendo así

comunicaciones de datos encriptadas entre dos computadoras conectadas a través de una red abierta, como Internet.

SSH opera bajo un modelo cliente-servidor, lo que significa que la conexión se inicia cuando el cliente SSH se conecta al servidor SSH. El cliente es responsable de manejar el proceso de configuración de la conexión, utilizando criptografía de clave pública para confirmar la identidad del servidor. Una vez establecida la conexión, el protocolo SSH emplea algoritmos de cifrado para asegurar la privacidad e integridad de los datos intercambiados entre el cliente y el servidor [29].

6.6.1. Cifrado SSH con claves públicas y privadas

Una clave SSH es una credencial esencial para acceder de manera segura a sistemas remotos a través del protocolo Secure Shell (SSH). Está basada en tecnología de infraestructura de clave pública (PKI), considerada el estándar más alto para la autenticación y el cifrado de identidades digitales, lo que permite un método de autenticación seguro y escalable.

Se basa en el uso de dos claves asimétricas (una pública y una privada). La clave privada es confidencial y debe protegerse, mientras que la clave pública se comparte con los servidores SSH. Este método es ampliamente utilizado en servicios en la nube, herramientas de transferencia de archivos y otras aplicaciones críticas para verificar identidades y protegerse contra accesos no autorizados o ataques. Algunos de los algoritmos matemáticos más comunes utilizados para la generación de estas claves son Rivest Shamir Adleman (RSA) y el algoritmo de firma digital de curva elíptica (ECDSA) [30].

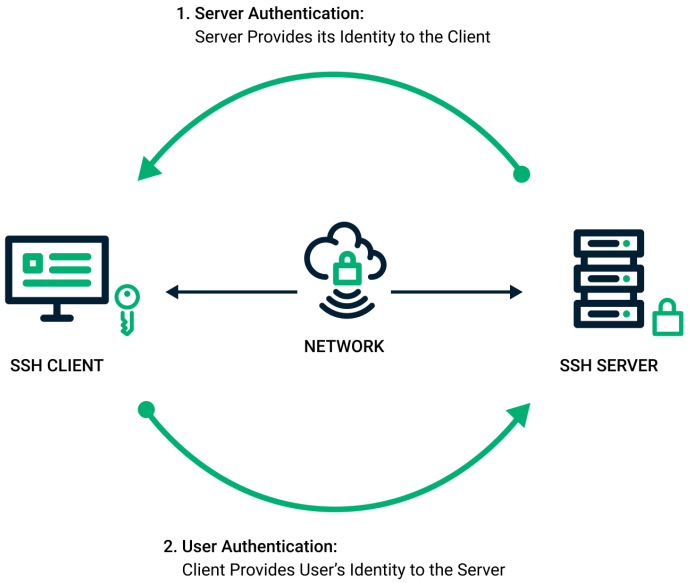


Figura 20: Funcionamiento de las claves SSH

6.7. SSH File Transfer Protocol (SFTP)

El protocolo de transferencia segura de archivos SFTP emplea el cifrado SSH para ofrecer un alto grado de seguridad en la transferencia de archivos. Esto asegura que tanto el envío como la recepción de archivos estén protegidos [31].

Algunas de las características del protocolo SFTP son las siguientes:

1. SFTP utiliza un único canal a través del puerto TCP 22 para transmitir datos, asegurando la protección criptográfica mediante SSH.
2. Admite la autenticación con claves públicas, que es más segura que la autenticación basada solo en contraseñas.
3. Es ideal para enviar archivos sensibles de manera segura a través de redes públicas como Internet.

6.8. Bourne Again Shell (Bash)

Para interactuar de manera más flexible con archivos del sistema operativo y realizar diversas acciones mediante la línea de comandos, se utiliza lo que se conoce como Shell. En Linux, uno de los intérpretes de comandos más comunes es el Bourne Again Shell, conocido como bash.

Este intérprete Unix permite la lectura y ejecución de múltiples comandos de manera secuencial, eliminando la necesidad de ejecutarlos manualmente uno por uno. En lugar de eso, es posible crear un archivo de *script* de bash para ejecutar esos comandos de forma automatizada [32].

6.9. *Script* Bash

Los *scripts* Bash se utilizan para automatizar la ejecución de secuencias de comandos. Una de sus principales ventajas es la flexibilidad que ofrecen en la organización de los comandos dentro de una secuencia, garantizando que cada vez que ejecutes el *script*, los comandos se ejecuten en el orden correcto.

Para que el shell reconozca que debe usar el intérprete Bash para ejecutar los comandos, se debe comenzar el *script* con la siguiente línea:

```
#!/bin/bash
```

Cuadro 8: Línea de encabezado para ejecutar *script* con Bash [32]

También es importante mencionar que, para ejecutar el *script*, hay dos métodos posibles:

1. Usar ./nombre_script.sh: para utilizar este método, primero se necesita conceder permisos de ejecución al *script* con el comando chmod +x nombre_script.sh.

2. Usar bash nombre_script.sh: este método no requiere permisos de ejecución previos para el *script*.

```
#!/bin/bash
echo "Hola, mundo"
```

Cuadro 9: Ejemplo básico de *script* Bash [32]

CAPÍTULO 7

Levantamiento de servidor compartido NFS

Uno de los objetivos de este trabajo fue la síntesis lógica de varios circuitos. Sin embargo, los demás integrantes del equipo, quienes estaban a cargo de la síntesis física y otras pruebas, necesitaban acceso a los archivos generados por mi parte del proyecto. Para facilitar la colaboración y compartir los archivos de manera eficiente, decidimos levantar un servidor NFS, lo que permitió que todos los miembros del equipo pudieran acceder a los archivos de manera centralizada y en tiempo real.

Un servidor NFS permite compartir archivos entre diferentes computadoras a través de una red, facilitando el acceso de varios usuarios a archivos y directorios.

Para ello, fue necesario realizar una serie de configuraciones, tanto en la máquina que se utilizó como servidor, como en las demás máquinas que se utilizaron para acceder a la carpeta compartida del servidor.

Es importante mencionar que se tuvo que elegir una máquina como servidor. En esta máquina se realizaron las configuraciones detalladas en la sección 7.1, mientras que las demás máquinas que necesitaban tener acceso a la carpeta compartida del servidor siguieron los pasos descritos en la sección 7.2. A continuación, se explican las configuraciones realizadas:

7.1. Pasos para levantar servidor NFS

1. **Creación de la carpeta a compartir:**

   el primer paso fue crear la carpeta en el servidor que se deseaba compartir. En nuestro caso, creamos una carpeta llamada compartir_archivos en home/nanoelectronica/Documents.

2. **Acceder como *root*:**

   luego, fue necesario abrir una terminal y obtener permisos de *root*, ya que todas las configuraciones que se muestran en los siguientes pasos deben realizarse con permisos de *root*. Para ello, se utiliza el comando:

   ```
   su -
   ```

   e ingresamos la contraseña de usuario.

3. **Instalación del paquete nfs-utils en usuario con permisos de *root*:**

   se instaló el paquete nfs-utils con el siguiente comando:

   ```
   dnf install nfs-utils A
   ```

4. **Acceso al archivo /etc/exports:**

   para configurar el servidor NFS, se accedió al archivo /etc/exports. En este archivo se deben especificar los directorios que serán exportados por el servidor NFS para que otros puedan montarlos y acceder a su contenido.

   Para acceder al archivo, se ingresó lo siguiente:

   ```
   nano /etc/exports
   ```

5. **Agregación del directorio compartido al archivo /etc/exports:**

   luego de haber abierto el archivo /etc/exports, se agregó el directorio a compartir de la siguiente manera:

```
/home/nanoelectronica/Documents/compartir_archivos/ *(rw,sync,no_root_squash)
```

   - Primero, se especifica el directorio a compartir. En este caso, /home/nanoelectronica/Documents/compartir_archivos/.
   - Luego, se coloca un espacio y la IP de las máquinas clientes que podrán acceder a la carpeta compartida. En nuestro caso, se utilizó el comodín * en lugar de especificar un rango de direcciones IP para mayor facilidad.
   - Finalmente, se definen los parámetros del directorio compartido: rw permite permisos de lectura y escritura a los clientes, sync asegura que los cambios se escriban en el disco de manera inmediata y no_root_squash permite que el usuario *root* en los clientes tenga acceso completo como *root*.

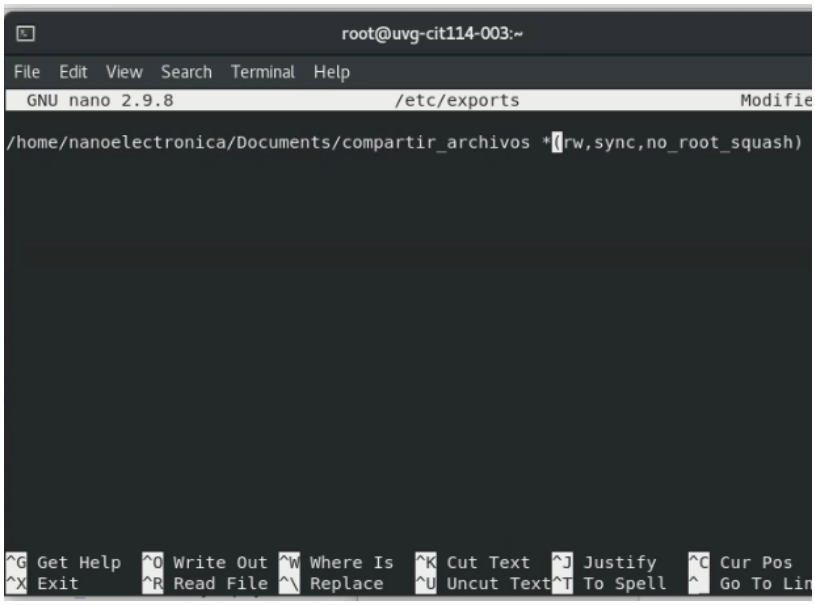


Figura 21: Agregación del directorio compartido al archivo /etc/exports

6. **Exportación del directorio compartido:**

   para aplicar los cambios realizados en el archivo /etc/exports, se ejecutó el comando:

   ```
   exportfs -a
   ```

   Esto permite que el sistema lea el archivo /etc/exports y exporte todos los directorios listados, asegurando que estén disponibles para los clientes.

7. **Reinicio del servidor NFS:**

   se reinició el servidor para asegurarse de que los cambios se apliquen, utilizando el comando:

   ```
   systemctl restart nfs-server
   ```

8. **Configuración de inicio automático del servidor NFS:**

   luego, se configuró el servicio NFS para que se inicie automáticamente al arrancar el sistema. Esto se realizó con el comando:

   ```
   systemctl enable –now nfs-server
   ```

9. **Verificación del estado del servicio NFS:**

   se verificó que el estado del servicio NFS estuviera activo (*active*) con el siguiente comando:

   ```
   systemctl status nfs-server
   ```

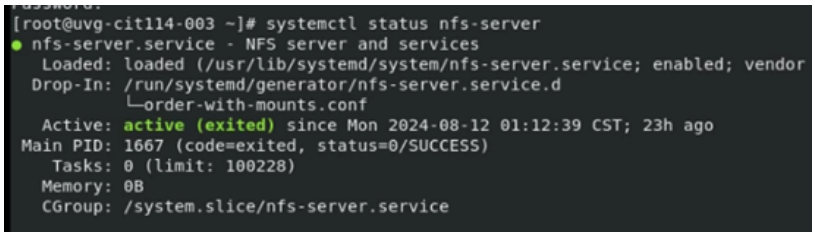


Figura 22: Verificación del estado del servicio NFS

Si el estado no aparece como *active*, se recomienda repetir los pasos anteriores.

10. **Verificación de la carpeta compartida:**

    para asegurarse de que la carpeta ingresada en el archivo /etc/exports esté compartida correctamente con las máquinas clientes, se ingresó el comando:

    ```
    exportfs -v
    ```

11. **Permisos de firewall:**

    finalmente, se verificó que el firewall de la universidad permitiera conexiones NFS con el siguiente comando:

    ```
    firewall-cmd –list-all
    ```

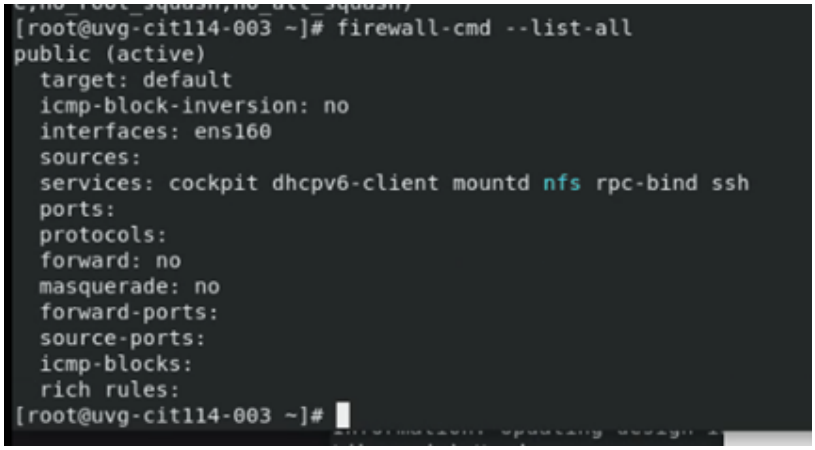


Figura 23: Verificación de permisos de firewall

Si en services no aparecen permisos para conexiones NFS, se deben ingresar los siguientes comandos:

```
firewall-cmd –add-service=nfs –permanent
firewall-cmd –add-service=rpc-bind –permanent
firewall-cmd –add-service=mountd –permanent
firewall-cmd –reload
```

7.2. Pasos para montar servidor compartido

Para montar la carpeta compartida desde otras computadoras, se deben seguir los siguientes pasos:

1. **Creación de la carpeta:**

   el primer paso fue crear la carpeta que se utilizará para montar la carpeta compartida. En nuestro caso, creamos una carpeta llamada archivos_compartidos en home/nanoelectronica/Documents.

2. **Acceder como *root*:**

   luego, fue necesario abrir una terminal y obtener permisos de *root*, ya que todas las configuraciones que se muestran en los siguientes pasos deben realizarse con permisos de *root*. Para ello, se utiliza el comando:

   ```
   su -
   ```

   e ingresamos la contraseña de usuario.

3. **Instalación del paquete nfs-utils:**

   se instaló el paquete nfs-utils con el siguiente comando:

   ```
   dnf install nfs-utils
   ```

4. **Montar la carpeta compartida:**

   existen dos formas de montar la carpeta compartida:

   - **Montar la carpeta manualmente:**
     este comando debe ejecutarse cada vez que la máquina se apague o reinicie para volver a montar la carpeta compartida:

   ```
   mount -t nfs IP_SERVIDOR:/home/nanoelectronica/Documents/compartir_archivos /home/nanoelectronica/Documents/archivos_compartidos

   Ejemplo:
   mount -t nfs 192.168.72.60:/home/nanoelectronica/Documents/compartir_archivos /home/nanoelectronica/Documents/archivos_compartidos
   ```

   - **Montar la carpeta automáticamente:**
     también se puede configurar el sistema para que monte la carpeta compartida del servidor automáticamente al arrancar el sistema.
     Para ello, se edita el archivo /etc/fstab con el siguiente comando:

     ```
     nano /etc/fstab
     ```

Luego, se agrega la línea correspondiente al archivo /etc/fstab para montar la carpeta automáticamente al inicio del sistema. El formato es el siguiente:

IP_SERVIDOR:directorio_Carpeta_Compartida_Servidor Directorio_Carpeta_Montada_Cliente nfs defaults 0 0

Ejemplo:
192.168.72.60:/home/nanoelectronica/Documents/compartir_archivos /home/nanoelectronica/Documents/archivos_compartidos nfs defaults 0 0

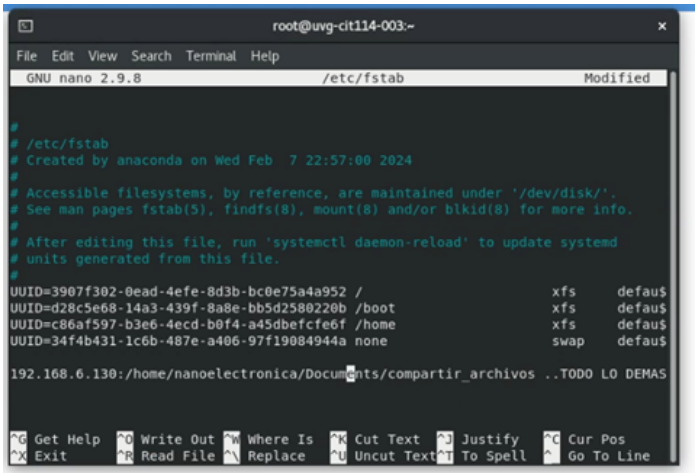


Figura 24: Configuración para montar la carpeta automáticamente

Se puede elegir cualquiera de las dos opciones para montar la carpeta compartida.

De esta forma, realizamos la configuración de un servidor y de las máquinas para poder acceder a la carpeta compartida del servidor, facilitando así la transferencia de archivos.

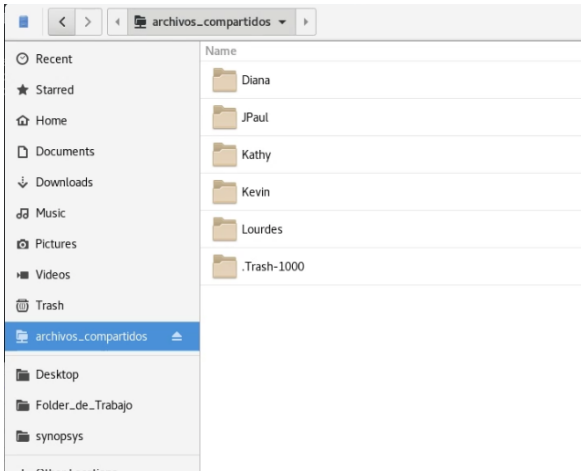


Figura 25: Carpeta montada en las máquinas clientes

CAPÍTULO 8

Identificación de librerías y archivos de TSMC de 180 nm

8.1. Librerías de 180 nm

Uno de los pasos importantes que se deben realizar antes de comenzar el proceso de síntesis lógica de un circuito es la identificación de las librerías de TSMC correspondientes a la tecnología de 180 nm, ya que en este trabajo se trabajará específicamente con las celdas de TSMC de dicha tecnología.

Al seleccionar las librerías, se deben tener en cuenta distintos aspectos, como el tipo de celdas que se necesitarán (para este proyecto bastan las celdas estándar y las celdas de pines de entrada y salida) y el nivel de voltaje adecuado para alimentar los circuitos (para la tecnología de 180 nm se recomienda utilizar un voltaje de 1.8 V). Esto es particularmente importante para el circuito del nanochip El Gran Jaguar, que se pretende mandar a fabricar.

A continuación, se explica cómo identificar y encontrar las librerías de TSMC de tecnología de 180 nm en los archivos proporcionados por TSMC.

1. ***Target library:***
   - TSMC > 180 > CMOS > G > stclib > 7-track > tcp018gbwp7t_290a_FE > tcb018gbwp7t_270a_nldm.tar.gz > tcb018gbwp7t_270a_nldm > TSMCHOME > digital > Front_End > timing_power_noise > NLDM > tcb018gbwp7t_270a > tcb018gbwp7ttc.db
2. ***Link library:***
   - TSMC > 180 > CMOS > G > stclib > 7-track > tcp018gbwp7t_290a_FE > tcb018gbwp7t_270a_nldm.tar.gz > tcb018gbwp7t_270a_nldm > TSMCHOME

> digital > Front_End > timing_power_noise > NLDM > tcb018gbwp7t_270a > tcb018gbwp7ttc.db

- TSMC > 180 > CMOS > G > stclib > 7-track > tcp018gbwp7t_290a_FE > tcb018gbwp7t_270a_nldm.tar.gz > tcb018gbwp7t_270a_nldm > TSMCHOME > digital > Front_End > timing_power_noise > NLDM > tcb018gbwp7t_270a > tcb018gbwp7twc.db
- TSMC > 180 > CMOS > G > stclib > 7-track > tcp018gbwp7t_290a_FE > tcb018gbwp7t_270a_nldm.tar.gz > tcb018gbwp7t_270a_nldm > TSMCHOME > digital > Front_End > timing_power_noise > NLDM > tcb018gbwp7t_270a > tcb018gbwp7tbc.db
- TSMC > 180 > CMOS > G > IO3.3V > iolib > LINEAR > tpd018nv_280a_-FE > tpd018nv_280a_nldm.tar.gz > tpd018nv_280a_nldm > TSMCHOME > digital > Front_End > timing_power_noise > NLDM > tpd018nv_280a_270a > tpd018nvtc.db

Luego de haber localizado las librerías que se emplearán, es necesario copiarlas y ubicarlas en un mismo directorio. Esto es importante porque, como se puede ver en los directorios mostrados anteriormente, algunos archivos se encuentran en diferentes ubicaciones. Para evitar problemas durante el proceso de síntesis, es fundamental que todos los archivos estén en un solo lugar.

Tanto los *scripts* que realizan la síntesis lógica como los que realizan la síntesis física hacen referencia a un único directorio donde están almacenadas las librerías. Si los archivos están dispersos, los *scripts* no podrán localizarlos correctamente, lo que podría provocar errores en el proceso de síntesis.

Por lo tanto, es esencial organizar las librerías en un solo directorio para asegurar que los *scripts* funcionen sin inconvenientes y puedan acceder a los archivos necesarios de manera correcta.

8.2. Archivos Verilog

Además de las librerías para la síntesis lógica y física, otros archivos cruciales que se deben localizar para la simulación de un circuito ya sintetizado son los archivos Verilog proporcionados por TSMC, que contienen las celdas estándar (compuertas, *flip-flops*, etc.) y las celdas de pines de entrada y salida.

1. **Archivos Verilog con celdas estándar de TSMC de 180 nm:**
   - TSMC > 180 > CMOS > G > stclib > 7-track > tcp018gbwp7t_290a_FE > tcb018gbwp7t_270a_vlg.tar.gz > tcb018gbwp7t_270a_vlg > TSMCHOME > digital > Front_End > verilog > tcb018gbwp7t_270a > tcb018gbwp7t.v
2. **Archivos Verilog con celdas de pines de entrada y salida de TSMC de 180 nm:**

- TSMC > 180 > CMOS > G > IO3.3V > iolib > LINEAR > tpd018nv_280a_FE > tpd018nv_260a_vlg.tar.gz > tpd018nv_260a_vlg > TSMCHOME > digital > Front_End > verilog > tpd018nv_260a > tpd018nv.v

8.3. Archivos PDF

También es esencial tener a disposición los archivos PDF que contienen la información técnica de las celdas estándar y las celdas de pines de entrada y salida. Estos documentos proporcionan detalles clave sobre la estructura, características eléctricas y parámetros de las celdas, lo cual es útil para analizar y comprender el funcionamiento de las mismas.

1. **PDF con información de las celdas estándar de TSMC de 180 nm:**
   - TSMC > 180 > CMOS > G > stclib > 7-track > tcp018gbwp7t_290a_FE > tcb018gbwp7t_270a_doc.tar.gz > tcb018gbwp7t_270a_doc > TSMCHOME > digital > Documentation > documents > tcb018gbwp7t_270a > name_pdfs*.pdf
2. **PDF con información de las celdas de pines de entrada y salida de TSMC de 180 nm:**
   - TSMC > 180 > CMOS > G > IO3.3V > iolib > LINEAR > tpd018nv_280a_FE > tpd018nv_280a_doc.tar.gz > tpd018nv_280a_doc > TSMCHOME > digital > Documentation > documents > tpd018nv_260a > name_pdfs*.pdf

Todos los archivos mencionados anteriormente ya se han organizado en sus respectivos directorios, siguiendo la jerarquía propuesta que se detalla en el capítulo 9. Se recomienda consultar dicho capítulo para comprender la estructura y el contenido de cada directorio.

CAPÍTULO 9

Jerarquía de directorios

En este capítulo se explica en detalle la jerarquía de directorios utilizada para realizar el proceso de síntesis lógica de manera organizada y eficiente. Esta estructura asegura que todos los archivos necesarios, como las librerías, *scripts* y resultados, estén correctamente ubicados, facilitando el acceso y manejo de cada archivo durante el proceso de síntesis.

En años anteriores, se ha logrado realizar la síntesis lógica y física de algunos circuitos, principalmente del nanochip El Gran Jaguar. Sin embargo, en el año 2023, el estudiante Noel Prado, al intentar utilizar el flujo de diseño que se había empleado en años anteriores con las librerías de diseño de Synopsys, se encontró con varias dificultades. Principalmente, esto se debía a la manera en que se referenciaban los archivos requeridos para realizar la síntesis [24].

Para solucionar ese problema, propuso una nueva jerarquía de directorios, en la cual todos los archivos necesarios para llevar a cabo ambas síntesis fueron colocados en un único directorio llamado "librerías". Además, creó una carpeta general que contenía un directorio para la síntesis lógica, otro para la síntesis física y otro llamado *"verilog_generation"*. Tanto los directorios de síntesis lógica como los de síntesis física fueron diseñados con el fin de no duplicar archivos, mientras que el directorio *"verilog_generation"* contenía los archivos necesarios para diseñar el nanochip. Esta parte de la jerarquía, en términos de funcionalidad, ya se había implementado en años anteriores, pero lo que se logró fue reducir el número de directorios para facilitar la comprensión.

Esto fue de gran importancia, ya que durante el proceso de ambas síntesis se generan varios archivos y, a su vez, se deben hacer referencias a las librerías en el momento de realizar la síntesis lógica o física. Por lo tanto, es crucial llevar a cabo todo de manera ordenada sin duplicar archivos, ya que tanto los archivos de las librerías como los archivos generados en la síntesis lógica y física suelen ocupar un espacio de memoria considerable.

Sin embargo, este diseño de directorios solo fue ideado para llevar a cabo el proceso de síntesis del nanochip. Así que, al realizar la síntesis lógica de otro circuito, los archivos de salida se reemplazaban por los del nuevo circuito. Esto era algo que se necesitaba mejorar, ya que en este trabajo se iban a sintetizar varios circuitos (incluyendo el nanochip) y se necesitaba que los archivos de salida no se reemplazaran al momento de sintetizar otro circuito. Además, algunos nombres de los directorios no eran muy entendibles, por lo que también se tenían que cambiar para esta nueva implementación.

A continuación, se explica cómo está distribuido el flujo general de directorios con los cambios realizados. Además, dado que el objetivo de este trabajo es realizar la síntesis lógica de varios circuitos, se ha creado una nueva jerarquía de directorios para la carpeta de síntesis lógica, la cual se muestra y explica más adelante.

9.1. Jerarquía de directorios general

En este diagrama se puede observar la jerarquía general de directorios que se utilizó. Es importante mencionar que solo se muestran los directorios y no los archivos, ya que en cada uno de ellos hay una gran cantidad de archivos.

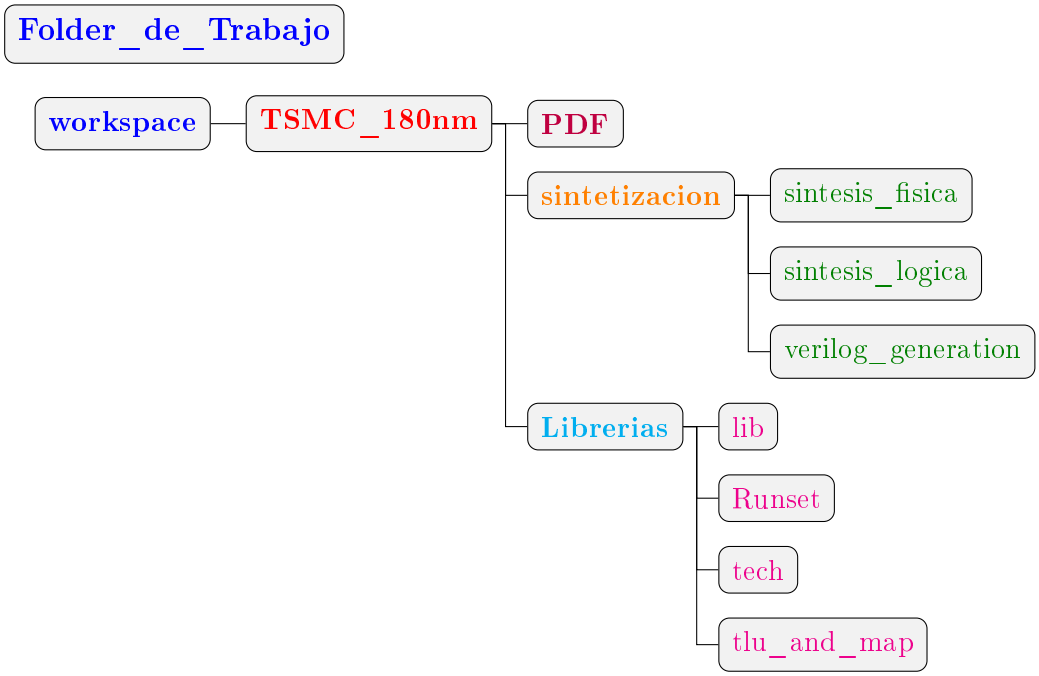


Figura 26: Jerarquía de directorios general

A continuación, se explica el contenido de cada directorio para una mejor comprensión:

- **Folder_de_Trabajo:** este directorio es el principal y contiene el directorio *"workspace"*. Se recomienda que, si se desea realizar otros tipos de pruebas diferentes a la sintetización de circuitos, se creen directorios dentro de este para mantener un espacio de trabajo organizado.

- ***workspace*:** el objetivo de este directorio es que aquí se creen los directorios para identificar la tecnología utilizada. Para este trabajo, solo se cuenta con el directorio "TSMC_180nm", pero aquí se podrían crear otros directorios que identifiquen la tecnología que se esté utilizando, como por ejemplo: "TSMC_90nm", "TSMC_65nm", etc.
- **TSMC_180nm:** este directorio contiene tres subdirectorios principales: PDF, sintetización y librerías. Se recomienda mantener esta estructura para cualquier directorio de otra tecnología, ya que facilita la identificación de los archivos importantes que se necesitan.
- **PDF:** en este directorio se encuentran los archivos PDF con información de las celdas de la tecnología TSMC de 180 nm. Incluye tanto los PDF de las celdas estándar de TSMC, que contienen información sobre compuertas, *flip-flops*, etc., como los PDF con información de los pines de entrada y salida. Es importante tener estos documentos en un solo lugar, ya que son útiles para analizar los resultados de la síntesis de un circuito.
- **sintetizacion:** este es el espacio principal para trabajar en la síntesis de los circuitos. Dentro de este, hay tres subdirectorios:
  - **sintesis_logica:** en este directorio se encuentra todo lo necesario para realizar la síntesis lógica de cualquier circuito, desde los archivos para la ejecución de la síntesis hasta los subdirectorios que se utilizan para organizar los archivos de entrada y salida generados durante el proceso. Esto se explica con más detalle posteriormente.
  - **sintesis_fisica:** este directorio contiene los archivos necesarios para la síntesis física de los circuitos, desde los archivos de ejecución hasta los subdirectorios correspondientes. Sin embargo, este trabajo se ha enfocado principalmente en la síntesis lógica, por lo que no se explica en detalle la jerarquía de directorios para la síntesis física. No obstante, la lógica de organización es la misma.
  - ***verilog_generation*:** este directorio contiene todos los archivos necesarios para diseñar el circuito del nanochip. Incluye el archivo Python que genera el archivo Verilog con el módulo del nanochip, así como otro archivo Python que genera el archivo Verilog con el módulo para los pines de entrada y salida del nanochip.
- **Librerias:** este directorio, específicamente en el subdirectorio "lib", contiene las librerías de 180 nm que se utilizan tanto para la síntesis lógica como para la física. En los demás subdirectorios se encuentran archivos (*runset*, ndm, etc.) que se utilizan específicamente para la síntesis física.

En la siguiente sección se explica en detalle la jerarquía de directorios utilizada para realizar la síntesis lógica.

9.2. Jerarquía de directorios para la síntesis lógica

En la sección anterior se mostró la jerarquía general de directorios que se empleó. Ahora se presenta la jerarquía específica del directorio de síntesis lógica.

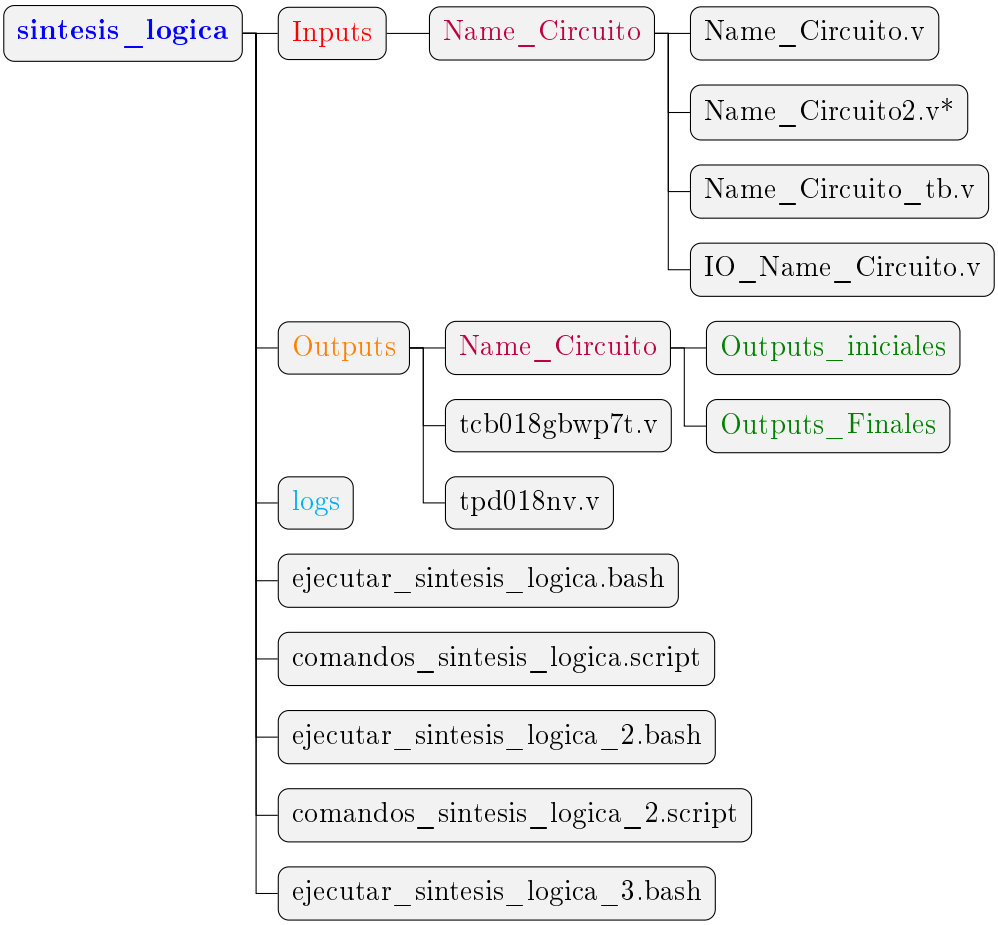


Figura 27: Diagrama de la jerarquía de directorios para la síntesis lógica

Esta nueva jerarquía, en términos de funcionalidad, sigue siendo la misma que la propuesta en años anteriores. Sin embargo, se eliminaron algunos directorios, se cambiaron ciertos nombres y se agregaron otros, dependiendo de los circuitos que se fueran a diseñar. A continuación, se explica cada directorio en detalle para facilitar su comprensión:

- ***inputs*:** en este directorio se crean manualmente los subdirectorios con el nombre del circuito principal a sintetizar. Para una correcta ejecución de la síntesis, dentro de cada subdirectorio deben colocarse algunos archivos importantes:
  - **Name_Circuito.v:** archivo Verilog que describe el circuito a sintetizar. Por ejemplo: not.v.
  - **Name_Circuito2.v:*** estos archivos están identificados con un asterisco (*) ya que pueden ser necesarios o no. Se deben incluir si el circuito a sintetizar contiene módulos separados. Por ejemplo, si se diseña un sumador de 8 bits con el nombre sumador_8bits.v y este se compone de otro módulo, como un sumador de 4 bits (sumador_4bits.v), ambos archivos deben estar en el directorio de sumador_8bits.
  - **Name_Circuito_tb.v:** *testbench* utilizado para simular el circuito.

- **IO_Name_Circuito.v:** archivo que contiene el módulo de los pines de entrada y salida para el circuito a sintetizar.

Es crucial que los archivos mencionados estén correctamente ubicados en el directorio Inputs/Name_Circuito, ya que, si falta alguno, el proceso de síntesis fallará. Esto se debe a que, para llevar a cabo la síntesis lógica, se utiliza un *script* llamado ejecutar_sintesis_logica_2.bash el cual sintetiza el circuito en dos fases: primero, el circuito original, y luego el circuito previamente sintetizado junto con el módulo que describe los pines de entrada y salida.

Sin embargo, para que este *script* funcione correctamente, es indispensable que todos los archivos estén ubicados en sus directorios correspondientes.

La importancia de realizar la síntesis en dos fases se explica en los capítulos 10 y 11, mientras que el funcionamiento del *script* se detalla en el capítulo 12.

- ***outputs*:** este directorio contiene los subdirectorios y archivos generados durante la síntesis lógica, además de dos archivos Verilog proporcionados por TSMC que hacen referencia a sus celdas.

  Al ejecutar el *script* ejecutar_sintesis_logica_2.bash, se crea un subdirectorio con el nombre del circuito en *outputs*, y dentro de este se generan automáticamente dos subdirectorios adicionales con sus respectivos archivos, los cuales son:

  - **Outputs_iniciales:** si el proceso de síntesis lógica se ejecuta correctamente, aquí se generan cuatro archivos: Name_Circuito_sintetizado.v (circuito original sintetizado), Name_Circuito_sintetizado_and_IO.v (circuito sintetizado unido con el módulo de pines de entrada y salida) y otros dos archivos con extensión .sdc y .ddc que corresponden al circuito sintetizado.
  - **Outputs_finales:** si el proceso de síntesis lógica finaliza correctamente, en este directorio se generan cuatro archivos: Name_Circuito_sintesis_final.ddc, Name_Circuito_sintesis_final.sdc, Name_Circuito_sintesis_final.v (netlist final del circuito sintetizado junto con el módulo de los pines de entrada y salida) y el archivo Name_Circuito_tb.v (*testbench* del circuito).

  Además, en esta carpeta de *outputs* se encuentran dos archivos de gran importancia, los cuales son:

  - **tcb018gbwp7t.v:** este archivo es necesario para la simulación del circuito sintetizado, ya que contiene las celdas estándar de TSMC con tecnología de 180 nm.
  - **tpd018nv.v:** este archivo contiene las celdas estándar de los pines de entrada y salida de TSMC con tecnología de 180 nm. Al igual que el archivo previamente mencionado, es crucial para la simulación del circuito sintetizado.

- ***logs*:** este directorio almacena los archivos de registro (*logs*) generados al ejecutar el *script* de síntesis lógica.

- **ejecutar_sintesis_logica.bash** este *script* es el que se realizó en años anteriores para el proceso de síntesis lógica de un circuito y debe ejecutarse en el directorio de sintesis_logica.

- **comandos_sintesis_logica.script:** este archivo contiene los comandos necesarios de Design Compiler, el cual se usa en ejecutar_sintesis_logica.bash.
- **ejecutar_sintesis_logica_2.bash** este *script* fue desarrollado en este trabajo para automatizar todo el proceso de síntesis lógica para un circuito tomando en cuenta los pines de I/O, el cual debe ejecutarse en el directorio de sintesis_logica.
- **comandos_sintesis_logica_2.script:** este archivo contiene los comandos necesarios de Design Compiler para ejecutar la síntesis de un circuito tomando en cuenta el módulo de pines de I/O.
- **ejecutar_sintesis_logica_3.bash** este *script* al igual que ejecutar_sintesis_logica_2.bash se encarga de automatizar todo el proceso de síntesis lógica para un circuito tomando en cuenta los pines de I/O, con la diferencia de que desde la terminal se puede ingresar el nombre de los circuitos a sintetizar.

CAPÍTULO 10

Flujo de diseño a seguir para la síntesis lógica

Para poder realizar el proceso de síntesis lógica en cualquier circuito descrito en Verilog, es necesario seguir los pasos de un flujo de diseño, con los cuales se podrá llegar a un *netlist* final en Verilog, el cual tendrá el circuito totalmente sintetizado lógicamente.

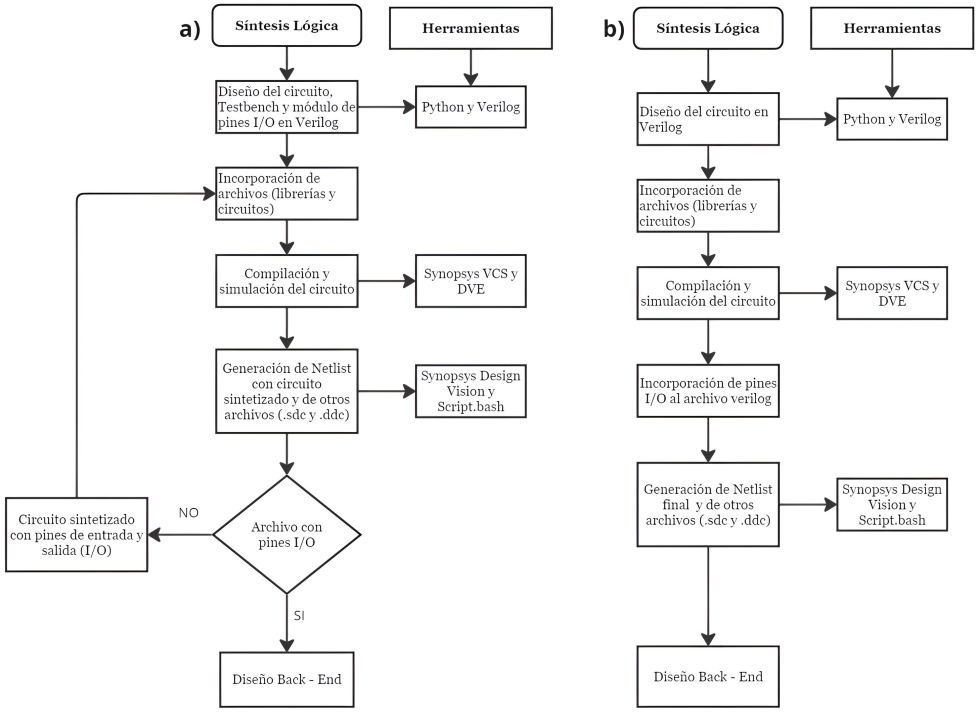


Figura 28: Flujo de diseño para la síntesis lógica. Flujo a) flujo propuesto en años anteriores modificado. b) flujo propuesto por el estudiante Noel Prado [24].

El flujo de diseño que se estará utilizando se muestra en la Figura 28 (a). El flujo a) fue establecido por los estudiantes que trabajaron previamente en este proyecto, con un pequeño cambio: ahora, como primer paso, se crean tanto el archivo Verilog del circuito como el archivo que contiene el módulo de pines de entrada y salida. Por otro lado, el flujo de diseño b) fue propuesto por el estudiante Noel Prado [24].

Lo que diferencia estos flujos es que en el a) se sigue el proceso de realizar la síntesis lógica dos veces. Primero, se sintetiza el circuito original; luego, en el archivo Verilog generado que tiene el circuito sintetizado, se le agrega el módulo de pines de I/O para el circuito, y se vuelve a sintetizar para obtener el *netlist* final.

En cambio, en el flujo de diseño b), el paso de agregar el módulo de los pines I/O después de haber sintetizado el circuito original ya no se realiza. Ahora, ese módulo con pines I/O se agrega en el mismo archivo que tiene el módulo del circuito original sin sintetizar. Esto se hace para evitar ejecutar la síntesis lógica dos veces como se hacía anteriormente.

Sin embargo, para esta implementación se utilizará el flujo de diseño a). Esto se debe a que, si bien utilizando el flujo de diseño b) solamente se requiere hacer una síntesis uniendo el módulo del circuito original con el módulo de pines I/O, y el proceso es más rápido, este método puede funcionar solamente en circuitos simples, como lo es una compuerta *not*. Esto ocurre por los siguientes motivos:

- Se pueden identificar de forma fácil los errores que pudieron haber ocurrido en el proceso de síntesis en circuitos sencillos.
- Se reduce la cantidad de archivos Verilog generados en el proceso de síntesis.

Pero para circuitos más complejos, como por ejemplo un divisor o multiplicador de 32 bits, podría no ser recomendable el flujo de diseño b) por estas razones:

- Si surgen problemas en el proceso de síntesis, puede ser más difícil identificar su origen debido a la falta de separación entre las etapas de síntesis.
- No permite un diseño de síntesis robusto y optimizado, ya que se estaría esperando que tanto la síntesis del circuito original como la del módulo de pines I/O se realicen exitosamente al mismo tiempo, lo cual podría no suceder.

En cambio, con el flujo de diseño a), a pesar de que el proceso de síntesis se realiza dos veces y toma un poco más de tiempo para ejecutarse, se tienen las siguientes ventajas:

- Al sintetizar primero el circuito original, se permite que el sintetizador aplique optimizaciones en el diseño lógico sin las complejidades adicionales del módulo de pines I/O, facilitando el proceso de síntesis.
- La separación de etapas facilita la identificación y corrección de problemas en el circuito original antes de agregar la lógica de pines de entrada y salida.

- Puede implementarse perfectamente para la síntesis de circuitos más complejos, ya que si el proceso de síntesis llegara a fallar, se obtienen informes tanto del circuito original sintetizado como del circuito sintetizado junto con los pines de I/O, lo que ayudaría a identificar el problema.

Por esa razón, se decidió emplear el flujo de diseño a). Con este flujo se verifica que el proceso de síntesis se haga correctamente y se pueden identificar de forma más fácil los posibles problemas que puedan ocurrir en el proceso.

Sin embargo, para que el proceso de síntesis no se tuviera que ejecutar dos veces de forma manual, se desarrollaron nuevos *scripts*, con el objetivo de que, al ejecutar un *script* solo una vez, se realice el proceso descrito en el flujo b) de forma automática. Esto se explica con mayor detalle en el capítulo 12.

CAPÍTULO 11

Diseño de circuitos en HDL

En este capítulo se explica el proceso de diseño de circuitos en Verilog, enfocándose en tres aspectos principales: el diseño del módulo que describe el comportamiento lógico del circuito, el diseño del *testbench* para la simulación tanto del circuito original como el sintetizado, y el diseño del módulo de pines de entrada y salida. Estos diseños de módulos son cruciales para el proceso de síntesis del circuito, el cual se realizará utilizando la tecnología de 180 nm de TSMC.

A lo largo del capítulo, se abordarán aspectos cruciales que se consideran para la creación e integración de estos módulos, con el objetivo de lograr una implementación eficiente y adecuada para su posterior síntesis lógica.

Es importante recalcar que, para la implementación realizada en este trabajo, el diseño del módulo de pines de entrada y salida también debe llevarse a cabo en primer lugar. Como se mencionó en el capítulo 10, el proceso de síntesis lógica se realizará mediante un *script* que automatiza todo el flujo de diseño descrito en dicho capítulo. Por lo tanto, es necesario tener creado previamente el archivo que contiene el módulo de pines de entrada y salida antes de la primera síntesis. La importancia de este módulo se explica posteriormente.

Para ilustrar el proceso, se tomará como ejemplo el diseño de un circuito *not*, lo cual permitirá mostrar de manera clara cómo se construyen tanto el módulo lógico del circuito, el *testbench* y el módulo de pines de entrada y salida. El proceso de síntesis de este mismo circuito se realizará en el capítulo 12.

11.1. Diseño del módulo de un circuito en Verilog

Como se observó en el flujo de diseño presentado en el capítulo anterior, el primer paso para llevar a cabo la síntesis lógica es la elaboración de un circuito utilizando un lenguaje descriptivo de *hardware*. En este caso, se empleó el lenguaje Verilog.

A continuación, se muestra el diseño de una compuerta *not*, que se realizó a nivel *behavioral*.

```
module Not(A,Y);
    input A;
    output Y;
    assign Y = ~A;
endmodule
```

Cuadro 10: Archivo not.v, diseño de una *not* en Verilog

Después de diseñar el circuito, es fundamental revisarlo para asegurarse de que no haya errores en la declaración de las entradas y salidas.

El diseño de un circuito puede llevarse a cabo tanto a nivel estructural como a nivel *behavioral*. Dependiendo del nivel de diseño elegido, la estructura del circuito sintetizado puede ser sencilla o más compleja. Esto se debe a que, en este caso, se utilizan librerías que contienen celdas de TSMC. Si durante el proceso de síntesis la aplicación Design Compiler no encuentra alguna configuración (por ejemplo, una compuerta *and, xor* o *nor* de tres entradas), el *software* reemplaza dicha configuración con una conexión lógica similar, pero más compleja.

Sin embargo, en el capítulo 13 se demuestra que es posible sintetizar circuitos descritos tanto a nivel *behavioral* como a nivel estructural. Como se mencionó anteriormente, si el *software* encuentra una configuración que no está disponible, la reemplaza por configuraciones que están disponibles y que cumplen con el mismo comportamiento lógico.

11.2. Diseño del *testbench* para simulación de un circuito

Para verificar que el circuito se haya diseñado correctamente, es necesario crear un archivo Verilog de *testbench*. Este archivo se utilizará para comprobar que el comportamiento del circuito, antes de la síntesis, sea el adecuado.

En el siguiente cuadro se muestra el *testbench* diseñado para simular la *not*.

```
module Not_tb;
wire Y;
reg A;
Not compuerta(.A(A), .Y(Y));

initial
begin
    $fsdbDumpfile("not.fsdb");
```

```
        $fsdbDumpvars(0, Not_tb);
        $dumpfile("not.vcd");
        $dumpvars(0, Not_tb);
        A = 1'b0;
        #1;
        A = 1'b1;
        #1;
        A = 1'b0;
        #1;
        A = 1'b1;

    end
    endmodule
```

Cuadro 11: Archivo not_tb.v, diseño del *testbench* para simulación de una *not* en Verilog

Al diseñar un *testbench* para circuitos en Verilog, hay algunos puntos importantes a considerar para asegurarse de que el circuito funcione correctamente:

- **Definir claramente las señales de prueba:**
  - ***inputs*:** hay que usar registros (*reg*) para simular las señales de entrada, ya que los registros permiten cambiar los valores durante la simulación.
  - ***outputs*:** hay que usar cables (*wire*) para las señales de salida, ya que estas no necesitan ser modificadas directamente.
- **Inicialización adecuada:** se tiene que inicializar todas las señales relevantes al inicio de la simulación para evitar estados indefinidos.
- **Simulación de tiempo:** se tiene que usar retrasos (#) para simular el tiempo de propagación y dar tiempo para reaccionar a los cambios de las entradas.
- **Generación de archivos de simulación:** se tiene que usar $dumpfile y $dumpvars para generar los archivos (VCD o FSDB). Esto es fundamental para poder analizar los resultados visualmente y depurar cualquier problema en la simulación.
  - En este ejemplo, $fsdbDumpfile("not.fsdb") y $fsdbDumpvars(0, Not_tb) son para FSDB.
  - $dumpfile("not.vcd") y $dumpvars(0, Not_tb) son para VCD.

Para simular el circuito, se emplea un simulador de Synopsys denominado DVE o WaveView, cuyo uso se explica en el siguiente capítulo.

11.3. Diseño del módulo de pines de entrada y salida de un circuito en Verilog

Otro aspecto que debe considerarse antes de sintetizar un circuito es el diseño de su módulo de pines de entrada y salida.

El diseño de este módulo es de gran importancia, ya que, aunque el circuito puede sintetizarse sin la implementación de este módulo, su relevancia se hace evidente en la implementación física. Este módulo permite la interacción entre el circuito interno del chip y el exterior a través de los pines de entrada y salida. Dado que se trabaja con circuitos nanométricos, es crucial contar con pines de entrada y salida que faciliten esta interacción con el circuito desde el exterior.

Para realizar el diseño de este módulo, es necesario definir la librería que se utilizará, en este caso, una de TSMC de 180 nm. Esta librería ya fue definida por los estudiantes que trabajaron en este proyecto, y en el capítulo 8 se explicó cómo ubicarla. Ellos establecieron las configuraciones necesarias para lograr una implementación exitosa en la tecnología de 180 nm. A continuación, se detalla la celda elegida y las configuraciones que se establecieron.

11.3.1. Selección y configuración del pin de entrada y salida

La celda seleccionada es la PDDW0204SCDG. El módulo correspondiente a esta celda se encuentra en el archivo Verilog llamado tpd018nv.v, y la información relacionada puede consultarse en los documentos PDF proporcionados por TSMC o en la carpeta titulada "PDF"(ver capítulo 9 para su ubicación).

Esta celda representa un pin diseñado por TSMC, que puede funcionar como entrada o salida, dependiendo de la configuración aplicada a sus entradas.

Las Figuras 29 y 30 muestran información sobre el esquema y la tabla de verdad del pin PDDW0204SCDG (esta información está disponible en su respectivo PDF). Esta tabla es de gran importancia, ya que permite definir el pin como entrada o salida, según las configuraciones seleccionadas.

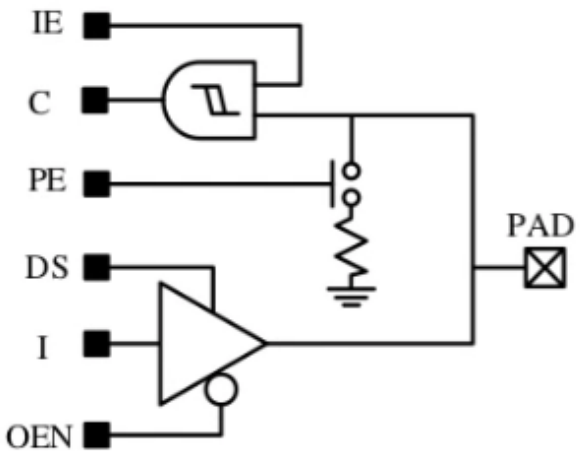


Figura 29: Esquemático del circuito del pin PDDW0204SCDG

Truth Table

INPUT						OUTPUT	
DS	OEN	I	PAD	PE	IE	PAD	C
0/1	0	0	-	0/1	0	0	0
0/1	0	0	-	0/1	1	0	0
0/1	0	1	-	0/1	0	1	0
0/1	0	1	-	0/1	1	1	1
0/1	1	0/1	0	0/1	0	-	0
0/1	1	0/1	0	0/1	1	-	0
0/1	1	0/1	1	0/1	0	-	0
0/1	1	0/1	1	0/1	1	-	1
0/1	1	0/1	Z	0	0	-	0
0/1	1	0/1	Z	0	1	-	X
0/1	1	0/1	Z	1	0	L	0
0/1	1	0/1	Z	1	1	L	L

Figura 30: Tabla de verdad de pin PDDW0204SCDG

En la Figura 31 y Cuadro 12, se puede observar que el módulo del pin, así como sus entradas y salidas, siguen el orden de (I, DS, OEN, PAD, C, PE, IE). Este es el orden que debe seguirse para configurar correctamente el pin, ya sea como entrada o salida.

```
`celldefine
module PDDW0204SCDG (I,DS,OEN,PAD,C,PE,IE);
   input I,DS,OEN,PE,IE;
   inout PAD;
   output C;
   wire  MG;
   parameter PullTime = 100000;
   reg lastPAD, pull_uen, pull_den,PS;
initial begin
  pull_uen = 0;
  pull_den = 0;
  PS = 0;
end
  bufif1 (weak0, weak1)(PAD_i, 1'b1, pull_uen);
  bufif1 (weak0, weak1)(PAD_i, 1'b0, pull_den);
  buf    (C, CO);
  and    (CO, C_buf, IE);
  nand   (PUEN, PS, PE);
  not    (PU, PUEN);
```

Figura 31: Módulo del pin PDDW0204SCDG definido en el archivo tpd018nv.v

```
module PDDW0204SCDG (I,DS,OEN,PAD,C,PE,IE);
    input I,DS,OEN,PE,IE;
    inout PAD;
```

```
   output C;
   wire MG;
   parameter PullTime = 100000;
   reg lastPAD, pull_uen, pull_den,PS;
initial begin
  pull_uen = 0;
  pull_den = 0;
  PS = 0;
end
  bufif1 (weak0, weak1)(PAD_i, 1'b1, pull_uen);
  bufif1 (weak0, weak1)(PAD_i, 1'b0, pull_den);
  buf (C, CO);
  and (CO, C_buf, IE);
  nand (PUEN, PS, PE);
  not (PU, PUEN);
  not (PSB, PS);
  and (PD, PE, PSB);
  bufif0 (PAD_q, I, OEN);
  ...
```

Cuadro 12: Módulo del pin PDDW0204SCDG definido en el archivo tpd018nv.v

Configuración de pin como entrada

Para configurar el pin como entrada, se debe utilizar PAD como entrada y C como salida. Tal como se observa en la Figura 29 del esquemático, PAD actuará como la entrada desde el exterior del pin, mientras que C funcionará como la salida del pin que estará conectada hacia el circuito. La representación de esto se puede ver en la Figura 32.

Figura 32: Representación de pin PDDW0204SCDG configurado como entrada

El objetivo es que, al ingresar un uno lógico al pin (en PAD), este se transmita como un uno lógico hacia el circuito (a través de C). De igual manera, al ingresar un cero lógico en el pin, este debe ser transmitido como cero hacia el circuito.

Tomando esto en cuenta, se procedió a analizar los casos listados en la tabla de verdad. En la Figura 30, se observa que los casos que se comportan como se desea están encerrados en el recuadro celeste. En la fila 6, tanto PAD como C son 0, y en la fila 8 son 1, lo cual coincide con el comportamiento deseado. Mientras tanto, la configuración de las demás entradas (DS, OEN, I, PE, IE) se mantiene igual en ambos casos.

Por lo tanto, para llevar a cabo la configuración del pin como entrada, se debe hacer de la siguiente manera:

```
PDDW0204SCDG U1(.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A), .C(A_w), .PE(1'b0), .IE
    (1'b1));
```

Cuadro 13: Configuración del pin PDDW0204SCDG como entrada

- **I:** según la tabla de verdad y los casos seleccionados, se le puede asignar un 1 o un 0. En este caso, se le asignó un 0.
- **DS:** también se le puede asignar un 1 o un 0. En este caso, se le asignó un 0.
- **OEN:** se debe asignar un valor de 1.
- **PAD:** a PAD se le debe asignar una entrada (en este caso, A), ya que este será el enlace hacia el mundo exterior.
- **C:** se debe asignar un *wire* (en este caso, A_w), que se conectará a la entrada del circuito a través de C.
- **PE:** se le puede asignar un 1 o un 0. En este caso, se le asignó un 0.
- **IE:** se le debe asignar un 1.

Cabe mencionar que, si el circuito está conformado por más entradas, el proceso a seguir es el mismo.

Configuración de pin como salida

Para configurar el pin como salida, se debe seguir la misma lógica que se utilizó para configurar el pin como entrada. Sin embargo, como se puede observar en el circuito mostrado en la Figura 29, se debe utilizar la terminal I como entrada y PAD como salida. En este caso, I (la entrada del pin) se comportará como la salida lógica del circuito, mientras que PAD actuará como salida hacia el mundo exterior.

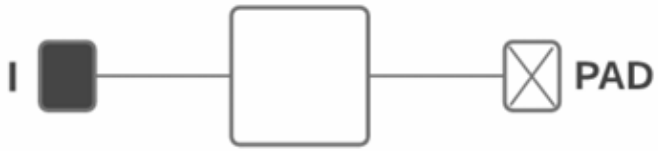


Figura 33: Representación de pin PDDW0204SCDG configurado como salida

El objetivo es que, cuando se ingrese un uno lógico al pin (en I), este se transmita como un uno lógico hacia el mundo exterior (a través de PAD). De igual manera, al ingresar un cero lógico en el pin, este debe transmitirse como cero hacia el mundo exterior.

Tomando esto en cuenta, se procedió a analizar nuevamente los casos listados en la tabla de verdad. En la Figura 30, se observa que los casos que se comportan de la manera deseada están encerrados en un recuadro verde, ya que en la fila 2 tanto I como PAD son 0, y en la fila 4 ambos son 1, lo que coincide con lo que se busca. Además, en este caso, la salida C no se tendrá en cuenta, ya que no se utiliza en la configuración del pin como salida, por lo que se considera como *"don't care"*. Esto permite que se puedan utilizar incluso las configuraciones de las filas 1 y 3, ya que tanto I como PAD presentan el comportamiento necesario. Mientras tanto, la configuración de las demás entradas (DS, OEN, PE, IE) permanece igual para ambos casos.

Por lo tanto, para llevar a cabo la configuración del pin como salida, se debe hacer de la siguiente manera:

```
PDDW0204SCDG U1(.I(Y_w), .DS(1'b0), .OEN(1'b0), .PAD(Y), .C(1'bx), .PE(1'b0), .IE
    (1'b1));

//o tambien se puede

PDDW0204SCDG U1(.I(Y_w), .DS(1'b0), .OEN(1'b0), .PAD(Y), .C(1'bx), .PE(1'b0), .IE
    (1'b0));
```

Cuadro 14: Configuración del pin PDDW0204SCDG como salida

- **I:** a I se le debe asignar un *wire* (en este caso Y_w), ya que por medio de esta terminal el circuito le asignará una señal a la entrada de ese pin (ya sea 0 o 1).
- **DS:** se le puede asignar ya sea un 1 o 0. En este caso, se le asignó un 0.
- **OEN:** se le debe asignar un valor de 0.
- **PAD:** a PAD se le debe asignar un *output* (en este caso Y), ya que este será la salida hacia el mundo exterior.
- **C:** se configura como *don't care* (x), ya que no se utiliza.
- **PE:** se le puede asignar ya sea un 1 o 0. En este caso, se le asignó un 0.
- **IE:** se le puede asignar un 0 o un 1 (según los casos seleccionados).

Si el circuito está conformado por más salidas, el proceso que se realiza es el mismo.

11.3.2. Declaración de pines de entrada y salida

Una vez especificado detalladamente qué pin se eligió de las celdas de TSMC y cómo se configura, se puede proceder a crear el módulo en Verilog.

Para declarar este módulo, es esencial visualizar cómo se conectaría físicamente. Es decir, se deben declarar tanto las entradas y salidas del circuito como los *wires*, que son los cables que se conectarán al exterior del circuito. En la Figura 34 se muestra una imagen representativa de la conexión entre los pines y el exterior del circuito.

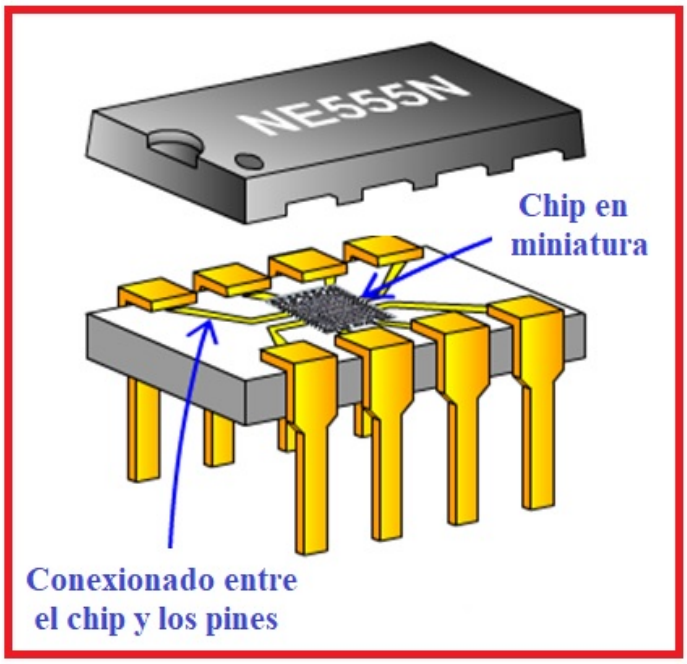


Figura 34: Circuito representativo de la conexión entre pines de entrada y salida con el exterior de un circuito

En el Cuadro 14 se muestra la parte inicial del módulo de pines de entrada y salida para la *not*. Se encuentran las configuraciones mencionadas anteriormente, es decir, las entradas, salidas y los *wires*.

En la última línea se realiza la instancia hacia el circuito; cabe mencionar que dicha instancia debe dirigirse al circuito que ha sido sintetizado previamente. Esta instancia se lleva a cabo primero indicando el nombre del módulo que se sintetizó previamente, seguido de un nombre representativo que hará referencia al mismo. Este recibe como entrada y salida los *wires*, ya que, como se mencionó anteriormente, es a través de estos *wires* que se realiza la conexión entre los pines de entrada y salida con el exterior del circuito.

```
module Not_IO(A, Y);
    input A;
    output Y;
    wire A_w, Y_w;
    Not Compuerta(A_w,Y_w);
```

Cuadro 15: Declaración de pines de entrada, salida y *wires* para una *not*

Luego de ello, es necesario completar la configuración de los pines, los cuales ya se explicaron anteriormente. En este caso, como la *not* está conformada solo por una entrada y una salida, solo se configura un pin de entrada y uno de salida. En el Cuadro 15 se muestran estas configuraciones.

```
    //Entradas:
    PDDW0204SCDG U10(.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A), .C(A_w), .PE(1'
        b0), .IE(1'b1));
    //Salidas:
```

```
    PDDW0204SCDG U11(.I(Y_w), .DS(1'b0), .OEN(1'b0), .PAD(Y), .C(1'bx), .PE(1'
        b0), .IE(1'b0));
```

Cuadro 16: Configuración de los pines de entrada y salida para una *not*

De esta manera, el diseño final del módulo queda como se muestra en el Cuadro 16. Es importante destacar que, como se observa en las últimas líneas del Cuadro 16, se realizaron dos configuraciones finales, que son las de PVDD1CDG PVDD() y PVSS1CDG PVSS(). Estas son de gran importancia para la síntesis física, ya que por medio de estas se crea la instancia a los pads de alimentación para el circuito. Existe una terminal para VDD y una para VSS; estas pertenecen a la misma librería de los pines de entrada y salida configurados anteriormente. El pin de voltaje se encuentra como PVDD1CDG y el de tierra como PVSS1CDG.

```
module Not_IO(A, Y);
    input A;
    output Y;
    wire A_w, Y_w;
    Not Compuerta(A_w,Y_w);

    //Entradas:
    PDDW0204SCDG U10(.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A), .C(A_w), .PE(1'
        b0), .IE(1'b1));
    //Salidas:
    PDDW0204SCDG U11(.I(Y_w), .DS(1'b0), .OEN(1'b0), .PAD(Y), .C(1'bx), .PE(1'
        b0), .IE(1'b0));

    //Pines de VDD y VSS
    PVDD1CDG PVDD();
    PVSS1CDG PVSS();
endmodule
```

Cuadro 17: Archivo not_IO.v: módulo final de pines de entrada y salida para una *not*

CAPÍTULO 12

Configuración y utilización de herramientas para simulación y sintetización de un circuito

En este capítulo, se llevará a cabo el proceso de simulación de un circuito descrito en Verilog utilizando las aplicaciones de Synopsys llamadas VCS y DVE, con el fin de entender cómo se emplean estas herramientas para simular el comportamiento de un circuito original y sintetizado. Además, se mostrará cómo utilizar las aplicaciones Design Compiler y Design Vision, las cuales son herramientas fundamentales para realizar la síntesis lógica de cualquier circuito. Esto se puede hacer mediante la interfaz gráfica o a través de comandos.

En la sección 12.2 se explica el proceso de síntesis mediante la interfaz gráfica de Design Vision. En la sección 12.3.1 se explicarán los *scripts* utilizados en años anteriores para la síntesis lógica y en la sección 12.3.2 se mostrará su funcionamiento.

Por otro lado, en la sección 12.3.3 se presentarán los nuevos *scripts* que fueron desarrollados, los cuales permiten la síntesis de cualquier circuito (tomando en cuenta el módulo de pines de entrada y salida) de forma ordenada, siguiendo la jerarquía de directorios para la síntesis lógica explicada en el capítulo 9 y el diagrama del flujo de diseño detallado en el capítulo 10, y se mostrará su funcionamiento en la sección 12.3.4.

Para entender mejor el uso y configuración de las herramientas y archivos, en este capítulo se simulará y sintetizará la compuerta *not* diseñada en el capítulo anterior. Sin embargo, el objetivo principal de este capítulo es aprender el uso y manejo de las herramientas de simulación y síntesis lógica.

En el capítulo 13 se explica con mayor detalle cómo realizar la síntesis lógica para cualquier circuito descrito en Verilog.

12.1. Uso de VCS y DVE para simulación

Después de diseñar un circuito en Verilog, es necesario simularlo para verificar que su funcionamiento sea el esperado. Para ello, se utilizan dos herramientas de Synopsys: VCS y DVE.

12.1.1. Uso de VCS

La primera herramienta a utilizar es VCS. Verilog Compiler Simulator (VCS) es una herramienta de Synopsys que se utiliza para analizar, compilar y simular circuitos desarrollados en lenguajes de descripción de *hardware* (HDL) como Verilog. Si el circuito contiene errores de diseño, VCS los detectará, y no se generará el simulador hasta que el diseño esté correcto.

VCS se utiliza mediante el siguiente comando:

```
vcs -v file_name.v file_name_tb.v -o name_simulacion -full64 -debug_access+all
```

Cuadro 18: Comando para compilar y simular un diseño de *hardware* descrito en Verilog utilizando VCS

- **vcs:** este comando de la herramienta VCS (Synopsys VCS), se utiliza para compilar y simular diseños de *hardware* descritos en lenguajes como Verilog.
- **file_name.v:** este es un archivo Verilog el cual se desea compilar y simular. Es importante mencionar que se pueden colocar más archivos Verilog si forman parte del diseño.
- **file_name_tb.v:** este es el archivo de *testbench* asociado con el diseño, el cual contiene las pruebas que se ejecutarán en el diseño durante la simulación.
- **-o nombre_simulacion:** con este comando se especifica el nombre del archivo de salida de la simulación.
- **-full64:** esta opción indica que la compilación y la simulación se realizarán en un entorno de 64 bits.
- **-debug_access+all:** esta opción habilita el modo de depuración y acceso a todas las señales en el diseño durante la simulación.

Por lo que, para compilar y simular el circuito *not* diseñado en el capítulo anterior, es necesario ejecutar el siguiente comando en una terminal abierta en el directorio que contiene el archivo Verilog del circuito y el *testbench*:

```
vcs -v not.v not_tb.v -o simulacion -full64 -debug_access+all
```

Cuadro 19: Comando para compilar y simular el circuito *not*

Después de ello, si no hay ningún error en el diseño, se generará un archivo simulador con el nombre que se le haya asignado, como se muestra en la Figura 35. Este archivo debe ejecutarse para generar los archivos .fsdb y .vcd (como se muestra en la Figura 36), siendo el archivo .vcd el que se utilizará para verificar el comportamiento del circuito. El comando para ello es el siguiente:

```
./nombre_simulacion

//en este caso

./simulacion
```

Cuadro 20: Comando para ejecutar la simulación

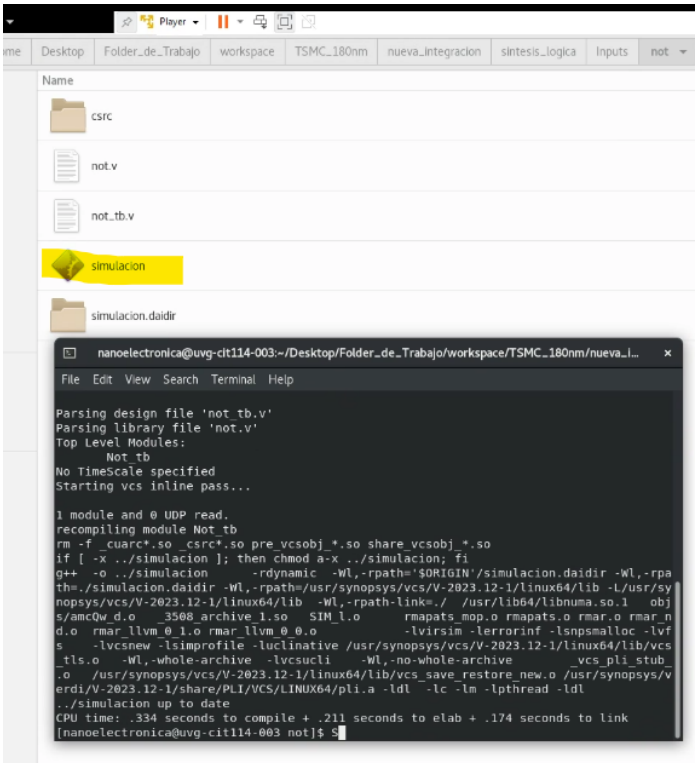


Figura 35: Generación de archivo simulador

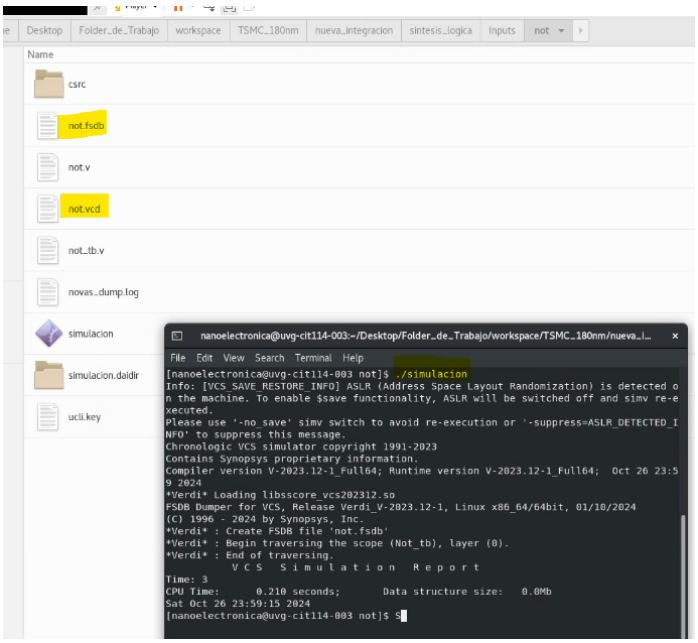


Figura 36: Ejecución de simulador

12.1.2. Uso de DVE

Una vez generados los archivos .fsdb y .vcd, se procede a utilizar la herramienta DVE, que sirve para verificar el comportamiento del circuito, empleando el archivo .vcd para la simulación.

Discovery Visual Environment (DVE) es una interfaz gráfica de usuario (GUI) interactiva que se utiliza para analizar y depurar los diseños de circuitos. Para abrir la interfaz gráfica, se utiliza el siguiente comando:

```
dve -full64
```

Cuadro 21: Comando para abrir la interfaz gráfica de DVE

Este comando puede ejecutarse en cualquier directorio, pero se recomienda abrir DVE en el directorio donde se encuentra el circuito que se desea verificar, para facilitar la localización del archivo .vcd.

Para visualizar el *waveform* de la simulación (archivo .vcd) a través de la interfaz gráfica de DVE, se deben seguir los siguientes pasos:

1. Ir a *File/Open Database* (véase Figura 37).

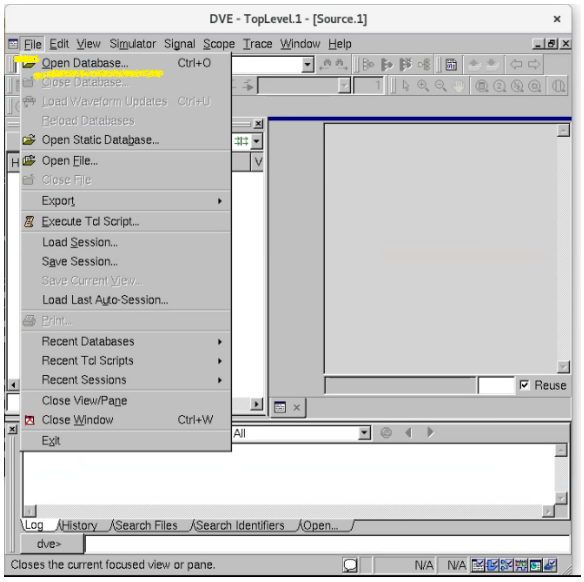


Figura 37: Interfaz gráfica DVE 1

2. Seleccionar el archivo ".vcd" que se desea visualizar de la carpeta del proyecto (véase Figura 38).

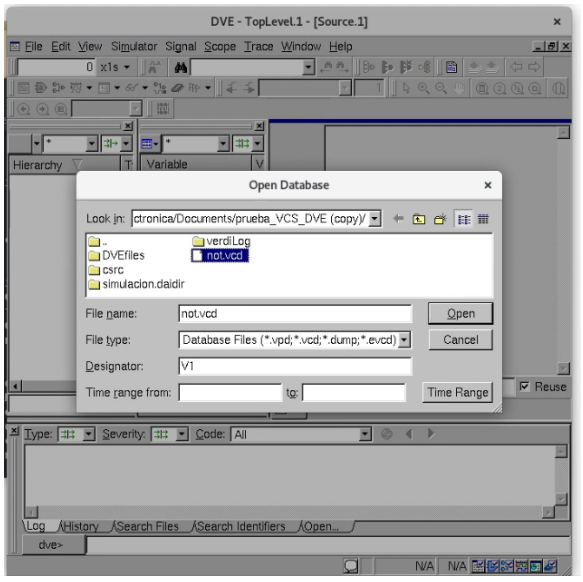


Figura 38: Interfaz gráfica DVE 2

3. Luego se encontrará el nombre del modelo *testbench* en *Hierarchy box* (véase Figura 39).

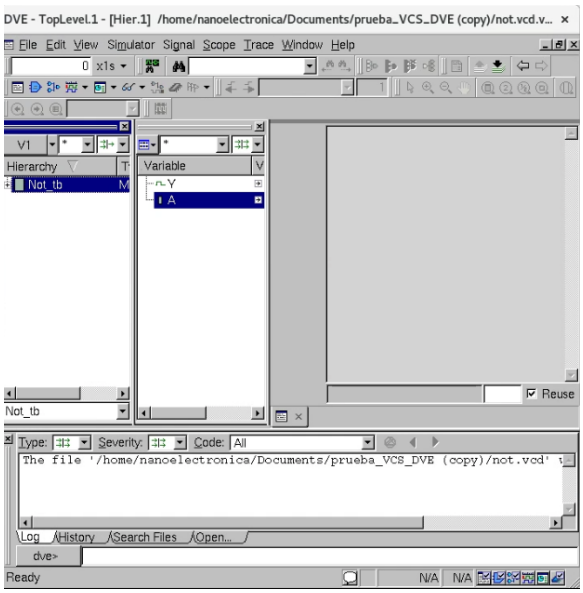


Figura 39: Interfaz gráfica DVE 3

4. Dar clic derecho al modelo de *testbench* → clic izquierdo en *Add to Waves* → clic izquierdo en *New Wave View* (véase Figura 40).

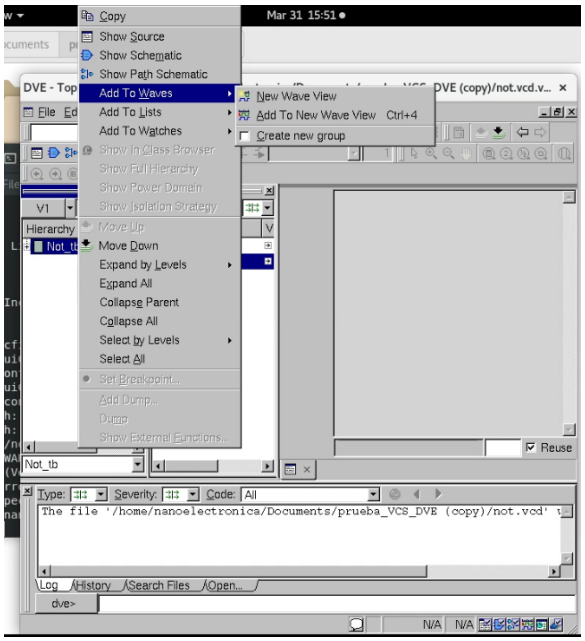


Figura 40: Interfaz gráfica DVE 4

5. Se generará una nueva ventana con el resultado del *waveforms* de los *inputs* y *outputs* (véase Figura 41).

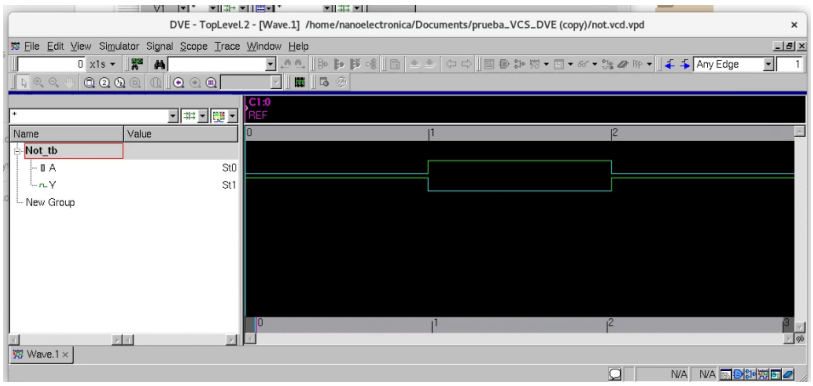


Figura 41: Visualización del *waveform* del circuito *not* realizado

De esta manera, se logra la visualización del comportamiento del circuito. Cabe mencionar que los pasos de simulación son los mismos para simular y visualizar el comportamiento de un circuito ya sintetizado; sin embargo, al ejecutar el comando de VCS, es necesario hacer referencia a los archivos Verilog de TSMC que contienen las celdas (tcb018gbwp7t.v y tpd018nv.v). Esto se abordará con mayor detalle en el capítulo 13.

12.2. Uso de Design Vision con la interfaz gráfica

Para utilizar la herramienta Design Vision, es necesario ejecutar el siguiente comando en cualquier terminal:

```
design_vision
```

Cuadro 22: Comando para abrir la interfaz gráfica de Design Vision

Sin embargo, se recomienda ejecutar este comando dentro de la carpeta sintesis_logica/outputs/name_circuito para mantener un correcto orden de los archivos generados durante el proceso.

Después de ejecutar el comando mostrado anteriormente, debería aparecer una ventana como la que se muestra en la Figura 42.

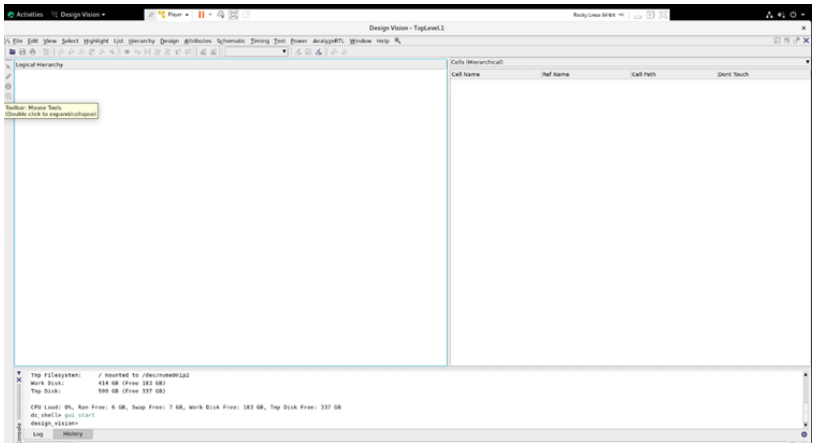


Figura 42: Interfaz gráfica de Design Vision

A continuación, se detallan los pasos a seguir para comenzar la síntesis de un circuito, en este caso, el circuito *not* diseñado en el capítulo anterior.

12.2.1. Importación de librerías

En el capítulo 8 se explicó cómo identificar las librerías de TSMC que se utilizarán, y en el capítulo 9 se detalló su ubicación dentro de la propuesta de jerarquía de directorios. Esto es fundamental, ya que para sintetizar un circuito es necesario hacer referencia a estas librerías; por lo tanto, es importante tenerlas localizadas en un único directorio.

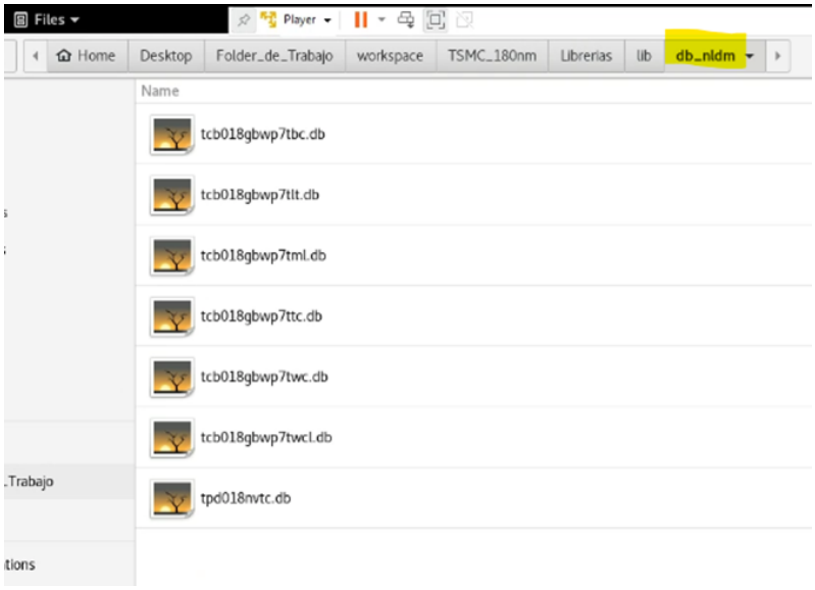


Figura 43: Ubicación de librerías agrupadas en un mismo directorio

Para instanciar las librerías, se deben seguir los siguientes pasos:

1. Ir a la pestaña *File* y hacer clic en *Setup*.

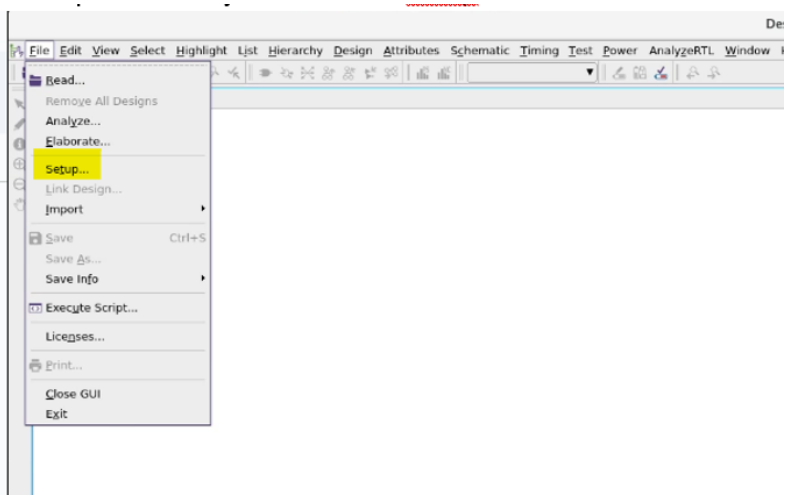


Figura 44: Configuración de librerías

2. En la ventana de variables de aplicación que se abrió, buscar las siguientes variables en el filtro:

   - search_path:

     en esta variable se debe especificar el directorio donde se encuentran ubicadas nuestras librerías. Todas las librerías a utilizar deben estar en el mismo directorio; de lo contrario, la herramienta no será capaz de localizar los archivos.

     Si se utiliza la jerarquía de directorios propuesta, las librerías deberían ubicarse en el directorio /home/nanoelectronica/Desktop/Folder_de_Trabajo/workspace/TSMC_180nm/Librerias/lib/db_nldm. Este directorio debe colocarse después del parámetro "ver".

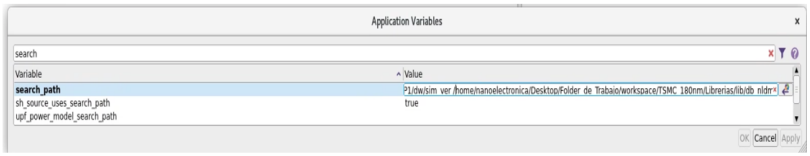


Figura 45: Variable search_path

   - target_library:

     en esta variable se debe especificar nuestra librería objetivo. En nuestro caso, esta librería es tcb018gbwp7ttc.db.

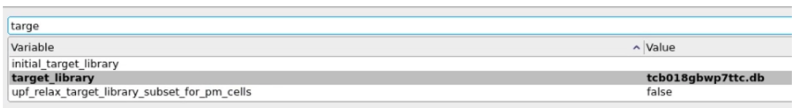


Figura 46: Variable target_library

- link_library:
  en esta variable se deben especificar las librerías que se utilizarán como enlace. En nuestro caso, las librerías son: tcb018gbwp7ttc.db, tcb018gbwp7twc.db, tcb018gbwp7tbc.db y tpd018nvtc.db.

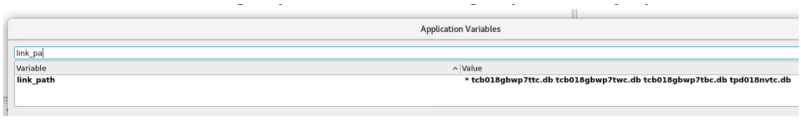


Figura 47: Variable link_library

Como se puede observar, este proceso es bastante largo, y si se comete algún error, la referencia a las librerías no se realizará correctamente. Por esta razón, una alternativa sería ejecutar los siguientes comandos en la terminal de Design Vision o en la misma terminal donde se ejecutó la herramienta:

```
# Importacion de librerias

lappend search_path "path donde estan ubicadas las librerias"
set link_library " * tcb018gbwp7ttc.db tcb018gbwp7twc.db tcb018gbwp7tbc.db
    tpd018nvtc.db"
set target_library "tcb018gbwp7ttc.db"
```

Cuadro 23: Importación de librerías

12.2.2. Importación de archivos

El siguiente paso consiste en importar el archivo que contiene la descripción del circuito en lenguaje HDL (en este caso, en Verilog). Para ello, se deben seguir los siguientes pasos:

1. Ir a la pestaña *File* y hacer clic en *Analyze*.

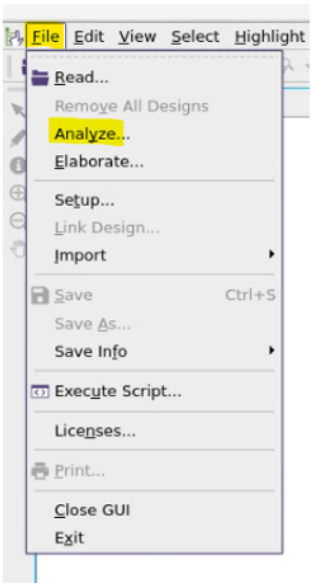


Figura 48: Importación de archivos 1

2. Luego, se abrirá una ventana similar a la mostrada en la Figura 49. En esta ventana, se debe hacer clic en el botón resaltado para acceder a otra ventana, como la que se observa en la Figura 50. A continuación, se debe presionar el botón *Add* para acceder al menú de búsqueda, lo que permitirá localizar el archivo que se desea cargar (véase la Figura 51). Por último, se debe hacer clic en el botón *OK* (véanse la Figura 52 y la Figura 53).

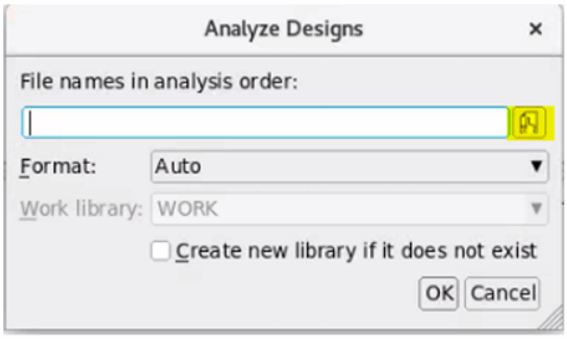


Figura 49: Importación de archivos 2

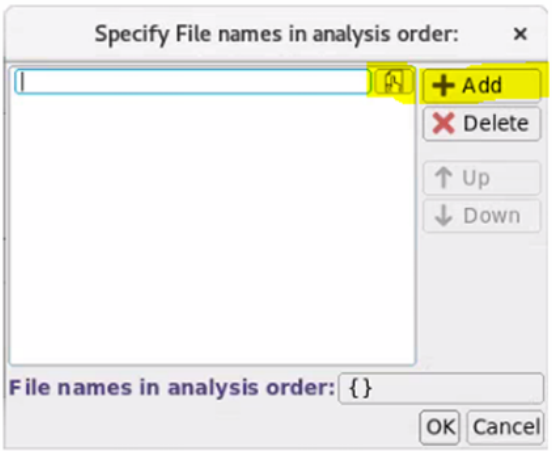


Figura 50: Importación de archivos 3

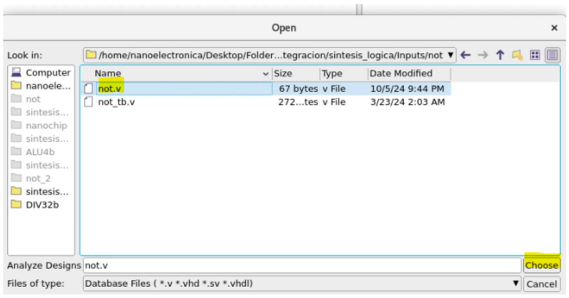


Figura 51: Importación de archivos 4

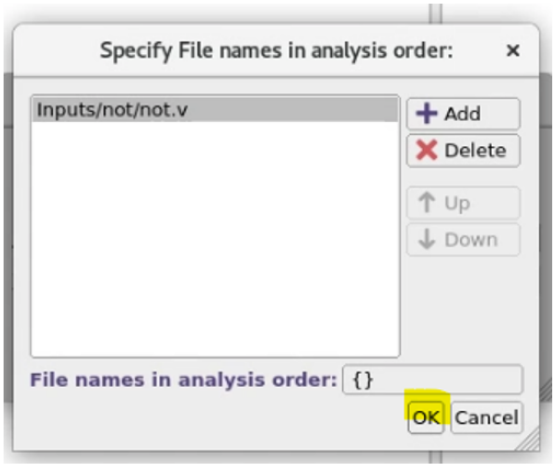


Figura 52: Importación de archivos 5

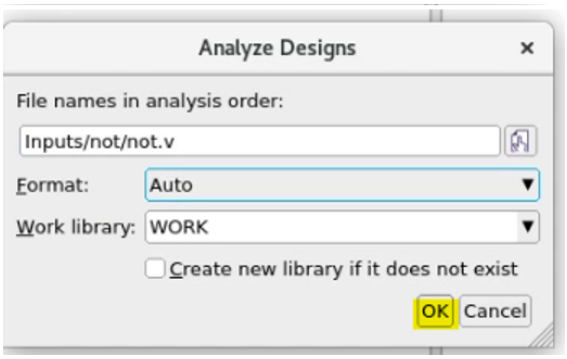


Figura 53: Importación de archivos 6

Si este paso se realizó correctamente, no debería aparecer ningún mensaje de error en la terminal de la herramienta, como se muestra en la Figura 54.

```
Compiling source file ./Inputs/not/not.v
Presto compilation completed successfully.
Loading db file '/home/nanoelectronica/Desktop/Folder_de_Trabajo/workspace/TSMC_180nm/Librerias/lib/db_nldm/tcb018gbwp7ttc.db'
Loading db file '/home/nanoelectronica/Desktop/Folder_de_Trabajo/workspace/TSMC_180nm/Librerias/lib/db_nldm/tcb018gbwp7twc.db'
Loading db file '/home/nanoelectronica/Desktop/Folder_de_Trabajo/workspace/TSMC_180nm/Librerias/lib/db_nldm/tcb018gbwp7tbc.db'
Loading db file '/home/nanoelectronica/Desktop/Folder_de_Trabajo/workspace/TSMC
design_vision>
```

Figura 54: Importación de archivos 7

3. Luego, se debe volver a la pestaña *File* y hacer clic en la opción *Elaborate*.

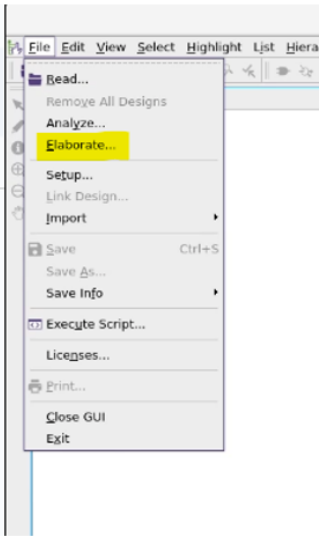


Figura 55: Importación de archivos 8

4. Aparecerá una ventana similar a la que se muestra en la Figura 56, en la cual se debe seleccionar la librería `work` y, a continuación, elegir el diseño con el nombre que se haya definido previamente para el módulo. Finalmente, se debe hacer clic en el botón *OK* para cargar el diseño en la herramienta.

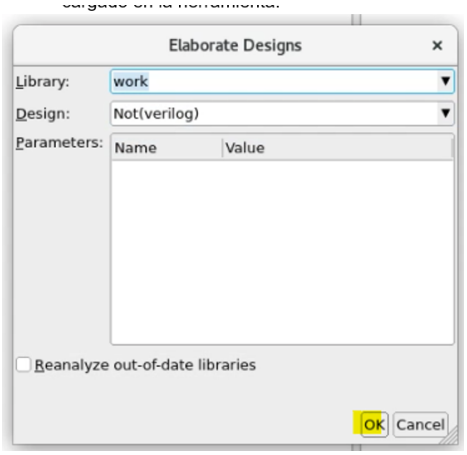


Figura 56: Importación de archivos 9

Si aparece el error mostrado en la Figura 57, se recomienda colocar manualmente el nombre del módulo del circuito utilizando el comando indicado en la Figura 58. Este error al intentar elaborar un diseño puede estar relacionado con un error en la versión de la herramienta utilizada al intentar elaborar el diseño.

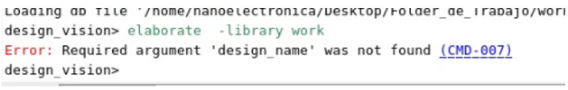


Figura 57: Importación de archivos 10

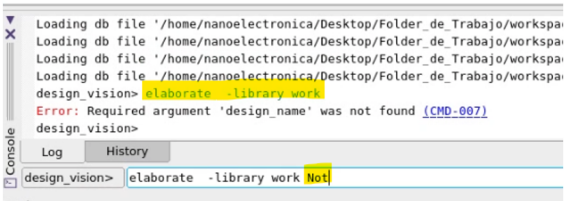


Figura 58: Importación de archivos 11

5. Si los pasos anteriores se realizaron correctamente, el circuito debería aparecer cargado en la herramienta, tal como se muestra en la Figura 59.

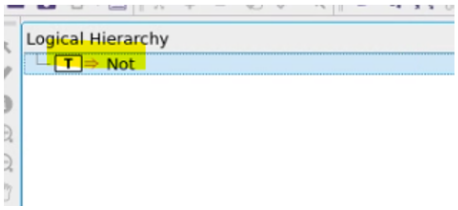


Figura 59: Importación de archivos 12

6. Para analizar visualmente el diseño del circuito, se debe hacer clic derecho en el diseño dentro de la sección *Logical Hierarchy* y seleccionar *Schematic View*, como se muestra

en la Figura 60. Esto abrirá una vista en forma de *black box* del circuito, lo que permite verificar que el diseño tenga las entradas y salidas correctas, tal como se observa en la Figura 61.

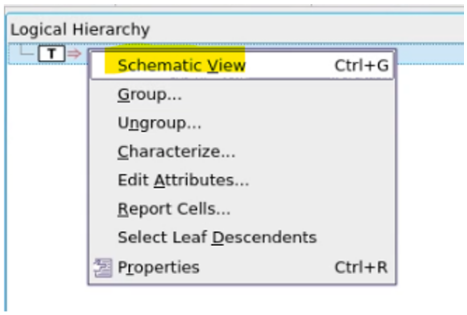


Figura 60: Vista al esquemático

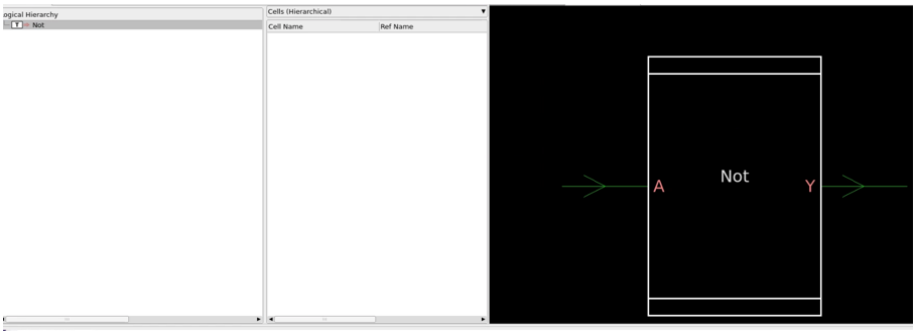


Figura 61: Esquemático de el circuito

También es posible visualizar el circuito en distintos niveles de jerarquía, como a nivel de compuertas, haciendo doble clic en el esquema de *Schematic View*. Esto permite revisar la estructura detallada del circuito y analizar cada componente de manera individual, tal como se muestra en la Figura 62.

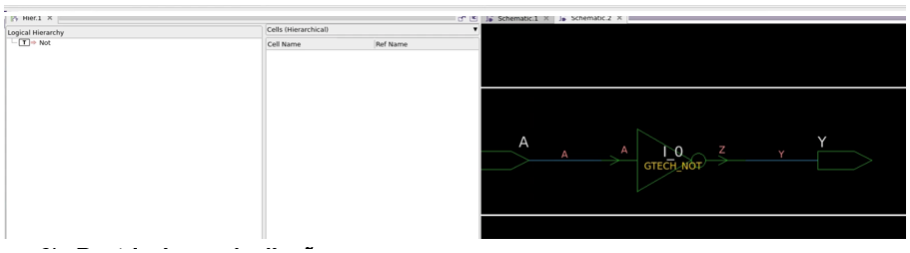


Figura 62: Esquemático de el circuito 2

12.2.3. Restricciones de diseño

El siguiente paso consiste en aplicar las restricciones de diseño, un proceso crucial para hacer que el diseño sea más realista. Estas restricciones permiten optimizar ciertos aspectos

del circuito, como los relojes que utiliza, el área de diseño, entre otros.

Antes de realizar cualquier cambio en las restricciones del diseño, es necesario llevar a cabo dos pasos de preparación:

1. Realizar el *LINK* del diseño. Esta acción se efectúa desde la pestaña *File*, haciendo clic en la opción *Link Design*, tal como se muestra en la Figura 63. Esto abrirá una ventana similar a la que se observa en la Figura 64, en la cual se debe verificar que las instancias de las librerías sean correctas.

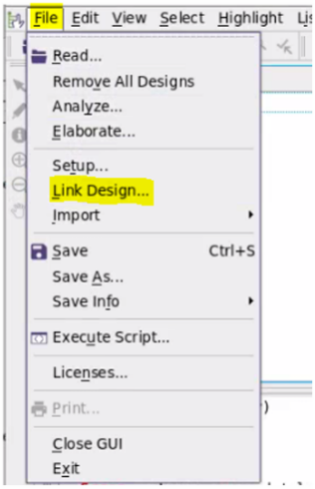


Figura 63: Enlace del diseño con las librerías

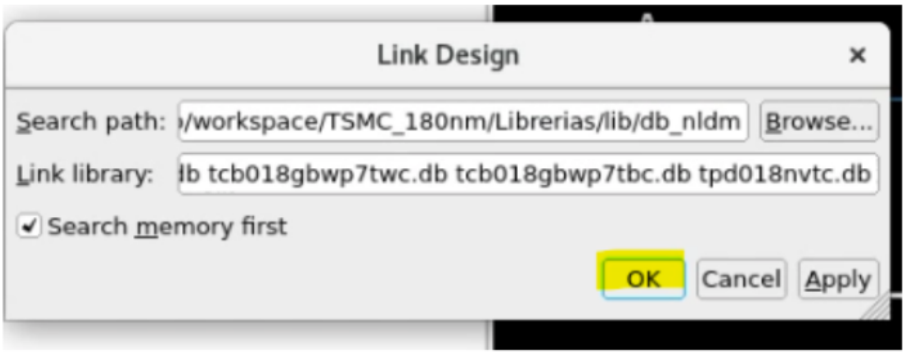


Figura 64: Enlace del diseño con las librerías 2

2. Realizar la unificación del diseño. Esta acción se lleva a cabo desde la pestaña *Hierarchy*, haciendo clic en la opción *Uniquify/Hierarchy*, como se muestra en la Figura 65. Esto abrirá una ventana similar a la que se observa en la Figura 66, donde se debe activar la opción *instances* y luego hacer clic en el botón *OK*. Este proceso es necesario para evitar que existan múltiples instancias de un mismo módulo en el diseño.

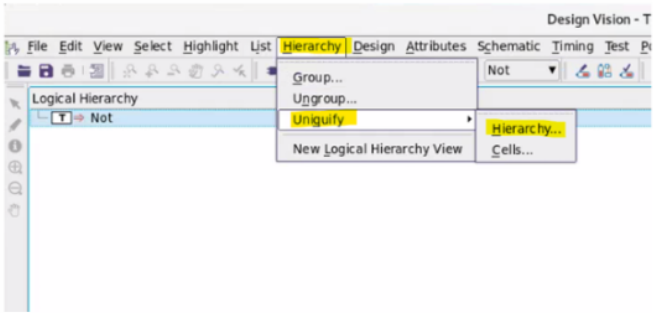


Figura 65: Unificación del diseño

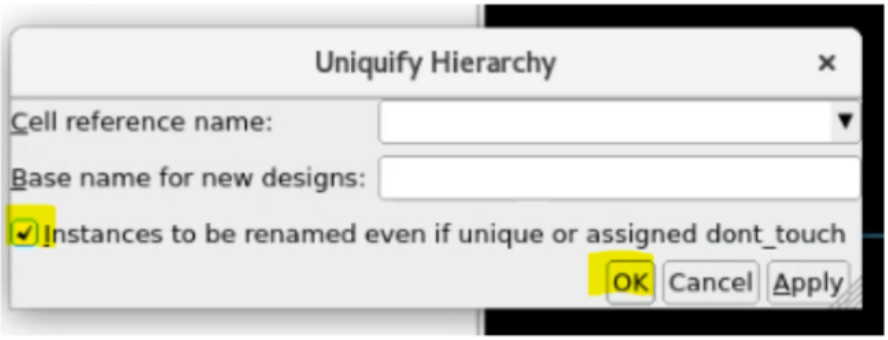


Figura 66: Unificación del diseño 2

Restricciones de tiempo

Para configurar un reloj en un circuito síncrono, es necesario ir a la pestaña *Attributes* y hacer clic en la opción *Specify Clock*. Esto abrirá una ventana como la que se muestra en la Figura 67, en la cual se pueden definir el nombre, el periodo deseado y el ciclo de trabajo del reloj que se desea implementar.

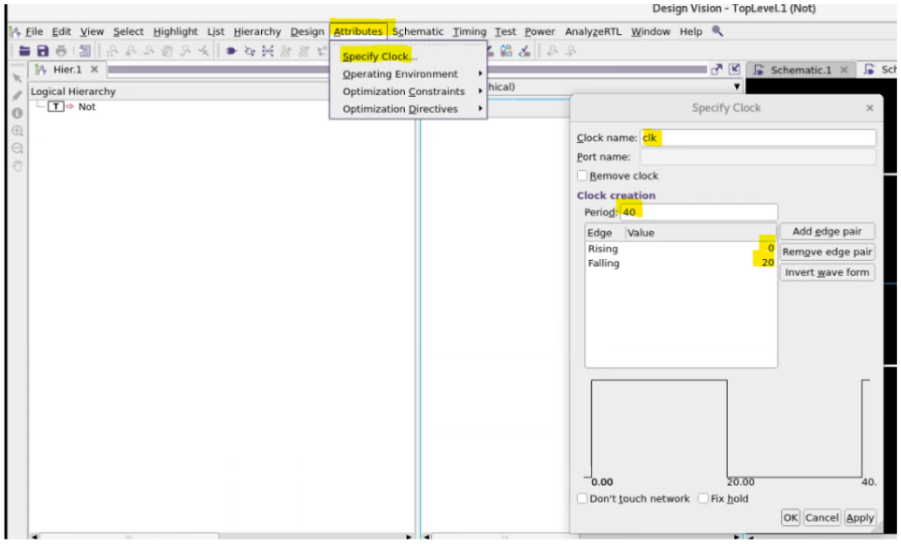


Figura 67: Creación de relojes

En este caso, se definió que el reloj tiene como nombre `clk`, con un borde ascendente en 0 ns, un borde descendente en 20 ns y un periodo de 40 ns (todas las medidas se expresan en nanosegundos). Esto genera un reloj con una frecuencia de 25 MHz y un ciclo de trabajo del 50 %, lo cual es típico para una señal de reloj de 25 MHz.

Esta restricción solo debe aplicarse si el circuito a sintetizar depende de algún reloj. Si el circuito no es puramente combinacional o no depende de ningún reloj (como en el caso de un inversor *not*), se creará un reloj virtual, como se observa en el *warning* mostrado en la Figura 68. En este caso, no tiene sentido crear un reloj, pero se realizó únicamente por fines ilustrativos.

```
design_vision>
Current design is 'Not'.
Current design is 'Not'.
design_vision> create_clock -name "clk" -period 40 -waveform { 0 20  }
Warning: Creating virtual clock named 'clk' with no sources. (UID-348)
1
design_vision>
```

Figura 68: *Warning* de creación de relojes virtuales para circuitos asíncronos

12.2.4. Verificación de diseño

Luego de haber realizado todas las configuraciones deseadas en el diseño, es necesario efectuar una verificación para corroborar que no existan problemas de conectividad, corto-circuitos o múltiples instancias de un mismo módulo. Esto se puede realizar utilizando el siguiente comando:

```
check_design
```

Cuadro 24: Comando para verificación del diseño

Si todo está configurado correctamente, Design Vision debería devolver un valor `1` en la terminal. Esto indica que no se encontraron problemas en el diseño del circuito.

12.2.5. Compilación de diseño

El siguiente paso, después de verificar el diseño, es la compilación, donde Design Vision se encargará de sintetizar el código Verilog original en base a las celdas de las librerías de TSMC. Para ello, se deben seguir los siguientes pasos:

1. Ir a la pestaña de *Design* y luego hacer clic en la opción de *Compile Design*.

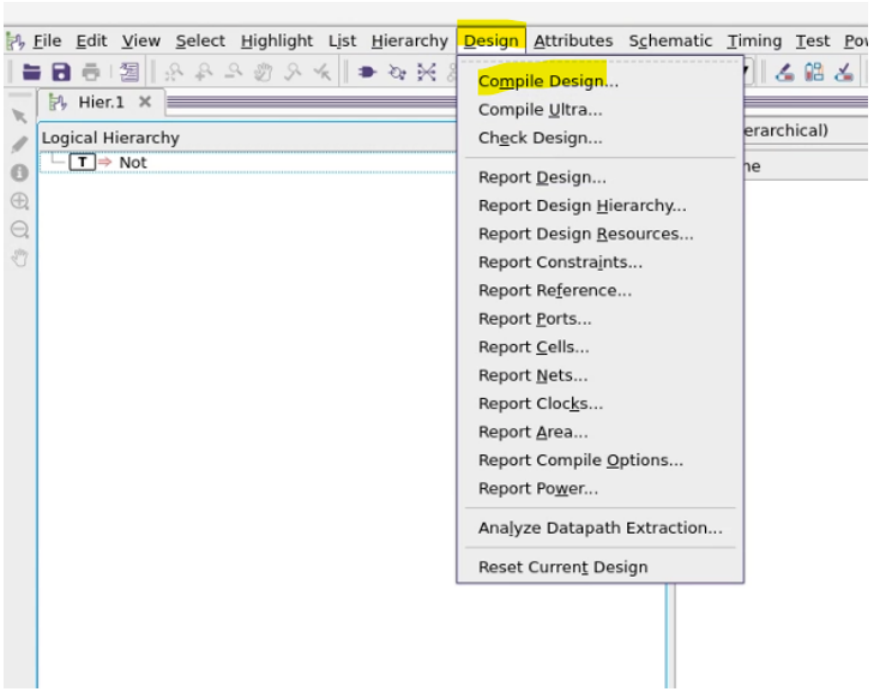


Figura 69: Síntesis del circuito

2. Se abrirá una ventana como la que se muestra en la Figura 70. En esta ventana, se debe establecer el nivel de esfuerzo y luego hacer clic en el botón *OK*.

   El parámetro `map_effort` se utiliza para definir el esfuerzo asignado a la etapa de mapeo (tiene dos estados posibles: *medium* y *high*). El siguiente parámetro es `area_effort`, que controla el esfuerzo para optimizar el área del diseño (cuenta con tres posibles estados: *none*, *medium* y *high*). Por último, el parámetro `power_effort` regula el esfuerzo de optimización de energía (este parámetro cuenta con cuatro estados posibles: *none*, *low*, *medium* y *high*).

Para la síntesis de la compuerta *not*, se establecieron los valores en *high*.

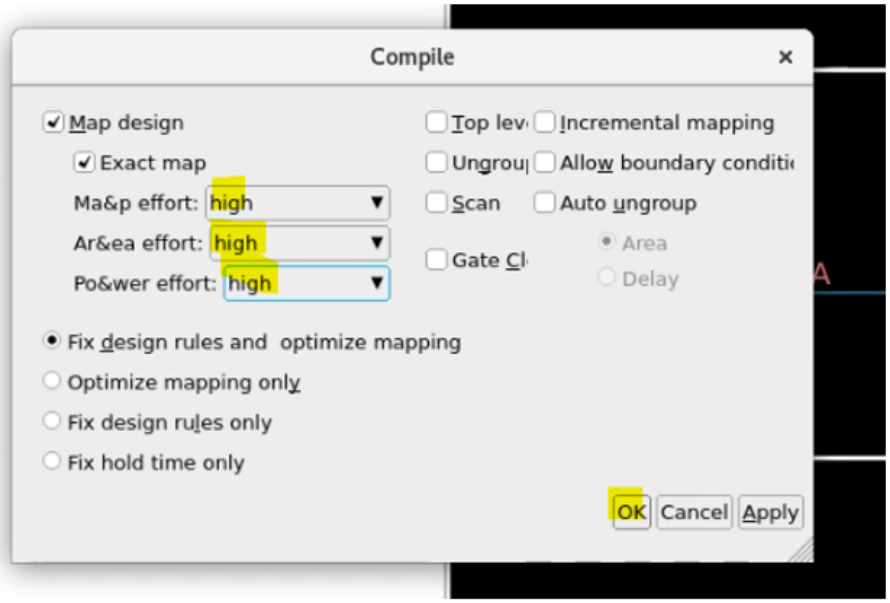


Figura 70: Síntesis del circuito

3. Luego, se actualizará automáticamente el diseño del circuito, el cual ahora estará sintetizado. Esto se puede comprobar haciendo nuevamente clic derecho en el diseño dentro de la sección *Logical Hierarchy* y seleccionando *Schematic View*. De esta forma, se muestra el circuito como una *black box*, y al hacer doble clic se visualiza el circuito a nivel de compuerta.

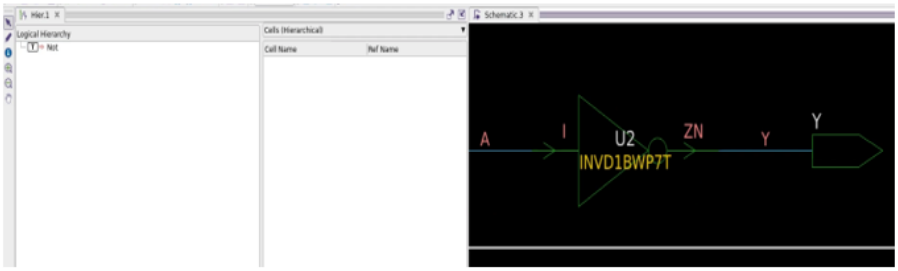


Figura 71: Esquemático del circuito sintetizado

Al comparar el esquemático de la Figura 62 con el obtenido en la Figura 71, se observa que la celda utilizada como inversor no es la misma. Esto se debe a que, durante la síntesis, las celdas genéricas se reemplazan por celdas de las librerías de TSMC de 180 nm. En este caso, la celda `INVD1BW97T` corresponde a una celda fabricable de TSMC.

12.2.6. Generación de reportes

Luego de realizar todos los pasos anteriores, ya se podrán generar los reportes de las características del diseño. Para ello, se debe ir a la opción *Design* y hacer clic en la opción correspondiente al reporte que se desea generar. Como se muestra en la Figura 72, existen varias alternativas para obtener reportes específicos, como el reporte de diseño, el de celdas o el de área, entre otros. En la Figura 73 se presenta un ejemplo de un reporte de área generado para el circuito sintetizado de la compuerta *not*.

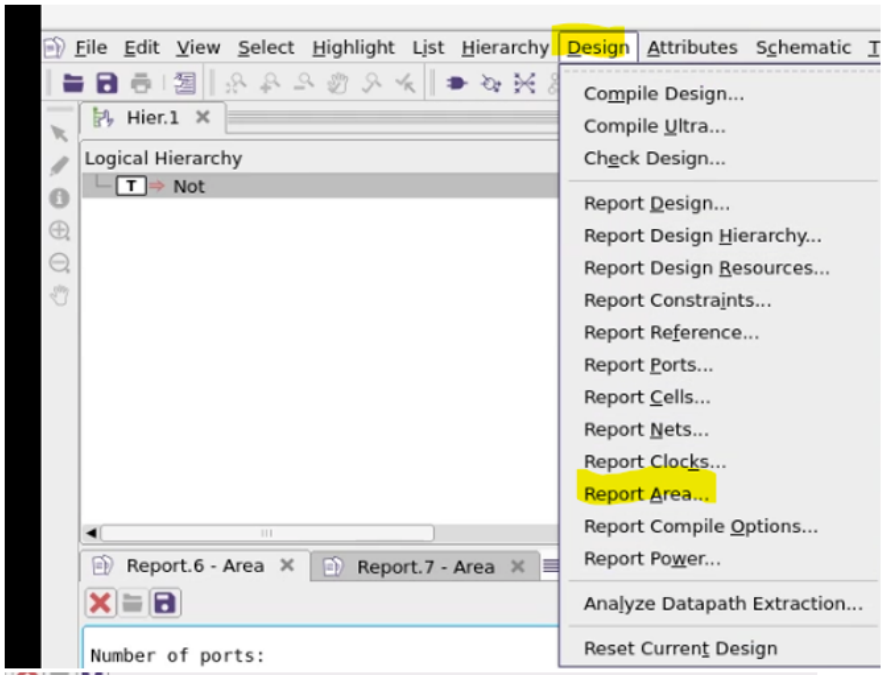


Figura 72: Reportes de Design Vision

```
Number of ports:                            2
Number of nets:                             2
Number of cells:                            1
Number of combinational cells:              1
Number of sequential cells:                 0
Number of macros/black boxes:               0
Number of buf/inv:                          1
Number of references:                       1

Combinational area:                 6.585600
Buf/Inv area:                       6.585600
Noncombinational area:              0.000000
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (Wire load has zero net area)
```

Figura 73: Reporte de área del circuito *not* sintetizado

12.2.7. Generación de archivos de salida

Por último, se deben generar los archivos de salida que se utilizarán en la etapa de síntesis física. Los archivos que se deben generar son tres:

1. Un archivo Verilog que contiene el circuito sintetizado a nivel de compuerta en formato *netlist*.
2. Un archivo en formato `.sdc` que incluye las restricciones del diseño.
3. Un archivo `.ddc` que contiene la información general del circuito.

Para ello, se debe hacer clic en *File* y luego en *Save As*, como se muestra en la Figura 74. A continuación, se abrirá una ventana como la que se observa en la Figura 75, en la cual se debe indicar el directorio de destino, la extensión y el nombre del archivo a guardar. En la Figura 76 se muestra el archivo Verilog que contiene el módulo de la compuerta *not* sintetizada, obtenido al guardar el archivo con la extensión `.v`.

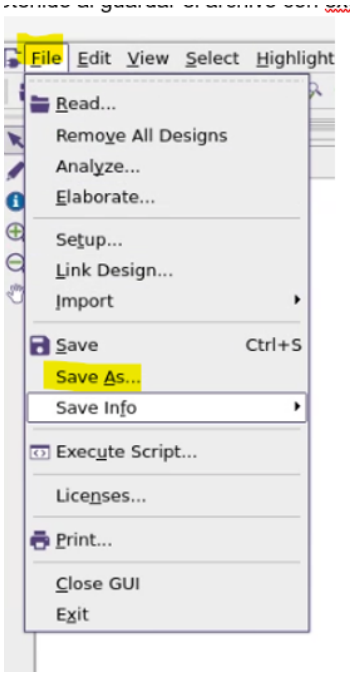


Figura 74: Generación de archivos de salida

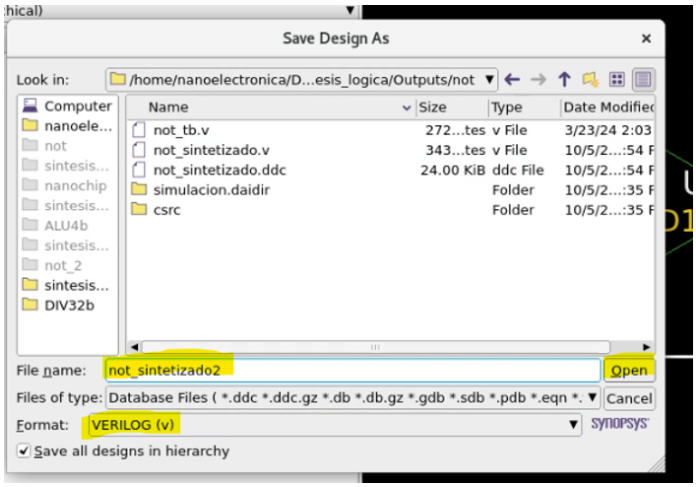


Figura 75: Generación de archivos de salida 2

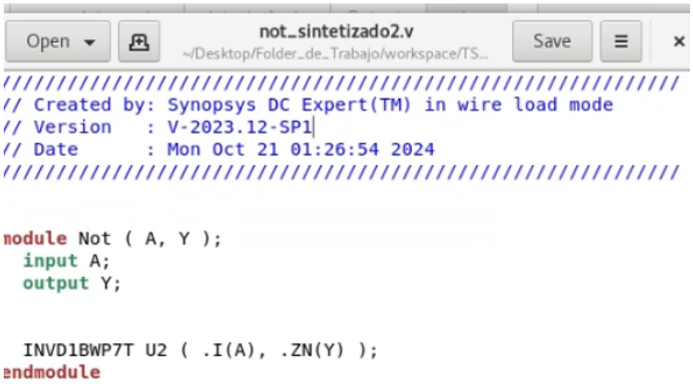


Figura 76: Archivo del *not* sintetizado

12.3. Uso de Design Vision con comandos

Una gran ventaja de este entorno es que todo el proceso de síntesis lógica puede realizarse mediante líneas de comando, lo que permite optimizar tiempo y recursos. Desde la terminal de comandos es posible abrir y cerrar la herramienta Design Compiler, importar las librerías, leer el archivo Verilog que describe el circuito a sintetizar, verificar el diseño del circuito, definir los parámetros de diseño y generar los archivos `.sdc`, `.ddc` y `.v` (*netlist*), los cuales serán utilizados en la etapa de síntesis física. Además, se pueden ejecutar prácticamente cualquier otra configuración o ajuste necesario.

Por esta razón, los exalumnos Luis Nájera, Steven Rubio, Sophia Cardona y Elmer Torres, quienes trabajaron en años anteriores en el proceso de síntesis lógica, desarrollaron y mejoraron un *script* que contiene todos los comandos necesarios para llevar a cabo la síntesis lógica sin errores.

12.3.1. Análisis de los *scripts* y explicación de comandos utilizados en años anteriores

A continuación, se muestra el *script* `ejecutar_sintesis_logica.bash`, que se debe ejecutar para realizar la síntesis lógica de un circuito. Este *script* se encarga principalmente de ejecutar el archivo `comandos_sintesis_logica.script` mediante el comando `dc_shell`, el cual se utiliza para iniciar el entorno de Design Compiler en modo de línea de comandos.

```
#!/bin/bash
echo " Corriendo sintesis logica con Design Vision:"
echo " Ejecutando la parte de Sintesis Cell"

dc_shell -f comandos_sintesis_logica.script

mv default.svf logs/
mv command.log logs/

echo "Finalizo la Sintesis Logica"
exit
```

Cuadro 25: *Script* ejecutar_sintesis_logica.bash utilizado para ejecutar la síntesis lógica

Este *script* automatiza el proceso de síntesis lógica, ya que ejecuta el archivo `comandos_sintesis_logica.script`, en el que se definen los comandos específicos del entorno Design Compiler para realizar la síntesis lógica del diseño.

En el Cuadro 26 se muestra el *script* denominado `comandos_sintesis_logica.script`, el cual contiene los comandos de Design Compiler que fueron utilizados en años anteriores y que han funcionado sin errores al sintetizar circuitos. La única modificación realizada fue la incorporación de la variable `PARAMETRO`, que permite controlar el nombre del archivo del circuito a sintetizar, el directorio donde se encuentran los archivos de entrada y los nombres de los archivos de salida (generados en el directorio `sintesis_logica/Outputs/"Name_Circuito"`).

```
##-----------------------------------------------------------
##------------COMANDOS PARA LA SINTESIS LOGICA----------
##-----------------------------------------------------------
# Definir el parametro
set PARAMETRO "not"
set PARAMETRO2 "${PARAMETRO}_sintetizado"

# IMPORTACION DE LIBRERIAS
lappend search_path cd ../../Librerias/lib/db_nldm
set link_library " * tcb018gbwp7ttc.db tcb018gbwp7twc.db tcb018gbwp7tbc.db
    tpd018nvtc.db"
set target_library "tcb018gbwp7ttc.db"

# LECTURA DEL ARCHIVO DESCRIPTIVO DEL CIRCUITO
# Usar el parametro en lugar del nombre del modulo
read_file -format verilog "Inputs/$PARAMETRO/$PARAMETRO.v"
#read_file -format verilog "verilog/$PARAMETRO/$PARAMETRO.v verilog/$PARAMETRO/
    ADDER4b.v verilog/$PARAMETRO/ADDER16b.v verilog/$PARAMETRO/SUB16b.v"
```

```
# Configuracion de restricciones
reset_design
set_max_area 0
link
uniquify
create_clock clk -period 40 -waveform {0 20}
# create_clock -period 40 -waveform {0 20} -name clk


# Verificacion del diseno
check_design

# Compilacion de circuito
compile -map_effort high -area_effort high -power_effort high

# generarion de reportes
report_area
report_design
report_power
report_cell

# Generacion de archivos en Outputs
write -format ddc -h -o "./Outputs/$PARAMETRO/$PARAMETRO2.ddc"
write -hierarchy -format verilog -output "./Outputs/$PARAMETRO/$PARAMETRO2.v"
write_sdc "./Outputs/$PARAMETRO/$PARAMETRO2.sdc"
exit
```

Cuadro 26: *Script* comandos_sintesis_logica.script con los comandos para realizar la síntesis lógica

A continuación, se explica el uso y la importancia de cada comando utilizado en `comandos_sintesis_logica.script`.

Importación de librerías

```
lappend search_path /PATH/
set link_library " * tcb018gbwp7ttc.db tcb018gbwp7twc.db tcb018gbwp7tbc.db
    tpd018nvtc.db"
set target_library "tcb018gbwp7ttc.db"
```

Cuadro 27: Comandos para importación de librerías

El primer comando que se utiliza, `lappend search_path`, sirve para especificar la ubicación donde se encuentran las librerías que se emplearán durante el proceso de síntesis lógica.

Posteriormente, el comando `set` define las variables `link_library` y `target_library`, a las cuales se les asigna la lista de librerías necesarias para la síntesis. Esta lista incluye archivos de base de datos `.db` que contienen información sobre las celdas y componentes del diseño requeridos para el proceso.

Importación del archivo Verilog del circuito

```
read_file -format verilog {verilog/not/not.v}
#read_file -format verilog "verilog/ADDER32b/ADDER32b.v verilog/ADDER32b/
    ADDER4b.v"
```

Cuadro 28: Comando para importación del archivo Verilog del circuito

El comando `read_file` se utiliza para leer el archivo que contiene la descripción lógica del circuito que se empleará en la síntesis lógica. La opción `-format` permite especificar el tipo de formato del archivo, en este caso, Verilog. Luego, se debe indicar la ubicación donde se encuentra el archivo Verilog a sintetizar.

En caso de que se necesite sintetizar un circuito compuesto por varios archivos, donde cada módulo está descrito en un archivo diferente, se puede emplear nuevamente el comando `read_file`, especificando la ubicación exacta de cada uno de los archivos Verilog (como se muestra en la línea comentada del Cuadro 28).

Configuración de restricciones

```
reset_design
set_max_area 0
link
uniquify
create_clock clk -period 40 -waveform {0 20}
# create_clock -period 40 -waveform {0 20} -name clk
```

Cuadro 29: Configuración de restricciones

Luego se realizan las configuraciones de restricciones, las cuales se explican a continuación:

- `reset_design`: el comando `reset_design` se utiliza para reiniciar el estado del diseño a su condición inicial. Se encarga de limpiar todas las configuraciones (como, por ejemplo, los relojes, los retrasos de entrada y salida, las condiciones de funcionamiento, etc.) que se hayan aplicado al diseño, con el fin de asegurarse de que el diseño se reinicie sin interferencias de configuraciones anteriores.
- `set_max_area 0`: el comando `set_max_area 0` se utiliza para establecer una restricción de área y velocidad. Sirve para definir que el circuito ocupe la menor área posible al momento de sintetizarlo. Cabe mencionar que el área máxima representa la cantidad de compuertas en el diseño, no el área física que ocupa el chip.
- `link` y `uniquify`: el comando `link` se utiliza para asegurar que en las librerías se encuentren las celdas que se utilizarán para sintetizar el circuito, mientras que el comando `uniquify` garantiza que cada instancia de un módulo tenga un nombre único en el diseño. Por ejemplo, si se tienen dos módulos `halfadder` en un diseño, `uniquify` podría renombrar las instancias como `halfadder_1` y `halfadder_2` para distinguirlas claramente.

- `create_clock`: el comando `create_clock` se utiliza para definir restricciones de *timing*. La opción `-period` se usa para especificar el periodo del reloj (en este caso, 40 ns), mientras que la opción `-waveform` permite definir los bordes del reloj. En este caso, se definió que el reloj tiene un borde ascendente en 0 ns y un borde descendente en 20 ns.

  Esto crea un reloj con una frecuencia de 25 MHz y un ciclo de trabajo del 50 %, lo cual es típico para una señal de reloj de 25 MHz.

  Es importante mencionar que, para crear un reloj, se tienen dos opciones, dependiendo de si el circuito es síncrono o no es puramente combinacional.

  - **Circuito síncrono:** en este caso, se debe utilizar el siguiente comando, en el que después de `create_clock` se coloca el nombre del reloj del circuito.

```
create_clock clk -period 40 -waveform {0 20}
```

Cuadro 30: Creación de reloj

  - **Circuito no puramente combinacional:** si el diseño tiene algunos componentes sincrónicos (por ejemplo, registros), se debe crear un reloj virtual, el cual será análogo a un reloj del sistema. El único cambio consiste en colocar el nombre del reloj después de especificar los bordes y agregar el parámetro `-name`.

```
create_clock -period 40 -waveform {0 20} -name clk
```

Cuadro 31: Creación de reloj virtual

Verificación del diseño

```
check_design
```

Cuadro 32: Comando para verificación del diseño

El comando `check_design` se utiliza para revisar el diseño. Al ejecutar este comando y verificar que no haya ningún error, la herramienta devuelve un valor de 1 en la terminal. Específicamente, realiza las siguientes verificaciones:

- **Verificación de la estructura del diseño:** asegura que todos los módulos y componentes del diseño estén conectados correctamente y que no existan errores estructurales.
- **Comprobación de las bibliotecas:** verifica que las librerías especificadas estén disponibles y sean compatibles con el diseño.
- **Detección de errores de sintaxis:** identifica posibles errores en el código del diseño, como errores de escritura o de formato.

Compilación del circuito

```
compile -map_effort high -area_effort high
```

Cuadro 33: Comando para ejecutar síntesis lógica al circuito

El comando `compile` se utiliza para efectuar la síntesis y optimización del circuito con base en las restricciones establecidas.

Las opciones que se pueden configurar incluyen `map_effort`, que define el nivel de esfuerzo empleado durante el mapeo al *netlist*, y `area_effort`, que establece el nivel de esfuerzo aplicado para optimizar el área del diseño sintetizado.

Ambas opciones se configuraron con el valor *high* para obtener una mayor precisión en el cumplimiento de las restricciones y en el correcto mapeo del diseño al *netlist*.

Generación de reportes

```
report_area
report_design
report_power
```

Cuadro 34: Comandos para la generación de reportes

Luego se generan algunos reportes para conocer diferentes características del circuito sintetizado. Es importante mencionar que se pueden generar otros tipos de reportes, según el aspecto que se desee analizar. Sin embargo, los siguientes son algunos de los más relevantes:

- `report_area`: proporciona información sobre las características físicas del diseño, como el número de celdas y el área total que ocupa.
- `report_design`: muestra las restricciones que se definieron para el circuito.
- `report_power`: presenta los valores de potencia del circuito sintetizado.

Generación de archivos de salida

```
write -format ddc -h -o ./PATH/DONDE/GUARDAR/name_sintesis.ddc
write_sdc ./PATH/DONDE/GUARDAR/name_sintesis.sdc
write -hierarchy -format verilog -output ./PATH/DONDE/GUARDAR/name_sintesis.v

exit
```

Cuadro 35: Comandos para la generación de archivos de salida

Por último, se generan los archivos de salida necesarios para simular el circuito sintetizado y para realizar la siguiente etapa del diseño, la cual es la síntesis física.

Para generar el archivo `.ddc`, se utiliza el comando *write*. Se debe colocar la opción `-format` y luego especificar el tipo de formato, que en este caso es `ddc`. Finalmente, se indica la ubicación donde se guardará el archivo de salida. El nombre del archivo corresponde al que se coloque al final de la ruta, por ejemplo: `name_sintesis.ddc`.

Luego, para generar el archivo `.sdc`, se utiliza el comando `write_sdc`, especificando la ubicación donde se guardará el archivo de salida y colocando al final de la ruta el nombre que tendrá el archivo, por ejemplo: `name_sintesis.sdc`.

Por último, para generar el archivo Verilog (`.v`), se utiliza nuevamente el comando *write*. Se emplea la opción `-hierarchy` para asegurar que la síntesis del circuito se realice correctamente y que la estructura jerárquica sea la esperada. También se utiliza la opción `-format` para especificar el tipo de formato, que en este caso es Verilog. Finalmente, se indica la ubicación donde se guardará el archivo y se coloca al final de la ruta el nombre que tendrá el archivo, por ejemplo: `name_sintesis.v`.

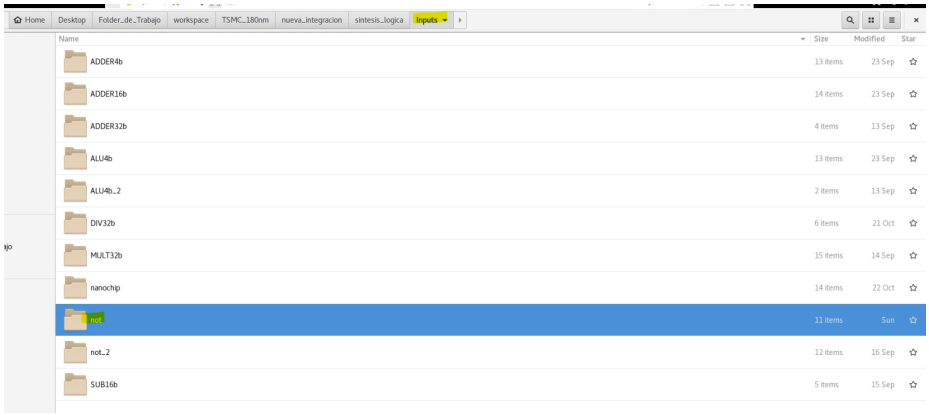
- **Archivo `.ddc`:** contiene información general del circuito, como la descripción del diseño y las restricciones aplicadas, entre otros datos.
- **Archivo `.sdc`:** incluye las restricciones utilizadas en el diseño.
- **Archivo `.v`:** contiene el circuito sintetizado a nivel de compuerta en formato *netlist*.

12.3.2. Uso de los *scripts* utilizados en años anteriores

En este caso, con fines ilustrativos se sintetizó el circuito de una compuerta *not*. Sin embargo, el proceso para sintetizar otros circuitos es el mismo.

Antes de ejecutar el *script* `ejecutar_sintesis_logica.bash`, es importante recordar que, para generar correctamente los archivos de salida, se debe crear manualmente un subdirectorio dentro de la carpeta *outputs* con el mismo nombre que el directorio del circuito en *inputs*.

Por ejemplo, si en *inputs* existe un directorio llamado *not*, en *outputs* también debe crearse una carpeta con el mismo nombre (*not*), según se explica en el capítulo 9, "Jerarquía de directorios de la síntesis lógica" (véase la Figura 77). Si no se realiza este paso, los archivos de salida no se generan.



(a) Carpeta del circuito creado en *inputs*

(b) Carpeta del circuito creado en *outputs*

Figura 77: Creación de carpetas del circuito

Los archivos que deben estar presentes en el directorio `Inputs/"Name_Circuito"` son:

- Archivo(s) Verilog que describen el circuito a sintetizar
- Archivo Verilog del *testbench*, utilizado para la simulación del circuito
- Archivo Verilog que contiene el módulo de pines de entrada y salida del circuito a sintetizar

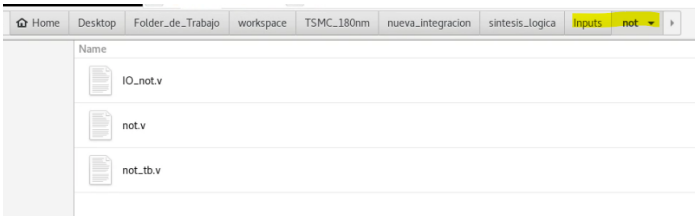


Figura 78: Archivos necesarios en la carpeta del circuito en *inputs*

Es importante recordar que tanto el archivo del circuito principal como la carpeta donde está guardado deben tener el mismo nombre, tal como se observa en la Figura 78. Si no se realiza este paso, el proceso de síntesis no se completará exitosamente.

Para sintetizar el circuito, primero se debe abrir el *script* comandos_sintesis_logica.script y definir en la variable PARAMETRO el nombre de la carpeta creada en *inputs* que contiene el circuito a sintetizar.

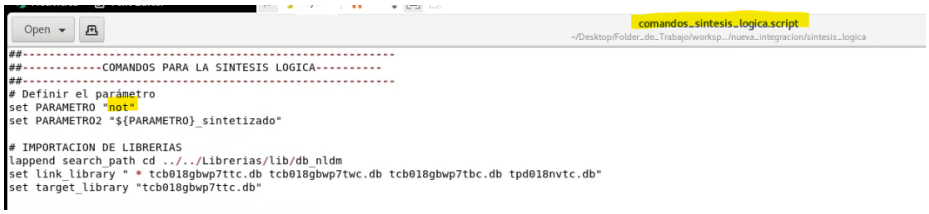


Figura 79: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

Luego, se debe ejecutar el *script* con el comando que se muestra en el Cuadro 36, desde una terminal ubicada en la carpeta `sintesis_logica`.

```
./ejecutar_sintesis_logica.bash
```

Cuadro 36: Comando para ejecutar el *script* ejecutar_sintesis_logica.bash

Si el proceso se ejecutó correctamente y se siguieron todos los pasos previos, en la terminal no debería aparecer ningún error, y los archivos de salida estarán en la carpeta correspondiente (véanse las Figuras 80 y 81).

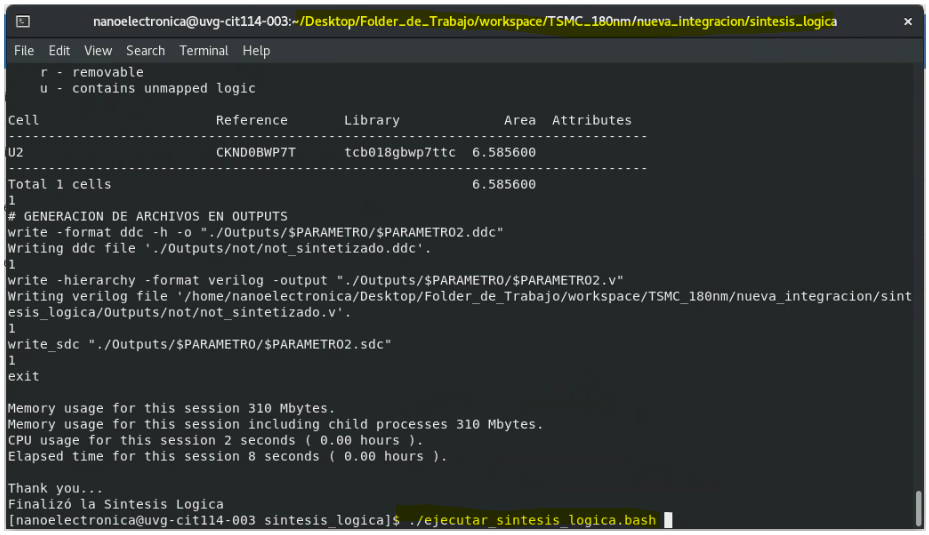


Figura 80: Ejecución del *script* ejecutar_sintesis_logica.bash sin errores

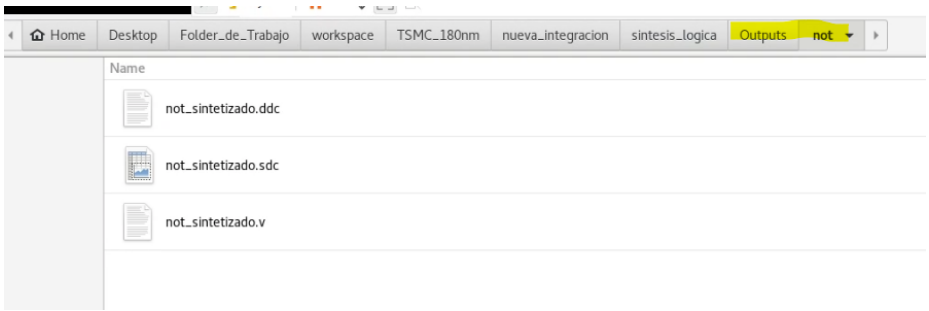


Figura 81: Archivos generados en la carpeta del circuito en *outputs*

En la Figura 82 se muestra el archivo Verilog obtenido, correspondiente al circuito sintetizado. Este archivo hace referencia a módulos que representan celdas de TSMC con tecnología de 180 nm.

Figura 82: Archivo not_sintetizado.v: circuito *not* sintetizado

Sin embargo, lo anterior solo comprueba que el proceso de síntesis se realizó correctamente, pero no garantiza que el circuito sintetizado se comporte como se desea. Para verificar su correcto funcionamiento, es necesario simular el circuito sintetizado. Este proceso de simulación es similar al que se explicó en la sección 12.1 para simular el circuito original. En el capítulo 13 se simularán los circuitos sintetizados de diferentes diseños.

Es importante mencionar que, si se desea sintetizar el circuito previamente generado junto con el módulo de pines de entrada y salida utilizando el *script* `ejecutar_sintesis_logica.bash`, se deben unir ambas partes en un mismo archivo Verilog. Luego, es necesario actualizar el nombre del archivo en el *script* mediante la variable `PARAMETRO` y volver a ejecutar la síntesis.

Todo este proceso se realizaría manualmente, lo cual puede resultar tardado. Por esta razón, se diseñó otro *script* que automatiza todo el procedimiento, el cual se explica en las siguientes secciones.

No obstante, para comprender el uso de los comandos y el proceso de síntesis, se recomienda utilizar este *script* únicamente si se desea sintetizar un circuito sin incluir los pines de entrada y salida.

12.3.3. Análisis de los nuevos *scripts* realizados

En el Cuadro 37 se muestra el *script* denominado `ejecutar_sintesis_logica_2.bash`, el cual debe ejecutarse para realizar la síntesis lógica de un circuito con la integración del módulo de pines de entrada y salida (I/O) de forma automática.

```
#!/bin/bash
PARAMETRO="not"

# paso 1: Creacion de carpetas de OUtputs
mkdir -p ~/Desktop/Folder_de_Trabajo/workspace/TSMC_180nm/nueva_integracion/
    sintesis_logica/Outputs/$PARAMETRO/Outputs_iniciales ~/Desktop/
    Folder_de_Trabajo/workspace/TSMC_180nm/nueva_integracion/sintesis_logica/
    Outputs/$PARAMETRO/Outputs_finales
```

```
cp ~/Desktop/Folder_de_Trabajo/workspace/TSMC_180nm/nueva_integracion/
    sintesis_logica/Inputs/$PARAMETRO/${PARAMETRO}_tb.v ~/Desktop/
    Folder_de_Trabajo/workspace/TSMC_180nm/nueva_integracion/sintesis_logica/
    Outputs/$PARAMETRO/Outputs_finales

echo " Corriendo sintesis logica con Design Vision:"
echo " Ejecutando la parte de Sintesis Cell"

# Paso 2: Ejecutar la parte 1 del \textit{script}
echo "------------------------------------------------------------"
echo "-------- EJECUTANDO SINTESIS LOGICA SOLO CON MODULO -------"
echo "---------------------- DEL CIRCUITO ORIGINAL ---------------"
dc_shell -f comandos_sintesis_logica_2.script -x "set script_step 1"

# Paso 3: Mover los archivos de log a la carpeta correspondiente
mv default.svf logs/
mv command.log logs/

# Paso 4: Combinar el archivo generado $PARAMETRO2.v con el archivo
    PINES_IO_$PARAMETRO.v
PARAMETRO2="${PARAMETRO}_sintetizado"
COMBINED_FILE="Outputs/$PARAMETRO/Outputs_iniciales/${PARAMETRO2}_and_IO.v"
cat "Inputs/$PARAMETRO/IO_$PARAMETRO.v" "Outputs/$PARAMETRO/Outputs_iniciales/
    $PARAMETRO2.v" > "$COMBINED_FILE"

# Paso 5: Ejecutar la parte 2 del \textit{script}
echo "------------------------------------------------------------"
echo "--------- EJECUTANDO SINTESIS LOGICA CON EL MODULO -------"
echo "---------------------- DE PINES I/O ----------------------"
dc_shell -f comandos_sintesis_logica_2.script -x "set script_step 2"

# Paso 6: Mover los archivos de log a la carpeta correspondiente
mv default.svf logs/
mv command.log logs/

# Finalizar
echo "Finalizo la sintesis logica"

exit
```

Cuadro 37: *Script* ejecutar_sintesis_logica_2.bash utilizado para ejecutar la síntesis lógica

Este *script* se divide en las siguientes partes:

- **Configuración inicial:** en esta parte se define la variable `PARAMETRO` con el valor del nombre del circuito a sintetizar, el cual coincide con el nombre del directorio del circuito tanto en la carpeta *inputs* como en *outputs*.
- **Paso 1 – Creación de carpetas:** esta parte del código se encarga de crear dos carpetas para almacenar los archivos de salida en rutas específicas: `Outputs_iniciales` y `Outputs_finales`, dentro de una carpeta en *outputs* que lleva el nombre definido en la variable `$PARAMETRO`. Si el directorio con el nombre de `$PARAMETRO` no existe, también se creará junto con los dos subdirectorios mencionados anteriormente.

Además, se copia el archivo `${PARAMETRO}_tb.v` desde la carpeta `Inputs/Name_Circuito` hacia `Outputs_finales`. Esto permite tener disponible el archivo de *testbench* cuando se desee realizar simulaciones del circuito sintetizado.

- **Paso 2 – Ejecución de la síntesis lógica inicial:** se ejecuta la parte del *script* `comandos_sintesis_logica_2.script` en `dc_shell`, la cual está configurada para la primera etapa de la síntesis lógica. Esto se realiza estableciendo la variable `script_step` en 1, que actúa como una bandera para el *script*, indicándole que se trata del primer proceso de síntesis (solo del circuito).
- **Paso 3 – Organización de archivos de registro (*logs*):** se mueven los archivos de registro generados (`default.svf` y `command.log`) a una carpeta llamada *logs* para mantenerlos organizados.
- **Paso 4 – Combinación de archivos de síntesis con pines I/O:** se define una variable `PARAMETRO2=${PARAMETRO}_sintetizado` para hacer referencia al primer archivo generado en la primera etapa de síntesis. Luego, se combina este archivo con el archivo de pines de I/O (`IO_$PARAMETRO.v`), el cual debe estar en la carpeta `Inputs/Name_Circuito`. Esto permite crear un nuevo archivo que une ambos anteriores, nombrado `${PARAMETRO2}_sintetizado_and_IO.v`, el cual se ubicará en la carpeta `Outputs_iniciales`.
- **Paso 5 – Ejecución de la síntesis lógica con pines de I/O:** se ejecuta nuevamente el *script* de síntesis lógica en `dc_shell`, esta vez configurando un segundo paso (`set` *`script`*`_step 2`). Esta configuración indica que se trata del segundo proceso de síntesis, en el cual se sintetiza el archivo que incluye el circuito previamente sintetizado junto con los pines de I/O, es decir, `${PARAMETRO2}_sintetizado_and_IO.v`.
- **Organización de *logs* finales:** se mueven los archivos de registro generados en esta segunda etapa a la carpeta *logs*.

Por otro lado, en el Cuadro 38 se muestra el *script* denominado `comandos_sintesis_logica_2.script`. Este *script* contiene los mismos comandos explicados en la sección 12.3.1; sin embargo, la diferencia es que automatiza el proceso de síntesis lógica en dos etapas para un módulo de circuito y su integración con un archivo de pines de I/O.

El *script* utiliza la variable mencionada en el *script* `ejecutar_sintesis_logica_2.bash`, denominada `script_step`, para ejecutar selectivamente una de las dos etapas del proceso de síntesis lógica.

```
# Definir el parametro
set PARAMETRO "not"
set PARAMETRO2 "${PARAMETRO}_sintetizado"
#set script_step 1 ;# Variable para controlar que parte del \textit{script} se
    ejecuta

lappend search_path cd ../../Librerias/lib/db_nldm
set link_library " * tcb018gbwp7ttc.db tcb018gbwp7twc.db tcb018gbwp7tbc.db
    tpd018nvtc.db"
set target_library "tcb018gbwp7ttc.db"
```

```
if { $script_step == 1 } {
    # Parte 1 - Sintesis Cell original

    # Usar el parametro en lugar del nombre del modulo
    read_file -format verilog "Inputs/$PARAMETRO/$PARAMETRO.v"
    #read_file -format verilog "Inputs/$PARAMETRO/$PARAMETRO.v Inputs/
        $PARAMETRO/ADDER4b.v Inputs/$PARAMETRO/ADDER16b.v Inputs/$PARAMETRO/
        SUB16b.v"
    #read_file -format verilog "Inputs/$PARAMETRO/$PARAMETRO.v Inputs/
        $PARAMETRO/ADDER4b.v Inputs/$PARAMETRO/ADDER16b.v"

    reset_design
    set_max_area 0
    link
    uniquify
    create_clock clk -period 40 -waveform {0 20}

    check_design

    compile -map_effort high -area_effort high -power_effort high

    report_design
    report_power
    report_area
    report_cell

    #write -format ddc -h -o "./verilog/$PARAMETRO/$PARAMETRO2.ddc"
    write -hierarchy -format verilog -output "./Outputs/$PARAMETRO/
        Outputs_iniciales/$PARAMETRO2.v"
    #write_sdc "./verilog/$PARAMETRO/$PARAMETRO2.sdc"

    exit
}

if { $script_step == 2 } {
    # Parte 2 - Sintesis combinada con el archivo PINES_IO

    #set PARAMETRO "not"
    set PARAMETRO3 "${PARAMETRO}_sintetizado_and_IO"
    set PARAMETRO4 "${PARAMETRO}_sintesis_final"


    # Usar el parametro en lugar del nombre del modulo
    read_file -format verilog "Outputs/$PARAMETRO/Outputs_iniciales/$PARAMETRO3
        .v"

    check_design

    compile -map_effort high -area_effort high -power_effort high

    report_design
    report_power
    report_area
```

```
    report_cell

    write -format ddc -h -o "./Outputs/$PARAMETRO/Outputs_finales/$PARAMETRO4.
        ddc"
    write -hierarchy -format verilog -output "./Outputs/$PARAMETRO/
        Outputs_finales/$PARAMETRO4.v"
    write_sdc "./Outputs/$PARAMETRO/Outputs_finales/$PARAMETRO4.sdc"

    exit
}
```

Cuadro 38: *Script* comandos_sintesis_logica_2.script con los comandos para realizar la síntesis lógica

Este nuevo *script* realiza lo siguiente:

- **Etapa 1 – Síntesis del módulo original (cuando `script_step == 1`):** esta primera etapa de síntesis realiza lo explicado anteriormente en la sección 12.3.1, que incluye la importación del archivo que contiene el circuito a sintetizar, las configuraciones de diseño, la compilación y la generación de reportes. La única diferencia es que en esta etapa se manejan dos parámetros:

  1. `PARAMETRO`, que indica el nombre del circuito y del directorio creado manualmente en *inputs*.
  2. `PARAMETRO2`, que se utiliza para asignar un nombre al archivo de salida (circuito sintetizado), cuyo nombre `${PARAMETRO}_sintetizado` depende de `PARAMETRO`.

  Además, en esta parte se genera únicamente el archivo de salida del circuito sintetizado en Verilog, que se guarda automáticamente en la ruta `/Outputs/$PARAMETRO/Outputs_iniciales/` con el nombre `$PARAMETRO2.v`.

- **Etapa 2 – Síntesis combinada con pines de I/O (cuando `script_step == 2`):** en esta segunda etapa de síntesis se realiza lo siguiente:

  1. Se introducen dos nuevos parámetros: `PARAMETRO3`, definido como `$PARAMETRO_sintetizado_and_IO`, que hace referencia al archivo que combina el circuito sintetizado con el módulo de pines de I/O; y `PARAMETRO4`, establecido como `$PARAMETRO_sintesis_final`, que indica el nombre del archivo de salida generado en esta etapa.
  2. Se carga el archivo creado en la primera etapa (`$PARAMETRO_sintetizado_and_IO.v`), que contiene el circuito sintetizado junto con los pines de I/O.
  3. Se lleva a cabo la síntesis lógica utilizando el archivo cargado (circuito previamente sintetizado unido con el módulo de pines de I/O).
  4. Se generan los reportes que incluyen información sobre el diseño completamente sintetizado, como el área utilizada, la potencia y el uso de celdas.
  5. Se generan tres archivos de salida esenciales para la síntesis física: `$PARAMETRO4.ddc`, que contiene el diseño digital; `$PARAMETRO4.v`, que corresponde al circuito sintetizado con pines de I/O; y `$PARAMETRO4.sdc`, que incluye las restricciones del diseño.

Estas acciones completan la segunda etapa de la síntesis lógica, integrando el módulo de pines de I/O y generando los archivos necesarios para las etapas posteriores del diseño.

12.3.4. Uso de los nuevos *scripts*

Nuevamente, en este caso se sintetizó el circuito de una *not*; sin embargo, el proceso para sintetizar otros circuitos es el mismo.

Además, utilizando estos nuevos *scripts*, no es necesario crear manualmente un subdirectorio dentro de la carpeta *outputs* con el mismo nombre que el directorio del circuito en *inputs*, ya que el *script* `ejecutar_sintesis_logica_2.bash` lo crea automáticamente con el mismo nombre que tiene el circuito principal a sintetizar.

Sin embargo, antes de ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, es esencial asegurarse de que los siguientes archivos estén presentes en el directorio `Inputs/Name_Circuito`:

- Archivo o archivos Verilog que describen el circuito a sintetizar.
- Archivo Verilog de *testbench*, utilizado para la simulación del circuito.
- Archivo Verilog que contiene el módulo de pines de entrada y salida del circuito a sintetizar. Si este archivo no se encuentra, solo se realizará la síntesis del circuito original.

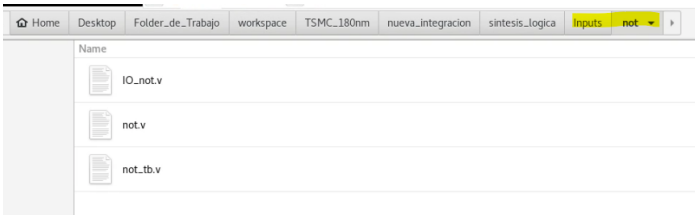
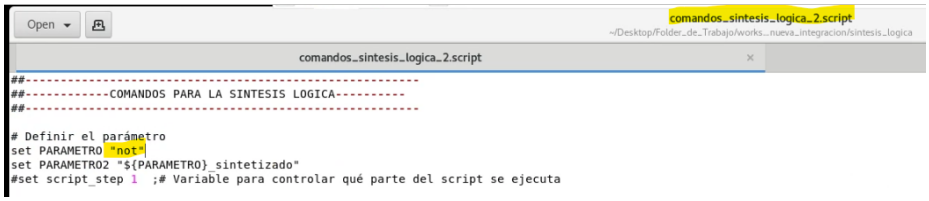


Figura 83: Archivos necesarios en la carpeta del circuito en *inputs*

Nuevamente, es importante recordar que tanto el archivo del circuito principal como la carpeta donde está guardado deben tener el mismo nombre, tal como se observa en la Figura 83. Si no se realiza este paso, el proceso de síntesis no se completará exitosamente.

Para sintetizar el circuito, primero se deben abrir los *scripts* `ejecutar_sintesis_logica_2.bash` y `comandos_sintesis_logica_2.script`, y definir en la variable `PARAMETRO` el nombre de la carpeta creada en *inputs* que contiene el circuito a sintetizar (véase la Figura 84).



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 84: Definición de la variable PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

Luego, se debe ejecutar el *script* con el comando que se muestra en el Cuadro 39, desde una terminal ubicada en la carpeta `sintesis_logica`.

```
./ejecutar_sintesis_logica_2.bash
```

Cuadro 39: Comando para ejecutar el *script* ejecutar_sintesis_logica_2.bash

Después de ejecutar el *script*, si el diseño fue realizado correctamente y se siguieron todos los pasos, no debería aparecer ningún error en la terminal. Deberían mostrarse los reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O, y los archivos de salida deberían haberse creado en sus respectivos directorios (véanse las Figuras 85 y 86).

En este caso, como se puede observar, se cuenta con archivos de *outputs* iniciales y finales. En los *outputs* iniciales se almacenará el archivo generado en la primera etapa de síntesis, que corresponde al circuito sintetizado sin los pines de I/O, así como el archivo que resulta de la unión del circuito sintetizado con el módulo de pines de I/O. Por otro lado, en los *outputs* finales se almacenan los archivos que se crean al sintetizar el circuito previamente sintetizado junto con el módulo de pines de I/O.

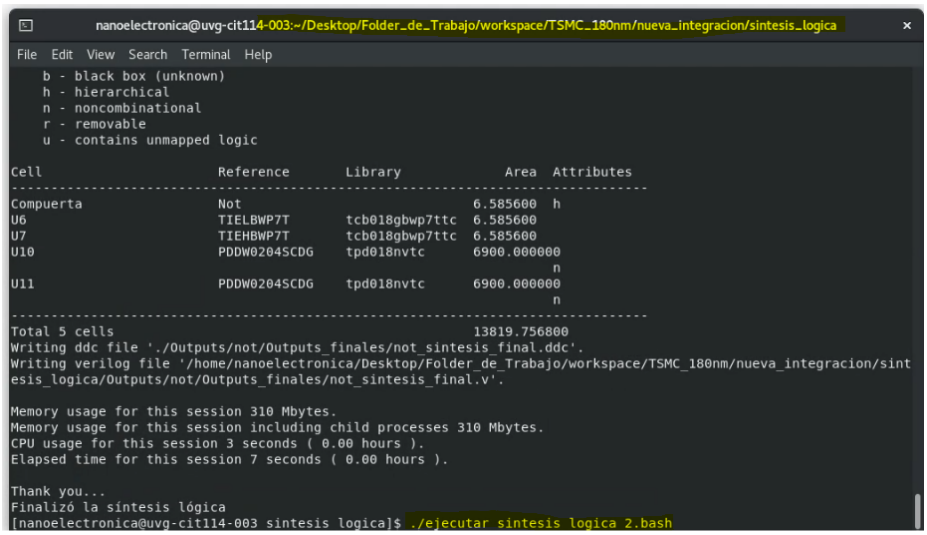
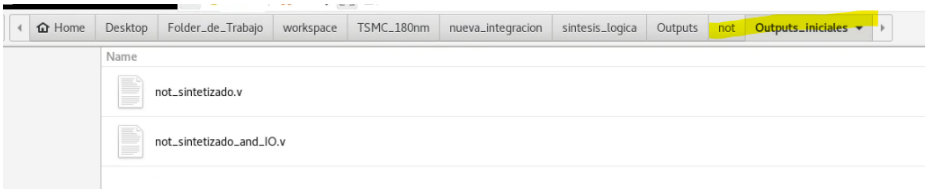
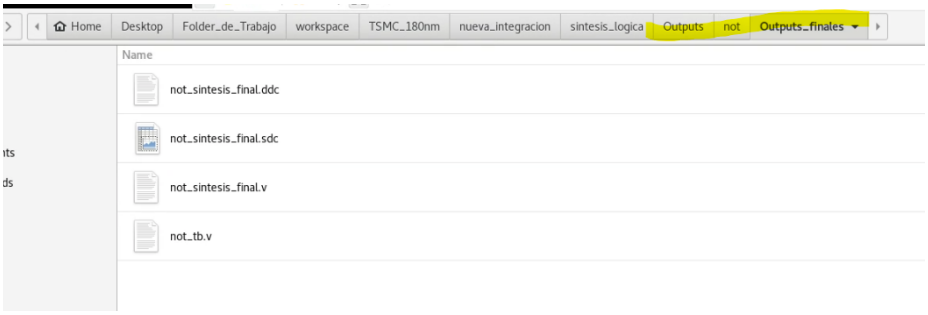


Figura 85: Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores

(a) Directorios generados en la carpeta del circuito en *outputs*



(b) Archivos generados en la carpeta del circuito en Outputs_iniciales



(c) Archivos generados en la carpeta del circuito en Outputs_finales

Figura 86: Archivos generados en la carpeta del circuito en *outputs*

En la Figura 87 se muestra el archivo de salida Verilog obtenido, que representa el

circuito sintetizado junto con el módulo de pines de I/O también sintetizado. Como se puede observar, este archivo hace referencia a módulos que utilizan celdas de TSMC con tecnología de 180 nm.

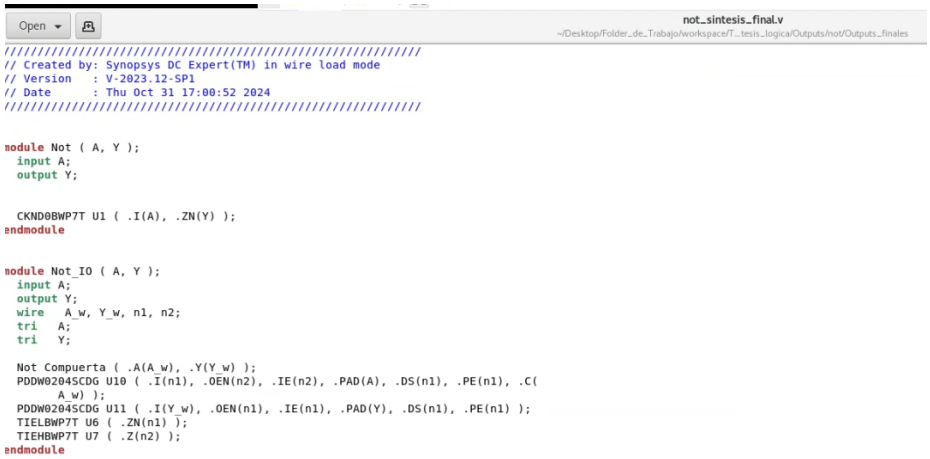


Figura 87: Archivo not_sintesis_final.v: circuito *not* con pines de I/O sintetizado

Una vez sintetizado el circuito, este también puede ser simulado, tal como se explicó en la sección 12.1. En el siguiente capítulo se detalla con mayor profundidad el proceso de simulación y la síntesis de distintos circuitos.

CAPÍTULO 13

Sintetización lógica de distintos circuitos descritos en Verilog utilizando el flujo de diseño A

En este capítulo se utilizaron diversos circuitos con el objetivo de comprobar que cualquier circuito descrito en Verilog puede ser sintetizado lógicamente. Para estas pruebas se emplearon los *scripts* desarrollados `comandos_sintesis_logica_2.script` y `ejecutar_sintesis_logica_2.bash`.

No se realizaron pruebas utilizando la interfaz gráfica de Design Vision, ya que este proceso es más largo. En cambio, al ejecutar un *script*, el circuito sintetizado se obtiene de manera más rápida y sencilla.

Se diseñaron los archivos que contienen el módulo del circuito a crear, así como su archivo de *testbench*, el cual se utilizó para la simulación tanto del circuito original como del sintetizado. Además, se diseñó el módulo que define los pines de entrada y salida. Asimismo, se realizaron simulaciones tanto de los circuitos originales como de los sintetizados.

13.1. Proceso para la síntesis lógica de una compuerta *not*

13.1.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench*

El primer paso a considerar es la creación de una carpeta dentro del directorio *inputs*, nombrada de acuerdo con el circuito principal que se va a sintetizar; en este caso, la carpeta fue llamada *not*. Posteriormente, se realizó el diseño de los archivos del circuito utilizando un lenguaje descriptivo de *hardware*; en este caso, Verilog, el cual permite construir un circuito en diferentes niveles de abstracción: *behavioral*, *structural* o RTL (*register transfer level*). Se

necesitó diseñar tres archivos:

- Archivo(s) Verilog que describen el circuito a sintetizar.
- Archivo Verilog de *testbench*, utilizado para la simulación del circuito.
- Archivo Verilog que contiene el módulo de pines de entrada y salida del circuito a sintetizar. Si este archivo no se encuentra, solo se realizará la síntesis del circuito original.

Para esta primera prueba se diseñó el circuito de una compuerta *not* a nivel de comportamiento (*behavioral*). Se optó por un circuito sencillo para facilitar la prueba inicial y así poder depurar posibles errores al realizar la síntesis lógica utilizando el *script* `ejecutar_sintesis_logica_2.bash`.

Si la síntesis lógica se completaba con éxito en esta primera prueba, el siguiente paso sería escalar a diseños y síntesis de circuitos más complejos, con el fin de comprobar que todo funcionara correctamente.

El archivo que contiene el módulo del circuito original se nombró `not.v`; el *testbench*, `not_tb.v`; y el archivo que contiene el módulo de entradas y salidas, `IO_not.v`, siguiendo la estructura explicada en el capítulo y en la sección 9.2. Es importante adherirse a esta estructura, ya que, de no hacerlo, el proceso de síntesis no se llevará a cabo correctamente.

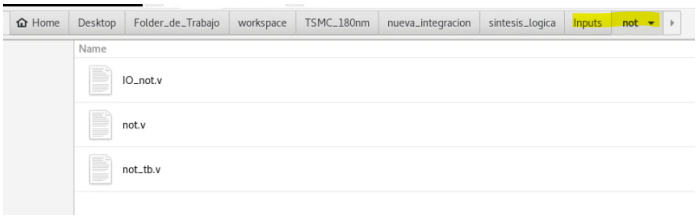


Figura 88: Archivos necesarios en la carpeta del circuito en *inputs*

En el Cuadro 40 se muestra el diseño del circuito, mientras que en el Cuadro 41 se presenta el diseño del módulo de pines de entrada y salida. Además, en el Cuadro 42 se muestra el *testbench* diseñado para la simulación tanto del circuito original como del circuito sintetizado.

```
module Not(A,Y);
    input A;
    output Y;
    assign Y = ~A;
endmodule
```

Cuadro 40: Archivo not.v: diseño de una una *not* en Verilog

```
    module Not_IO(A, Y);
        input A;
        output Y;
        wire A_w, Y_w;
        Not Compuerta(A_w,Y_w);

        //Entradas:
        PDDW0204SCDG U10(.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A), .C(A_w), .PE(1'
            b0), .IE(1'b1));
        //Salidas:
        PDDW0204SCDG U11(.I(Y_w), .DS(1'b0), .OEN(1'b0), .PAD(Y), .C(1'bx), .PE(1'
            b0), .IE(1'b0));

        //Pines de VDD y VSS
        PVDD1CDG PVDD();
        PVSS1CDG PVSS();
endmodule
```

Cuadro 41: Archivo IO_not.v: diseño de módulo de entradas y salidas para una *not* en Verilog

```
    module Not_tb;
    wire Y;
    reg A;
    Not compuerta(.A(A), .Y(Y));

    initial
    begin
        $fsdbDumpfile("not.fsdb");
        $fsdbDumpvars(0, Not_tb);
        $dumpfile("not.vcd");
        $dumpvars(0, Not_tb);
        A = 1'b0;
        #1;
        A = 1'b1;
        #1;
        A = 1'b0;
        #1;
        A = 1'b1;

    end
    endmodule
```

Cuadro 42: Archivo not_tb.v: diseño del *testbench* para simulación de una *not* en Verilog

13.1.2. Simulación del diseño original del circuito

Como siguiente paso, se simuló el circuito original con el objetivo de verificar que no haya errores de compilación y que el diseño del circuito sea correcto. Si no se realiza este paso, existe el riesgo de encontrar errores debido a posibles fallos en el diseño del circuito, lo que impediría proceder a la ejecución de la síntesis lógica.

Para simular el comportamiento del circuito, se utilizó la herramienta VCS, además de DVE para analizar el *waveform* del comportamiento del circuito. El funcionamiento de estas herramientas ya se explicó en el capítulo anterior.

Los comandos necesarios para generar y visualizar la simulación del circuito se presentan en el Cuadro 43.

```
vcs -v not.v not_tb.v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 43: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Siguiendo los pasos descritos en el capítulo y en la sección 12.1 para la visualización del archivo `.vcd` generado al ejecutar el archivo de simulación, se obtiene la Figura 89.

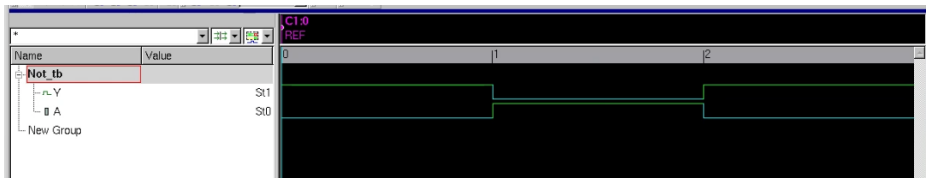
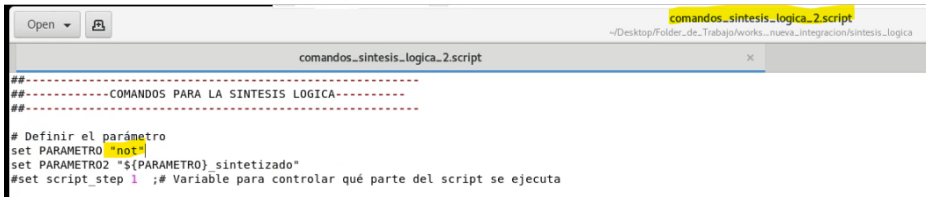


Figura 89: Simulación de una *not* en su diseño original

Como se puede observar en la Figura 89, el comportamiento del circuito es el esperado, ya que cuando la entrada `A` está en 1, la salida `Y` está en 0, y viceversa. Por lo tanto, se procedió a ejecutar el *script* para llevar a cabo la síntesis lógica del circuito.

13.1.3. Síntesis lógica del diseño original del circuito

Antes de proceder a ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, fue necesario abrir tanto este *script* como el `comandos_sintesis_logica_2.script` para definir en la variable `PARAMETRO` el nombre de la carpeta que se creó en *inputs*, la cual contiene el circuito a sintetizar. En este caso, se definió como *not*.



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 90: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

Luego, se ejecutó el *script* utilizando el comando que se muestra en el Cuadro 44, en una terminal abierta en el directorio sintesis_logica.

```
./ejecutar_sintesis_logica_2.bash
```

Cuadro 44: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica.bash

Al ejecutar este comando, en la terminal no apareció ningún error y se generaron los reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O. Además, en la carpeta not/Outputs_iniciales se creó un archivo llamado `not_sintetizado.v`, el cual corresponde al archivo resultante de la primera etapa de la síntesis lógica (ver Figuras 91 y 92).

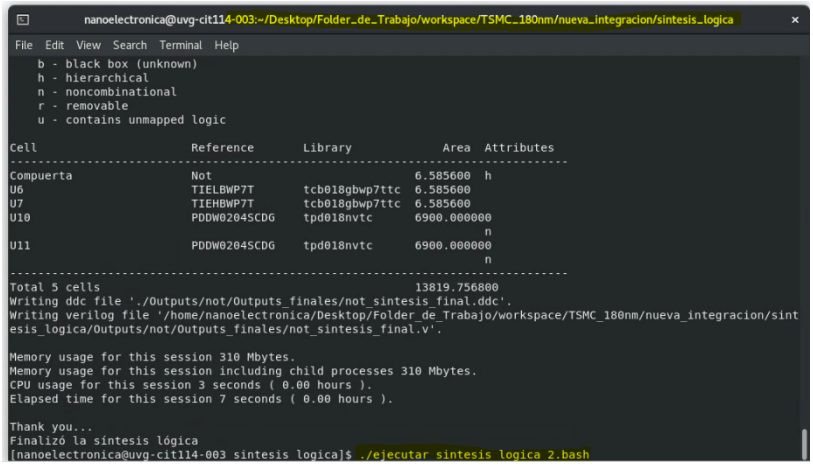


Figura 91: Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores

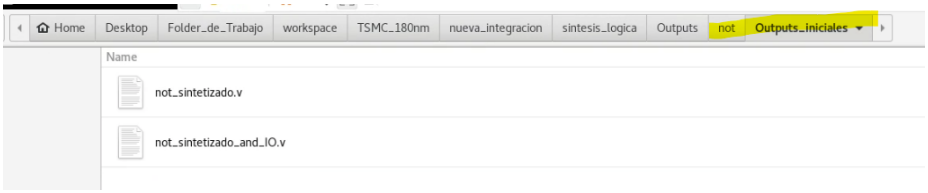


Figura 92: Archivos generados en la carpeta del circuito en Outputs_iniciales

El archivo `not_sintetizado.v`, que corresponde al circuito sintetizado, se muestra en el Cuadro 45.

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Fri Sep 13 19:30:10 2024
////////////////////////////////////////////////////////////

module Not ( A, Y );
  input A;
  output Y;


  CKND0BWP7T U2 ( .I(A), .ZN(Y) );
endmodule
```

Cuadro 45: Archivo not_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito

13.1.4. Simulación del circuito sintetizado

El siguiente paso que se realizó, y que es muy importante, fue la simulación del circuito sintetizado. Como se explicó en el capítulo anterior, aunque se puede llegar a sintetizar un circuito, eso no garantiza que el comportamiento del circuito sintetizado sea el esperado. Por lo tanto, es fundamental simularlo, tal como se hizo con el circuito original.

Para ello, se ejecutaron los siguientes comandos:

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v not_sintetizado.v not_tb.v -o
    simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 46: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Como se puede observar en el Cuadro 46, para simular el circuito sintetizado es importante hacer referencia a los archivos `tcb018gbwp7t.v` y `tpd018nv.v`, ya que estos contienen las celdas a las cuales los módulos del archivo `not_sintetizado.v` hacen referencia. Dado

que estos archivos se encuentran ubicados en el directorio de *outputs* y se está ejecutando el comando en una terminal abierta en el directorio `Outputs/not/Outputs_iniciales`, es necesario subir dos niveles en el sistema de directorios para poder encontrarlos; por esa razón, se utiliza `../../`.

Luego de haber ejecutado los comandos mencionados anteriormente, se logró observar el mismo comportamiento obtenido con el circuito original, lo que comprueba que la síntesis del circuito se realizó correctamente (ver Figura 93).

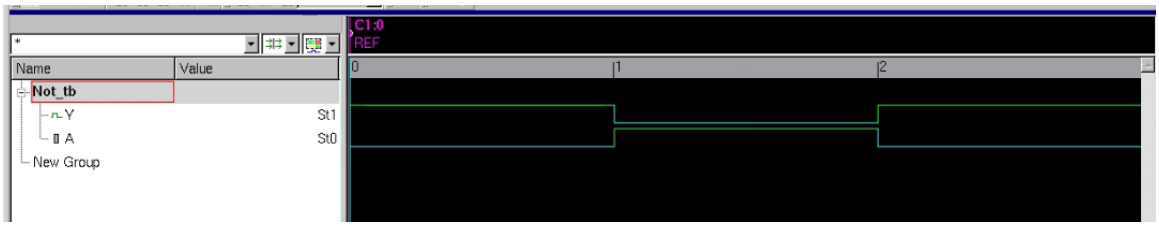


Figura 93: Simulación de la *not* en su diseño sintetizado

13.1.5. Unión del circuito sintetizado con el módulo de pines de I/O

El archivo `not_sintetizado_and_IO.v` se creó uniendo el archivo `not_sintetizado.v` con el archivo `IO_not.v`. Este proceso también lo realiza automáticamente el *script* antes de comenzar el segundo proceso de síntesis lógica.

Por ello, es fundamental tener diseñado desde el principio el archivo `IO_not.v` y guardarlo en la carpeta `Inputs/not`. Si este archivo no está presente, el *script* no podrá localizarlo y el nuevo archivo generado contendrá únicamente el contenido de `not_sintetizado.v`. Esto ocasionará que, en el siguiente paso, se realice nuevamente la síntesis sin incluir el módulo de pines de entrada y salida, obteniéndose un resultado similar al archivo `not_sintetizado.v`.

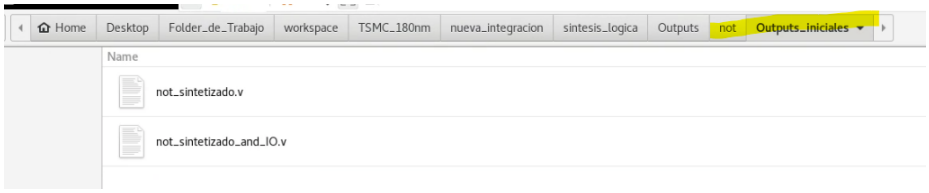


Figura 94: Archivos generados en la carpeta del circuito en Outputs_iniciales

En el Cuadro 47 se muestra el archivo generado. Este es el archivo que se utilizará para obtener el circuito completamente sintetizado con pines de entrada y salida.

```
module Not_IO(A, Y);
       input A;
       output Y;
       wire A_w, Y_w;
       Not Compuerta(A_w,Y_w);
       //Configuracion de PIN 1/0
       //Entradas (C):
       //PDDW0204SCDG U10(1'b0,1'b0, 1'b1,A,A_w,1'b0,1'b1);
```

```
        //Salidas (I):
        //PDDW0204SCDG U11(Y_w,1'b0, 1'b0, Y, 1'bx, 1'b0,1'b0);

        //Entradas:
        PDDW0204SCDG U10(.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A), .C(A_w), .PE(1'
            b0), .IE(1'b1));
        //Salidas:
        PDDW0204SCDG U11(.I(Y_w), .DS(1'b0), .OEN(1'b0), .PAD(Y), .C(1'bx), .PE(1'
            b0), .IE(1'b0));

        //Pines de VDD y VSS
        PVDD1CDG PVDD();
        PVSS1CDG PVSS();
endmodule


////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Fri Sep 13 19:30:10 2024
////////////////////////////////////////////////////////////


module Not ( A, Y );
  input A;
  output Y;


  CKND0BWP7T U2 ( .I(A), .ZN(Y) );
endmodule
```

Cuadro 47: Archivo not_sintetizado_and_IO.v: circuito sintetizado con pines de I/O

13.1.6. Síntesis lógica del circuito sintetizado con pines de I/O

A través del archivo `not_sintetizado_and_IO.v` se obtuvo el circuito completamente sintetizado. Este archivo se generó en el directorio `Outputs/not/Outputs_finales` con el nombre `not_sintesis_final.v`. Además, se generaron otros dos archivos que también servirán para el siguiente proceso de síntesis física.

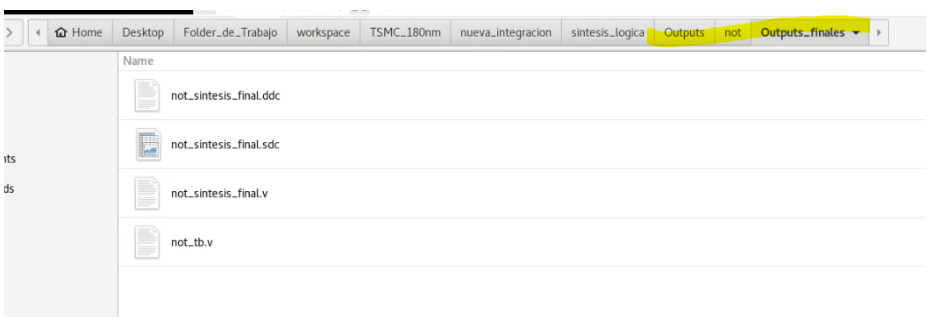


Figura 95: Archivos generados en la carpeta del circuito en Outputs_finales

En el Cuadro 48 se muestra el archivo Verilog obtenido. Como se puede observar, este último archivo incluye el módulo del circuito *not* sintetizado y también contiene el módulo de los pines de I/O sintetizados, ya que ambos hacen referencia a celdas de TSMC.

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Fri Sep 13 19:30:17 2024
////////////////////////////////////////////////////////////

module Not ( A, Y );
    input A;
    output Y;
    CKND0BWP7T U1 ( .I(A), .ZN(Y) );
endmodule

module Not_IO ( A, Y );
    input A;
    output Y;
    wire A_w, Y_w, n1, n2;
    tri A;
    tri Y;

    Not Compuerta ( .A(A_w), .Y(Y_w) );
    PDDW0204SCDG U10 ( .I(n1), .OEN(n2), .IE(n2), .PAD(A), .DS(n1), .PE(n1), .C(
        A_w) );
    PDDW0204SCDG U11 ( .I(Y_w), .OEN(n1), .IE(n1), .PAD(Y), .DS(n1), .PE(n1) );
    TIELBWP7T U6 ( .ZN(n1) );
    TIEHBWP7T U7 ( .Z(n2) );
endmodule
```

Cuadro 48: Archivo not_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O

13.1.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O

El último paso que se realizó fue la simulación del circuito completamente sintetizado junto con los pines de I/O. Esto se llevó a cabo nuevamente para comprobar que el circuito completamente sintetizado se comporte correctamente.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 49.

```
    vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v not_sintesis_final.v not_tb.v -o
        simulacion -full64 -debug_access+all

    ./simulacion

    dve -full64
```

Cuadro 49: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en la Figura 96, el comportamiento es idéntico al que se obtuvo anteriormente, lo que confirma que el *netlist* final obtenido del circuito sintetizado con pines de I/O es correcto.

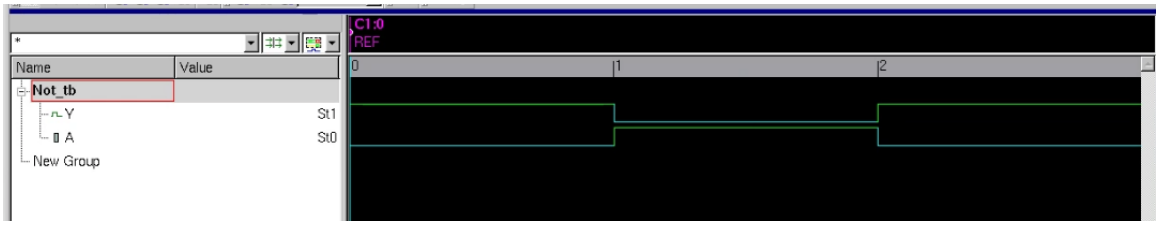


Figura 96: Simulación del *netlist* final

13.2. Proceso para la síntesis lógica de una ALU de 4 bits

13.2.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench*

Después de diseñar, sintetizar y simular un circuito sencillo, como una *not*, se procedió a escalar el proyecto al diseño y síntesis de un circuito más complejo: una ALU (unidad aritmético-lógica) de 4 bits.

El primer paso fue crear una carpeta en el directorio de *inputs*, nombrada según el circuito principal a sintetizar, en este caso, `ALU4b` (tanto la carpeta como el archivo Verilog que contiene el circuito principal deben tener el mismo nombre). Luego, se diseñaron los archivos necesarios para la simulación y síntesis del circuito.

Siguiendo la estructura explicada en el capítulo y sección 9.2, el archivo que contiene el módulo de la ALU de 4 bits se nombró `ALU4b.v`. El archivo de *testbench*, utilizado para simular y verificar el funcionamiento del circuito, se denominó `ALU4b_tb.v`. Por último, el archivo que define el módulo de entradas y salidas se nombró `IO_ALU4b.v`.

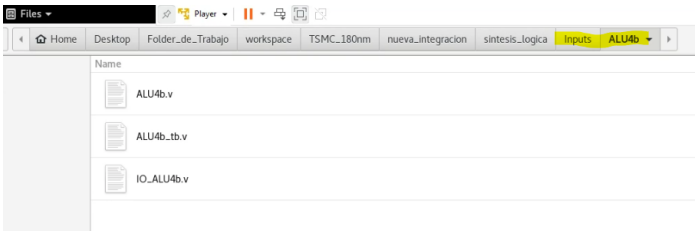


Figura 97: Archivos necesarios en la carpeta del circuito en *inputs*

En el Cuadro 50 se muestra el diseño del circuito, mientras que en el Cuadro 51 se presenta el diseño del módulo de pines de entrada y salida. Además, en el Cuadro 52 se muestra el *testbench* diseñado para la simulación del circuito original y del circuito sintetizado.

Dado que el diseño del circuito incluye múltiples entradas y salidas, en el módulo de pines es necesario hacer referencia a cada una de ellas de forma individual. Esto es necesario

porque la ALU es de 4 bits, por lo que tanto las entradas A y B como la salida son de 4 bits. Adicionalmente, se cuenta con un selector de entrada de 3 bits y una salida de acarreo. Sin embargo, el proceso de creación de cada pin es el mismo para cada entrada y salida (ver Cuadro 51).

```
module ALU_4bit (
    input [3:0] A, B, // Entradas de 4 bits
    input [2:0] ALU_Sel, // Seleccin de operacion (3 bits para seleccionar
        entre 8 operaciones)
    output reg [3:0] ALU_Out, // Salida de 4 bits
    output reg Carry_Out // Bandera de acarreo para operaciones de suma/resta
);

always @(*) begin
    case (ALU_Sel)
        3'b000: {Carry_Out, ALU_Out} = A + B; // Suma
        3'b001: {Carry_Out, ALU_Out} = A - B; // Resta
        3'b010: ALU_Out = A & B; // AND
        3'b011: ALU_Out = A | B; // OR
        3'b100: ALU_Out = A ^ B; // XOR
        3'b101: ALU_Out = ~A; // NOT
        3'b110: ALU_Out = A << 1; // Shift left
        3'b111: ALU_Out = A >> 1; // Shift right
        default: ALU_Out = 4'b0000; // Default case
    endcase
end
endmodule
```

Cuadro 50: Archivo ALU4b.v: diseño de una ALU de 4 bits en Verilog

```
module ALU_4bit_IO (
    input [3:0] A,
    input [3:0] B,
    input [2:0] ALU_Sel,
    output [3:0] ALU_Out,
    output Carry_Out
);
    wire [3:0] A_w, B_w, ALU_Out_w;
    wire [2:0] ALU_Sel_w;
    wire Carry_Out_w;

    // Instancia del modulo ALU_4bit
    ALU_4bit ALU_inst (
        .A(A_w),
        .B(B_w),
        .ALU_Sel(ALU_Sel_w),
        .ALU_Out(ALU_Out_w),
        .Carry_Out(Carry_Out_w)
    );

    // Interconexion de pines de entrada (A, B, ALU_Sel)
    // Entradas A
    PDDW0204SCDG U_A0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A[0]), .C(A_w[0]),
         .PE(1'b0), .IE(1'b1));
```

```
    PDDW0204SCDG U_A1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A[1]), .C(A_w[1]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_A2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A[2]), .C(A_w[2]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_A3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A[3]), .C(A_w[3]),
        .PE(1'b0), .IE(1'b1));

    // Entradas B
    PDDW0204SCDG U_B0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(B[0]), .C(B_w[0]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_B1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(B[1]), .C(B_w[1]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_B2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(B[2]), .C(B_w[2]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_B3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(B[3]), .C(B_w[3]),
        .PE(1'b0), .IE(1'b1));

    // Entrada ALU_Sel
    PDDW0204SCDG U_S0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(ALU_Sel[0]), .C(
        ALU_Sel_w[0]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_S1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(ALU_Sel[1]), .C(
        ALU_Sel_w[1]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_S2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(ALU_Sel[2]), .C(
        ALU_Sel_w[2]), .PE(1'b0), .IE(1'b1));

    // Interconexion de pines de salida (ALU_Out, Carry_Out)
    // Salidas ALU_Out
    PDDW0204SCDG U_Out0 (.I(ALU_Out_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(ALU_Out
        [0]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Out1 (.I(ALU_Out_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(ALU_Out
        [1]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Out2 (.I(ALU_Out_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(ALU_Out
        [2]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Out3 (.I(ALU_Out_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(ALU_Out
        [3]), .C(1'bx), .PE(1'b0), .IE(1'b0));

    // Salida Carry_Out
    PDDW0204SCDG U_Carry (.I(Carry_Out_w), .DS(1'b0), .OEN(1'b0), .PAD(
        Carry_Out), .C(1'bx), .PE(1'b0), .IE(1'b0));


//Pines de VDD y VSS
PVDD1CDG PVDD();
PVSS1CDG PVSS();

endmodule
```

Cuadro 51: Archivo IO_ALU4b.v: diseño de módulo de entradas y salidas para una ALU de 4 bits en Verilog

```
module ALU_4bit_tb;
    reg [3:0] A, B; // Entradas de 4 bits
    reg [2:0] ALU_Sel; // Signal de seleccion de operacion
    wire [3:0] ALU_Out; // Salida de 4 bits
    wire Carry_Out; // Bandera de acarreo
    // Instancia del modulo ALU_4bit
    ALU_4bit uut (
        .A(A),
        .B(B),
        .ALU_Sel(ALU_Sel),
        .ALU_Out(ALU_Out),
        .Carry_Out(Carry_Out)
    );
    initial begin
        $fsdbDumpfile("alu.fsdb");
        $fsdbDumpvars(0, ALU_4bit_tb);
        $dumpfile("alu.vcd");
        $dumpvars(0, ALU_4bit_tb);

        // Inicializacion de las senales
        A = 4'b0000;
        B = 4'b0000;
        ALU_Sel = 3'b000;

        // Prueba de la operacion de suma
        A = 4'b0011; B = 4'b0101; ALU_Sel = 3'b000; // A + B
        #10;
        // Prueba de la operacion de resta
        A = 4'b0100; B = 4'b0011; ALU_Sel = 3'b001; // A - B
        #10;
        // Prueba de la operacion AND
        A = 4'b1100; B = 4'b1010; ALU_Sel = 3'b010; // A & B
        #10;
        // Prueba de la operacion OR
        A = 4'b1100; B = 4'b1010; ALU_Sel = 3'b011; // A | B
        #10;
        // Prueba de la operacion XOR
        A = 4'b1100; B = 4'b1010; ALU_Sel = 3'b100; // A ^ B
        #10;
        // Prueba de la operacion NOT
        A = 4'b1100; ALU_Sel = 3'b101; // ~A
        #10;
        // Prueba del desplazamiento a la izquierda
        A = 4'b1100; ALU_Sel = 3'b110; // A << 1
        #10;
        // Prueba del desplazamiento a la derecha
        A = 4'b1100; ALU_Sel = 3'b111; // A >> 1
        #10;
        $finish;
    end
endmodule
```

Cuadro 52: Archivo ALU4b_tb.v: diseño del *testbench* para simulación de una ALU de 4 bits en Verilog

13.2.2. Simulación del diseño original del circuito

Como siguiente paso, se simuló el circuito original con el objetivo de verificar que no haya errores de compilación y que el diseño del circuito sea correcto. Como se explicó anteriormente, si no se realiza este paso, existe el riesgo de encontrar errores debido a posibles fallos en el diseño del circuito, lo que impediría proceder a la ejecución de la síntesis lógica.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 53 en una terminal abierta en `Inputs/ALU4b/`.

```
vcs -v ALU4b.v ALU4b_tb.v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 53: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Siguiendo los pasos descritos en el capítulo y sección 12.1 para la visualización del archivo `.vcd` generado al ejecutar el archivo de simulación, se obtuvo la Figura 98.

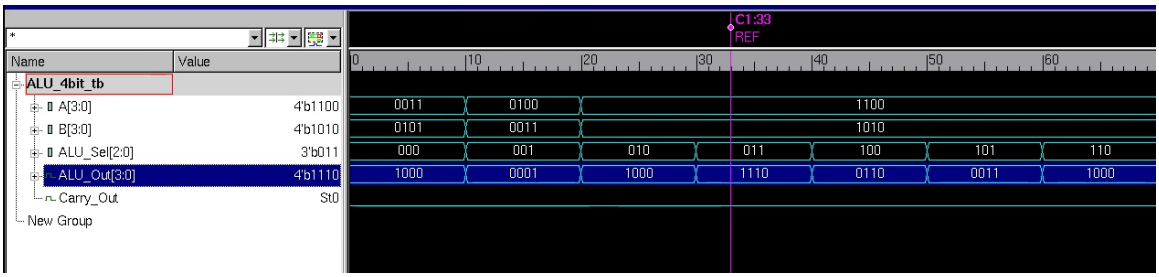


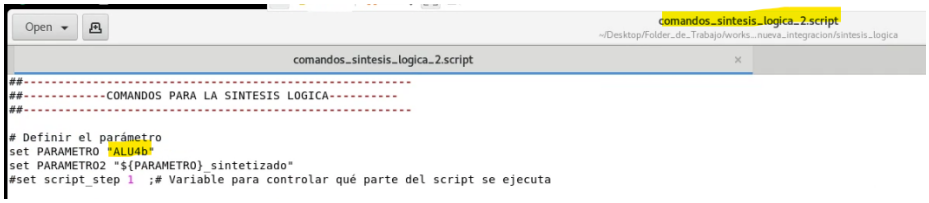
Figura 98: Simulación de una ALU de 4 bits en su diseño original

Como se puede observar en la Figura 98, el comportamiento del circuito es el esperado, ya que, dependiendo del selector (`ALU_Sel`), se cumple cierta operación entre las entradas A y B. Por ejemplo, en la primera operación, cuando `ALU_Sel` = 3'b000, `A` = 4'b0011 y `B` = 4'b0101, en la salida `ALU_Out` se obtiene 4'b1000, resultado de la suma entre `A` y `B`. Los resultados continúan de manera similar para las demás operaciones.

Por lo tanto, se procedió a ejecutar el *script* para llevar a cabo la síntesis lógica del circuito.

13.2.3. Síntesis lógica del diseño original del circuito

Antes de proceder a ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, fue necesario abrir tanto este *script* como el archivo `comandos_sintesis_logica_2.script` para definir en la variable `PARAMETRO` el nombre de la carpeta creada en *inputs*, la cual contiene el circuito a sintetizar. En este caso, se definió como `ALU4b`.



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 99: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

Luego, se ejecutó el *script* utilizando el comando que se muestra en el Cuadro 54, en una terminal abierta en el directorio `sintesis_logica`.

```
./ejecutar_sintesis_logica_2.bash
```

Cuadro 54: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash

Al ejecutar este comando, se creó una carpeta en *outputs* con el nombre `ALU4b`. Además, se generaron dos subdirectorios dentro de esta, denominados `outputs_iniciales` y `outputs_finales`. En la terminal no apareció ningún error, y se generaron los reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O.

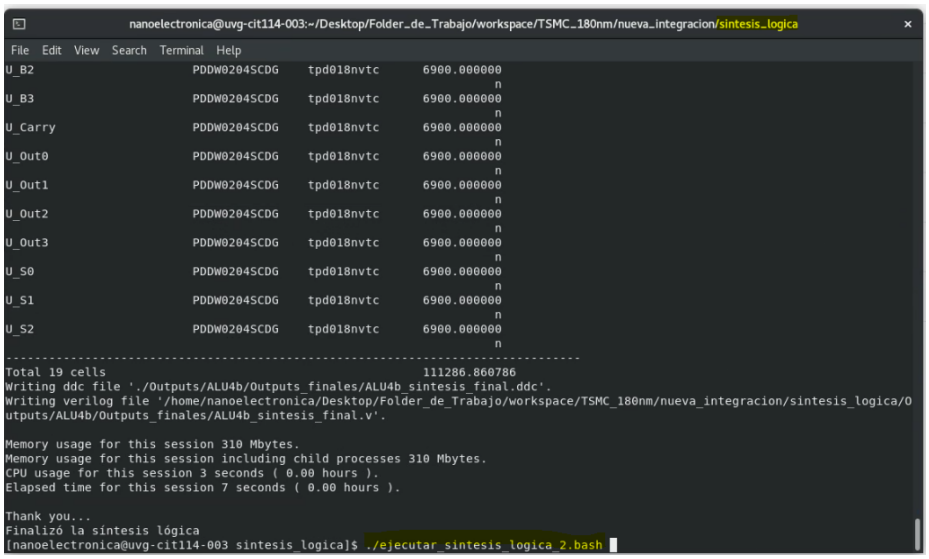


Figura 100: Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores

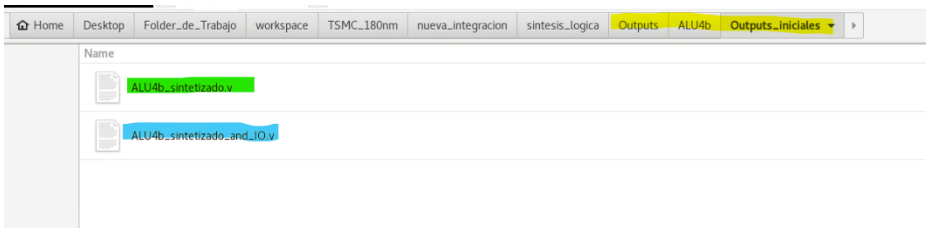


Figura 101: Archivos generados en la carpeta del circuito en Outputs_iniciales

El archivo `ALU4b_sintetizado.v`, que corresponde al circuito sintetizado, se muestra en el Cuadro 55.

```
////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Fri Sep 13 19:35:05 2024
////////////////////////////////////////////////////////////////


module ALU_4bit ( A, B, ALU_Sel, ALU_Out, Carry_Out );
  input [3:0] A;
  input [3:0] B;
  input [2:0] ALU_Sel;
  output [3:0] ALU_Out;
  output Carry_Out;
  wire N57, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58,
        n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72,
        n73, n74, n75, n76, n77, n78, n79, n80, n81, n82, n83, n84, n85, n86,
        n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100,
        n101, n102, n103, n104, n105, n106, n107, n108, n109, n110, n111,
        n112, n113, n114, n115, n116, n117;
```

```
    LHQD1BWP7T Carry_Out_reg ( .E(N57), .D(n117), .Q(Carry_Out) );
    OAI21D0BWP7T U51 ( .A1(n46), .A2(n47), .B(n48), .ZN(n117) );
    MUX2ND0BWP7T U52 ( .I0(n49), .I1(n50), .S(n51), .ZN(n48) );
    NR2D0BWP7T U53 ( .A1(n52), .A2(n53), .ZN(n50) );
    NR2D0BWP7T U54 ( .A1(A[3]), .A2(n54), .ZN(n49) );
    MUX2ND0BWP7T U55 ( .I0(ALU_Sel[0]), .I1(n55), .S(A[3]), .ZN(n47) );
    CKND2D0BWP7T U56 ( .A1(ALU_Sel[0]), .A2(n54), .ZN(n55) );
    ND3D0BWP7T U57 ( .A1(n56), .A2(n57), .A3(n58), .ZN(ALU_Out[3]) );
    OA32D0BWP7T U58 ( .A1(n59), .A2(ALU_Sel[0]), .A3(n60), .B1(n61), .B2(n53),
         .Z(n58) );
    MUX2ND0BWP7T U59 ( .I0(n62), .I1(n63), .S(A[3]), .ZN(n57) );
    NR2D0BWP7T U60 ( .A1(n46), .A2(n64), .ZN(n63) );
    CKND0BWP7T U61 ( .I(B[3]), .ZN(n46) );
    MUX2ND0BWP7T U62 ( .I0(n65), .I1(n66), .S(n67), .ZN(n56) );
    AOI21D0BWP7T U63 ( .A1(B[3]), .A2(A[3]), .B(n53), .ZN(n67) );
    NR2D0BWP7T U64 ( .A1(B[3]), .A2(A[3]), .ZN(n53) );
    OAI221D0BWP7T U65 ( .A1(n68), .A2(n69), .B1(n70), .B2(n71), .C(n72), .ZN(n66)
        );
    CKND0BWP7T U66 ( .I(n54), .ZN(n68) );
    OAI22D0BWP7T U67 ( .A1(n52), .A2(n70), .B1(n54), .B2(n69), .ZN(n65) );
    OAI21D0BWP7T U68 ( .A1(n73), .A2(n59), .B(n74), .ZN(n54) );
    OAI21D0BWP7T U69 ( .A1(A[2]), .A2(n75), .B(n76), .ZN(n74) );
    CKND0BWP7T U70 ( .I(n71), .ZN(n52) );
    OAI22D0BWP7T U71 ( .A1(n59), .A2(n76), .B1(n77), .B2(n78), .ZN(n71) );
    ND3D0BWP7T U72 ( .A1(n79), .A2(n80), .A3(n81), .ZN(ALU_Out[2]) );
    AOI22D0BWP7T U73 ( .A1(n82), .A2(n83), .B1(n62), .B2(n59), .ZN(n81) );
    CKMUX2D0BWP7T U74 ( .I0(A[1]), .I1(A[3]), .S(ALU_Sel[0]), .Z(n82) );
    MUX2ND0BWP7T U75 ( .I0(n84), .I1(n85), .S(n76), .ZN(n80) );
    CKND0BWP7T U76 ( .I(B[2]), .ZN(n76) );
    NR2D0BWP7T U77 ( .A1(n86), .A2(n59), .ZN(n85) );
    MUX2ND0BWP7T U78 ( .I0(n87), .I1(n86), .S(n59), .ZN(n84) );
    OA221D0BWP7T U79 ( .A1(n73), .A2(n69), .B1(n70), .B2(n88), .C(n72), .Z(n86)
         );
    CKND0BWP7T U80 ( .I(n75), .ZN(n73) );
    NR2D0BWP7T U81 ( .A1(n89), .A2(n90), .ZN(n87) );
    MUX2ND0BWP7T U82 ( .I0(n91), .I1(n90), .S(n78), .ZN(n79) );
    NR2D0BWP7T U83 ( .A1(B[2]), .A2(A[2]), .ZN(n78) );
    OAI22D0BWP7T U84 ( .A1(n77), .A2(n70), .B1(n75), .B2(n69), .ZN(n90) );
    OAI21D0BWP7T U85 ( .A1(n92), .A2(n93), .B(n94), .ZN(n75) );
    OAI21D0BWP7T U86 ( .A1(A[1]), .A2(n95), .B(n96), .ZN(n94) );
    CKND0BWP7T U87 ( .I(n88), .ZN(n77) );
    OAI22D0BWP7T U88 ( .A1(n93), .A2(n96), .B1(n97), .B2(n98), .ZN(n88) );
    ND3D0BWP7T U89 ( .A1(n99), .A2(n100), .A3(n101), .ZN(ALU_Out[1]) );
    AOI22D0BWP7T U90 ( .A1(n102), .A2(n83), .B1(n62), .B2(n93), .ZN(n101) );
    CKND0BWP7T U91 ( .I(n60), .ZN(n83) );
    MUX2ND0BWP7T U92 ( .I0(n59), .I1(n103), .S(n51), .ZN(n102) );
    CKND0BWP7T U93 ( .I(A[2]), .ZN(n59) );
    MUX2ND0BWP7T U94 ( .I0(n104), .I1(n105), .S(n96), .ZN(n100) );
    CKND0BWP7T U95 ( .I(B[1]), .ZN(n96) );
    NR2D0BWP7T U96 ( .A1(n106), .A2(n93), .ZN(n105) );
    MUX2ND0BWP7T U97 ( .I0(n107), .I1(n106), .S(n93), .ZN(n104) );
    OA221D0BWP7T U98 ( .A1(n92), .A2(n69), .B1(n108), .B2(n70), .C(n72), .Z(n106)
        );
    NR2D0BWP7T U99 ( .A1(n89), .A2(n109), .ZN(n107) );
```

```
    CKND0BWP7T U100 ( .I(n64), .ZN(n89) );
    MUX2ND0BWP7T U101 ( .I0(n91), .I1(n109), .S(n97), .ZN(n99) );
    NR2D0BWP7T U102 ( .A1(B[1]), .A2(A[1]), .ZN(n97) );
    OAI22D0BWP7T U103 ( .A1(n98), .A2(n70), .B1(n95), .B2(n69), .ZN(n109) );
    CKND0BWP7T U104 ( .I(n92), .ZN(n95) );
    OAI211D0BWP7T U105 ( .A1(n110), .A2(n61), .B(n111), .C(n112), .ZN(ALU_Out[0])
        );
    AOI21D0BWP7T U106 ( .A1(n92), .A2(n113), .B(n114), .ZN(n112) );
    OAI32D0BWP7T U107 ( .A1(n60), .A2(n93), .A3(n51), .B1(n98), .B2(n64), .ZN(
        n114) );
    CKND0BWP7T U108 ( .I(n108), .ZN(n98) );
    NR2D0BWP7T U109 ( .A1(n103), .A2(n110), .ZN(n108) );
    CKND0BWP7T U110 ( .I(A[1]), .ZN(n93) );
    CKND2D0BWP7T U111 ( .A1(ALU_Sel[2]), .A2(ALU_Sel[1]), .ZN(n60) );
    NR2D0BWP7T U112 ( .A1(n110), .A2(A[0]), .ZN(n92) );
    MUX2ND0BWP7T U113 ( .I0(n115), .I1(n62), .S(n103), .ZN(n111) );
    CKND0BWP7T U114 ( .I(A[0]), .ZN(n103) );
    NR3D0BWP7T U115 ( .A1(n51), .A2(ALU_Sel[1]), .A3(n116), .ZN(n62) );
    IOA21D0BWP7T U116 ( .A1(n110), .A2(n113), .B(n61), .ZN(n115) );
    ND3D0BWP7T U117 ( .A1(n70), .A2(n72), .A3(n69), .ZN(n113) );
    CKND2D0BWP7T U118 ( .A1(N57), .A2(ALU_Sel[0]), .ZN(n69) );
    IND3D0BWP7T U119 ( .A1(ALU_Sel[1]), .B1(n51), .B2(ALU_Sel[2]), .ZN(n72) );
    CKND2D0BWP7T U120 ( .A1(N57), .A2(n51), .ZN(n70) );
    NR2D0BWP7T U121 ( .A1(ALU_Sel[1]), .A2(ALU_Sel[2]), .ZN(N57) );
    CKND0BWP7T U122 ( .I(n91), .ZN(n61) );
    NR2D0BWP7T U123 ( .A1(n64), .A2(n51), .ZN(n91) );
    CKND0BWP7T U124 ( .I(ALU_Sel[0]), .ZN(n51) );
    CKND2D0BWP7T U125 ( .A1(ALU_Sel[1]), .A2(n116), .ZN(n64) );
    CKND0BWP7T U126 ( .I(ALU_Sel[2]), .ZN(n116) );
    CKND0BWP7T U127 ( .I(B[0]), .ZN(n110) );
endmodule
```

Cuadro 55: Archivo ALU4b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito

13.2.4. Simulación del circuito sintetizado

El siguiente paso que se llevó a cabo, y que es fundamental para corroborar que el circuito sintetizado se comporte correctamente, fue la simulación del circuito sintetizado.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 56 en una terminal abierta en `Outputs/ALU4b/Outputs_iniciales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v ALU4b_sintetizado.v ALU4b_tb.v -o
    simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 56: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en el Cuadro 56, para simular el circuito sintetizado es importante hacer referencia a los archivos `tcb018gbwp7t.v` y `tpd018nv.v`, ya que estos contienen las celdas a las cuales los módulos del archivo `ALU4b_sintetizado.v` hacen referencia. Dado que estos archivos se encuentran ubicados en el directorio de *outputs*, y se está ejecutando el comando en una terminal abierta en el directorio `Outputs/ALU4b/Outputs_iniciales`, es necesario subir dos niveles en el sistema de directorios para poder encontrarlos; por esta razón, se utiliza la ruta `../../`.

Luego de haber ejecutado los comandos mencionados anteriormente, se obtuvo la Figura 102. Se observó el mismo comportamiento que en el circuito original, lo que confirma que la síntesis del circuito se realizó correctamente.

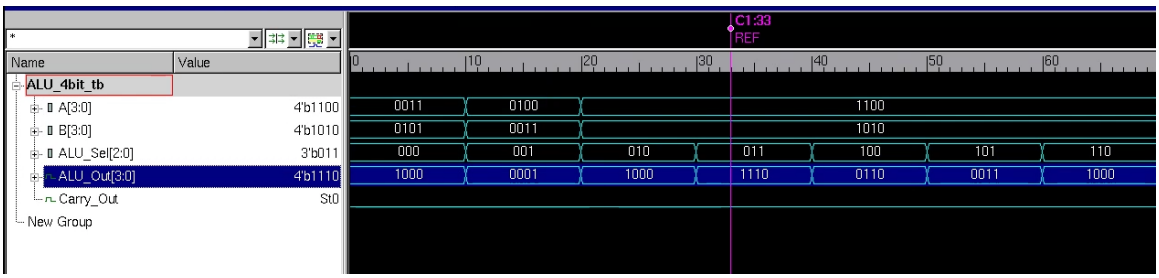


Figura 102: Simulación de la ALU de 4 bits en su diseño sintetizado

13.2.5. Unión del circuito sintetizado con el módulo de pines de I/O

El archivo denominado `ALU4b_sintetizado_and_IO.v`, se creó uniendo el archivo `ALU4b_sintetizado.v` con el archivo `IO_ALU4b.v`. Este proceso también lo ejecuta automáticamente el *script* antes de comenzar el segundo proceso de síntesis lógica.

Por ello, es fundamental tener diseñado desde el inicio el archivo `IO_ALU4b.v` y guardarlo en la carpeta `Inputs/ALU4b`. Si este archivo no está presente, el *script* no podrá localizarlo y el nuevo archivo generado contendrá únicamente el contenido de `ALU4b_sintetizado.v`. Esto ocasionará que, en el siguiente paso, se realice nuevamente la síntesis sin incluir el módulo de pines de entrada y salida, obteniéndose un resultado similar al archivo `ALU4b_sintetizado.v`.

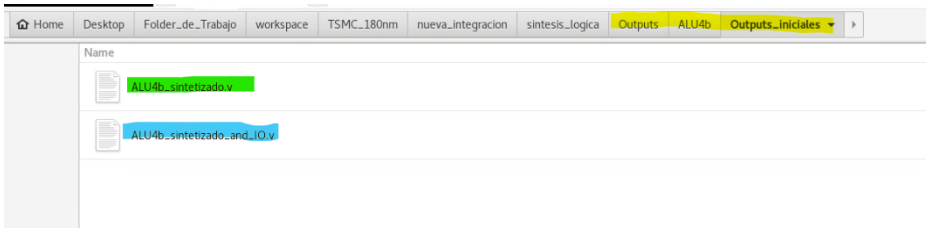


Figura 103: Archivos generados en la carpeta del circuito en Outputs_iniciales

En el Cuadro 57 se muestra el archivo generado. Este archivo se utilizará para obtener el circuito completamente sintetizado con pines de entrada y salida.

```
module ALU_4bit_IO (
    input [3:0] A,
    input [3:0] B,
    input [2:0] ALU_Sel,
    output [3:0] ALU_Out,
    output Carry_Out
);
    wire [3:0] A_w, B_w, ALU_Out_w;
    wire [2:0] ALU_Sel_w;
    wire Carry_Out_w;

    // Instancia del modulo ALU_4bit
    ALU_4bit ALU_inst (
        .A(A_w),
        .B(B_w),
        .ALU_Sel(ALU_Sel_w),
        .ALU_Out(ALU_Out_w),
        .Carry_Out(Carry_Out_w)
    );

    // Interconexion de pines de entrada (A, B, ALU_Sel)
    // Entradas A
    PDDW0204SCDG U_A0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A[0]), .C(A_w[0]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_A1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A[1]), .C(A_w[1]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_A2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A[2]), .C(A_w[2]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_A3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(A[3]), .C(A_w[3]),
         .PE(1'b0), .IE(1'b1));

    // Entradas B
    PDDW0204SCDG U_B0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(B[0]), .C(B_w[0]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_B1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(B[1]), .C(B_w[1]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_B2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(B[2]), .C(B_w[2]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_B3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(B[3]), .C(B_w[3]),
         .PE(1'b0), .IE(1'b1));

    // Entrada ALU_Sel
    PDDW0204SCDG U_S0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(ALU_Sel[0]), .C(
        ALU_Sel_w[0]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_S1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(ALU_Sel[1]), .C(
        ALU_Sel_w[1]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_S2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(ALU_Sel[2]), .C(
        ALU_Sel_w[2]), .PE(1'b0), .IE(1'b1));

    // Interconexion de pines de salida (ALU_Out, Carry_Out)
    // Salidas ALU_Out
    PDDW0204SCDG U_Out0 (.I(ALU_Out_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(ALU_Out
        [0]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Out1 (.I(ALU_Out_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(ALU_Out
```

```
        [1]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Out2 (.I(ALU_Out_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(ALU_Out
        [2]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Out3 (.I(ALU_Out_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(ALU_Out
        [3]), .C(1'bx), .PE(1'b0), .IE(1'b0));

    // Salida Carry_Out
    PDDW0204SCDG U_Carry (.I(Carry_Out_w), .DS(1'b0), .OEN(1'b0), .PAD(
        Carry_Out), .C(1'bx), .PE(1'b0), .IE(1'b0));


//Pines de VDD y VSS
PVDD1CDG PVDD();
PVSS1CDG PVSS();

endmodule

////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Fri Sep 13 19:35:05 2024
////////////////////////////////////////////////////////////

module ALU_4bit ( A, B, ALU_Sel, ALU_Out, Carry_Out );
  input [3:0] A;
  input [3:0] B;
  input [2:0] ALU_Sel;
  output [3:0] ALU_Out;
  output Carry_Out;
  wire N57, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58,
        n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72,
        n73, n74, n75, n76, n77, n78, n79, n80, n81, n82, n83, n84, n85, n86,
        n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100,
        n101, n102, n103, n104, n105, n106, n107, n108, n109, n110, n111,
        n112, n113, n114, n115, n116, n117;

  LHQD1BWP7T Carry_Out_reg ( .E(N57), .D(n117), .Q(Carry_Out) );
  OAI21D0BWP7T U51 ( .A1(n46), .A2(n47), .B(n48), .ZN(n117) );
  MUX2ND0BWP7T U52 ( .I0(n49), .I1(n50), .S(n51), .ZN(n48) );
  NR2D0BWP7T U53 ( .A1(n52), .A2(n53), .ZN(n50) );
  NR2D0BWP7T U54 ( .A1(A[3]), .A2(n54), .ZN(n49) );
  MUX2ND0BWP7T U55 ( .I0(ALU_Sel[0]), .I1(n55), .S(A[3]), .ZN(n47) );
  CKND2D0BWP7T U56 ( .A1(ALU_Sel[0]), .A2(n54), .ZN(n55) );
  ND3D0BWP7T U57 ( .A1(n56), .A2(n57), .A3(n58), .ZN(ALU_Out[3]) );
  OA32D0BWP7T U58 ( .A1(n59), .A2(ALU_Sel[0]), .A3(n60), .B1(n61), .B2(n53),
       .Z(n58) );
  MUX2ND0BWP7T U59 ( .I0(n62), .I1(n63), .S(A[3]), .ZN(n57) );
  NR2D0BWP7T U60 ( .A1(n46), .A2(n64), .ZN(n63) );
  CKND0BWP7T U61 ( .I(B[3]), .ZN(n46) );
  MUX2ND0BWP7T U62 ( .I0(n65), .I1(n66), .S(n67), .ZN(n56) );
  AOI21D0BWP7T U63 ( .A1(B[3]), .A2(A[3]), .B(n53), .ZN(n67) );
  NR2D0BWP7T U64 ( .A1(B[3]), .A2(A[3]), .ZN(n53) );
  OAI221D0BWP7T U65 ( .A1(n68), .A2(n69), .B1(n70), .B2(n71), .C(n72), .ZN(n66)
      );
```

```
    CKND0BWP7T U66 ( .I(n54), .ZN(n68) );
    OAI22D0BWP7T U67 ( .A1(n52), .A2(n70), .B1(n54), .B2(n69), .ZN(n65) );
    OAI21D0BWP7T U68 ( .A1(n73), .A2(n59), .B(n74), .ZN(n54) );
    OAI21D0BWP7T U69 ( .A1(A[2]), .A2(n75), .B(n76), .ZN(n74) );
    CKND0BWP7T U70 ( .I(n71), .ZN(n52) );
    OAI22D0BWP7T U71 ( .A1(n59), .A2(n76), .B1(n77), .B2(n78), .ZN(n71) );
    ND3D0BWP7T U72 ( .A1(n79), .A2(n80), .A3(n81), .ZN(ALU_Out[2]) );
    AOI22D0BWP7T U73 ( .A1(n82), .A2(n83), .B1(n62), .B2(n59), .ZN(n81) );
    CKMUX2D0BWP7T U74 ( .I0(A[1]), .I1(A[3]), .S(ALU_Sel[0]), .Z(n82) );
    MUX2ND0BWP7T U75 ( .I0(n84), .I1(n85), .S(n76), .ZN(n80) );
    CKND0BWP7T U76 ( .I(B[2]), .ZN(n76) );
    NR2D0BWP7T U77 ( .A1(n86), .A2(n59), .ZN(n85) );
    MUX2ND0BWP7T U78 ( .I0(n87), .I1(n86), .S(n59), .ZN(n84) );
    OA221D0BWP7T U79 ( .A1(n73), .A2(n69), .B1(n70), .B2(n88), .C(n72), .Z(n86)
          );
    CKND0BWP7T U80 ( .I(n75), .ZN(n73) );
    NR2D0BWP7T U81 ( .A1(n89), .A2(n90), .ZN(n87) );
    MUX2ND0BWP7T U82 ( .I0(n91), .I1(n90), .S(n78), .ZN(n79) );
    NR2D0BWP7T U83 ( .A1(B[2]), .A2(A[2]), .ZN(n78) );
    OAI22D0BWP7T U84 ( .A1(n77), .A2(n70), .B1(n75), .B2(n69), .ZN(n90) );
    OAI21D0BWP7T U85 ( .A1(n92), .A2(n93), .B(n94), .ZN(n75) );
    OAI21D0BWP7T U86 ( .A1(A[1]), .A2(n95), .B(n96), .ZN(n94) );
    CKND0BWP7T U87 ( .I(n88), .ZN(n77) );
    OAI22D0BWP7T U88 ( .A1(n93), .A2(n96), .B1(n97), .B2(n98), .ZN(n88) );
    ND3D0BWP7T U89 ( .A1(n99), .A2(n100), .A3(n101), .ZN(ALU_Out[1]) );
    AOI22D0BWP7T U90 ( .A1(n102), .A2(n83), .B1(n62), .B2(n93), .ZN(n101) );
    CKND0BWP7T U91 ( .I(n60), .ZN(n83) );
    MUX2ND0BWP7T U92 ( .I0(n59), .I1(n103), .S(n51), .ZN(n102) );
    CKND0BWP7T U93 ( .I(A[2]), .ZN(n59) );
    MUX2ND0BWP7T U94 ( .I0(n104), .I1(n105), .S(n96), .ZN(n100) );
    CKND0BWP7T U95 ( .I(B[1]), .ZN(n96) );
    NR2D0BWP7T U96 ( .A1(n106), .A2(n93), .ZN(n105) );
    MUX2ND0BWP7T U97 ( .I0(n107), .I1(n106), .S(n93), .ZN(n104) );
    OA221D0BWP7T U98 ( .A1(n92), .A2(n69), .B1(n108), .B2(n70), .C(n72), .Z(n106)
         );
    NR2D0BWP7T U99 ( .A1(n89), .A2(n109), .ZN(n107) );
    CKND0BWP7T U100 ( .I(n64), .ZN(n89) );
    MUX2ND0BWP7T U101 ( .I0(n91), .I1(n109), .S(n97), .ZN(n99) );
    NR2D0BWP7T U102 ( .A1(B[1]), .A2(A[1]), .ZN(n97) );
    OAI22D0BWP7T U103 ( .A1(n98), .A2(n70), .B1(n95), .B2(n69), .ZN(n109) );
    CKND0BWP7T U104 ( .I(n92), .ZN(n95) );
    OAI211D0BWP7T U105 ( .A1(n110), .A2(n61), .B(n111), .C(n112), .ZN(ALU_Out[0])
         );
    AOI21D0BWP7T U106 ( .A1(n92), .A2(n113), .B(n114), .ZN(n112) );
    OAI32D0BWP7T U107 ( .A1(n60), .A2(n93), .A3(n51), .B1(n98), .B2(n64), .ZN(
         n114) );
    CKND0BWP7T U108 ( .I(n108), .ZN(n98) );
    NR2D0BWP7T U109 ( .A1(n103), .A2(n110), .ZN(n108) );
    CKND0BWP7T U110 ( .I(A[1]), .ZN(n93) );
    CKND2D0BWP7T U111 ( .A1(ALU_Sel[2]), .A2(ALU_Sel[1]), .ZN(n60) );
    NR2D0BWP7T U112 ( .A1(n110), .A2(A[0]), .ZN(n92) );
    MUX2ND0BWP7T U113 ( .I0(n115), .I1(n62), .S(n103), .ZN(n111) );
    CKND0BWP7T U114 ( .I(A[0]), .ZN(n103) );
    NR3D0BWP7T U115 ( .A1(n51), .A2(ALU_Sel[1]), .A3(n116), .ZN(n62) );
```

```
    IOA21D0BWP7T U116 ( .A1(n110), .A2(n113), .B(n61), .ZN(n115) );
    ND3D0BWP7T U117 ( .A1(n70), .A2(n72), .A3(n69), .ZN(n113) );
    CKND2D0BWP7T U118 ( .A1(N57), .A2(ALU_Sel[0]), .ZN(n69) );
    IND3D0BWP7T U119 ( .A1(ALU_Sel[1]), .B1(n51), .B2(ALU_Sel[2]), .ZN(n72) );
    CKND2D0BWP7T U120 ( .A1(N57), .A2(n51), .ZN(n70) );
    NR2D0BWP7T U121 ( .A1(ALU_Sel[1]), .A2(ALU_Sel[2]), .ZN(N57) );
    CKND0BWP7T U122 ( .I(n91), .ZN(n61) );
    NR2D0BWP7T U123 ( .A1(n64), .A2(n51), .ZN(n91) );
    CKND0BWP7T U124 ( .I(ALU_Sel[0]), .ZN(n51) );
    CKND2D0BWP7T U125 ( .A1(ALU_Sel[1]), .A2(n116), .ZN(n64) );
    CKND0BWP7T U126 ( .I(ALU_Sel[2]), .ZN(n116) );
    CKND0BWP7T U127 ( .I(B[0]), .ZN(n110) );
endmodule
```

Cuadro 57: Archivo ALU4b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O

13.2.6. Síntesis lógica del circuito sintetizado con pines de I/O

A través del archivo `ALU4b_sintetizado_and_IO.v` se obtuvo el circuito completamente sintetizado. Este archivo se generó en el directorio `Outputs/ALU4b/Outputs_finales` con el nombre `ALU4b_sintesis_final.v`. Además, se generaron otros dos archivos que también servirán para el siguiente proceso de síntesis física.

Figura 104: Archivos generados en la carpeta del circuito en Outputs_finales

En el Cuadro 58 se muestra el archivo Verilog obtenido.

```
////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Fri Sep 13 19:35:13 2024
////////////////////////////////////////////////////////////////
module ALU_4bit ( A, B, ALU_Sel, ALU_Out, Carry_Out );
    input [3:0] A;
    input [3:0] B;
    input [2:0] ALU_Sel;
    output [3:0] ALU_Out;
    output Carry_Out;
    wire N57, n117, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13,
          n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27,
          n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41,
```

```
        n42, n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55,
        n56, n57, n58, n59, n60, n61, n62;

  LHQD1BWP7T Carry_Out_reg ( .E(N57), .D(n117), .Q(Carry_Out) );
  OAI21D0BWP7T U1 ( .A1(n1), .A2(n2), .B(n3), .ZN(n117) );
  MUX2ND0BWP7T U2 ( .I0(n4), .I1(n5), .S(n6), .ZN(n3) );
  OAI21D0BWP7T U3 ( .A1(n7), .A2(n8), .B(n2), .ZN(n5) );
  IAO21D0BWP7T U4 ( .A1(B[3]), .A2(n9), .B(A[3]), .ZN(n4) );
  OAI211D0BWP7T U5 ( .A1(A[3]), .A2(n10), .B(n11), .C(n12), .ZN(ALU_Out[3]) );
  IAO21D0BWP7T U6 ( .A1(n13), .A2(n7), .B(n14), .ZN(n12) );
  OAI32D0BWP7T U7 ( .A1(n15), .A2(ALU_Sel[0]), .A3(n16), .B1(n2), .B2(n17),
     .ZN(n14) );
  MUX2ND0BWP7T U8 ( .I0(n18), .I1(n19), .S(n20), .ZN(n11) );
  INR2D0BWP7T U9 ( .A1(n2), .B1(n7), .ZN(n20) );
  NR2D0BWP7T U10 ( .A1(B[3]), .A2(A[3]), .ZN(n7) );
  CKND2D0BWP7T U11 ( .A1(B[3]), .A2(A[3]), .ZN(n2) );
  OAI221D0BWP7T U12 ( .A1(n21), .A2(n22), .B1(n9), .B2(n23), .C(n24), .ZN(n19)
        );
  CKND0BWP7T U13 ( .I(n1), .ZN(n9) );
  CKND0BWP7T U14 ( .I(n8), .ZN(n21) );
  OAI22D0BWP7T U15 ( .A1(n1), .A2(n23), .B1(n8), .B2(n22), .ZN(n18) );
  MAOI222D0BWP7T U16 ( .A(B[2]), .B(A[2]), .C(n25), .ZN(n8) );
  AOI21D0BWP7T U17 ( .A1(n26), .A2(n27), .B(n28), .ZN(n25) );
  AOI21D0BWP7T U18 ( .A1(n29), .A2(A[1]), .B(B[1]), .ZN(n28) );
  OAI21D0BWP7T U19 ( .A1(n30), .A2(n16), .B(n31), .ZN(n1) );
  AO21D0BWP7T U20 ( .A1(n16), .A2(n30), .B(B[2]), .Z(n31) );
  CKND0BWP7T U21 ( .I(n32), .ZN(n10) );
  OAI211D0BWP7T U22 ( .A1(n15), .A2(n33), .B(n34), .C(n35), .ZN(ALU_Out[2]) );
  MUX3ND0BWP7T U23 ( .I0(n36), .I1(n37), .I2(n38), .S0(n16), .S1(B[2]), .ZN(
     n35) );
  CKND2D0BWP7T U24 ( .A1(n39), .A2(n13), .ZN(n38) );
  MUX2ND0BWP7T U25 ( .I0(n40), .I1(n36), .S(n16), .ZN(n39) );
  IND2D0BWP7T U26 ( .A1(n37), .B1(n17), .ZN(n40) );
  OAI22D0BWP7T U27 ( .A1(n41), .A2(n23), .B1(n42), .B2(n22), .ZN(n37) );
  OAI221D0BWP7T U28 ( .A1(n43), .A2(n22), .B1(n30), .B2(n23), .C(n24), .ZN(n36) )
     ;
  CKND0BWP7T U29 ( .I(n41), .ZN(n30) );
  OAI21D0BWP7T U30 ( .A1(n44), .A2(n27), .B(n45), .ZN(n41) );
  AO21D0BWP7T U31 ( .A1(n27), .A2(n44), .B(B[1]), .Z(n45) );
  CKND0BWP7T U32 ( .I(n42), .ZN(n43) );
  MAOI222D0BWP7T U33 ( .A(B[1]), .B(n46), .C(A[1]), .ZN(n42) );
  AN2D0BWP7T U34 ( .A1(B[0]), .A2(A[0]), .Z(n46) );
  MUX2ND0BWP7T U35 ( .I0(n47), .I1(n32), .S(n16), .ZN(n34) );
  CKND0BWP7T U36 ( .I(A[2]), .ZN(n16) );
  MUX2ND0BWP7T U37 ( .I0(A[3]), .I1(A[1]), .S(n6), .ZN(n33) );
  OAI211D0BWP7T U38 ( .A1(n15), .A2(n48), .B(n49), .C(n50), .ZN(ALU_Out[1]) );
  MUX3ND0BWP7T U39 ( .I0(n51), .I1(n52), .I2(n53), .S0(n27), .S1(B[1]), .ZN(
     n50) );
  CKND2D0BWP7T U40 ( .A1(n54), .A2(n13), .ZN(n53) );
  MUX2ND0BWP7T U41 ( .I0(n55), .I1(n51), .S(n27), .ZN(n54) );
  IND2D0BWP7T U42 ( .A1(n52), .B1(n17), .ZN(n55) );
  OAI22D0BWP7T U43 ( .A1(n56), .A2(n23), .B1(n26), .B2(n22), .ZN(n52) );
  OAI221D0BWP7T U44 ( .A1(n29), .A2(n22), .B1(n44), .B2(n23), .C(n24), .ZN(n51) )
     ;
```

```
    CKND0BWP7T U45 ( .I(n26), .ZN(n29) );
    MUX2ND0BWP7T U46 ( .I0(n47), .I1(n32), .S(n27), .ZN(n49) );
    MUX2ND0BWP7T U47 ( .I0(A[2]), .I1(A[0]), .S(n6), .ZN(n48) );
    OAI211D0BWP7T U48 ( .A1(n57), .A2(n56), .B(n58), .C(n59), .ZN(ALU_Out[0]) );
    AOI21D0BWP7T U49 ( .A1(n47), .A2(B[0]), .B(n60), .ZN(n59) );
    OAI32D0BWP7T U50 ( .A1(n15), .A2(n27), .A3(n6), .B1(n26), .B2(n17), .ZN(n60)
           );
    CKND2D0BWP7T U51 ( .A1(A[0]), .A2(B[0]), .ZN(n26) );
    CKND0BWP7T U52 ( .I(A[1]), .ZN(n27) );
    CKND2D0BWP7T U53 ( .A1(ALU_Sel[2]), .A2(ALU_Sel[1]), .ZN(n15) );
    MUX2ND0BWP7T U54 ( .I0(n32), .I1(n61), .S(A[0]), .ZN(n58) );
    OAI21D0BWP7T U55 ( .A1(B[0]), .A2(n57), .B(n13), .ZN(n61) );
    CKND0BWP7T U56 ( .I(n47), .ZN(n13) );
    NR2D0BWP7T U57 ( .A1(n17), .A2(n6), .ZN(n47) );
    CKND2D0BWP7T U58 ( .A1(ALU_Sel[1]), .A2(n62), .ZN(n17) );
    NR3D0BWP7T U59 ( .A1(n6), .A2(ALU_Sel[1]), .A3(n62), .ZN(n32) );
    CKND0BWP7T U60 ( .I(ALU_Sel[2]), .ZN(n62) );
    CKND0BWP7T U61 ( .I(n44), .ZN(n56) );
    INR2D0BWP7T U62 ( .A1(B[0]), .B1(A[0]), .ZN(n44) );
    AN3D0BWP7T U63 ( .A1(n22), .A2(n23), .A3(n24), .Z(n57) );
    IND3D0BWP7T U64 ( .A1(ALU_Sel[1]), .B1(n6), .B2(ALU_Sel[2]), .ZN(n24) );
    CKND2D0BWP7T U65 ( .A1(N57), .A2(ALU_Sel[0]), .ZN(n23) );
    CKND2D0BWP7T U66 ( .A1(N57), .A2(n6), .ZN(n22) );
    CKND0BWP7T U67 ( .I(ALU_Sel[0]), .ZN(n6) );
    NR2D0BWP7T U68 ( .A1(ALU_Sel[1]), .A2(ALU_Sel[2]), .ZN(N57) );
endmodule

module ALU_4bit_IO ( A, B, ALU_Sel, ALU_Out, Carry_Out );
    input [3:0] A;
    input [3:0] B;
    input [2:0] ALU_Sel;
    output [3:0] ALU_Out;
    output Carry_Out;
    wire Carry_Out_w, n1, n2;
    wire [3:0] A_w;
    wire [3:0] B_w;
    wire [2:0] ALU_Sel_w;
    wire [3:0] ALU_Out_w;
    tri [3:0] A;
    tri [3:0] B;
    tri [2:0] ALU_Sel;
    tri [3:0] ALU_Out;
    tri Carry_Out;

    ALU_4bit ALU_inst ( .A(A_w), .B(B_w), .ALU_Sel(ALU_Sel_w), .ALU_Out(
        ALU_Out_w), .Carry_Out(Carry_Out_w) );
    PDDW0204SCDG U_A0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(A[0]), .DS(n1), .PE(n1),
        .C(A_w[0]) );
    PDDW0204SCDG U_A1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(A[1]), .DS(n1), .PE(n1),
        .C(A_w[1]) );
    PDDW0204SCDG U_A2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(A[2]), .DS(n1), .PE(n1),
        .C(A_w[2]) );
    PDDW0204SCDG U_A3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(A[3]), .DS(n1), .PE(n1),
        .C(A_w[3]) );
```

```
    PDDW0204SCDG U_B0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(B[0]), .DS(n1), .PE(n1),
        .C(B_w[0]) );
    PDDW0204SCDG U_B1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(B[1]), .DS(n1), .PE(n1),
        .C(B_w[1]) );
    PDDW0204SCDG U_B2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(B[2]), .DS(n1), .PE(n1),
        .C(B_w[2]) );
    PDDW0204SCDG U_B3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(B[3]), .DS(n1), .PE(n1),
        .C(B_w[3]) );
    PDDW0204SCDG U_S0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(ALU_Sel[0]), .DS(n1),
        .PE(n1), .C(ALU_Sel_w[0]) );
    PDDW0204SCDG U_S1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(ALU_Sel[1]), .DS(n1),
        .PE(n1), .C(ALU_Sel_w[1]) );
    PDDW0204SCDG U_S2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(ALU_Sel[2]), .DS(n1),
        .PE(n1), .C(ALU_Sel_w[2]) );
    PDDW0204SCDG U_Out0 ( .I(ALU_Out_w[0]), .OEN(n1), .IE(n1), .PAD(ALU_Out[0]),
        .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Out1 ( .I(ALU_Out_w[1]), .OEN(n1), .IE(n1), .PAD(ALU_Out[1]),
        .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Out2 ( .I(ALU_Out_w[2]), .OEN(n1), .IE(n1), .PAD(ALU_Out[2]),
        .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Out3 ( .I(ALU_Out_w[3]), .OEN(n1), .IE(n1), .PAD(ALU_Out[3]),
        .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Carry ( .I(Carry_Out_w), .OEN(n1), .IE(n1), .PAD(Carry_Out),
        .DS(n1), .PE(n1) );
    TIELBWP7T U6 ( .ZN(n1) );
    TIEHBWP7T U7 ( .Z(n2) );
endmodule
```

Cuadro 58: Archivo ALU4b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del diseño original del circuito con pines de I/O

13.2.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O

El último paso que se realizó fue la simulación del circuito completamente sintetizado junto con los pines de I/O. Esto se llevó a cabo nuevamente para comprobar que el circuito completamente sintetizado se comportara correctamente.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 59 en una terminal abierta en `Outputs/ALU4b/Outputs_finales`.

```
    vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v ALU4b_sintesis_final.v ALU4b_tb.v
        -o simulacion -full64 -debug_access+all

    ./simulacion

    dve -full64
```

Cuadro 59: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en la Figura 105, el comportamiento es idéntico al que se obtuvo anteriormente, lo que confirma que el *netlist* final obtenido del circuito sintetizado con pines de I/O es correcto.

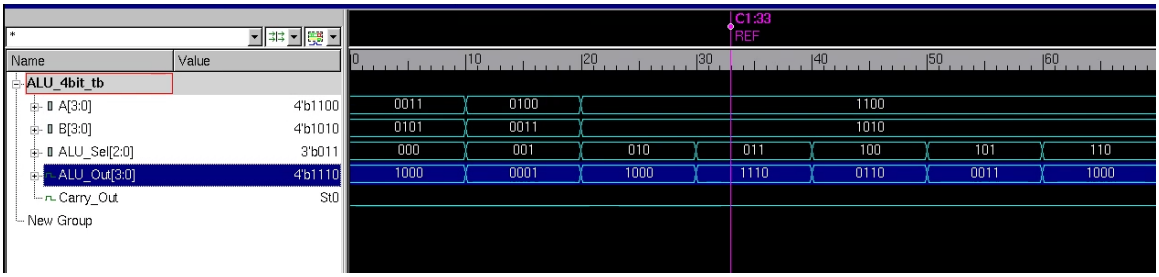


Figura 105: Simulación del *netlist* final

13.3. Proceso para la síntesis lógica de un *carry look-ahead adder* de 4 bits

13.3.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench*

El primer paso fue crear una carpeta en el directorio de *inputs*, nombrada según el circuito principal a sintetizar; en este caso, `ADDER4b` (tanto la carpeta como el archivo Verilog que contiene el circuito principal deben tener el mismo nombre). Luego, se diseñaron los archivos necesarios para la simulación y síntesis del circuito.

Siguiendo la estructura explicada en el capítulo y sección 9.2, el archivo que contiene el módulo principal del sumador de 4 bits se nombró `ADDER4b.v`. El archivo de *testbench*, utilizado para simular y verificar el funcionamiento del circuito, se denominó `ADDER4b_tb.v`. Por último, el archivo que define el módulo de entradas y salidas se nombró `IO_ADDER4b.v`.

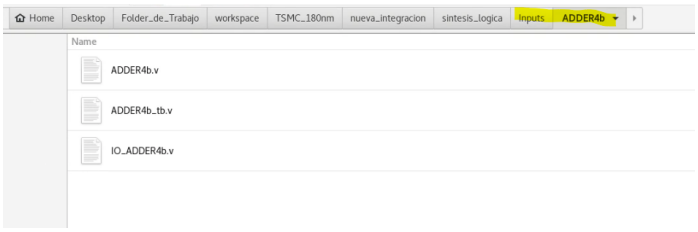


Figura 106: Archivos necesarios en la carpeta del circuito en *inputs*

En el Cuadro 60 se muestra el diseño del circuito, mientras que en el Cuadro 61 se presenta el diseño del módulo de pines de entrada y salida. Además, en el Cuadro 62 se muestra el *testbench* diseñado para la simulación del circuito original y del circuito sintetizado.

Dado que el diseño del circuito incluye múltiples entradas y salidas, en el módulo de pines es necesario hacer referencia a cada una de ellas de forma individual. Sin embargo, el proceso de creación de cada pin es el mismo para todas las entradas y salidas (ver Cuadro 61).

```
module carry_look_ahead_4bit(a,b, cin, S,cout);
input [3:0] a,b;
input cin;

output [3:0] S;
output cout;

wire [3:0] p,g,c;
wire w1, w2, w3, w4, w5, w6, w7;

    xor G1(p[0], a[0], b[0]);
    xor G17(p[1], a[1], b[1]);
    xor G18(p[2], a[2], b[2]);
    xor G19(p[3], a[3], b[3]);
    and G2(g[0], a[0], b[0]);
    and G20(g[1], a[1], b[1]);
    and G21(g[2], a[2], b[2]);
    and G22(g[3], a[3], b[3]);

    and G3(w1, p[0], cin);
    or G4(c[1], g[0], w1);
    and G5(w2, p[1], g[0]);
    and G6(w3, p[1], p[0], cin);
    or G7(c[2], g[1], w2, w3);
    and G8(w4, p[2], g[1]);
    and G16(w5, p[2], p[1], g[0]);
    and G9(w6, p[2], p[1], p[0], cin);
    or G10(c[3], g[2], w4, w5, w6);
    and G11(w7, p[3], g[2]);
    and G12(w8, p[3], p[2], g[1]);
    and G13(w9, p[3], p[2], p[1], g[0]);
    and G14(w10, p[3], p[2], p[1], p[0], cin);
    or G15(cout, g[3], w7, w8, w9, w10);

    xor G23(S[0], p[0], cin);
    xor G24(S[1], p[1], c[1]);
    xor G25(S[2], p[2], c[2]);
    xor G26(S[3], p[3], c[3]);

endmodule
```

Cuadro 60: Archivo ADDER4b.v: diseño de un sumador de 4 bits en Verilog

```
module carry_look_ahead_4bit_IO (
    input [3:0] a,
    input [3:0] b,
    input cin,
    output [3:0] S,
    output cout
);
    wire [3:0] a_w, b_w, S_w;
    wire cin_w, cout_w;
    wire [3:0] p, g, c;
```

```
    wire w1, w2, w3, w4, w5, w6, w7, w8, w9, w10;

    // Instancia del modulo carry_look_ahead_4bit
    carry_look_ahead_4bit U (
        .a(a_w),
        .b(b_w),
        .cin(cin_w),
        .S(S_w),
        .cout(cout_w)
    );

    // Interconexion de pines de entrada (a, b, cin)
    // Entradas a
    PDDW0204SCDG U_A0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[0]), .C(a_w[0]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_A1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[1]), .C(a_w[1]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_A2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[2]), .C(a_w[2]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_A3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[3]), .C(a_w[3]),
        .PE(1'b0), .IE(1'b1));

    // Entradas b
    PDDW0204SCDG U_B0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[0]), .C(b_w[0]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_B1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[1]), .C(b_w[1]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_B2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[2]), .C(b_w[2]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_B3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[3]), .C(b_w[3]),
        .PE(1'b0), .IE(1'b1));

    // Entrada cin
    PDDW0204SCDG U_Cin (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(cin), .C(cin_w),
        .PE(1'b0), .IE(1'b1));

    // Interconexion de pines de salida (S, cout)
    // Salidas S
    PDDW0204SCDG U_S0 (.I(S_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(S[0]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S1 (.I(S_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(S[1]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S2 (.I(S_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(S[2]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S3 (.I(S_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(S[3]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));

    // Salida cout
    PDDW0204SCDG U_Cout (.I(cout_w), .DS(1'b0), .OEN(1'b0), .PAD(cout), .C(1'bx
        ), .PE(1'b0), .IE(1'b0));

    // Pines de VDD y VSS
    PVDD1CDG PVDD();
    PVSS1CDG PVSS();
```

```
endmodule
```

Cuadro 61: Archivo IO_ADDER4b.v: diseño de módulo de entradas y salidas para un sumador de 4 bits en Verilog

```
module adder4bit;
reg [3:0] A, B;
reg cin;
wire [3:0] S;
wire Co;

// Instanciar el mdulo carry_look_ahead_4bit
carry_look_ahead_4bit add(.a(A), .b(B), .cin(cin), .S(S), .cout(Co));

initial
begin
    // Inicializacion de archivos de volcado de datos
    $fsdbDumpfile("adder4b.fsdb");
    $fsdbDumpvars(0, adder4bit);
    $dumpfile("adder4b.vcd");
    $dumpvars(0, adder4bit);

    // Pruebas con diferentes valores de entrada
    cin = 0; A = 4'd5; B = 4'd3;
    #100;

    A = 4'd6;
    B = 4'd9;
    #100;

    A = 4'd12;
    B = 4'd4;
    #100;

    A = 4'd7;
    B = 4'd8;
    #100;

    A = 4'd15;
    B = 4'd15;
    #100;

    // Finalizar simulacion
    $finish;
end
endmodule
```

Cuadro 62: Archivo ADDER4b_tb.v: diseño del *testbench* para simulación de un sumador de 4 bits en Verilog

13.3.2. Simulación del diseño original del circuito

Como siguiente paso, se simuló el circuito original con el objetivo de verificar que no existan errores de compilación y que el diseño del circuito sea correcto. Como se explicó anteriormente, si no se realiza este paso, existe el riesgo de encontrar errores debido a posibles fallos en el diseño del circuito, lo que impediría proceder con la ejecución de la síntesis lógica.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 63 en una terminal abierta en `Inputs/ADDER4b/`.

```
vcs -v ADDER4b.v ADDER4b_tb.v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 63: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Siguiendo los pasos descritos en el capítulo y sección 12.1 para la visualización del archivo `.vcd` generado al ejecutar el archivo de simulación, se obtuvo la Figura 107.

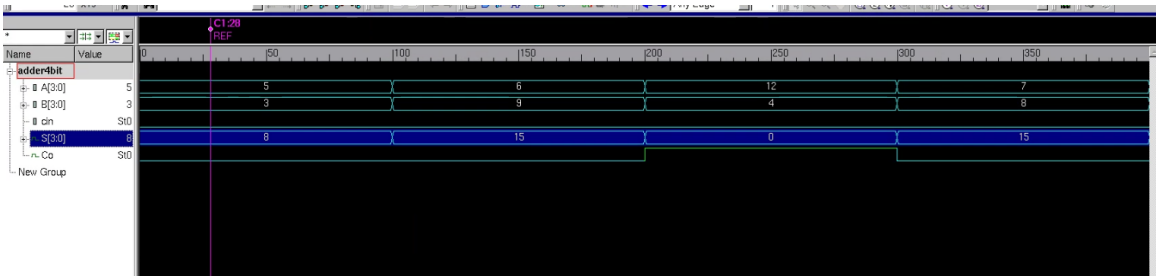
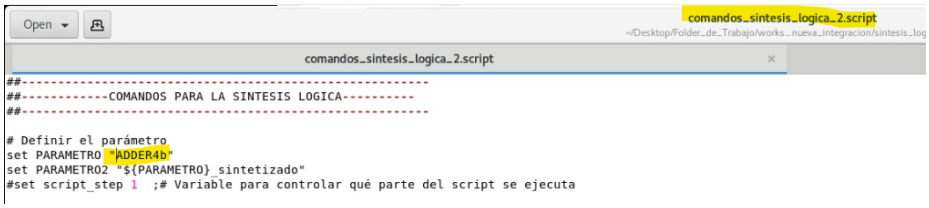


Figura 107: Simulación del sumador de 4 bits en su diseño original

Como se observa en la Figura 107, el comportamiento del circuito es el esperado, ya que se realiza la suma entre las entradas `A` y `B`, y el resultado se obtiene en la salida `S`. Por lo tanto, se procedió a ejecutar el *script* para llevar a cabo la síntesis lógica del circuito.

13.3.3. Síntesis lógica del diseño original del circuito

Antes de proceder a ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, fue necesario abrir tanto este *script* como el archivo `comandos_sintesis_logica_2.script` para definir en la variable `PARAMETRO` el nombre de la carpeta creada en *inputs*, la cual contiene el circuito a sintetizar. En este caso, se definió como `ADDER4b`.



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 108: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

Luego, se ejecutó el *script* utilizando el comando que se muestra en el Cuadro 64, en una terminal abierta en el directorio `sintesis_logica`.

```
./ejecutar_sintesis_logica_2.bash
```

Cuadro 64: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash

Al ejecutar este comando, se creó una carpeta en *outputs* con el nombre `ADDER4b`. Además, se generaron dos subdirectorios dentro de esta, denominados `Outputs_iniciales` y `Outputs_finales`. En la terminal no apareció ningún error, y se generaron los reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O. Asimismo, en la carpeta `Outputs/ADDER4b/Outputs_iniciales` se creó un archivo denominado `ADDER4b_sintetizado.v`, el cual corresponde al resultado de la primera etapa de la síntesis lógica (ver Figuras 109 y 110).

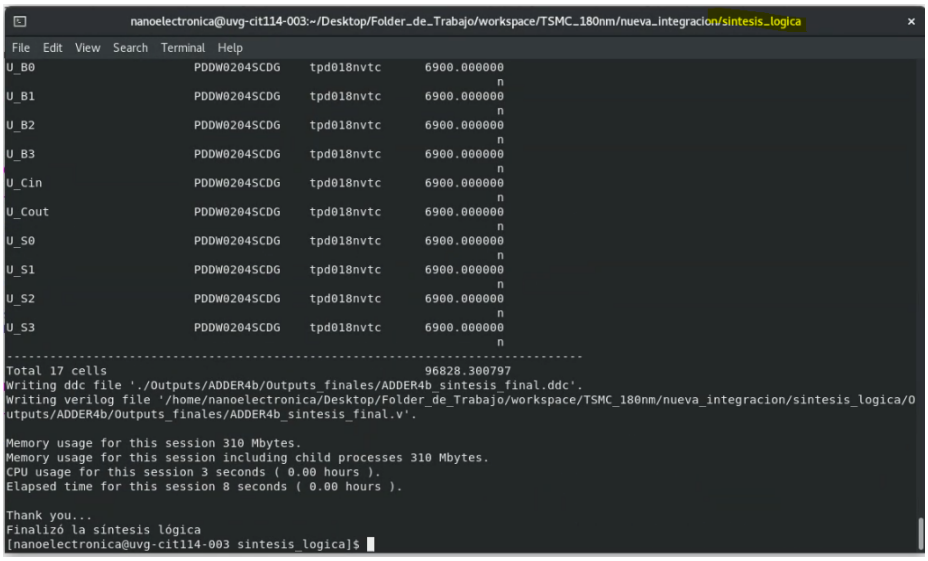


Figura 109: Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores

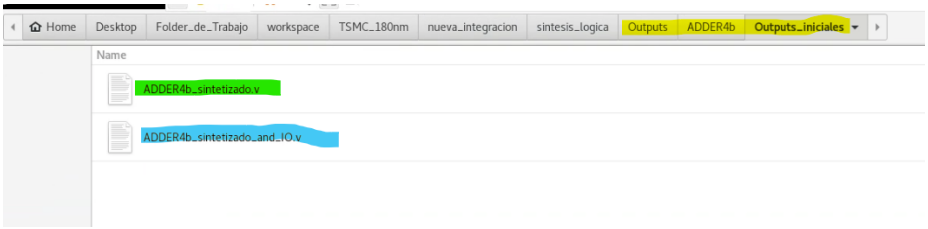


Figura 110: Archivos generados en la carpeta del circuito en Outputs_iniciales

El archivo `ADDER4b_sintetizado.v`, que corresponde al circuito sintetizado, se muestra en el Cuadro 65.

```
////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Sun Sep 15 23:29:53 2024
////////////////////////////////////////////////////////////////


module carry_look_ahead_4bit ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n8, n9, n10, n11, n12, n13, n14;

  MOAI22D0BWP7T U13 ( .A1(n8), .A2(n9), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U14 ( .A1(n8), .A2(n9), .Z(S[3]) );
  XNR2D1BWP7T U15 ( .A1(b[3]), .A2(a[3]), .ZN(n9) );
  AOI22D0BWP7T U16 ( .A1(b[2]), .A2(a[2]), .B1(n10), .B2(n11), .ZN(n8) );
  CKXOR2D0BWP7T U17 ( .A1(n11), .A2(n10), .Z(S[2]) );
```

```
    MOAI22D0BWP7T U18 ( .A1(n12), .A2(n13), .B1(b[1]), .B2(a[1]), .ZN(n10) );
    CKXOR2D0BWP7T U19 ( .A1(b[2]), .A2(a[2]), .Z(n11) );
    CKXOR2D0BWP7T U20 ( .A1(n12), .A2(n13), .Z(S[1]) );
    XNR2D1BWP7T U21 ( .A1(b[1]), .A2(a[1]), .ZN(n13) );
    AOI22D0BWP7T U22 ( .A1(n14), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n12) );
    CKXOR2D0BWP7T U23 ( .A1(cin), .A2(n14), .Z(S[0]) );
    CKXOR2D0BWP7T U24 ( .A1(b[0]), .A2(a[0]), .Z(n14) );
  endmodule
```

Cuadro 65: Archivo ADDER4b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito

13.3.4. Simulación del circuito sintetizado

El siguiente paso que se llevó a cabo, y que es fundamental para corroborar que el circuito sintetizado se comporte correctamente, fue la simulación del circuito sintetizado.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 66 en una terminal abierta en `Outputs/ADDER4b/Outputs_iniciales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v ADDER4b_sintetizado.v ADDER4b_tb.v
     -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 66: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en el Cuadro 66, para simular el circuito sintetizado es importante hacer referencia a los archivos `tcb018gbwp7t.v` y `tpd018nv.v`, ya que estos contienen las celdas a las cuales los módulos del archivo `ADDER4b_sintetizado.v` hacen referencia. Dado que estos archivos se encuentran ubicados en el directorio de *outputs*, y se está ejecutando el comando en una terminal abierta en el directorio `Outputs/ADDER4b/Outputs_iniciales`, es necesario subir dos niveles en el sistema de directorios para poder encontrarlos; por esta razón, se utiliza la ruta `../../`.

Luego de haber ejecutado los comandos mencionados anteriormente, se obtuvo la Figura 111. Se observó el mismo comportamiento que en el circuito original, lo que confirma que la síntesis del circuito se realizó correctamente.

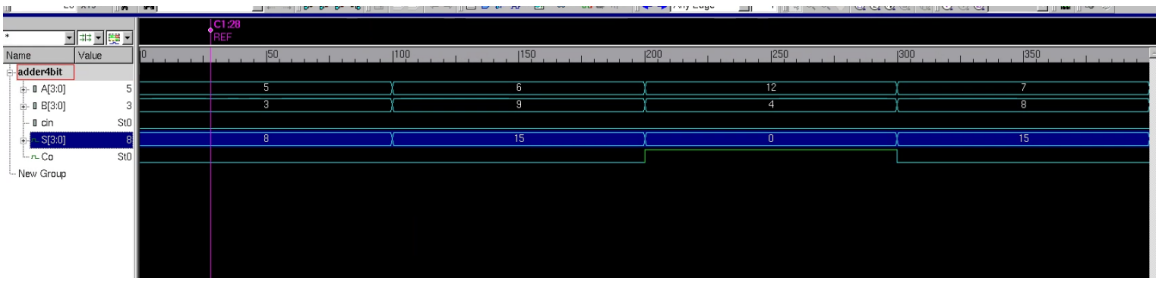


Figura 111: Simulación del sumador de 4 bits en su diseño sintetizado

13.3.5. Unión del circuito sintetizado con el módulo de pines de I/O

El archivo denominado `ADDER4b_sintetizado_and_IO.v` se creó uniendo el archivo ADDER4b_sintetizado.v con el archivo `IO_ADDER4b.v`. Este proceso también lo ejecuta automáticamente el *script* antes de comenzar el segundo proceso de síntesis lógica.

Por ello, es fundamental tener diseñado desde el inicio el archivo `IO_ADDER4b.v` y guardarlo en la carpeta `Inputs/ADDER4b`. Si este archivo no está presente, el *script* no podrá localizarlo y el nuevo archivo generado contendrá únicamente el contenido de `ADDER4b_sintetizado.v`. Esto ocasionará que, en el siguiente paso, se realice nuevamente la síntesis sin incluir el módulo de pines de entrada y salida, obteniéndose un resultado similar al archivo `ADDER4b_sintetizado.v`.

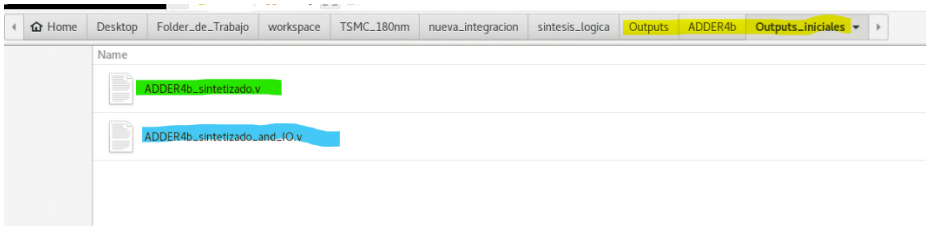


Figura 112: Archivos generados en la carpeta del circuito en Outputs_iniciales

En el Cuadro 67 se muestra el archivo generado. Este archivo se utilizará para obtener el circuito completamente sintetizado con pines de entrada y salida.

```
module carry_look_ahead_4bit_IO (
    input [3:0] a,
    input [3:0] b,
    input cin,
    output [3:0] S,
    output cout
);
    wire [3:0] a_w, b_w, S_w;
    wire cin_w, cout_w;
    wire [3:0] p, g, c;
    wire w1, w2, w3, w4, w5, w6, w7, w8, w9, w10;

    // Instancia del modulo carry_look_ahead_4bit
    carry_look_ahead_4bit U (
```

```
        .a(a_w),
        .b(b_w),
        .cin(cin_w),
        .S(S_w),
        .cout(cout_w)
    );

    // Interconexion de pines de entrada (a, b, cin)
    // Entradas a
    PDDW0204SCDG U_A0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[0]), .C(a_w[0]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_A1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[1]), .C(a_w[1]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_A2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[2]), .C(a_w[2]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_A3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[3]), .C(a_w[3]), .PE
        (1'b0), .IE(1'b1));

    // Entradas b
    PDDW0204SCDG U_B0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[0]), .C(b_w[0]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_B1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[1]), .C(b_w[1]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_B2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[2]), .C(b_w[2]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_B3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[3]), .C(b_w[3]), .PE
        (1'b0), .IE(1'b1));

    // Entrada cin
    PDDW0204SCDG U_Cin (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(cin), .C(cin_w), .PE
        (1'b0), .IE(1'b1));

    // Interconexion de pines de salida (S, cout)
    // Salidas S
    PDDW0204SCDG U_S0 (.I(S_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(S[0]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S1 (.I(S_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(S[1]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S2 (.I(S_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(S[2]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S3 (.I(S_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(S[3]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));

    // Salida cout
    PDDW0204SCDG U_Cout (.I(cout_w), .DS(1'b0), .OEN(1'b0), .PAD(cout), .C(1'bx), .
        PE(1'b0), .IE(1'b0));

    // Pines de VDD y VSS
    PVDD1CDG PVDD();
    PVSS1CDG PVSS();

endmodule

////////////////////////////////////////////////////////////
```

```
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Sun Sep 15 23:29:53 2024
////////////////////////////////////////////////////////////////


module carry_look_ahead_4bit ( a, b, cin, S, cout );
    input [3:0] a;
    input [3:0] b;
    output [3:0] S;
    input cin;
    output cout;
    wire n8, n9, n10, n11, n12, n13, n14;

    MOAI22D0BWP7T U13 ( .A1(n8), .A2(n9), .B1(b[3]), .B2(a[3]), .ZN(cout) );
    CKXOR2D0BWP7T U14 ( .A1(n8), .A2(n9), .Z(S[3]) );
    XNR2D1BWP7T U15 ( .A1(b[3]), .A2(a[3]), .ZN(n9) );
    AOI22D0BWP7T U16 ( .A1(b[2]), .A2(a[2]), .B1(n10), .B2(n11), .ZN(n8) );
    CKXOR2D0BWP7T U17 ( .A1(n11), .A2(n10), .Z(S[2]) );
    MOAI22D0BWP7T U18 ( .A1(n12), .A2(n13), .B1(b[1]), .B2(a[1]), .ZN(n10) );
    CKXOR2D0BWP7T U19 ( .A1(b[2]), .A2(a[2]), .Z(n11) );
    CKXOR2D0BWP7T U20 ( .A1(n12), .A2(n13), .Z(S[1]) );
    XNR2D1BWP7T U21 ( .A1(b[1]), .A2(a[1]), .ZN(n13) );
    AOI22D0BWP7T U22 ( .A1(n14), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n12) );
    CKXOR2D0BWP7T U23 ( .A1(cin), .A2(n14), .Z(S[0]) );
    CKXOR2D0BWP7T U24 ( .A1(b[0]), .A2(a[0]), .Z(n14) );
endmodule
```

Cuadro 67: Archivo ADDER4b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O

13.3.6. Síntesis lógica del circuito sintetizado con pines de I/O

A través del archivo `ADDER4b_sintetizado_and_IO.v` se obtuvo el circuito completamente sintetizado. Este archivo se generó en el directorio `Outputs/ADDER4b/Outputs_finales` con el nombre `ADDER4b_sintesis_final.v`. Además, se generaron otros dos archivos que también servirán para el siguiente proceso de síntesis física.

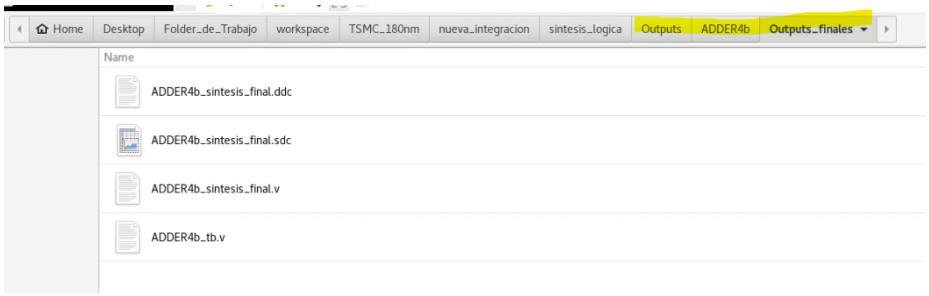


Figura 113: Archivos generados en la carpeta del circuito en Outputs_finales

En el Cuadro 68 se muestra el archivo Verilog obtenido.

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Sun Sep 15 23:30:01 2024
////////////////////////////////////////////////////////////


module carry_look_ahead_4bit ( a, b, cin, S, cout );
    input [3:0] a;
    input [3:0] b;
    output [3:0] S;
    input cin;
    output cout;
    wire n1, n2, n3, n4, n5, n6, n7;

    MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
    CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
    XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
    AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
    CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
    MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
    CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
    CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
    XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
    AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
    CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
    CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_IO ( a, b, cin, S, cout );
    input [3:0] a;
    input [3:0] b;
    output [3:0] S;
    input cin;
    output cout;
    wire cin_w, cout_w, n1, n2;
    wire [3:0] a_w;
    wire [3:0] b_w;
    wire [3:0] S_w;
    tri [3:0] a;
    tri [3:0] b;
    tri cin;
    tri [3:0] S;
    tri cout;

    carry_look_ahead_4bit U ( .a(a_w), .b(b_w), .cin(cin_w), .S(S_w), .cout(
        cout_w) );
    PDDW0204SCDG U_A0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[0]), .DS(n1), .PE(n1),
        .C(a_w[0]) );
    PDDW0204SCDG U_A1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[1]), .DS(n1), .PE(n1),
        .C(a_w[1]) );
    PDDW0204SCDG U_A2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[2]), .DS(n1), .PE(n1),
```

```
        .C(a_w[2]) );
    PDDW0204SCDG U_A3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[3]), .DS(n1), .PE(n1),
        .C(a_w[3]) );
    PDDW0204SCDG U_B0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[0]), .DS(n1), .PE(n1),
        .C(b_w[0]) );
    PDDW0204SCDG U_B1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[1]), .DS(n1), .PE(n1),
        .C(b_w[1]) );
    PDDW0204SCDG U_B2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[2]), .DS(n1), .PE(n1),
        .C(b_w[2]) );
    PDDW0204SCDG U_B3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[3]), .DS(n1), .PE(n1),
        .C(b_w[3]) );
    PDDW0204SCDG U_Cin ( .I(n1), .OEN(n2), .IE(n2), .PAD(cin), .DS(n1), .PE(n1),
        .C(cin_w) );
    PDDW0204SCDG U_S0 ( .I(S_w[0]), .OEN(n1), .IE(n1), .PAD(S[0]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S1 ( .I(S_w[1]), .OEN(n1), .IE(n1), .PAD(S[1]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S2 ( .I(S_w[2]), .OEN(n1), .IE(n1), .PAD(S[2]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S3 ( .I(S_w[3]), .OEN(n1), .IE(n1), .PAD(S[3]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_Cout ( .I(cout_w), .OEN(n1), .IE(n1), .PAD(cout), .DS(n1),
        .PE(n1) );
    TIELBWP7T U6 ( .ZN(n1) );
    TIEHBWP7T U7 ( .Z(n2) );
endmodule
```

Cuadro 68: Archivo ADDER4b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O

13.3.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O

El último paso que se realizó fue la simulación del circuito completamente sintetizado junto con los pines de I/O. Esto se llevó a cabo nuevamente para comprobar que el circuito completamente sintetizado se comportara correctamente.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 69 en una terminal abierta en `Outputs/ADDER4b/Outputs_finales`.

```
    vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v ADDER4b_sintesis_final.v
        ADDER4b_tb.v -o simulacion -full64 -debug_access+all

    ./simulacion

    dve -full64
```

Cuadro 69: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en la Figura 114, el comportamiento es idéntico al que se obtuvo

anteriormente, lo que confirma que el *netlist* final obtenido del circuito sintetizado con pines de I/O es correcto.

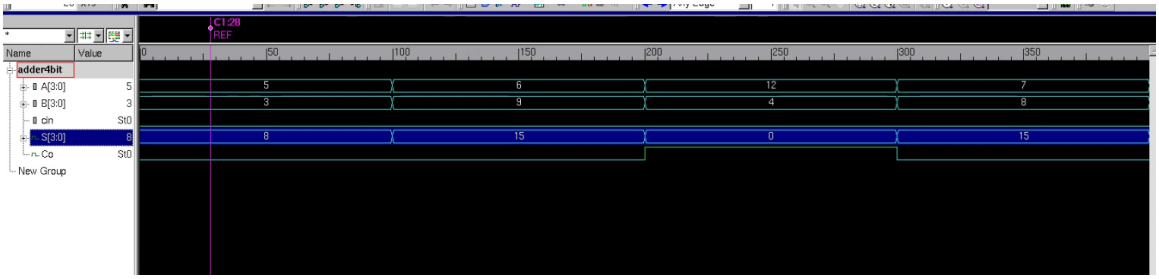


Figura 114: Simulación del *netlist* final

13.4. Proceso para la síntesis lógica de una *carry look-ahead adder* de 16 bits

13.4.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench*

El primer paso fue crear una carpeta en el directorio de *inputs*, nombrada según el circuito principal a sintetizar; en este caso, `ADDER16b`. (tanto la carpeta como el nombre del archivo Verilog que contiene el circuito principal deben tener el mismo nombre). Luego, se diseñaron los archivos necesarios para la simulación y síntesis del circuito.

Siguiendo la estructura explicada en el capítulo y sección 9.2, el archivo que contiene el módulo principal del sumador de 16 bits se nombró `ADDER16b.v`. El archivo de *testbench*, utilizado para simular y verificar el funcionamiento del circuito, se denominó `ADDER16b_tb.v`. Por último, el archivo que define el módulo de entradas y salidas se nombró `IO_ADDER16b.v`.

Sin embargo, para este circuito se realizó algo que no se había hecho con los anteriores, ya que se utilizó otro circuito para construir el circuito principal. En este caso, se usó el sumador de 4 bits diseñado anteriormente como base para crear el sumador de 16 bits. Por lo tanto, el archivo Verilog del sumador de 4 bits también se encuentra en la carpeta `Inputs/ADDER16b/`, ya que para los procesos de simulación y síntesis es importante hacer referencia a todos los archivos que se utilicen.

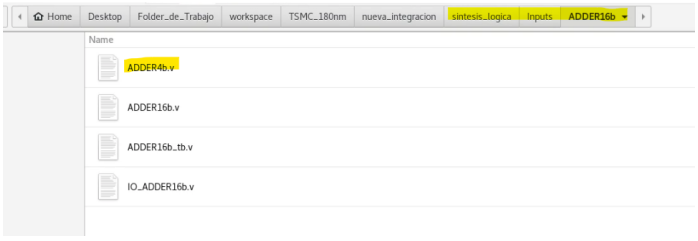


Figura 115: Archivos necesarios en la carpeta del circuito en *inputs*

En el Cuadro 70 se muestra el diseño del circuito, el cual hace referencia al módulo del sumador de 16 bits para su implementación. Por otro lado, en el Cuadro 71 se presenta el diseño del módulo de pines de entrada y salida. Además, en el Cuadro 72 se muestra el *testbench* diseñado para la simulación del circuito original y del circuito sintetizado.

Dado que el diseño del circuito incluye múltiples entradas y salidas, en el módulo de pines es necesario hacer referencia a cada una de ellas de forma individual. Sin embargo, el proceso de creación de cada pin es el mismo para todas las entradas y salidas (ver Cuadro 71).

```
module carry_look_ahead_16bit(a,b, cin, S, cout);

input [15:0] a,b;
input cin;

output [15:0] S;
output cout;

wire w1, w2, w3;


    carry_look_ahead_4bit U1 (a[3:0], b[3:0], 1'b0, S[3:0], w1);
    carry_look_ahead_4bit U2 (a[7:4], b[7:4], w1, S[7:4], w2);
    carry_look_ahead_4bit U3 (a[11:8], b[11:8], w2, S[11:8], w3);
    carry_look_ahead_4bit U4 (a[15:12], b[15:12], w3, S[15:12], cout);
    //assign S = {s8, s7, s6, s5, s4, s3, s2, s1};


endmodule
```

Cuadro 70: Archivo ADDER16b.v: diseño de un sumador de 16 bits en Verilog

```
module carry_look_ahead_16bit_IO(
    input [15:0] a,
    input [15:0] b,
    input cin,
    output [15:0] S,
    output cout
);
    wire [15:0] a_w, b_w, S_w;
    wire cin_w, cout_w;
    //wire [6:0] carries;

    // Instancia del modulo carry_look_ahead_16bit
    carry_look_ahead_16bit U (
        .a(a_w),
        .b(b_w),
        .cin(cin_w),
        .S(S_w),
        .cout(cout_w)
    );

    // Interconexion de pines de entrada (a, b, cin)
    // Entradas a
```

```
        PDDW0204SCDG U_a0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[0]), .C(a_w[0]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[1]), .C(a_w[1]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[2]), .C(a_w[2]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[3]), .C(a_w[3]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[4]), .C(a_w[4]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[5]), .C(a_w[5]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[6]), .C(a_w[6]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[7]), .C(a_w[7]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[8]), .C(a_w[8]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[9]), .C(a_w[9]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[10]), .C(a_w
            [10]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[11]), .C(a_w
            [11]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[12]), .C(a_w
            [12]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[13]), .C(a_w
            [13]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[14]), .C(a_w
            [14]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[15]), .C(a_w
            [15]), .PE(1'b0), .IE(1'b1));

        // Entradas b
        PDDW0204SCDG U_b0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[0]), .C(b_w[0]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[1]), .C(b_w[1]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[2]), .C(b_w[2]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[3]), .C(b_w[3]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[4]), .C(b_w[4]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[5]), .C(b_w[5]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[6]), .C(b_w[6]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[7]), .C(b_w[7]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[8]), .C(b_w[8]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[9]), .C(b_w[9]),
            .PE(1'b0), .IE(1'b1));
```

```
    PDDW0204SCDG U_b10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[10]), .C(b_w
        [10]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[11]), .C(b_w
        [11]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[12]), .C(b_w
        [12]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[13]), .C(b_w
        [13]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[14]), .C(b_w
        [14]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[15]), .C(b_w
        [15]), .PE(1'b0), .IE(1'b1));

    // Entrada cin
    PDDW0204SCDG U_cin (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(cin), .C(cin_w),
        .PE(1'b0), .IE(1'b1));

    // Interconexion de pines de salida (S, cout)
    // Salidas S
    PDDW0204SCDG U_S0 (.I(S_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(S[0]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S1 (.I(S_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(S[1]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S2 (.I(S_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(S[2]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S3 (.I(S_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(S[3]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S4 (.I(S_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(S[4]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S5 (.I(S_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(S[5]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S6 (.I(S_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(S[6]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S7 (.I(S_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(S[7]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S8 (.I(S_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(S[8]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S9 (.I(S_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(S[9]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S10 (.I(S_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(S[10]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S11 (.I(S_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(S[11]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S12 (.I(S_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(S[12]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S13 (.I(S_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(S[13]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S14 (.I(S_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(S[14]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S15 (.I(S_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(S[15]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));


    // Salida cout
```

```
    PDDW0204SCDG U_cout (.I(cout_w), .DS(1'b0), .OEN(1'b0), .PAD(cout), .C(1'bx
        ), .PE(1'b0), .IE(1'b0));
    //Pines de VDD y VSS
    PVDD1CDG PVDD();
    PVSS1CDG PVSS();
endmodule
```

Cuadro 71: Archivo IO_ADDER16b.v: diseño de módulo de entradas y salidas para un sumador de 16 bits en Verilog

```
module adder16bit;
reg [15:0] A, B;
reg cin;
wire [15:0] S;
wire Co;
// Instanciar el modulo carry_look_ahead_16bit
carry_look_ahead_16bit add(.a(A), .b(B), .cin(cin), .S(S), .cout(Co));
initial
begin
    // Inicializacion de archivos de volcado de datos
    $fsdbDumpfile("adder16b.fsdb");
    $fsdbDumpvars(0, adder16bit);
    $dumpfile("adder16b.vcd");
    $dumpvars(0, adder16bit);
    // Pruebas con diferentes valores de entrada
    cin = 0; A = 16'd100; B = 16'd50;
    #100;

    A = 16'd500;
    B = 16'd200;
    #100;

    A = 16'd1024;
    B = 16'd512;
    #100;

    A = 16'd4096;
    B = 16'd2048;
    #100;

    A = 16'd8192;
    B = 16'd4096;
    #100;

    A = 16'd16384;
    B = 16'd8192;
    #100;
    // Finalizar simulacion
    $finish;
end
endmodule
```

Cuadro 72: Archivo ADDER16b_tb.v: diseño del *testbench* para simulación de un sumador de 16 bits en Verilog

13.4.2. Simulación del diseño original del circuito

Como siguiente paso, se simuló el circuito original con el objetivo de verificar que no existan errores de compilación y que el diseño del circuito sea correcto. Como se explicó anteriormente, si no se realiza este paso, existe el riesgo de encontrar errores debido a posibles fallos en el diseño del circuito, lo que impediría proceder con la ejecución de la síntesis lógica.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 73 en una terminal abierta en `Inputs/ADDER16b/`.

```
vcs -v ADDER16b.v ADDER4b.v ADDER16b_tb.v -o simulacion -full64 -debug_access+
    all

./simulacion

dve -full64
```

Cuadro 73: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Como se observa en el Cuadro 73, es necesario colocar el nombre de todos los archivos Verilog que se están utilizando. Si no se hace esto, se producirá un error, ya que la aplicación no encontrará el módulo al que el circuito original hace referencia.

Siguiendo los pasos descritos en el capítulo y sección 12.1, para la visualización del archivo `.vcd` generado al ejecutar el archivo de simulación, se obtuvo la Figura 116.

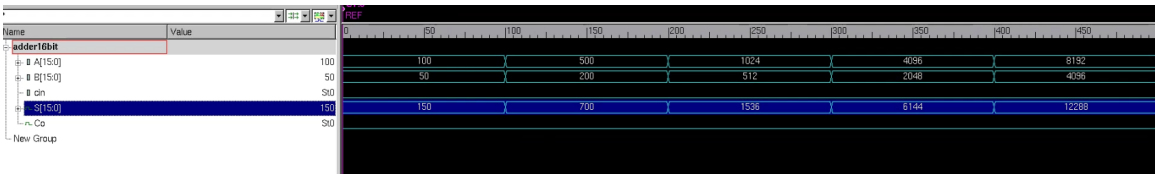


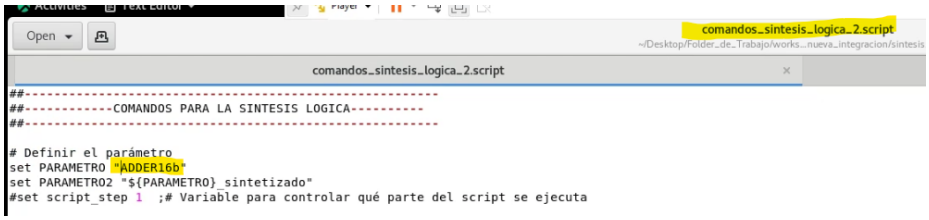
Figura 116: Simulación del sumador de 16 bits en su diseño original

Como se observa en la Figura 116, el comportamiento del circuito es el esperado, ya que se realiza la suma entre las entradas `A` y `B`, y el resultado se obtiene en la salida `S`.

Por lo tanto, se procedió a ejecutar el *script* para llevar a cabo la síntesis lógica del circuito.

13.4.3. Síntesis lógica del diseño original del circuito

Antes de proceder a ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, fue necesario abrir tanto este *script* como el archivo `comandos_sintesis_logica_2.script` para definir en la variable `PARAMETRO` el nombre de la carpeta creada en *inputs*, la cual contiene el circuito a sintetizar. En este caso, se definió como `ADDER16b`.



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 117: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

De igual manera, fue necesario modificar una parte del *script* `comandos_sintesis_logica_2.script`. Dado que el diseño del sumador de 16 bits utiliza un módulo que se encuentra en otro archivo Verilog, es necesario hacer referencia a dicho archivo para sintetizar correctamente el diseño.

La modificación debe realizarse en el primer paso de síntesis, en la sección donde se leen los archivos que describen el circuito. En la línea que contiene el comando `read_file`, se deben agregar los archivos adicionales necesarios. En este caso, la modificación quedaría de la siguiente forma:

```
if { $script_step == 1 } {
# Parte 1 - Sintesis Cell original

# Usar el parametro en lugar del nombre del modulo
read_file -format verilog "Inputs/$PARAMETRO/$PARAMETRO.v Inputs/$PARAMETRO/
    ADDER4b.v"
```

Cuadro 74: Modificación del *script* comandos_sintesis_logica_2.script

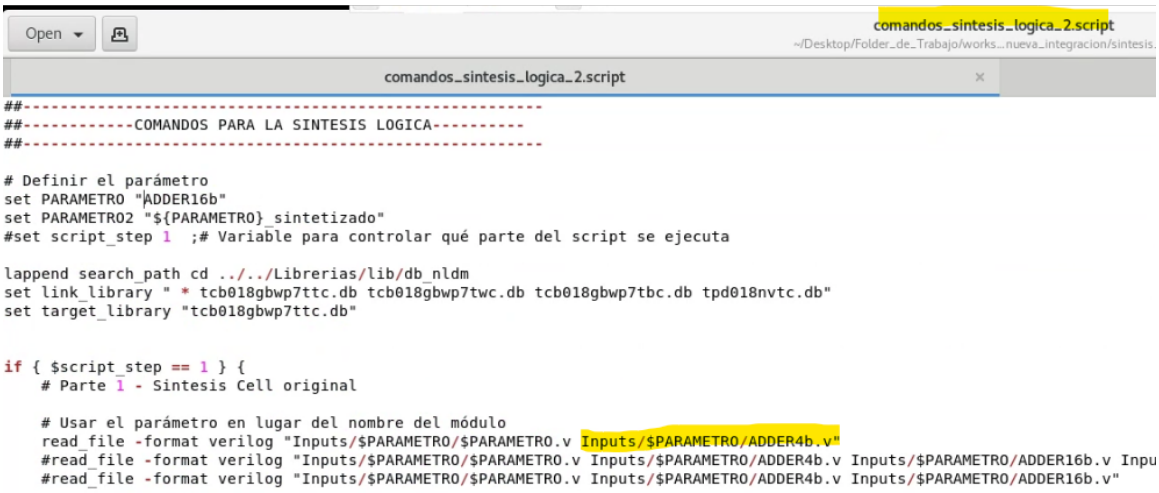


Figura 118: Modificación del *script* comandos_sintesis_logica_2.script

Como se observa en el Cuadro 74 y la Figura 118, la única modificación realizada fue agregar la lectura del archivo `ADDER4b.v` en el comando `read_file`, ya que el módulo contenido en ese archivo es utilizado en `ADDER16b.v` y, por lo tanto, también debe ser sintetizado. Si no se realiza, se producirá un error en el proceso de síntesis.

Luego, se ejecutó el *script* utilizando el comando que se muestra en el Cuadro 75, en una terminal abierta en el directorio `sintesis_logica`.

```
./ejecutar_sintesis_logica_2.bash
```

Cuadro 75: Comando para ejecutar la síntesis lógica del circuito con el *script* `ejecutar_sintesis_logica_2.bash`

Al ejecutar este comando, se creó una carpeta en *outputs* con el nombre `ADDER16b`. Además, se generaron dos subdirectorios dentro de esta, denominados `Outputs_iniciales` y `Outputs_finales`. En la terminal no apareció ningún error, y se generaron los reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O. Asimismo, en la carpeta `Outputs/ADDER16b/Outputs_iniciales` se creó un archivo denominado `ADDER16b_sintetizado.v`, el cual corresponde al resultado de la primera etapa de la síntesis lógica (ver Figuras 119 y 120).

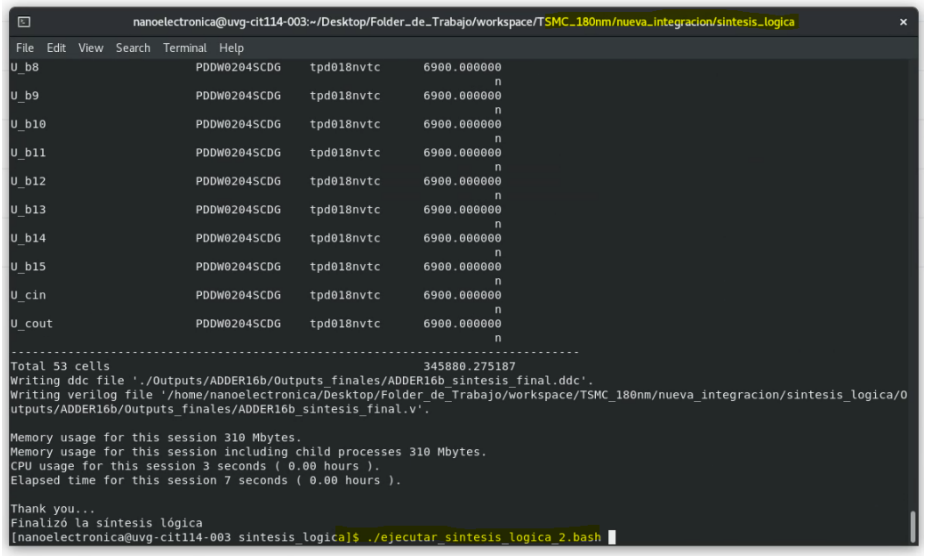


Figura 119: Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores

Figura 120: Archivos generados en la carpeta del circuito en Outputs_iniciales

El archivo `ADDER16b_sintetizado.v`, que corresponde al circuito sintetizado, se muestra en el Cuadro 76.

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Sun Sep 15 23:36:02 2024
////////////////////////////////////////////////////////////


module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
```

```
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
```

```
endmodule


module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire \*Logic0* , w1, w2, w3;

  carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(\*Logic0* ), .S(
       S[3:0]), .cout(w1) );
  carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(cout) );
  TIELBWP7T U6 ( .ZN(\*Logic0* ) );
endmodule
```

Cuadro 76: Archivo ADDER16b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito

13.4.4. Simulación del circuito sintetizado

El siguiente paso que se llevó a cabo, y que es fundamental para corroborar que el circuito sintetizado se comporte correctamente, fue la simulación del circuito sintetizado.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 77 en una terminal abierta en `Outputs/ADDER16b/Outputs_iniciales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v ADDER16b_sintetizado.v ADDER16b_tb.v
    -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 77: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se observa en el Cuadro 77, para simular el circuito sintetizado es importante hacer referencia a los archivos `tcb018gbwp7t.v` y `tpd018nv.v`, ya que estos contienen las celdas a las cuales los módulos del archivo `ADDER16b_sintetizado.v` hacen referencia. Dado que estos archivos se encuentran ubicados en el directorio de *outputs* y se está ejecutando el comando en una terminal abierta en el directorio `Outputs/ADDER16b/Outputs_iniciales`, es necesario subir dos niveles en el sistema de directorios para poder encontrarlos; por esa razón, se utiliza `../../`.

Luego de haber ejecutado los comandos mencionados anteriormente, se obtuvo la Figura 121. Se observó el mismo comportamiento que con el circuito original, lo que confirma que la síntesis del circuito se realizó correctamente.

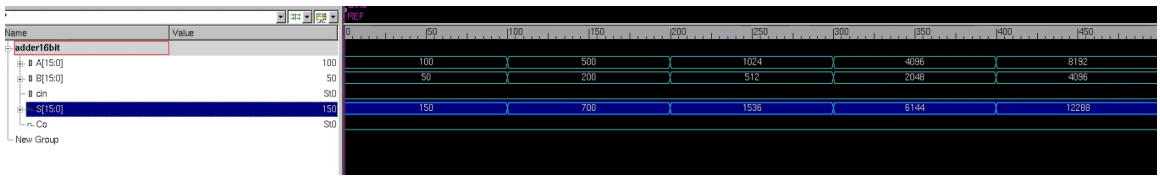


Figura 121: Simulación del sumador de 16 bits en su diseño sintetizado

13.4.5. Unión del circuito sintetizado con el módulo de pines de I/O

El archivo denominado `ADDER16b_sintetizado_and_IO.v` se creó uniendo el archivo `ADDER16b_sintetizado.v` con el archivo `IO_ADDER16b.v`. Este proceso también lo realiza automáticamente el *script* antes de comenzar el segundo proceso de síntesis lógica.

Por ello, es fundamental tener diseñado desde el principio el archivo `IO_ADDER16b.v` y guardarlo en la carpeta `Inputs/ADDER16b`. Si este archivo no está presente, el *script* no podrá localizarlo y el nuevo archivo generado contendrá únicamente el contenido de `ADDER16b_sintetizado.v`. Esto ocasionará que, en el siguiente paso, se realice nuevamente la síntesis sin incluir el módulo de pines de entrada y salida, obteniéndose un resultado similar al archivo `ADDER16b_sintetizado.v`.

Figura 122: Archivos generados en la carpeta del circuito en Outputs_iniciales

En el Cuadro 78 se muestra el archivo generado. Este archivo se utilizará para obtener el circuito completamente sintetizado con pines de entrada y salida.

```
module carry_look_ahead_16bit_IO(
    input [15:0] a,
    input [15:0] b,
    input cin,
    output [15:0] S,
    output cout
);
    wire [15:0] a_w, b_w, S_w;
    wire cin_w, cout_w;
    //wire [6:0] carries;

    // Instancia del modulo carry_look_ahead_16bit
    carry_look_ahead_16bit U (
        .a(a_w),
        .b(b_w),
        .cin(cin_w),
        .S(S_w),
        .cout(cout_w)
    );
    // Interconexion de pines de entrada (a, b, cin)
    // Entradas a
    PDDW0204SCDG U_a0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[0]), .C(a_w[0]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[1]), .C(a_w[1]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[2]), .C(a_w[2]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[3]), .C(a_w[3]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[4]), .C(a_w[4]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[5]), .C(a_w[5]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[6]), .C(a_w[6]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[7]), .C(a_w[7]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[8]), .C(a_w[8]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[9]), .C(a_w[9]), .PE
```

```
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_a10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[10]), .C(a_w[10]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_a11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[11]), .C(a_w[11]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_a12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[12]), .C(a_w[12]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_a13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[13]), .C(a_w[13]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_a14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[14]), .C(a_w[14]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_a15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[15]), .C(a_w[15]),
    .PE(1'b0), .IE(1'b1));

// Entradas b
PDDW0204SCDG U_b0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[0]), .C(b_w[0]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[1]), .C(b_w[1]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[2]), .C(b_w[2]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[3]), .C(b_w[3]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[4]), .C(b_w[4]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[5]), .C(b_w[5]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[6]), .C(b_w[6]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[7]), .C(b_w[7]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[8]), .C(b_w[8]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[9]), .C(b_w[9]), .PE
    (1'b0), .IE(1'b1));
PDDW0204SCDG U_b10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[10]), .C(b_w[10]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_b11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[11]), .C(b_w[11]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_b12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[12]), .C(b_w[12]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_b13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[13]), .C(b_w[13]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_b14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[14]), .C(b_w[14]),
    .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_b15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[15]), .C(b_w[15]),
    .PE(1'b0), .IE(1'b1));

// Entrada cin
PDDW0204SCDG U_cin (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(cin), .C(cin_w), .PE
    (1'b0), .IE(1'b1));

// Interconexion de pines de salida (S, cout)
// Salidas S
```

```
    PDDW0204SCDG U_S0 (.I(S_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(S[0]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S1 (.I(S_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(S[1]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S2 (.I(S_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(S[2]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S3 (.I(S_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(S[3]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S4 (.I(S_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(S[4]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S5 (.I(S_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(S[5]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S6 (.I(S_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(S[6]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S7 (.I(S_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(S[7]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S8 (.I(S_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(S[8]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S9 (.I(S_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(S[9]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S10 (.I(S_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(S[10]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S11 (.I(S_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(S[11]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S12 (.I(S_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(S[12]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S13 (.I(S_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(S[13]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S14 (.I(S_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(S[14]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S15 (.I(S_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(S[15]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));

    // Salida cout
    PDDW0204SCDG U_cout (.I(cout_w), .DS(1'b0), .OEN(1'b0), .PAD(cout), .C(1'bx), .
        PE(1'b0), .IE(1'b0));

//Pines de VDD y VSS
PVDD1CDG PVDD();
PVSS1CDG PVSS();
endmodule

////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Sun Sep 15 23:36:02 2024
////////////////////////////////////////////////////////////////

module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
    input [3:0] a;
    input [3:0] b;
    output [3:0] S;
    input cin;
    output cout;
```

```
    wire n1, n2, n3, n4, n5, n6, n7;

    MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
    CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
    XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
    AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
    CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
    MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
    CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
    CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
    XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
    AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
    CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
    CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
    input [3:0] a;
    input [3:0] b;
    output [3:0] S;
    input cin;
    output cout;
    wire n1, n2, n3, n4, n5, n6, n7;

    MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
    CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
    XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
    AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
    CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
    MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
    CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
    CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
    XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
    AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
    CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
    CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
    input [3:0] a;
    input [3:0] b;
    output [3:0] S;
    input cin;
    output cout;
    wire n1, n2, n3, n4, n5, n6, n7;

    MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
    CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
    XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
    AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
    CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
    MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
    CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
    CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
```

```
    XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
    AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
    CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
    CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
    input [3:0] a;
    input [3:0] b;
    output [3:0] S;
    input cin;
    output cout;
    wire n1, n2, n3, n4, n5, n6, n7;

    MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
    CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
    XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
    AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
    CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
    MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
    CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
    CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
    XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
    AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
    CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
    CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_16bit ( a, b, cin, S, cout );
    input [15:0] a;
    input [15:0] b;
    output [15:0] S;
    input cin;
    output cout;
    wire \*Logic0* , w1, w2, w3;

    carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(\*Logic0* ), .S(
        S[3:0]), .cout(w1) );
    carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
        .cout(w2) );
    carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
        .cout(w3) );
    carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
        S[15:12]), .cout(cout) );
    TIELBWP7T U6 ( .ZN(\*Logic0* ) );
endmodule
```

Cuadro 78: Archivo ADDER16b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O

13.4.6. Síntesis lógica del circuito sintetizado con pines de I/O

A través del archivo `ADDER16b_sintetizado_and_IO.v` se obtuvo el circuito completamente sintetizado. Este archivo se generó en el directorio `Outputs/ADDER16b/Outputs_finales` con el nombre `ADDER16b_sintesis_final.v`. Además, se generaron otros dos archivos que también servirán para el siguiente proceso de síntesis física.

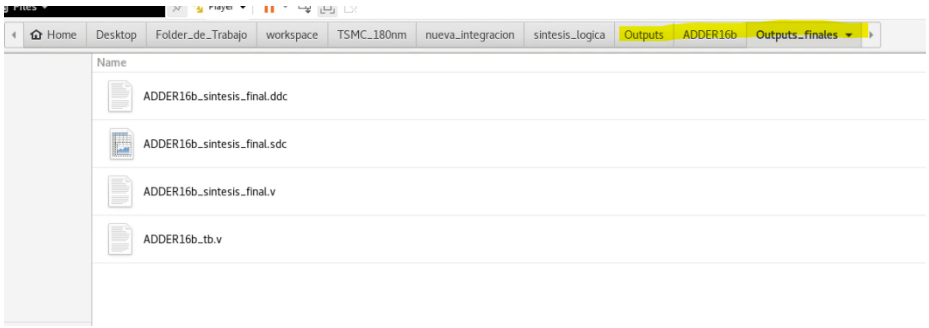


Figura 123: Archivos generados en la carpeta del circuito en Outputs_finales

En el Cuadro 79 se muestra el archivo Verilog obtenido.

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Sun Sep 15 23:36:09 2024
////////////////////////////////////////////////////////////

module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
input [3:0] a;
input [3:0] b;
output [3:0] S;
input cin;
output cout;
wire n1, n2, n3, n4, n5, n6, n7;

MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
input [3:0] a;
input [3:0] b;
```

```
output [3:0] S;
input cin;
output cout;
wire n1, n2, n3, n4, n5, n6, n7;

MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
input [3:0] a;
input [3:0] b;
output [3:0] S;
input cin;
output cout;
wire n1, n2, n3, n4, n5, n6, n7;

MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
input [3:0] a;
input [3:0] b;
output [3:0] S;
input cin;
output cout;
wire n1, n2, n3, n4, n5, n6, n7;

MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
```

```
MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_16bit ( a, b, cin, S, cout );
input [15:0] a;
input [15:0] b;
output [15:0] S;
input cin;
output cout;
wire w1, w2, w3, n1;

carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(n1), .S(S[3:0]),
       .cout(w1) );
carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(cout) );
TIELBWP7T U5 ( .ZN(n1) );
endmodule

module carry_look_ahead_16bit_IO ( a, b, cin, S, cout );
input [15:0] a;
input [15:0] b;
output [15:0] S;
input cin;
output cout;
wire cin_w, cout_w, n1, n2;
wire [15:0] a_w;
wire [15:0] b_w;
wire [15:0] S_w;
tri [15:0] a;
tri [15:0] b;
tri cin;
tri [15:0] S;
tri cout;

carry_look_ahead_16bit U ( .a(a_w), .b(b_w), .cin(cin_w), .S(S_w), .cout(
       cout_w) );
PDDW0204SCDG U_a0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[0]), .DS(n1), .PE(n1),
       .C(a_w[0]) );
PDDW0204SCDG U_a1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[1]), .DS(n1), .PE(n1),
       .C(a_w[1]) );
PDDW0204SCDG U_a2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[2]), .DS(n1), .PE(n1),
       .C(a_w[2]) );
PDDW0204SCDG U_a3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[3]), .DS(n1), .PE(n1),
       .C(a_w[3]) );
```

```
PDDW0204SCDG U_a4 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[4]), .DS(n1), .PE(n1),
       .C(a_w[4]) );
PDDW0204SCDG U_a5 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[5]), .DS(n1), .PE(n1),
       .C(a_w[5]) );
PDDW0204SCDG U_a6 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[6]), .DS(n1), .PE(n1),
       .C(a_w[6]) );
PDDW0204SCDG U_a7 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[7]), .DS(n1), .PE(n1),
       .C(a_w[7]) );
PDDW0204SCDG U_a8 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[8]), .DS(n1), .PE(n1),
       .C(a_w[8]) );
PDDW0204SCDG U_a9 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[9]), .DS(n1), .PE(n1),
       .C(a_w[9]) );
PDDW0204SCDG U_a10 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[10]), .DS(n1), .PE(n1), .C(
    a_w[10]) );
PDDW0204SCDG U_a11 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[11]), .DS(n1), .PE(n1), .C(
    a_w[11]) );
PDDW0204SCDG U_a12 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[12]), .DS(n1), .PE(n1), .C(
    a_w[12]) );
PDDW0204SCDG U_a13 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[13]), .DS(n1), .PE(n1), .C(
    a_w[13]) );
PDDW0204SCDG U_a14 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[14]), .DS(n1), .PE(n1), .C(
    a_w[14]) );
PDDW0204SCDG U_a15 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[15]), .DS(n1), .PE(n1), .C(
    a_w[15]) );
PDDW0204SCDG U_b0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[0]), .DS(n1), .PE(n1),
       .C(b_w[0]) );
PDDW0204SCDG U_b1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[1]), .DS(n1), .PE(n1),
       .C(b_w[1]) );
PDDW0204SCDG U_b2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[2]), .DS(n1), .PE(n1),
       .C(b_w[2]) );
PDDW0204SCDG U_b3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[3]), .DS(n1), .PE(n1),
       .C(b_w[3]) );
PDDW0204SCDG U_b4 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[4]), .DS(n1), .PE(n1),
       .C(b_w[4]) );
PDDW0204SCDG U_b5 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[5]), .DS(n1), .PE(n1),
       .C(b_w[5]) );
PDDW0204SCDG U_b6 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[6]), .DS(n1), .PE(n1),
       .C(b_w[6]) );
PDDW0204SCDG U_b7 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[7]), .DS(n1), .PE(n1),
       .C(b_w[7]) );
PDDW0204SCDG U_b8 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[8]), .DS(n1), .PE(n1),
       .C(b_w[8]) );
PDDW0204SCDG U_b9 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[9]), .DS(n1), .PE(n1),
       .C(b_w[9]) );
PDDW0204SCDG U_b10 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[10]), .DS(n1), .PE(n1), .C(
    b_w[10]) );
PDDW0204SCDG U_b11 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[11]), .DS(n1), .PE(n1), .C(
    b_w[11]) );
PDDW0204SCDG U_b12 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[12]), .DS(n1), .PE(n1), .C(
    b_w[12]) );
PDDW0204SCDG U_b13 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[13]), .DS(n1), .PE(n1), .C(
    b_w[13]) );
PDDW0204SCDG U_b14 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[14]), .DS(n1), .PE(n1), .C(
    b_w[14]) );
```

```
PDDW0204SCDG U_b15 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[15]), .DS(n1), .PE(n1), .C(
    b_w[15]) );
PDDW0204SCDG U_cin ( .I(n1), .OEN(n2), .IE(n2), .PAD(cin), .DS(n1), .PE(n1),
       .C(cin_w) );
PDDW0204SCDG U_S0 ( .I(S_w[0]), .OEN(n1), .IE(n1), .PAD(S[0]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S1 ( .I(S_w[1]), .OEN(n1), .IE(n1), .PAD(S[1]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S2 ( .I(S_w[2]), .OEN(n1), .IE(n1), .PAD(S[2]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S3 ( .I(S_w[3]), .OEN(n1), .IE(n1), .PAD(S[3]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S4 ( .I(S_w[4]), .OEN(n1), .IE(n1), .PAD(S[4]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S5 ( .I(S_w[5]), .OEN(n1), .IE(n1), .PAD(S[5]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S6 ( .I(S_w[6]), .OEN(n1), .IE(n1), .PAD(S[6]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S7 ( .I(S_w[7]), .OEN(n1), .IE(n1), .PAD(S[7]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S8 ( .I(S_w[8]), .OEN(n1), .IE(n1), .PAD(S[8]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S9 ( .I(S_w[9]), .OEN(n1), .IE(n1), .PAD(S[9]), .DS(n1), .PE(
       n1) );
PDDW0204SCDG U_S10 ( .I(S_w[10]), .OEN(n1), .IE(n1), .PAD(S[10]), .DS(n1),
       .PE(n1) );
PDDW0204SCDG U_S11 ( .I(S_w[11]), .OEN(n1), .IE(n1), .PAD(S[11]), .DS(n1),
       .PE(n1) );
PDDW0204SCDG U_S12 ( .I(S_w[12]), .OEN(n1), .IE(n1), .PAD(S[12]), .DS(n1),
       .PE(n1) );
PDDW0204SCDG U_S13 ( .I(S_w[13]), .OEN(n1), .IE(n1), .PAD(S[13]), .DS(n1),
       .PE(n1) );
PDDW0204SCDG U_S14 ( .I(S_w[14]), .OEN(n1), .IE(n1), .PAD(S[14]), .DS(n1),
       .PE(n1) );
PDDW0204SCDG U_S15 ( .I(S_w[15]), .OEN(n1), .IE(n1), .PAD(S[15]), .DS(n1),
       .PE(n1) );
PDDW0204SCDG U_cout ( .I(cout_w), .OEN(n1), .IE(n1), .PAD(cout), .DS(n1),
       .PE(n1) );
TIELBWP7T U6 ( .ZN(n1) );
TIEHBWP7T U7 ( .Z(n2) );
endmodule
```

Cuadro 79: Archivo ADDER16b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O

13.4.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O

El último paso que se realizó fue la simulación del circuito completamente sintetizado junto con los pines de I/O. Esto se llevó a cabo nuevamente para comprobar que el circuito completamente sintetizado se comporte correctamente.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 80 en una terminal abierta en `Outputs/ADDER16b/Outputs_finales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v ADDER16b_sintesis_final.v
    ADDER16b_tb.v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 80: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se observa en la Figura 124, el comportamiento es idéntico al obtenido anteriormente, lo que confirma que el *netlist* final del circuito sintetizado con pines de I/O es correcto.

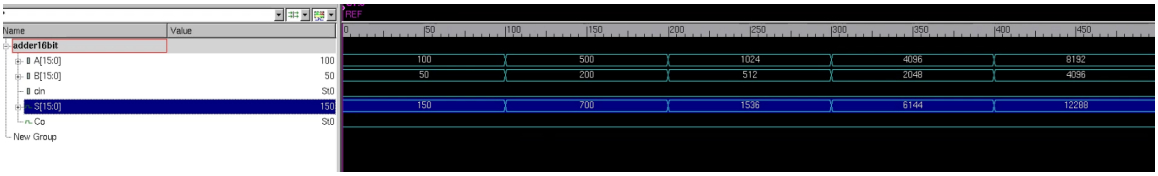


Figura 124: Simulación del *netlist* final

13.5. Proceso para la síntesis lógica de una *carry look-ahead adder* de 32 bits

13.5.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench*

El primer paso fue crear una carpeta en el directorio de *inputs*, nombrada según el circuito principal a sintetizar; en este caso, `ADDER32b` (tanto la carpeta como el nombre del archivo Verilog que contiene el circuito principal deben tener el mismo nombre). Luego, se diseñaron los archivos necesarios para la simulación y síntesis del circuito.

Siguiendo la estructura explicada en el capítulo y sección 9.2, el archivo que contiene el módulo principal del sumador de 32 bits se nombró `ADDER32b.v`. El archivo de *testbench*, utilizado para simular y verificar el funcionamiento del circuito, se denominó `ADDER32b_tb.v`. Por último, el archivo que define el módulo de entradas y salidas se nombró `IO_ADDER32b.v`.

Al igual que en el caso del sumador de 16 bits, se utilizó otro circuito para construir el circuito principal. En este caso, se usó el sumador de 4 bits diseñado anteriormente como base para crear el sumador de 32 bits. Por lo tanto, el archivo Verilog del sumador de 4 bits también se encuentra en la carpeta `Inputs/ADDER32b/`, ya que para los procesos de simulación y síntesis es importante hacer referencia a todos los archivos que se utilicen.

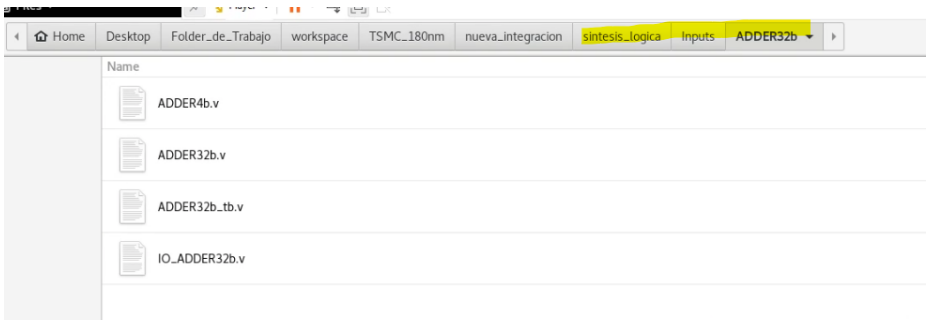


Figura 125: Archivos necesarios en la carpeta del circuito en *inputs*

En el Cuadro 81 se muestra el diseño del circuito, el cual hace referencia al módulo del sumador de 4 bits para su implementación. Por otro lado, en el Cuadro 82 se presenta el diseño del módulo de pines de entrada y salida. Además, en el Cuadro 83 se muestra el *testbench* diseñado para la simulación del circuito original y del circuito sintetizado.

Dado que el diseño del circuito incluye múltiples entradas y salidas, en el módulo de pines es necesario hacer referencia a cada una de ellas de forma individual. Sin embargo, el proceso de creación de cada pin es el mismo para todas las entradas y salidas (ver Cuadro 82).

```
    module carry_look_ahead_32bit(a,b, cin, S,cout);
        input [31:0] a,b;
        input cin;

        output [31:0] S;
        output cout;

        wire w1, w2, w3, w4, w5, w6, w7;

        carry_look_ahead_4bit U1 (a[3:0], b[3:0], 1'b0, S[3:0], w1);
        carry_look_ahead_4bit U2 (a[7:4], b[7:4], w1, S[7:4], w2);
        carry_look_ahead_4bit U3 (a[11:8], b[11:8], w2, S[11:8], w3);
        carry_look_ahead_4bit U4 (a[15:12], b[15:12], w3, S[15:12], w4);
        carry_look_ahead_4bit U5 (a[19:16], b[19:16], w4, S[19:16], w5);
        carry_look_ahead_4bit U6 (a[23:20], b[23:20], w5, S[23:20], w6);
        carry_look_ahead_4bit U7 (a[27:24], b[27:24], w6, S[27:24], w7);
        carry_look_ahead_4bit U8 (a[31:28], b[31:28], w7, S[31:28], cout);

        //assign S = {s8, s7, s6, s5, s4, s3, s2, s1};
    endmodule
```

Cuadro 81: Archivo ADDER32b.v: diseño de un sumador de 32 bits en Verilog

```
module carry_look_ahead_32bit_IO(
    input [31:0] a,
    input [31:0] b,
    input cin,
    output [31:0] S,
    output cout
);
    wire [31:0] a_w, b_w, S_w;
```

```
    wire cin_w, cout_w;
    wire [6:0] carries;

    // Instancia del modulo carry_look_ahead_32bit
    carry_look_ahead_32bit U (
        .a(a_w),
        .b(b_w),
        .cin(cin_w),
        .S(S_w),
        .cout(cout_w)
    );

    // Interconexion de pines de entrada (a, b, cin)
    // Entradas a
    PDDW0204SCDG U_a0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[0]), .C(a_w[0]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[1]), .C(a_w[1]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[2]), .C(a_w[2]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[3]), .C(a_w[3]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[4]), .C(a_w[4]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[5]), .C(a_w[5]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[6]), .C(a_w[6]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[7]), .C(a_w[7]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[8]), .C(a_w[8]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[9]), .C(a_w[9]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_a10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[10]), .C(a_w[10]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[11]), .C(a_w[11]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[12]), .C(a_w[12]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[13]), .C(a_w[13]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[14]), .C(a_w[14]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[15]), .C(a_w[15]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a16 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[16]), .C(a_w[16]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a17 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[17]), .C(a_w[17]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a18 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[18]), .C(a_w[18]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a19 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[19]), .C(a_w[19]),
        .PE(1'b0), .IE(1'b1));
```

```
    PDDW0204SCDG U_a20 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[20]), .C(a_w[20]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a21 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[21]), .C(a_w[21]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a22 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[22]), .C(a_w[22]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a23 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[23]), .C(a_w[23]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a24 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[24]), .C(a_w[24]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a25 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[25]), .C(a_w[25]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a26 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[26]), .C(a_w[26]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a27 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[27]), .C(a_w[27]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a28 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[28]), .C(a_w[28]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a29 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[29]), .C(a_w[29]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a30 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[30]), .C(a_w[30]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a31 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[31]), .C(a_w[31]),
        .PE(1'b0), .IE(1'b1));

    // Entradas b
    PDDW0204SCDG U_b0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[0]), .C(b_w[0]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[1]), .C(b_w[1]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[2]), .C(b_w[2]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[3]), .C(b_w[3]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[4]), .C(b_w[4]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[5]), .C(b_w[5]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[6]), .C(b_w[6]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[7]), .C(b_w[7]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[8]), .C(b_w[8]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[9]), .C(b_w[9]), .PE
        (1'b0), .IE(1'b1));
    PDDW0204SCDG U_b10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[10]), .C(b_w[10]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[11]), .C(b_w[11]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[12]), .C(b_w[12]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[13]), .C(b_w[13]),
        .PE(1'b0), .IE(1'b1));
```

```
    PDDW0204SCDG U_b14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[14]), .C(b_w[14]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[15]), .C(b_w[15]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b16 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[16]), .C(b_w[16]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b17 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[17]), .C(b_w[17]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b18 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[18]), .C(b_w[18]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b19 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[19]), .C(b_w[19]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b20 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[20]), .C(b_w[20]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b21 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[21]), .C(b_w[21]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b22 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[22]), .C(b_w[22]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b23 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[23]), .C(b_w[23]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b24 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[24]), .C(b_w[24]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b25 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[25]), .C(b_w[25]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b26 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[26]), .C(b_w[26]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b27 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[27]), .C(b_w[27]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b28 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[28]), .C(b_w[28]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b29 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[29]), .C(b_w[29]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b30 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[30]), .C(b_w[30]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b31 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[31]), .C(b_w[31]),
        .PE(1'b0), .IE(1'b1));

    // Entrada cin
    PDDW0204SCDG U_cin (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(cin), .C(cin_w), .PE
        (1'b0), .IE(1'b1));

    // Interconexion de pines de salida (S, cout)
    // Salidas S
    PDDW0204SCDG U_S0 (.I(S_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(S[0]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S1 (.I(S_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(S[1]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S2 (.I(S_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(S[2]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S3 (.I(S_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(S[3]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S4 (.I(S_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(S[4]), .C(1'bx), .PE
        (1'b0), .IE(1'b0));
    PDDW0204SCDG U_S5 (.I(S_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(S[5]), .C(1'bx), .PE
```

```
    (1'b0), .IE(1'b0));
PDDW0204SCDG U_S6 (.I(S_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(S[6]), .C(1'bx), .PE
    (1'b0), .IE(1'b0));
PDDW0204SCDG U_S7 (.I(S_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(S[7]), .C(1'bx), .PE
    (1'b0), .IE(1'b0));
PDDW0204SCDG U_S8 (.I(S_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(S[8]), .C(1'bx), .PE
    (1'b0), .IE(1'b0));
PDDW0204SCDG U_S9 (.I(S_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(S[9]), .C(1'bx), .PE
    (1'b0), .IE(1'b0));
PDDW0204SCDG U_S10 (.I(S_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(S[10]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S11 (.I(S_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(S[11]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S12 (.I(S_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(S[12]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S13 (.I(S_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(S[13]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S14 (.I(S_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(S[14]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S15 (.I(S_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(S[15]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S16 (.I(S_w[16]), .DS(1'b0), .OEN(1'b0), .PAD(S[16]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S17 (.I(S_w[17]), .DS(1'b0), .OEN(1'b0), .PAD(S[17]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S18 (.I(S_w[18]), .DS(1'b0), .OEN(1'b0), .PAD(S[18]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S19 (.I(S_w[19]), .DS(1'b0), .OEN(1'b0), .PAD(S[19]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S20 (.I(S_w[20]), .DS(1'b0), .OEN(1'b0), .PAD(S[20]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S21 (.I(S_w[21]), .DS(1'b0), .OEN(1'b0), .PAD(S[21]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S22 (.I(S_w[22]), .DS(1'b0), .OEN(1'b0), .PAD(S[22]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S23 (.I(S_w[23]), .DS(1'b0), .OEN(1'b0), .PAD(S[23]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S24 (.I(S_w[24]), .DS(1'b0), .OEN(1'b0), .PAD(S[24]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S25 (.I(S_w[25]), .DS(1'b0), .OEN(1'b0), .PAD(S[25]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S26 (.I(S_w[26]), .DS(1'b0), .OEN(1'b0), .PAD(S[26]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S27 (.I(S_w[27]), .DS(1'b0), .OEN(1'b0), .PAD(S[27]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S28 (.I(S_w[28]), .DS(1'b0), .OEN(1'b0), .PAD(S[28]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S29 (.I(S_w[29]), .DS(1'b0), .OEN(1'b0), .PAD(S[29]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S30 (.I(S_w[30]), .DS(1'b0), .OEN(1'b0), .PAD(S[30]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_S31 (.I(S_w[31]), .DS(1'b0), .OEN(1'b0), .PAD(S[31]), .C(1'bx),
    .PE(1'b0), .IE(1'b0));
```

```
    // Salida cout
    PDDW0204SCDG U_cout (.I(cout_w), .DS(1'b0), .OEN(1'b0), .PAD(cout), .C(1'bx), .
        PE(1'b0), .IE(1'b0));

//Pines de VDD y VSS
PVDD1CDG PVDD();
PVSS1CDG PVSS();
endmodule
```

Cuadro 82: Archivo IO_ADDER32b.v: diseño de módulo de entradas y salidas para un sumador de 32 bits en Verilog

```
    module adder32bit;
    reg [31:0] A, B;
    reg cin;
    wire [31:0] S;
    wire Co;

    carry_look_ahead_32bit add(.a(A), .b(B), .cin(cin), .S(S), .cout(Co));

    initial
    begin
        $fsdbDumpfile("adder32b.fsdb");
        $fsdbDumpvars(0, adder32bit);
        $dumpfile("adder32b.vcd");
        $dumpvars(0, adder32bit);
        cin= 0; A = 32'd85; B = 32'd1;
        #100
        A = 32'd5;
        B = 32'd7;

        #100
        A = 32'd6;
        B = 32'd8;

        #100
        A = 32'd100;
        B = 32'd100;

        #100
        A = 32'd4096;
        B = 32'd2048;

        #100
        A = 32'd512;
        B = 32'd500;

           $finish;
    end
    endmodule
```

Cuadro 83: Archivo ADDER32b_tb.v: diseño del *testbench* para simulación de un sumador de 32 bits en Verilog

13.5.2. Simulación del diseño original del circuito

Como siguiente paso, se simuló el circuito original con el objetivo de verificar que no existan errores de compilación y que el diseño del circuito sea correcto. Como se explicó anteriormente, si no se realiza este paso, existe el riesgo de encontrar errores debido a posibles fallos en el diseño del circuito, lo que impediría proceder con la ejecución de la síntesis lógica.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 84 en una terminal abierta en `Inputs/ADDER32b/`.

```
vcs -v ADDER32b.v ADDER4b.v ADDER32b_tb.v -o simulacion -full64 -debug_access+
    all

./simulacion

dve -full64
```

Cuadro 84: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Como se observa en el Cuadro 84, es necesario colocar el nombre de todos los archivos Verilog que se están utilizando. Si no se hace esto, se producirá un error, ya que la aplicación no encontrará el módulo al que el circuito original hace referencia.

Siguiendo los pasos descritos en el capítulo y sección 12.1, para la visualización del archivo `.vcd` generado al ejecutar el archivo de simulación, se obtuvo la Figura 126.

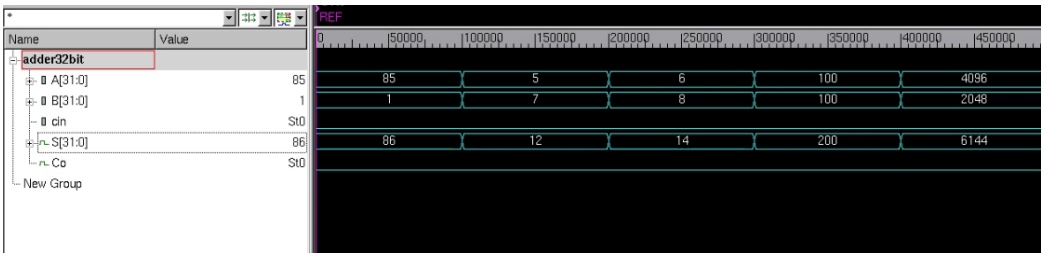


Figura 126: Simulación de sumador de 32 bits en su diseño original

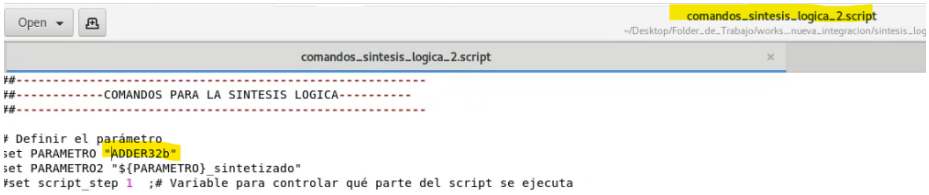
Como se observa en la Figura 126, el comportamiento del circuito es el esperado, ya que se realiza la suma entre las entradas `A` y `B`, y el resultado se obtiene en la salida `S`.

Por lo tanto, se procedió a ejecutar el *script* para llevar a cabo la síntesis lógica del circuito.

13.5.3. Síntesis lógica del diseño original del circuito

Antes de proceder a ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, fue necesario abrir tanto este *script* como el archivo `comandos_sintesis_logica_2.script` para definir

en la variable `PARAMETRO` el nombre de la carpeta creada en *inputs*, la cual contiene el circuito a sintetizar. En este caso, se definió como `ADDER32b`.



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 127: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

De igual manera, fue necesario modificar una parte del *script* `comandos_sintesis_logica_2.script`. Dado que el diseño del sumador de 32 bits utiliza un módulo que se encuentra en otro archivo Verilog, es necesario hacer referencia a dicho archivo para sintetizar correctamente el diseño.

La modificación debe realizarse en el primer paso de síntesis, en la sección donde se leen los archivos que describen el circuito. En la línea que contiene el comando `read_file`, se deben agregar los archivos adicionales necesarios. En este caso, la modificación quedaría de la siguiente forma:

```
if { $script_step == 1 } {
# Parte 1 - Sintesis Cell original

# Usar el parametro en lugar del nombre del modulo
read_file -format verilog "Inputs/$PARAMETRO/$PARAMETRO.v Inputs/$PARAMETRO/
    ADDER4b.v"
```

Cuadro 85: Modificación del *script* comandos_sintesis_logica_2.script

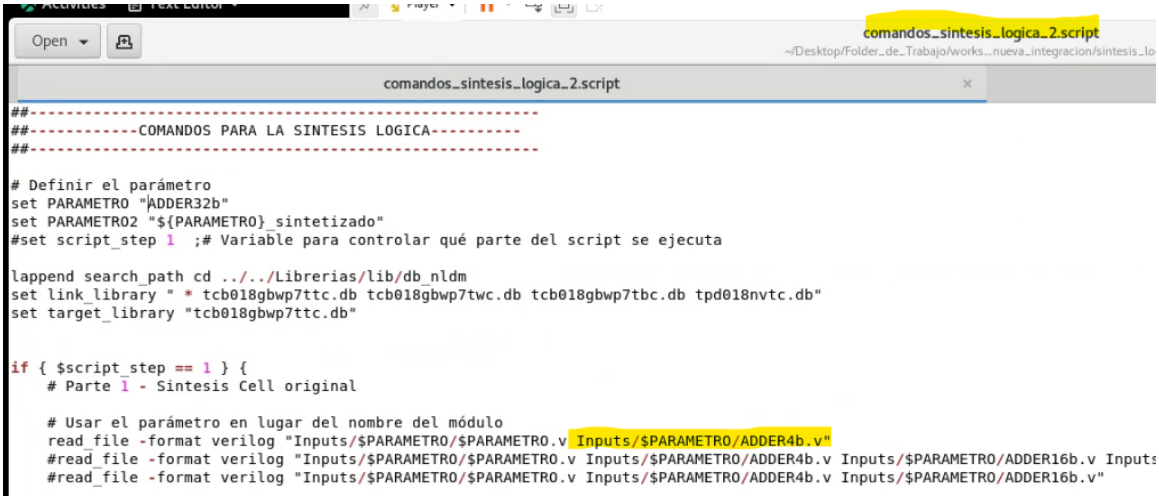


Figura 128: Modificación del *script* comandos_sintesis_logica_2.script

Como se observa en el Cuadro 85 y la Figura 128, la única modificación realizada fue agregar la lectura del archivo `ADDER4b.v` en el comando `read_file`, ya que el módulo contenido en ese archivo es utilizado en `ADDER32b.v` y, por lo tanto, también debe ser sintetizado. Si no se realiza, se producirá un error en el proceso de síntesis.

Luego, se ejecutó el *script* utilizando el comando que se muestra en el Cuadro 86, en una terminal abierta en el directorio `sintesis_logica`.

```
./ejecutar_sintesis_logica_2.bash
```

Cuadro 86: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash

Al ejecutar este comando, se creó una carpeta en *outputs* con el nombre `ADDER32b`. Además, se generaron dos subdirectorios dentro de esta, denominados `Outputs_iniciales` y `Outputs_finales`. En la terminal no apareció ningún error y se generaron los reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O. Asimismo, en la carpeta `Outputs/ADDER32b/Outputs_iniciales` se creó un archivo denominado `ADDER32b_sintetizado.v`, el cual corresponde al resultado de la primera etapa de la síntesis lógica (ver Figuras 129 y 130).

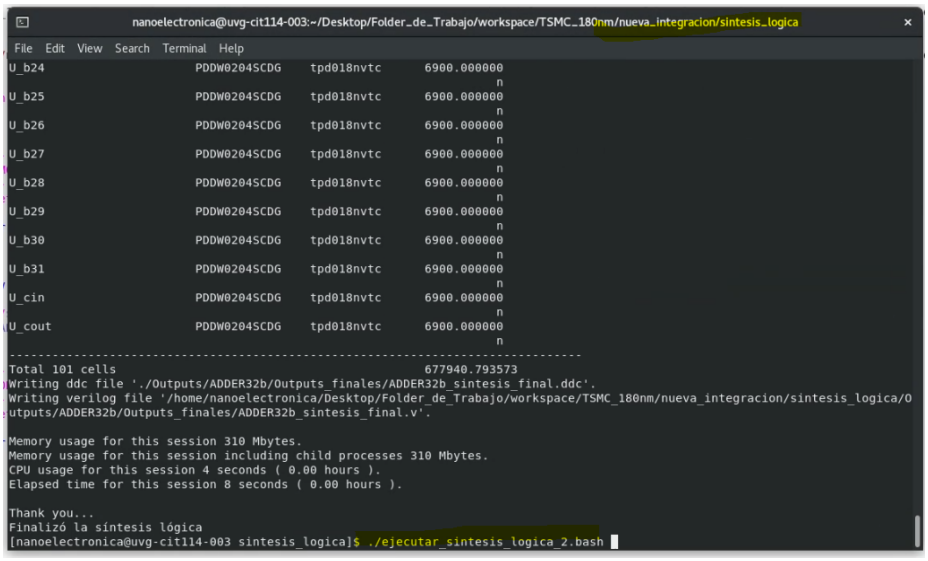


Figura 129: Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores

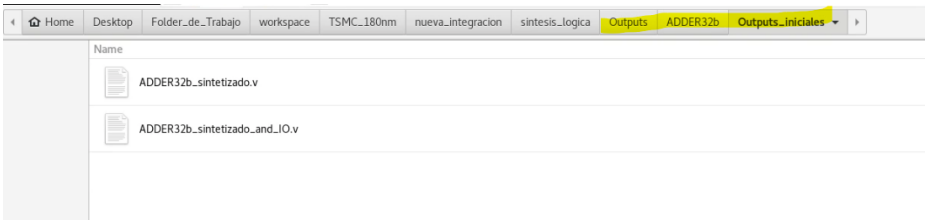


Figura 130: Archivos generados en la carpeta del circuito en Outputs_iniciales

El archivo `ADDER32b_sintetizado.v`, que corresponde al circuito sintetizado, se muestra en el Cuadro 87.

```
////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date  : Fri Nov 1 23:03:49 2024
////////////////////////////////////////////////////////////////


module carry_look_ahead_4bit_7 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
```

```
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule
```

```
module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_4 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
```

```
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_5 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_6 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
```

```
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_32bit ( a, b, cin, S, cout );
  input [31:0] a;
  input [31:0] b;
  output [31:0] S;
  input cin;
  output cout;
  wire \*Logic0* , w1, w2, w3, w4, w5, w6, w7;

  carry_look_ahead_4bit_7 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(\*Logic0* ), .S(
       S[3:0]), .cout(w1) );
  carry_look_ahead_4bit_6 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_5 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_4 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(w4) );
  carry_look_ahead_4bit_3 U5 ( .a(a[19:16]), .b(b[19:16]), .cin(w4), .S(
       S[19:16]), .cout(w5) );
  carry_look_ahead_4bit_2 U6 ( .a(a[23:20]), .b(b[23:20]), .cin(w5), .S(
       S[23:20]), .cout(w6) );
  carry_look_ahead_4bit_1 U7 ( .a(a[27:24]), .b(b[27:24]), .cin(w6), .S(
       S[27:24]), .cout(w7) );
  carry_look_ahead_4bit_0 U8 ( .a(a[31:28]), .b(b[31:28]), .cin(w7), .S(
       S[31:28]), .cout(cout) );
  TIELBWP7T U10 ( .ZN(\*Logic0* ) );
endmodule
```

Cuadro 87: Archivo ADDER32b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito

13.5.4. Simulación del circuito sintetizado

El siguiente paso que se llevó a cabo, y que es fundamental para corroborar que el circuito sintetizado se comporte correctamente, fue la simulación del circuito sintetizado.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 88 en una terminal abierta en `Outputs/ADDER32b/Outputs_iniciales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v ADDER32b_sintetizado.v ADDER32b_tb
    .v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 88: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Luego de haber ejecutado los comandos mencionados anteriormente, se obtuvo la Figura 131. Se observó el mismo comportamiento que con el circuito original, lo que confirma que la síntesis del circuito se realizó correctamente.

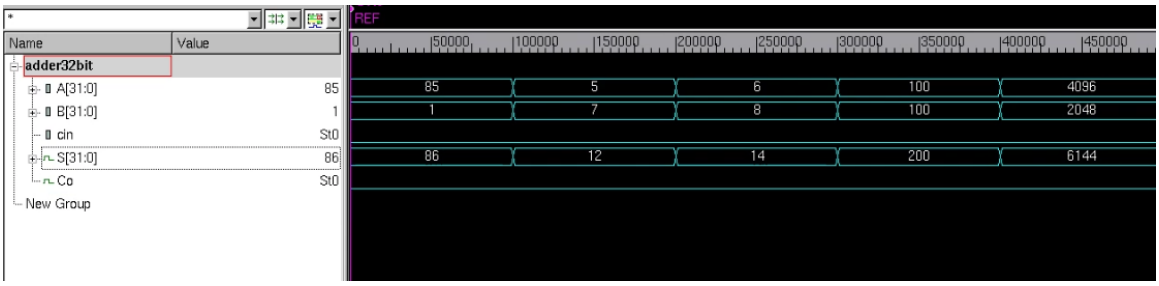


Figura 131: Simulación del sumador de 32 bits en su diseño sintetizado

13.5.5. Unión del circuito sintetizado con el módulo de pines de I/O

El archivo denominado `ADDER32b_sintetizado_and_IO.v` se creó uniendo el archivo `ADDER32b_sintetizado.v` con el archivo `IO_ADDER32b.v`. Este proceso también lo realiza automáticamente el *script* antes de comenzar el segundo proceso de síntesis lógica.

Por ello, es fundamental tener diseñado desde el principio el archivo `IO_ADDER32b.v` y guardarlo en la carpeta `Inputs/ADDER32b`. Si este archivo no está presente, el *script* no podrá localizarlo y el nuevo archivo generado contendrá únicamente el contenido de `ADDER32b_sintetizado.v`. Esto ocasionará que, en el siguiente paso, se realice nuevamente la síntesis sin incluir el módulo de pines de entrada y salida, obteniéndose un resultado similar al archivo `ADDER32b_sintetizado.v`.

Figura 132: Archivos generados en la carpeta del circuito en Outputs_iniciales

En el Cuadro 89 se muestra el archivo generado. Este archivo se utilizará para obtener

el circuito completamente sintetizado con pines de entrada y salida.

```
module carry_look_ahead_32bit_IO(
    input [31:0] a,
    input [31:0] b,
    input cin,
    output [31:0] S,
    output cout
);
    wire [31:0] a_w, b_w, S_w;
    wire cin_w, cout_w;
    wire [6:0] carries;
    // Instancia del modulo carry_look_ahead_32bit
    carry_look_ahead_32bit U (
        .a(a_w),
        .b(b_w),
        .cin(cin_w),
        .S(S_w),
        .cout(cout_w)
    );
    // Interconexion de pines de entrada (a, b, cin)
    // Entradas a
    PDDW0204SCDG U_a0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[0]), .C(a_w[0]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[1]), .C(a_w[1]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[2]), .C(a_w[2]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[3]), .C(a_w[3]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[4]), .C(a_w[4]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[5]), .C(a_w[5]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[6]), .C(a_w[6]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[7]), .C(a_w[7]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[8]), .C(a_w[8]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[9]), .C(a_w[9]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[10]), .C(a_w
        [10]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[11]), .C(a_w
        [11]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[12]), .C(a_w
        [12]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[13]), .C(a_w
        [13]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[14]), .C(a_w
        [14]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[15]), .C(a_w
        [15]), .PE(1'b0), .IE(1'b1));
```

```
    PDDW0204SCDG U_a16 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[16]), .C(a_w
        [16]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a17 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[17]), .C(a_w
        [17]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a18 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[18]), .C(a_w
        [18]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a19 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[19]), .C(a_w
        [19]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a20 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[20]), .C(a_w
        [20]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a21 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[21]), .C(a_w
        [21]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a22 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[22]), .C(a_w
        [22]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a23 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[23]), .C(a_w
        [23]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a24 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[24]), .C(a_w
        [24]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a25 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[25]), .C(a_w
        [25]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a26 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[26]), .C(a_w
        [26]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a27 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[27]), .C(a_w
        [27]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a28 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[28]), .C(a_w
        [28]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a29 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[29]), .C(a_w
        [29]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a30 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[30]), .C(a_w
        [30]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a31 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[31]), .C(a_w
        [31]), .PE(1'b0), .IE(1'b1));

    // Entradas b
    PDDW0204SCDG U_b0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[0]), .C(b_w[0]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[1]), .C(b_w[1]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[2]), .C(b_w[2]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[3]), .C(b_w[3]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[4]), .C(b_w[4]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[5]), .C(b_w[5]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[6]), .C(b_w[6]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[7]), .C(b_w[7]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[8]), .C(b_w[8]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[9]), .C(b_w[9]),
         .PE(1'b0), .IE(1'b1));
```

```
        PDDW0204SCDG U_b10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[10]), .C(b_w
            [10]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[11]), .C(b_w
            [11]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[12]), .C(b_w
            [12]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[13]), .C(b_w
            [13]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[14]), .C(b_w
            [14]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[15]), .C(b_w
            [15]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b16 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[16]), .C(b_w
            [16]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b17 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[17]), .C(b_w
            [17]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b18 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[18]), .C(b_w
            [18]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b19 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[19]), .C(b_w
            [19]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b20 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[20]), .C(b_w
            [20]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b21 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[21]), .C(b_w
            [21]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b22 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[22]), .C(b_w
            [22]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b23 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[23]), .C(b_w
            [23]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b24 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[24]), .C(b_w
            [24]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b25 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[25]), .C(b_w
            [25]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b26 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[26]), .C(b_w
            [26]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b27 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[27]), .C(b_w
            [27]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b28 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[28]), .C(b_w
            [28]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b29 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[29]), .C(b_w
            [29]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b30 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[30]), .C(b_w
            [30]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b31 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[31]), .C(b_w
            [31]), .PE(1'b0), .IE(1'b1));
        // Entrada cin
        PDDW0204SCDG U_cin (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(cin), .C(cin_w),
            .PE(1'b0), .IE(1'b1));

        // Interconexion de pines de salida (S, cout)
        // Salidas S
        PDDW0204SCDG U_S0 (.I(S_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(S[0]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S1 (.I(S_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(S[1]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
```

```
        PDDW0204SCDG U_S2 (.I(S_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(S[2]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S3 (.I(S_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(S[3]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S4 (.I(S_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(S[4]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S5 (.I(S_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(S[5]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S6 (.I(S_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(S[6]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S7 (.I(S_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(S[7]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S8 (.I(S_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(S[8]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S9 (.I(S_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(S[9]), .C(1'bx),
             .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S10 (.I(S_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(S[10]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S11 (.I(S_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(S[11]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S12 (.I(S_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(S[12]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S13 (.I(S_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(S[13]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S14 (.I(S_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(S[14]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S15 (.I(S_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(S[15]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S16 (.I(S_w[16]), .DS(1'b0), .OEN(1'b0), .PAD(S[16]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S17 (.I(S_w[17]), .DS(1'b0), .OEN(1'b0), .PAD(S[17]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S18 (.I(S_w[18]), .DS(1'b0), .OEN(1'b0), .PAD(S[18]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S19 (.I(S_w[19]), .DS(1'b0), .OEN(1'b0), .PAD(S[19]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S20 (.I(S_w[20]), .DS(1'b0), .OEN(1'b0), .PAD(S[20]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S21 (.I(S_w[21]), .DS(1'b0), .OEN(1'b0), .PAD(S[21]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S22 (.I(S_w[22]), .DS(1'b0), .OEN(1'b0), .PAD(S[22]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S23 (.I(S_w[23]), .DS(1'b0), .OEN(1'b0), .PAD(S[23]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S24 (.I(S_w[24]), .DS(1'b0), .OEN(1'b0), .PAD(S[24]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S25 (.I(S_w[25]), .DS(1'b0), .OEN(1'b0), .PAD(S[25]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S26 (.I(S_w[26]), .DS(1'b0), .OEN(1'b0), .PAD(S[26]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S27 (.I(S_w[27]), .DS(1'b0), .OEN(1'b0), .PAD(S[27]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S28 (.I(S_w[28]), .DS(1'b0), .OEN(1'b0), .PAD(S[28]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
```

```
    PDDW0204SCDG U_S29 (.I(S_w[29]), .DS(1'b0), .OEN(1'b0), .PAD(S[29]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S30 (.I(S_w[30]), .DS(1'b0), .OEN(1'b0), .PAD(S[30]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S31 (.I(S_w[31]), .DS(1'b0), .OEN(1'b0), .PAD(S[31]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));

    // Salida cout
    PDDW0204SCDG U_cout (.I(cout_w), .DS(1'b0), .OEN(1'b0), .PAD(cout), .C(1'bx
        ), .PE(1'b0), .IE(1'b0));

//Pines de VDD y VSS
PVDD1CDG PVDD();
PVSS1CDG PVSS();
endmodule

////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date  : Fri Nov 1 23:03:49 2024
////////////////////////////////////////////////////////////////

module carry_look_ahead_4bit_7 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
```

```
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
```

```
endmodule

module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_4 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_5 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;
```

```
  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_6 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_32bit ( a, b, cin, S, cout );
  input [31:0] a;
  input [31:0] b;
  output [31:0] S;
  input cin;
  output cout;
  wire \*Logic0* , w1, w2, w3, w4, w5, w6, w7;

  carry_look_ahead_4bit_7 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(\*Logic0* ), .S(
       S[3:0]), .cout(w1) );
  carry_look_ahead_4bit_6 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_5 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_4 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(w4) );
  carry_look_ahead_4bit_3 U5 ( .a(a[19:16]), .b(b[19:16]), .cin(w4), .S(
       S[19:16]), .cout(w5) );
```

```
  carry_look_ahead_4bit_2 U6 ( .a(a[23:20]), .b(b[23:20]), .cin(w5), .S(
       S[23:20]), .cout(w6) );
  carry_look_ahead_4bit_1 U7 ( .a(a[27:24]), .b(b[27:24]), .cin(w6), .S(
       S[27:24]), .cout(w7) );
  carry_look_ahead_4bit_0 U8 ( .a(a[31:28]), .b(b[31:28]), .cin(w7), .S(
       S[31:28]), .cout(cout) );
  TIELBWP7T U10 ( .ZN(\*Logic0* ) );
endmodule
```

Cuadro 89: Archivo ADDER32b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O

13.5.6. Síntesis lógica del circuito sintetizado con pines de I/O

A través del archivo `ADDER32b_sintetizado_and_IO.v` se obtuvo el circuito completamente sintetizado. Este archivo se generó en el directorio `Outputs/ADDER32b/Outputs_finales` con el nombre `ADDER32b_sintesis_final.v`. Además, se generaron otros dos archivos que también servirán para el siguiente proceso de síntesis física.

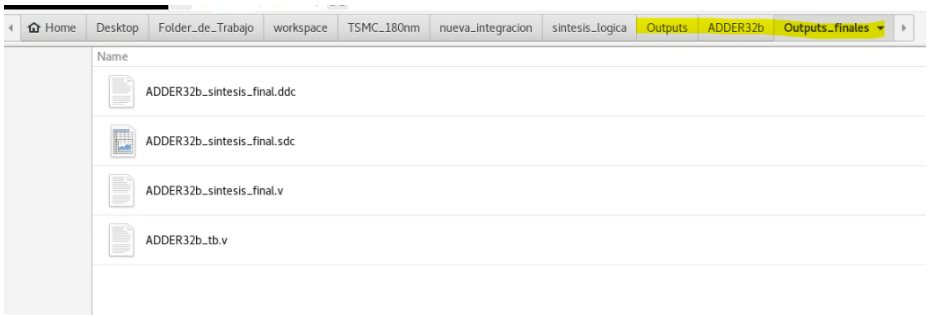


Figura 133: Archivos generados en la carpeta del circuito en Outputs_finales

En el Cuadro 90 se muestra el archivo Verilog obtenido.

```
////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date  : Fri Nov 1 23:03:57 2024
////////////////////////////////////////////////////////////////

module carry_look_ahead_4bit_7 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
```

```
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_6 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_5 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule
```

```
module carry_look_ahead_4bit_4 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
```

```
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;
  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule
```

```
module carry_look_ahead_32bit ( a, b, cin, S, cout );
  input [31:0] a;
  input [31:0] b;
  output [31:0] S;
  input cin;
  output cout;
  wire w1, w2, w3, w4, w5, w6, w7, n1;
  carry_look_ahead_4bit_7 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(n1), .S(S[3:0]),
       .cout(w1) );
  carry_look_ahead_4bit_6 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_5 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_4 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(w4) );
  carry_look_ahead_4bit_3 U5 ( .a(a[19:16]), .b(b[19:16]), .cin(w4), .S(
       S[19:16]), .cout(w5) );
  carry_look_ahead_4bit_2 U6 ( .a(a[23:20]), .b(b[23:20]), .cin(w5), .S(
       S[23:20]), .cout(w6) );
  carry_look_ahead_4bit_1 U7 ( .a(a[27:24]), .b(b[27:24]), .cin(w6), .S(
       S[27:24]), .cout(w7) );
  carry_look_ahead_4bit_0 U8 ( .a(a[31:28]), .b(b[31:28]), .cin(w7), .S(
       S[31:28]), .cout(cout) );
  TIELBWP7T U9 ( .ZN(n1) );
endmodule

module carry_look_ahead_32bit_IO ( a, b, cin, S, cout );
  input [31:0] a;
  input [31:0] b;
  output [31:0] S;
  input cin;
  output cout;
  wire cin_w, cout_w, n1, n2;
  wire [31:0] a_w;
  wire [31:0] b_w;
  wire [31:0] S_w;
  tri [31:0] a;
  tri [31:0] b;
  tri cin;
  tri [31:0] S;
  tri cout;
  carry_look_ahead_32bit U ( .a(a_w), .b(b_w), .cin(cin_w), .S(S_w), .cout(
       cout_w) );
  PDDW0204SCDG U_a0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[0]), .DS(n1), .PE(n1),
       .C(a_w[0]) );
  PDDW0204SCDG U_a1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[1]), .DS(n1), .PE(n1),
       .C(a_w[1]) );
  PDDW0204SCDG U_a2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[2]), .DS(n1), .PE(n1),
       .C(a_w[2]) );
  PDDW0204SCDG U_a3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[3]), .DS(n1), .PE(n1),
       .C(a_w[3]) );
  PDDW0204SCDG U_a4 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[4]), .DS(n1), .PE(n1),
       .C(a_w[4]) );
```

```
    PDDW0204SCDG U_a5 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[5]), .DS(n1), .PE(n1),
        .C(a_w[5]) );
    PDDW0204SCDG U_a6 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[6]), .DS(n1), .PE(n1),
        .C(a_w[6]) );
    PDDW0204SCDG U_a7 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[7]), .DS(n1), .PE(n1),
        .C(a_w[7]) );
    PDDW0204SCDG U_a8 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[8]), .DS(n1), .PE(n1),
        .C(a_w[8]) );
    PDDW0204SCDG U_a9 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[9]), .DS(n1), .PE(n1),
        .C(a_w[9]) );
    PDDW0204SCDG U_a10 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[10]), .DS(n1), .PE(n1)
        , .C(a_w[10]) );
    PDDW0204SCDG U_a11 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[11]), .DS(n1), .PE(n1)
        , .C(a_w[11]) );
    PDDW0204SCDG U_a12 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[12]), .DS(n1), .PE(n1)
        , .C(a_w[12]) );
    PDDW0204SCDG U_a13 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[13]), .DS(n1), .PE(n1)
        , .C(a_w[13]) );
    PDDW0204SCDG U_a14 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[14]), .DS(n1), .PE(n1)
        , .C(a_w[14]) );
    PDDW0204SCDG U_a15 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[15]), .DS(n1), .PE(n1)
        , .C(a_w[15]) );
    PDDW0204SCDG U_a16 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[16]), .DS(n1), .PE(n1)
        , .C(a_w[16]) );
    PDDW0204SCDG U_a17 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[17]), .DS(n1), .PE(n1)
        , .C(a_w[17]) );
    PDDW0204SCDG U_a18 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[18]), .DS(n1), .PE(n1)
        , .C(a_w[18]) );
    PDDW0204SCDG U_a19 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[19]), .DS(n1), .PE(n1)
        , .C(a_w[19]) );
    PDDW0204SCDG U_a20 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[20]), .DS(n1), .PE(n1)
        , .C(a_w[20]) );
    PDDW0204SCDG U_a21 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[21]), .DS(n1), .PE(n1)
        , .C(a_w[21]) );
    PDDW0204SCDG U_a22 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[22]), .DS(n1), .PE(n1)
        , .C(a_w[22]) );
    PDDW0204SCDG U_a23 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[23]), .DS(n1), .PE(n1)
        , .C(a_w[23]) );
    PDDW0204SCDG U_a24 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[24]), .DS(n1), .PE(n1)
        , .C(a_w[24]) );
    PDDW0204SCDG U_a25 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[25]), .DS(n1), .PE(n1)
        , .C(a_w[25]) );
    PDDW0204SCDG U_a26 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[26]), .DS(n1), .PE(n1)
        , .C(a_w[26]) );
    PDDW0204SCDG U_a27 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[27]), .DS(n1), .PE(n1)
        , .C(a_w[27]) );
    PDDW0204SCDG U_a28 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[28]), .DS(n1), .PE(n1)
        , .C(a_w[28]) );
    PDDW0204SCDG U_a29 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[29]), .DS(n1), .PE(n1)
        , .C(a_w[29]) );
    PDDW0204SCDG U_a30 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[30]), .DS(n1), .PE(n1)
        , .C(a_w[30]) );
    PDDW0204SCDG U_a31 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[31]), .DS(n1), .PE(n1)
        , .C(a_w[31]) );
```

```
    PDDW0204SCDG U_b0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[0]), .DS(n1), .PE(n1),
        .C(b_w[0]) );
    PDDW0204SCDG U_b1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[1]), .DS(n1), .PE(n1),
        .C(b_w[1]) );
    PDDW0204SCDG U_b2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[2]), .DS(n1), .PE(n1),
        .C(b_w[2]) );
    PDDW0204SCDG U_b3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[3]), .DS(n1), .PE(n1),
        .C(b_w[3]) );
    PDDW0204SCDG U_b4 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[4]), .DS(n1), .PE(n1),
        .C(b_w[4]) );
    PDDW0204SCDG U_b5 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[5]), .DS(n1), .PE(n1),
        .C(b_w[5]) );
    PDDW0204SCDG U_b6 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[6]), .DS(n1), .PE(n1),
        .C(b_w[6]) );
    PDDW0204SCDG U_b7 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[7]), .DS(n1), .PE(n1),
        .C(b_w[7]) );
    PDDW0204SCDG U_b8 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[8]), .DS(n1), .PE(n1),
        .C(b_w[8]) );
    PDDW0204SCDG U_b9 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[9]), .DS(n1), .PE(n1),
        .C(b_w[9]) );
    PDDW0204SCDG U_b10 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[10]), .DS(n1), .PE(n1)
        , .C(b_w[10]) );
    PDDW0204SCDG U_b11 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[11]), .DS(n1), .PE(n1)
        , .C(b_w[11]) );
    PDDW0204SCDG U_b12 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[12]), .DS(n1), .PE(n1)
        , .C(b_w[12]) );
    PDDW0204SCDG U_b13 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[13]), .DS(n1), .PE(n1)
        , .C(b_w[13]) );
    PDDW0204SCDG U_b14 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[14]), .DS(n1), .PE(n1)
        , .C(b_w[14]) );
    PDDW0204SCDG U_b15 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[15]), .DS(n1), .PE(n1)
        , .C(b_w[15]) );
    PDDW0204SCDG U_b16 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[16]), .DS(n1), .PE(n1)
        , .C(b_w[16]) );
    PDDW0204SCDG U_b17 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[17]), .DS(n1), .PE(n1)
        , .C(b_w[17]) );
    PDDW0204SCDG U_b18 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[18]), .DS(n1), .PE(n1)
        , .C(b_w[18]) );
    PDDW0204SCDG U_b19 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[19]), .DS(n1), .PE(n1)
        , .C(b_w[19]) );
    PDDW0204SCDG U_b20 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[20]), .DS(n1), .PE(n1)
        , .C(b_w[20]) );
    PDDW0204SCDG U_b21 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[21]), .DS(n1), .PE(n1)
        , .C(b_w[21]) );
    PDDW0204SCDG U_b22 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[22]), .DS(n1), .PE(n1)
        , .C(b_w[22]) );
    PDDW0204SCDG U_b23 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[23]), .DS(n1), .PE(n1)
        , .C(b_w[23]) );
    PDDW0204SCDG U_b24 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[24]), .DS(n1), .PE(n1)
        , .C(b_w[24]) );
    PDDW0204SCDG U_b25 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[25]), .DS(n1), .PE(n1)
        , .C(b_w[25]) );
    PDDW0204SCDG U_b26 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[26]), .DS(n1), .PE(n1)
        , .C(b_w[26]) );
```

```
    PDDW0204SCDG U_b27 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[27]), .DS(n1), .PE(n1)
        , .C(b_w[27]) );
    PDDW0204SCDG U_b28 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[28]), .DS(n1), .PE(n1)
        , .C(b_w[28]) );
    PDDW0204SCDG U_b29 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[29]), .DS(n1), .PE(n1)
        , .C(b_w[29]) );
    PDDW0204SCDG U_b30 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[30]), .DS(n1), .PE(n1)
        , .C(b_w[30]) );
    PDDW0204SCDG U_b31 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[31]), .DS(n1), .PE(n1)
        , .C(b_w[31]) );
    PDDW0204SCDG U_cin ( .I(n1), .OEN(n2), .IE(n2), .PAD(cin), .DS(n1), .PE(n1),
         .C(cin_w) );
    PDDW0204SCDG U_S0 ( .I(S_w[0]), .OEN(n1), .IE(n1), .PAD(S[0]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S1 ( .I(S_w[1]), .OEN(n1), .IE(n1), .PAD(S[1]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S2 ( .I(S_w[2]), .OEN(n1), .IE(n1), .PAD(S[2]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S3 ( .I(S_w[3]), .OEN(n1), .IE(n1), .PAD(S[3]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S4 ( .I(S_w[4]), .OEN(n1), .IE(n1), .PAD(S[4]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S5 ( .I(S_w[5]), .OEN(n1), .IE(n1), .PAD(S[5]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S6 ( .I(S_w[6]), .OEN(n1), .IE(n1), .PAD(S[6]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S7 ( .I(S_w[7]), .OEN(n1), .IE(n1), .PAD(S[7]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S8 ( .I(S_w[8]), .OEN(n1), .IE(n1), .PAD(S[8]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S9 ( .I(S_w[9]), .OEN(n1), .IE(n1), .PAD(S[9]), .DS(n1), .PE(
         n1) );
    PDDW0204SCDG U_S10 ( .I(S_w[10]), .OEN(n1), .IE(n1), .PAD(S[10]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S11 ( .I(S_w[11]), .OEN(n1), .IE(n1), .PAD(S[11]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S12 ( .I(S_w[12]), .OEN(n1), .IE(n1), .PAD(S[12]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S13 ( .I(S_w[13]), .OEN(n1), .IE(n1), .PAD(S[13]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S14 ( .I(S_w[14]), .OEN(n1), .IE(n1), .PAD(S[14]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S15 ( .I(S_w[15]), .OEN(n1), .IE(n1), .PAD(S[15]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S16 ( .I(S_w[16]), .OEN(n1), .IE(n1), .PAD(S[16]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S17 ( .I(S_w[17]), .OEN(n1), .IE(n1), .PAD(S[17]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S18 ( .I(S_w[18]), .OEN(n1), .IE(n1), .PAD(S[18]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S19 ( .I(S_w[19]), .OEN(n1), .IE(n1), .PAD(S[19]), .DS(n1),
         .PE(n1) );
    PDDW0204SCDG U_S20 ( .I(S_w[20]), .OEN(n1), .IE(n1), .PAD(S[20]), .DS(n1),
         .PE(n1) );
```

```
    PDDW0204SCDG U_S21 ( .I(S_w[21]), .OEN(n1), .IE(n1), .PAD(S[21]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S22 ( .I(S_w[22]), .OEN(n1), .IE(n1), .PAD(S[22]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S23 ( .I(S_w[23]), .OEN(n1), .IE(n1), .PAD(S[23]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S24 ( .I(S_w[24]), .OEN(n1), .IE(n1), .PAD(S[24]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S25 ( .I(S_w[25]), .OEN(n1), .IE(n1), .PAD(S[25]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S26 ( .I(S_w[26]), .OEN(n1), .IE(n1), .PAD(S[26]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S27 ( .I(S_w[27]), .OEN(n1), .IE(n1), .PAD(S[27]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S28 ( .I(S_w[28]), .OEN(n1), .IE(n1), .PAD(S[28]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S29 ( .I(S_w[29]), .OEN(n1), .IE(n1), .PAD(S[29]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S30 ( .I(S_w[30]), .OEN(n1), .IE(n1), .PAD(S[30]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_S31 ( .I(S_w[31]), .OEN(n1), .IE(n1), .PAD(S[31]), .DS(n1),
          .PE(n1) );
    PDDW0204SCDG U_cout ( .I(cout_w), .OEN(n1), .IE(n1), .PAD(cout), .DS(n1),
          .PE(n1) );
    TIELBWP7T U6 ( .ZN(n1) );
    TIEHBWP7T U7 ( .Z(n2) );
  endmodule
```

Cuadro 90: Archivo ADDER32b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O

13.5.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O

El último paso que se realizó fue la simulación del circuito completamente sintetizado junto con los pines de I/O. Esto se llevó a cabo nuevamente para comprobar que el circuito completamente sintetizado se comporte correctamente.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 91 en una terminal abierta en `Outputs/ADDER32b/Outputs_finales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v ADDER32b_sintesis_final.v
    ADDER32b_tb.v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 91: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se observa en la Figura 134, el comportamiento es idéntico al que se obtuvo anteriormente, lo que confirma que el *netlist* final obtenido del circuito sintetizado con pines de I/O es correcto.

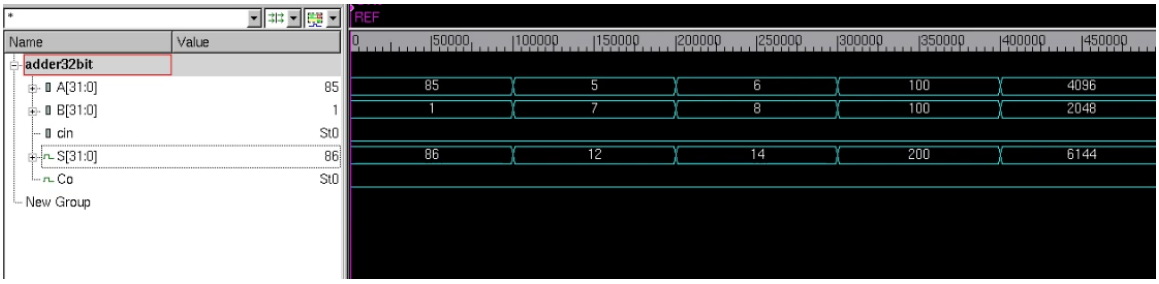


Figura 134: Simulación del *netlist* final

13.6. Proceso para la síntesis lógica de una *carry look-ahead subtractor* de 16 bits

13.6.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench*

El primer paso fue crear una carpeta en el directorio de *inputs*, nombrada según el circuito principal a sintetizar; en este caso, `SUB16b` (tanto la carpeta como el nombre del archivo Verilog que contiene el circuito principal deben tener el mismo nombre). Luego, se diseñaron los archivos necesarios para la simulación y síntesis del circuito.

Siguiendo la estructura explicada en el capítulo y sección 9.2, el archivo que contiene el módulo principal del restador de 16 bits se nombró `SUB16b.v`. El archivo de *testbench*, utilizado para simular y verificar el funcionamiento del circuito, se denominó `SUB16b_tb.v`. Por último, el archivo que define el módulo de entradas y salidas se nombró `IO_SUB16b.v`.

En este caso, se utilizaron otros circuitos para construir el circuito principal. Se emplearon los sumadores de 4 y 16 bits diseñados anteriormente como base para crear el restador de 16 bits. Por lo tanto, los archivos Verilog de los sumadores de 4 y 16 bits también se encuentran en la carpeta `Inputs/SUB16b/`, ya que para los procesos de simulación y síntesis es importante hacer referencia a todos los archivos utilizados.

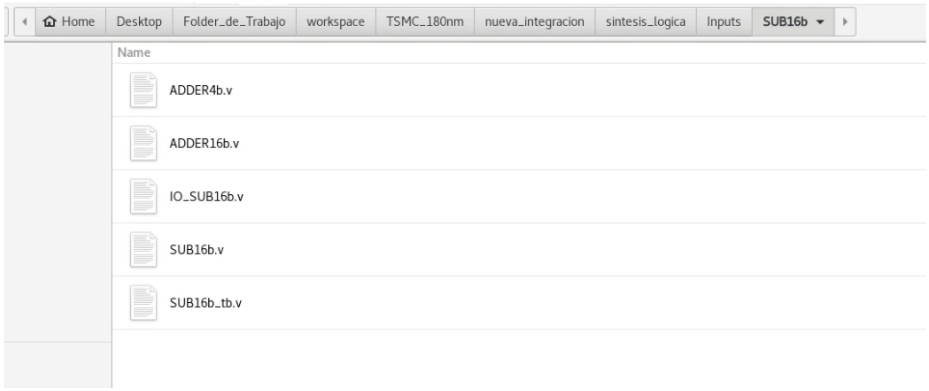


Figura 135: Archivos necesarios en la carpeta del circuito en *inputs*

En el Cuadro 92 se muestra el diseño del circuito, el cual hace referencia al módulo del sumador de 16 bits para su implementación. Por otro lado, en el Cuadro 93 se presenta el diseño del módulo de pines de entrada y salida. Además, en el Cuadro 94 se muestra el *testbench* diseñado para la simulación del circuito original y del circuito sintetizado.

Dado que el diseño del circuito incluye múltiples entradas y salidas, en el módulo de pines es necesario hacer referencia a cada una de ellas de forma individual. Sin embargo, el proceso de creación de cada pin es el mismo para todas las entradas y salidas (ver Cuadro 93).

```
module substractor_16bit(a,b, cin, S,cout);

    input [15:0] a,b;
    input cin;

    output [15:0] S;
    output cout;

    wire [15:0] b_not;
    wire [15:0] b_comp;

    not(b_not[0], b[0]);
    not(b_not[1], b[1]);
    not(b_not[2], b[2]);
    not(b_not[3], b[3]);
    not(b_not[4], b[4]);
    not(b_not[5], b[5]);
    not(b_not[6], b[6]);
    not(b_not[7], b[7]);
    not(b_not[8], b[8]);
    not(b_not[9], b[9]);
    not(b_not[10], b[10]);
    not(b_not[11], b[11]);
    not(b_not[12], b[12]);
    not(b_not[13], b[13]);
    not(b_not[14], b[14]);
    not(b_not[15], b[15]);

    assign b_comp = b_not + 1'b1;
```

```
        carry_look_ahead_16bit U35(a, b_comp, 1'b0, S, cout);
        //assign S = {s8, s7, s6, s5, s4, s3, s2, s1};
endmodule
```

Cuadro 92: Archivo SUB16b.v: diseño de un restador de 16 bits en Verilog

```
module substractor_16bit_IO(
    input [15:0] a,
    input [15:0] b,
    input cin,
    output [15:0] S,
    output cout
);
    wire [15:0] a_w, b_w, S_w;
    wire cin_w, cout_w;

    // Instancia del modulo substractor_16bit
    substractor_16bit U (
        .a(a_w),
        .b(b_w),
        .cin(cin_w),
        .S(S_w),
        .cout(cout_w)
    );

    // Interconexion de pines de entrada (a, b, cin)
    // Entradas a
    PDDW0204SCDG U_a0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[0]), .C(a_w[0]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[1]), .C(a_w[1]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[2]), .C(a_w[2]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[3]), .C(a_w[3]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[4]), .C(a_w[4]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[5]), .C(a_w[5]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[6]), .C(a_w[6]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[7]), .C(a_w[7]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[8]), .C(a_w[8]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[9]), .C(a_w[9]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[10]), .C(a_w
        [10]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[11]), .C(a_w
        [11]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[12]), .C(a_w
        [12]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[13]), .C(a_w
```

```
        [13]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[14]), .C(a_w
        [14]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[15]), .C(a_w
        [15]), .PE(1'b0), .IE(1'b1));

    // Entradas b
    PDDW0204SCDG U_b0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[0]), .C(b_w[0]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[1]), .C(b_w[1]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[2]), .C(b_w[2]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[3]), .C(b_w[3]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[4]), .C(b_w[4]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[5]), .C(b_w[5]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[6]), .C(b_w[6]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[7]), .C(b_w[7]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[8]), .C(b_w[8]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[9]), .C(b_w[9]),
         .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[10]), .C(b_w
        [10]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[11]), .C(b_w
        [11]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[12]), .C(b_w
        [12]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[13]), .C(b_w
        [13]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[14]), .C(b_w
        [14]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_b15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[15]), .C(b_w
        [15]), .PE(1'b0), .IE(1'b1));

    // Entrada cin
    PDDW0204SCDG U_cin (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(cin), .C(cin_w),
        .PE(1'b0), .IE(1'b1));

    // Interconexion de pines de salida (S, cout)
    // Salidas S
    PDDW0204SCDG U_S0 (.I(S_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(S[0]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S1 (.I(S_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(S[1]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S2 (.I(S_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(S[2]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S3 (.I(S_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(S[3]), .C(1'bx),
         .PE(1'b0), .IE(1'b0));
```

```
        PDDW0204SCDG U_S4 (.I(S_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(S[4]), .C(1'bx),
            .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S5 (.I(S_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(S[5]), .C(1'bx),
            .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S6 (.I(S_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(S[6]), .C(1'bx),
            .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S7 (.I(S_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(S[7]), .C(1'bx),
            .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S8 (.I(S_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(S[8]), .C(1'bx),
            .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S9 (.I(S_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(S[9]), .C(1'bx),
            .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S10 (.I(S_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(S[10]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S11 (.I(S_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(S[11]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S12 (.I(S_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(S[12]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S13 (.I(S_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(S[13]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S14 (.I(S_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(S[14]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));
        PDDW0204SCDG U_S15 (.I(S_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(S[15]), .C(1'
            bx), .PE(1'b0), .IE(1'b0));


        // Salida cout
        PDDW0204SCDG U_cout (.I(cout_w), .DS(1'b0), .OEN(1'b0), .PAD(cout), .C(1'bx
            ), .PE(1'b0), .IE(1'b0));

    //Pines de VDD y VSS
    PVDD1CDG PVDD();
    PVSS1CDG PVSS();

    endmodule
```

Cuadro 93: Archivo IO_SUB16b.v: diseño de módulo de entradas y salidas para un restador de 16 bits en Verilog

```
module testbench_substractor_16bit;
        reg [15:0] a, b;
        reg cin;
        wire [15:0] S;
        wire cout;

        // Instanciar el modulo substractor_16bit
        substractor_16bit sub(.a(a), .b(b), .cin(cin), .S(S), .cout(cout));

        initial begin
        // Inicializacion de archivos de volcado de datos
        $fsdbDumpfile("substractor16b.fsdb");
        $fsdbDumpvars(0, testbench_substractor_16bit);
        $dumpfile("substractor16b.vcd");
        $dumpvars(0, testbench_substractor_16bit);
```

```
        // Pruebas con diferentes valores de entrada
        cin = 0; a = 16'd100; b = 16'd50;
        #100;

        a = 16'd500; b = 16'd200;
        #100;

        a = 16'd1024; b = 16'd512;
        #100;

        a = 16'd4096; b = 16'd2048;
        #100;

        a = 16'd8192; b = 16'd4096;
        #100;

        a = 16'd16384; b = 16'd8192;
        #100;

        // Finalizar simulacion
        $finish;
        end
endmodule
```

Cuadro 94: Archivo SUB16b_tb.v: diseño del *testbench* para simulación de un restador de 16 bits en Verilog

13.6.2. Simulación del diseño original del circuito

Como siguiente paso, se simuló el circuito original con el objetivo de verificar que no existieran errores de compilación y que el diseño del circuito fuera correcto. Como se explicó anteriormente, si no se realiza este paso, existe el riesgo de encontrar errores debido a posibles fallos en el diseño del circuito, lo que impediría continuar con la ejecución de la síntesis lógica.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 95 en una terminal abierta en el directorio `Inputs/SUB16b/`.

```
vcs -v SUB16b.v ADDER16b.v ADDER4b.v SUB16b_tb.v -o simulacion -full64 -
    debug_access+all

./simulacion

dve -full64
```

Cuadro 95: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Como se observa en el Cuadro 95, es necesario colocar el nombre de todos los archivos Verilog que se están utilizando. Si no se realiza este paso, se producirá un error, ya que la aplicación no encontrará el módulo al que el circuito original hace referencia.

Siguiendo los pasos descritos en el capítulo y sección 12.1, para la visualización del archivo `.vcd` generado al ejecutar el archivo de simulación, se obtuvo la Figura 136.

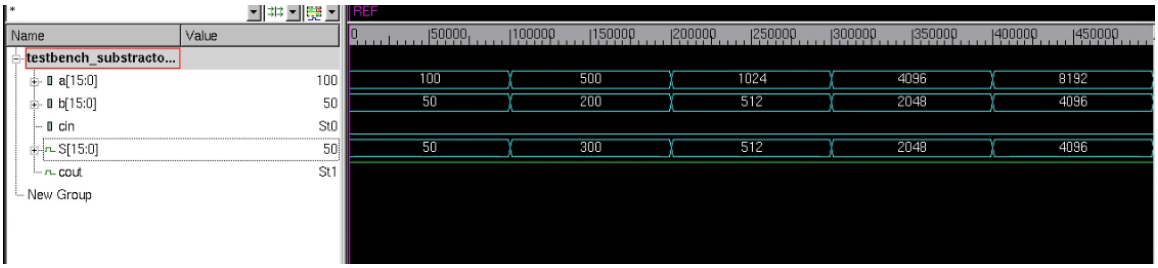


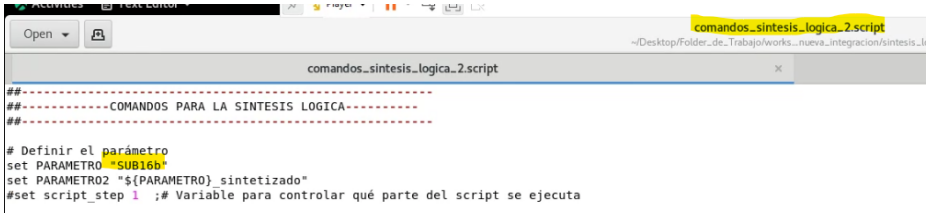
Figura 136: Simulación del sumador de 16 bits en su diseño original

Como se observa en la Figura 136, el comportamiento del circuito es el esperado, ya que se realiza la resta entre las entradas A y B, y el resultado se obtiene en la salida (S).

Por lo tanto, se procedió a ejecutar el *script* para llevar a cabo la síntesis lógica del circuito.

13.6.3. Síntesis lógica del diseño original del circuito

Antes de proceder a ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, fue necesario abrir tanto este *script* como el archivo `comandos_sintesis_logica_2.script` para definir en la variable `PARAMETRO` el nombre de la carpeta que se creó en *inputs*, la cual contiene el circuito a sintetizar. En este caso, se definió como `SUB16b`.



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 137: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

De igual manera, fue necesario modificar una parte del *script* `comandos_sintesis_-`

`logica_2.script`. Dado que el diseño del restador de 16 bits utiliza los módulos que se encuentran en otros archivos Verilog, es necesario hacer referencia a estos archivos para sintetizar correctamente el diseño.

La modificación debe realizarse en el primer paso de síntesis, en la parte donde se leen los archivos que describen el circuito. En la línea que contiene el comando `read_file`, se deben agregar los archivos adicionales necesarios. En este caso, la modificación quedaría de la siguiente forma:

```
    if { $script_step == 1 } {
    # Parte 1 - Sintesis Cell original

    # Usar el parametro en lugar del nombre del modulo
    read_file -format verilog "Inputs/$PARAMETRO/$PARAMETRO.v Inputs/$PARAMETRO/
        ADDER4b.v Inputs/$PARAMETRO/ADDER16b.v"
```

Cuadro 96: Modificación del *script* comandos_sintesis_logica_2.script

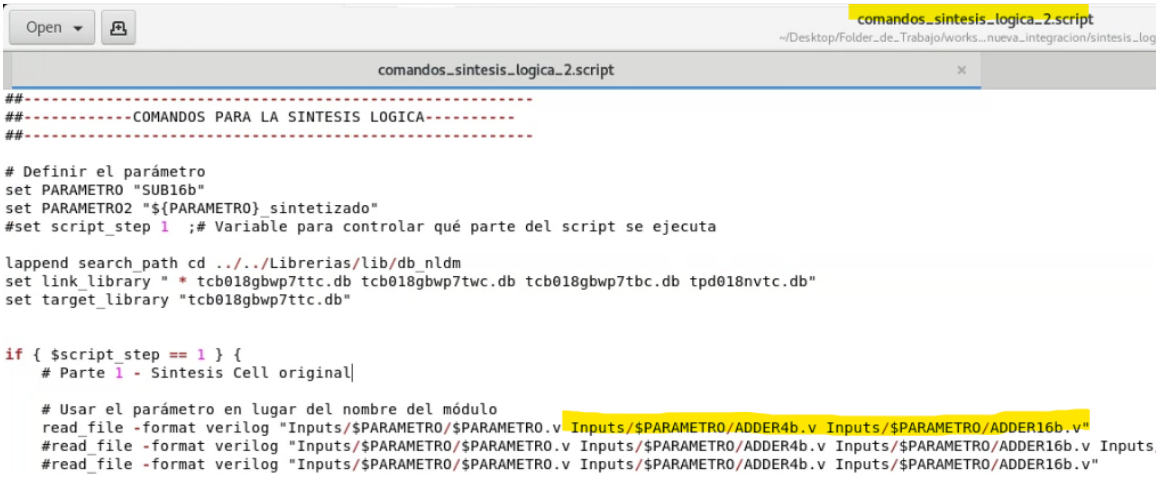


Figura 138: Modificación del *script* comandos_sintesis_logica_2.script

Como se observa en el Cuadro 96 y en la Figura 138, la única modificación realizada fue agregar la lectura de los archivos `ADDER4b.v` y `ADDER16b.v` en el comando `read_file`, ya que los módulos contenidos en esos archivos son utilizados en `SUB16b.v` y, por lo tanto, también deben ser sintetizados. Si no se realiza esta modificación, se producirá un error en el proceso de síntesis.

Luego, se ejecutó el *script* utilizando el comando que se muestra en el Cuadro 97, en una terminal abierta en el directorio `sintesis_logica`.

```
    ./ejecutar_sintesis_logica_2.bash
```

Cuadro 97: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash

Al ejecutar este comando, se creó una carpeta en *outputs* con el nombre `SUB16b`. Además, se generaron dos subdirectorios dentro de esta, llamados `Outputs_iniciales` y `Outputs_-`

`finales`. En la terminal no apareció ningún error y se generaron los reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O. Asimismo, en la carpeta `Outputs/SUB16b/Outputs_iniciales` se creó un archivo llamado `SUB16b_sintetizado.v`, el cual corresponde al archivo resultante de la primera etapa de la síntesis lógica (ver Figuras 139 y 140).

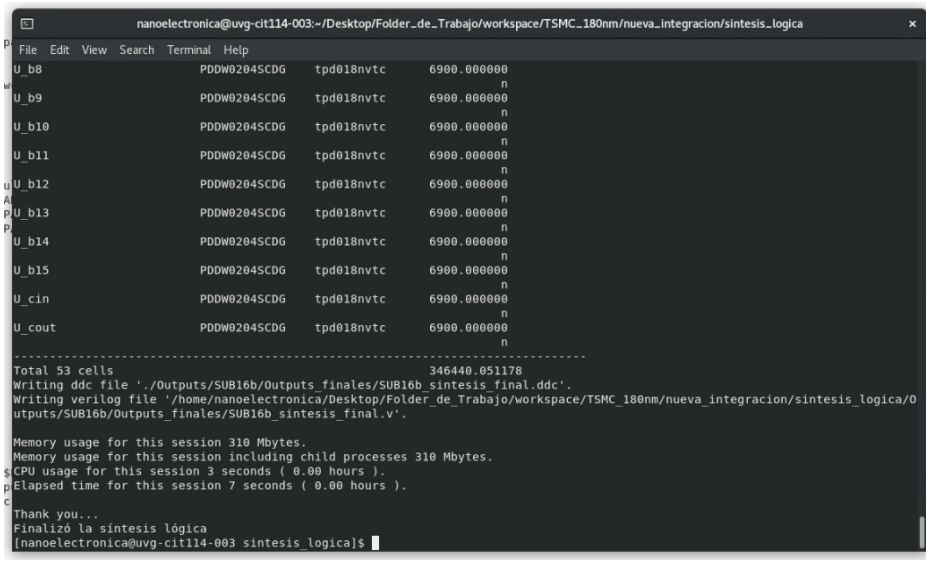


Figura 139: Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores

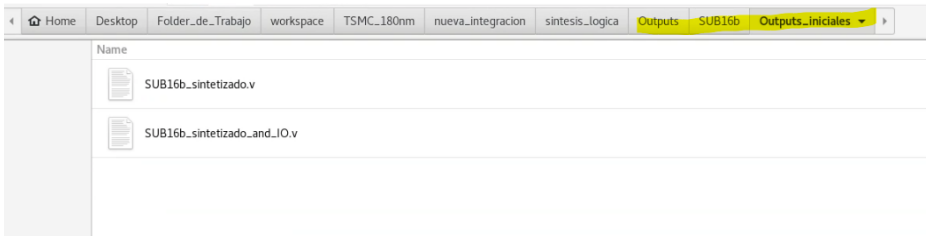


Figura 140: Archivos generados en la carpeta del circuito en Outputs_iniciales

El archivo `SUB16b_sintetizado.v`, que corresponde al circuito sintetizado, se muestra en el Cuadro 98.

```
////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date  : Sat Nov 2 10:26:30 2024
////////////////////////////////////////////////////////////////

module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
```

```
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
```

```
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire \*Logic0* , w1, w2, w3;

  carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(\*Logic0* ), .S(
       S[3:0]), .cout(w1) );
  carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(cout) );
  TIELBWP7T U6 ( .ZN(\*Logic0* ) );
endmodule

module substractor_16bit_DW01_inc_0 ( A, SUM );
  input [15:0] A;
  output [15:0] SUM;

  wire [15:2] carry;

  HA1D0BWP7T U1_1_14 ( .A(A[14]), .B(carry[14]), .CO(carry[15]), .S(SUM[14])
```

```
        );
  HA1D0BWP7T U1_1_13 ( .A(A[13]), .B(carry[13]), .CO(carry[14]), .S(SUM[13])
        );
  HA1D0BWP7T U1_1_12 ( .A(A[12]), .B(carry[12]), .CO(carry[13]), .S(SUM[12])
        );
  HA1D0BWP7T U1_1_11 ( .A(A[11]), .B(carry[11]), .CO(carry[12]), .S(SUM[11])
        );
  HA1D0BWP7T U1_1_10 ( .A(A[10]), .B(carry[10]), .CO(carry[11]), .S(SUM[10])
        );
  HA1D0BWP7T U1_1_9 ( .A(A[9]), .B(carry[9]), .CO(carry[10]), .S(SUM[9]) );
  HA1D0BWP7T U1_1_8 ( .A(A[8]), .B(carry[8]), .CO(carry[9]), .S(SUM[8]) );
  HA1D0BWP7T U1_1_7 ( .A(A[7]), .B(carry[7]), .CO(carry[8]), .S(SUM[7]) );
  HA1D0BWP7T U1_1_6 ( .A(A[6]), .B(carry[6]), .CO(carry[7]), .S(SUM[6]) );
  HA1D0BWP7T U1_1_5 ( .A(A[5]), .B(carry[5]), .CO(carry[6]), .S(SUM[5]) );
  HA1D0BWP7T U1_1_4 ( .A(A[4]), .B(carry[4]), .CO(carry[5]), .S(SUM[4]) );
  HA1D0BWP7T U1_1_3 ( .A(A[3]), .B(carry[3]), .CO(carry[4]), .S(SUM[3]) );
  HA1D0BWP7T U1_1_2 ( .A(A[2]), .B(carry[2]), .CO(carry[3]), .S(SUM[2]) );
  HA1D0BWP7T U1_1_1 ( .A(A[1]), .B(A[0]), .CO(carry[2]), .S(SUM[1]) );
  INVD1BWP7T U1 ( .I(A[0]), .ZN(SUM[0]) );
  CKXOR2D0BWP7T U2 ( .A1(carry[15]), .A2(A[15]), .Z(SUM[15]) );
endmodule

module substractor_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire \*Logic0* , n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27,
       n28, n29, n30, n31, n32;
  wire [15:0] b_comp;

  carry_look_ahead_16bit U35 ( .a(a), .b(b_comp), .cin(\*Logic0* ), .S(S),
       .cout(cout) );
  substractor_16bit_DW01_inc_0 add_29 ( .A({n17, n18, n19, n20, n21, n22, n23,
       n24, n25, n26, n27, n28, n29, n30, n31, n32}), .SUM(b_comp) );
  TIELBWP7T U19 ( .ZN(\*Logic0* ) );
  CKND0BWP7T U20 ( .I(b[15]), .ZN(n17) );
  CKND0BWP7T U21 ( .I(b[14]), .ZN(n18) );
  CKND0BWP7T U22 ( .I(b[13]), .ZN(n19) );
  CKND0BWP7T U23 ( .I(b[12]), .ZN(n20) );
  CKND0BWP7T U24 ( .I(b[11]), .ZN(n21) );
  CKND0BWP7T U25 ( .I(b[10]), .ZN(n22) );
  CKND0BWP7T U26 ( .I(b[9]), .ZN(n23) );
  CKND0BWP7T U27 ( .I(b[8]), .ZN(n24) );
  CKND0BWP7T U28 ( .I(b[7]), .ZN(n25) );
  CKND0BWP7T U29 ( .I(b[6]), .ZN(n26) );
  CKND0BWP7T U30 ( .I(b[5]), .ZN(n27) );
  CKND0BWP7T U31 ( .I(b[4]), .ZN(n28) );
  CKND0BWP7T U32 ( .I(b[3]), .ZN(n29) );
  CKND0BWP7T U33 ( .I(b[2]), .ZN(n30) );
  CKND0BWP7T U34 ( .I(b[1]), .ZN(n31) );
  CKND0BWP7T U36 ( .I(b[0]), .ZN(n32) );
endmodule
```

Cuadro 98: Archivo SUB16b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito

13.6.4. Simulación del circuito sintetizado

El siguiente paso que se llevó a cabo, y que es fundamental para corroborar que el circuito sintetizado se comporte correctamente, fue la simulación del circuito sintetizado.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 99 en una terminal abierta en `Outputs/SUB16b/Outputs_iniciales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v SUB16b_sintetizado.v SUB16b_tb.v -
    o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 99: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Luego de haber ejecutado los comandos mencionados anteriormente, se obtuvo la Figura 141. Se observó el mismo comportamiento que con el circuito original, lo que confirma que la síntesis del circuito se realizó correctamente.

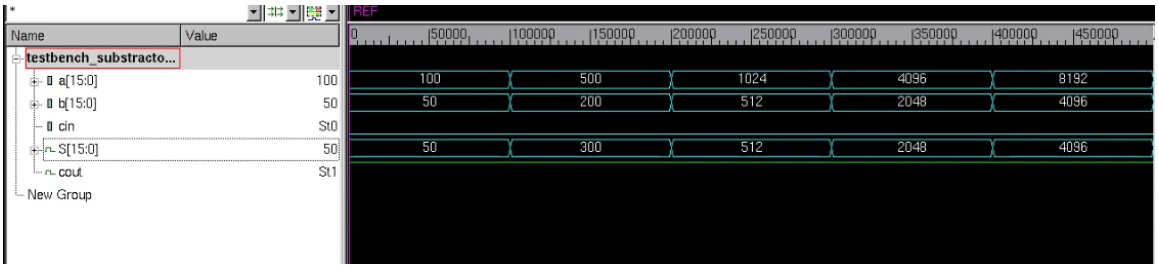


Figura 141: Simulación del restador de 16 bits en su diseño sintetizado

13.6.5. Unión del circuito sintetizado con el módulo de pines de I/O

El archivo denominado `SUB16b_sintetizado_and_IO.v` se creó uniendo el archivo `SUB16b_sintetizado.v` con el archivo `IO_SUB16b.v`. Este proceso también lo realiza automáticamente el *script* antes de comenzar el segundo proceso de síntesis lógica.

Por ello, es fundamental tener diseñado desde el principio el archivo `IO_SUB16b.v` y guardarlo en la carpeta `Inputs/SUB16b`. Si este archivo no está presente, el *script* no podrá localizarlo y el nuevo archivo generado contendrá únicamente el contenido de `SUB16b_sintetizado.v`. Esto ocasionará que, en el siguiente paso, se realice la síntesis nuevamente

sin incluir el módulo de pines de entrada y salida, obteniéndose un resultado similar al del archivo `SUB16b_sintetizado.v`.

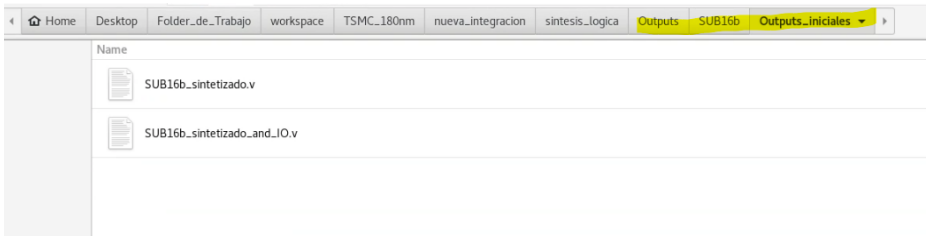


Figura 142: Archivos generados en la carpeta del circuito en Outputs_iniciales

En el Cuadro 100 se muestra el archivo generado. Este archivo se utilizará para obtener el circuito completamente sintetizado con los pines de entrada y salida.

```
module substractor_16bit_IO(
    input [15:0] a,
    input [15:0] b,
    input cin,
    output [15:0] S,
    output cout
);
    wire [15:0] a_w, b_w, S_w;
    wire cin_w, cout_w;

    // Instancia del modulo substractor_16bit
    substractor_16bit U (
        .a(a_w),
        .b(b_w),
        .cin(cin_w),
        .S(S_w),
        .cout(cout_w)
    );

    // Interconexion de pines de entrada (a, b, cin)
    // Entradas a
    PDDW0204SCDG U_a0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[0]), .C(a_w[0]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[1]), .C(a_w[1]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[2]), .C(a_w[2]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[3]), .C(a_w[3]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[4]), .C(a_w[4]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[5]), .C(a_w[5]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[6]), .C(a_w[6]),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_a7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[7]), .C(a_w[7]),
        .PE(1'b0), .IE(1'b1));
```

```
        PDDW0204SCDG U_a8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[8]), .C(a_w[8]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[9]), .C(a_w[9]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[10]), .C(a_w
            [10]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[11]), .C(a_w
            [11]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[12]), .C(a_w
            [12]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[13]), .C(a_w
            [13]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[14]), .C(a_w
            [14]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_a15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(a[15]), .C(a_w
            [15]), .PE(1'b0), .IE(1'b1));

        // Entradas b
        PDDW0204SCDG U_b0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[0]), .C(b_w[0]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[1]), .C(b_w[1]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[2]), .C(b_w[2]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[3]), .C(b_w[3]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[4]), .C(b_w[4]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[5]), .C(b_w[5]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[6]), .C(b_w[6]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[7]), .C(b_w[7]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[8]), .C(b_w[8]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[9]), .C(b_w[9]),
            .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[10]), .C(b_w
            [10]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[11]), .C(b_w
            [11]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[12]), .C(b_w
            [12]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[13]), .C(b_w
            [13]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[14]), .C(b_w
            [14]), .PE(1'b0), .IE(1'b1));
        PDDW0204SCDG U_b15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(b[15]), .C(b_w
            [15]), .PE(1'b0), .IE(1'b1));

        // Entrada cin
        PDDW0204SCDG U_cin (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(cin), .C(cin_w),
            .PE(1'b0), .IE(1'b1));
```

```
    // Interconexion de pines de salida (S, cout)
    // Salidas S
    PDDW0204SCDG U_S0 (.I(S_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(S[0]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S1 (.I(S_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(S[1]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S2 (.I(S_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(S[2]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S3 (.I(S_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(S[3]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S4 (.I(S_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(S[4]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S5 (.I(S_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(S[5]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S6 (.I(S_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(S[6]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S7 (.I(S_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(S[7]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S8 (.I(S_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(S[8]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S9 (.I(S_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(S[9]), .C(1'bx),
        .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S10 (.I(S_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(S[10]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S11 (.I(S_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(S[11]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S12 (.I(S_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(S[12]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S13 (.I(S_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(S[13]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S14 (.I(S_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(S[14]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_S15 (.I(S_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(S[15]), .C(1'
        bx), .PE(1'b0), .IE(1'b0));


    // Salida cout
    PDDW0204SCDG U_cout (.I(cout_w), .DS(1'b0), .OEN(1'b0), .PAD(cout), .C(1'bx
        ), .PE(1'b0), .IE(1'b0));

//Pines de VDD y VSS
PVDD1CDG PVDD();
PVSS1CDG PVSS();

endmodule

////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date  : Sat Nov 2 10:26:30 2024
////////////////////////////////////////////////////////////////
```

```
module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;
```

```
  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire \*Logic0* , w1, w2, w3;

  carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(\*Logic0* ), .S(
       S[3:0]), .cout(w1) );
  carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(cout) );
```

```
  TIELBWP7T U6 ( .ZN(\*Logic0* ) );
endmodule


module substractor_16bit_DW01_inc_0 ( A, SUM );
  input [15:0] A;
  output [15:0] SUM;

  wire [15:2] carry;

  HA1D0BWP7T U1_1_14 ( .A(A[14]), .B(carry[14]), .CO(carry[15]), .S(SUM[14])
        );
  HA1D0BWP7T U1_1_13 ( .A(A[13]), .B(carry[13]), .CO(carry[14]), .S(SUM[13])
        );
  HA1D0BWP7T U1_1_12 ( .A(A[12]), .B(carry[12]), .CO(carry[13]), .S(SUM[12])
        );
  HA1D0BWP7T U1_1_11 ( .A(A[11]), .B(carry[11]), .CO(carry[12]), .S(SUM[11])
        );
  HA1D0BWP7T U1_1_10 ( .A(A[10]), .B(carry[10]), .CO(carry[11]), .S(SUM[10])
        );
  HA1D0BWP7T U1_1_9 ( .A(A[9]), .B(carry[9]), .CO(carry[10]), .S(SUM[9]) );
  HA1D0BWP7T U1_1_8 ( .A(A[8]), .B(carry[8]), .CO(carry[9]), .S(SUM[8]) );
  HA1D0BWP7T U1_1_7 ( .A(A[7]), .B(carry[7]), .CO(carry[8]), .S(SUM[7]) );
  HA1D0BWP7T U1_1_6 ( .A(A[6]), .B(carry[6]), .CO(carry[7]), .S(SUM[6]) );
  HA1D0BWP7T U1_1_5 ( .A(A[5]), .B(carry[5]), .CO(carry[6]), .S(SUM[5]) );
  HA1D0BWP7T U1_1_4 ( .A(A[4]), .B(carry[4]), .CO(carry[5]), .S(SUM[4]) );
  HA1D0BWP7T U1_1_3 ( .A(A[3]), .B(carry[3]), .CO(carry[4]), .S(SUM[3]) );
  HA1D0BWP7T U1_1_2 ( .A(A[2]), .B(carry[2]), .CO(carry[3]), .S(SUM[2]) );
  HA1D0BWP7T U1_1_1 ( .A(A[1]), .B(A[0]), .CO(carry[2]), .S(SUM[1]) );
  INVD1BWP7T U1 ( .I(A[0]), .ZN(SUM[0]) );
  CKXOR2D0BWP7T U2 ( .A1(carry[15]), .A2(A[15]), .Z(SUM[15]) );
endmodule


module substractor_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire \*Logic0* , n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27,
        n28, n29, n30, n31, n32;
  wire [15:0] b_comp;

  carry_look_ahead_16bit U35 ( .a(a), .b(b_comp), .cin(\*Logic0* ), .S(S),
       .cout(cout) );
  substractor_16bit_DW01_inc_0 add_29 ( .A({n17, n18, n19, n20, n21, n22, n23,
       n24, n25, n26, n27, n28, n29, n30, n31, n32}), .SUM(b_comp) );
  TIELBWP7T U19 ( .ZN(\*Logic0* ) );
  CKND0BWP7T U20 ( .I(b[15]), .ZN(n17) );
  CKND0BWP7T U21 ( .I(b[14]), .ZN(n18) );
  CKND0BWP7T U22 ( .I(b[13]), .ZN(n19) );
  CKND0BWP7T U23 ( .I(b[12]), .ZN(n20) );
  CKND0BWP7T U24 ( .I(b[11]), .ZN(n21) );
```

```
  CKND0BWP7T U25 ( .I(b[10]), .ZN(n22) );
  CKND0BWP7T U26 ( .I(b[9]), .ZN(n23) );
  CKND0BWP7T U27 ( .I(b[8]), .ZN(n24) );
  CKND0BWP7T U28 ( .I(b[7]), .ZN(n25) );
  CKND0BWP7T U29 ( .I(b[6]), .ZN(n26) );
  CKND0BWP7T U30 ( .I(b[5]), .ZN(n27) );
  CKND0BWP7T U31 ( .I(b[4]), .ZN(n28) );
  CKND0BWP7T U32 ( .I(b[3]), .ZN(n29) );
  CKND0BWP7T U33 ( .I(b[2]), .ZN(n30) );
  CKND0BWP7T U34 ( .I(b[1]), .ZN(n31) );
  CKND0BWP7T U36 ( .I(b[0]), .ZN(n32) );
endmodule
```

Cuadro 100: Archivo SUB16b_sintetizado_and_IO.v: circuito sintetizado con pines de I/O

13.6.6. Síntesis lógica del circuito sintetizado con pines de I/O

A través del archivo `SUB16b_sintetizado_and_IO.v` se obtuvo el circuito completamente sintetizado. Este archivo se generó en el directorio `Outputs/SUB16b/Outputs_finales` con el nombre `SUB16b_sintesis_final.v`. Además, se generaron otros dos archivos que también servirán para el siguiente proceso de síntesis física.

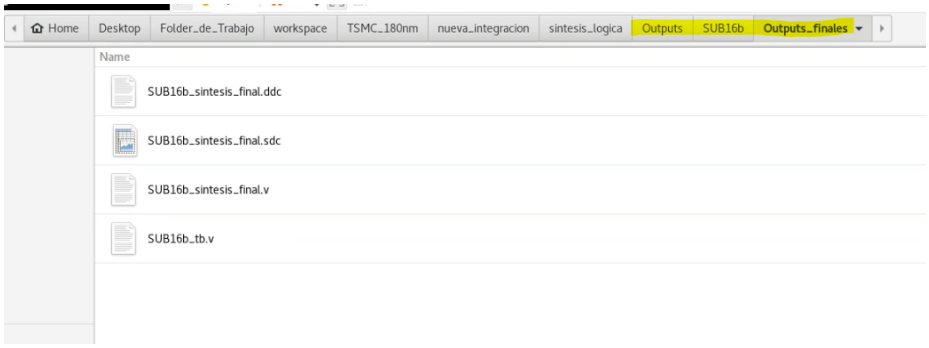


Figura 143: Archivos generados en la carpeta del circuito en Outputs_finales

En el Cuadro 101 se muestra el archivo Verilog obtenido.

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date  : Sat Nov 2 10:26:37 2024
////////////////////////////////////////////////////////////


module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
```

```
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
```

```
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire w1, w2, w3, n1;

  carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(n1), .S(S[3:0]),
       .cout(w1) );
  carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(cout) );
  TIELBWP7T U5 ( .ZN(n1) );
endmodule


module substractor_16bit_DW01_inc_0 ( A, SUM );
  input [15:0] A;
```

```
  output [15:0] SUM;
  wire n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14;

  CKXOR2D0BWP7T U1 ( .A1(A[9]), .A2(n1), .Z(SUM[9]) );
  XNR2D1BWP7T U2 ( .A1(n2), .A2(A[8]), .ZN(SUM[8]) );
  CKND2D0BWP7T U3 ( .A1(A[7]), .A2(n3), .ZN(n2) );
  CKXOR2D0BWP7T U4 ( .A1(n3), .A2(A[7]), .Z(SUM[7]) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(A[6]), .Z(SUM[6]) );
  XNR2D1BWP7T U6 ( .A1(n5), .A2(A[5]), .ZN(SUM[5]) );
  XNR2D1BWP7T U7 ( .A1(n6), .A2(A[4]), .ZN(SUM[4]) );
  CKND2D0BWP7T U8 ( .A1(A[3]), .A2(n7), .ZN(n6) );
  CKXOR2D0BWP7T U9 ( .A1(A[3]), .A2(n7), .Z(SUM[3]) );
  XNR2D1BWP7T U10 ( .A1(n8), .A2(A[2]), .ZN(SUM[2]) );
  XNR2D1BWP7T U11 ( .A1(A[1]), .A2(SUM[0]), .ZN(SUM[1]) );
  CKXOR2D0BWP7T U12 ( .A1(A[15]), .A2(n9), .Z(SUM[15]) );
  INR2D0BWP7T U13 ( .A1(A[14]), .B1(n10), .ZN(n9) );
  XNR2D1BWP7T U14 ( .A1(n10), .A2(A[14]), .ZN(SUM[14]) );
  IND2D0BWP7T U15 ( .A1(n11), .B1(A[13]), .ZN(n10) );
  XNR2D1BWP7T U16 ( .A1(n11), .A2(A[13]), .ZN(SUM[13]) );
  ND3D0BWP7T U17 ( .A1(A[11]), .A2(n12), .A3(A[12]), .ZN(n11) );
  XNR2D1BWP7T U18 ( .A1(n13), .A2(A[12]), .ZN(SUM[12]) );
  CKND2D0BWP7T U19 ( .A1(A[11]), .A2(n12), .ZN(n13) );
  CKXOR2D0BWP7T U20 ( .A1(A[11]), .A2(n12), .Z(SUM[11]) );
  INR2D0BWP7T U21 ( .A1(A[10]), .B1(n14), .ZN(n12) );
  XNR2D1BWP7T U22 ( .A1(n14), .A2(A[10]), .ZN(SUM[10]) );
  CKND2D0BWP7T U23 ( .A1(n1), .A2(A[9]), .ZN(n14) );
  AN3D0BWP7T U24 ( .A1(A[7]), .A2(n3), .A3(A[8]), .Z(n1) );
  AN2D0BWP7T U25 ( .A1(A[6]), .A2(n4), .Z(n3) );
  INR2D0BWP7T U26 ( .A1(A[5]), .B1(n5), .ZN(n4) );
  ND3D0BWP7T U27 ( .A1(A[3]), .A2(n7), .A3(A[4]), .ZN(n5) );
  INR2D0BWP7T U28 ( .A1(A[2]), .B1(n8), .ZN(n7) );
  CKND2D0BWP7T U29 ( .A1(A[1]), .A2(A[0]), .ZN(n8) );
  CKND0BWP7T U30 ( .I(A[0]), .ZN(SUM[0]) );
endmodule


module substractor_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16,
       n17;
  wire [15:0] b_comp;

  carry_look_ahead_16bit U35 ( .a(a), .b(b_comp), .cin(n17), .S(S), .cout(cout)
      );
  substractor_16bit_DW01_inc_0 add_29 ( .A({n1, n2, n3, n4, n5, n6, n7, n8, n9,
       n10, n11, n12, n13, n14, n15, n16}), .SUM(b_comp) );
  TIELBWP7T U1 ( .ZN(n17) );
  CKND0BWP7T U2 ( .I(b[7]), .ZN(n9) );
  CKND0BWP7T U3 ( .I(b[8]), .ZN(n8) );
  CKND0BWP7T U4 ( .I(b[9]), .ZN(n7) );
```

```
  CKND0BWP7T U5 ( .I(b[10]), .ZN(n6) );
  CKND0BWP7T U6 ( .I(b[11]), .ZN(n5) );
  CKND0BWP7T U7 ( .I(b[12]), .ZN(n4) );
  CKND0BWP7T U8 ( .I(b[13]), .ZN(n3) );
  CKND0BWP7T U9 ( .I(b[14]), .ZN(n2) );
  CKND0BWP7T U10 ( .I(b[0]), .ZN(n16) );
  CKND0BWP7T U11 ( .I(b[1]), .ZN(n15) );
  CKND0BWP7T U12 ( .I(b[2]), .ZN(n14) );
  CKND0BWP7T U13 ( .I(b[3]), .ZN(n13) );
  CKND0BWP7T U14 ( .I(b[4]), .ZN(n12) );
  CKND0BWP7T U15 ( .I(b[5]), .ZN(n11) );
  CKND0BWP7T U16 ( .I(b[6]), .ZN(n10) );
  CKND0BWP7T U17 ( .I(b[15]), .ZN(n1) );
endmodule


module substractor_16bit_IO ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire cin_w, cout_w, n1, n2;
  wire [15:0] a_w;
  wire [15:0] b_w;
  wire [15:0] S_w;
  tri [15:0] a;
  tri [15:0] b;
  tri cin;
  tri [15:0] S;
  tri cout;

  substractor_16bit U ( .a(a_w), .b(b_w), .cin(cin_w), .S(S_w), .cout(cout_w)
       );
  PDDW0204SCDG U_a0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[0]), .DS(n1), .PE(n1),
       .C(a_w[0]) );
  PDDW0204SCDG U_a1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[1]), .DS(n1), .PE(n1),
       .C(a_w[1]) );
  PDDW0204SCDG U_a2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[2]), .DS(n1), .PE(n1),
       .C(a_w[2]) );
  PDDW0204SCDG U_a3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[3]), .DS(n1), .PE(n1),
       .C(a_w[3]) );
  PDDW0204SCDG U_a4 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[4]), .DS(n1), .PE(n1),
       .C(a_w[4]) );
  PDDW0204SCDG U_a5 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[5]), .DS(n1), .PE(n1),
       .C(a_w[5]) );
  PDDW0204SCDG U_a6 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[6]), .DS(n1), .PE(n1),
       .C(a_w[6]) );
  PDDW0204SCDG U_a7 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[7]), .DS(n1), .PE(n1),
       .C(a_w[7]) );
  PDDW0204SCDG U_a8 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[8]), .DS(n1), .PE(n1),
       .C(a_w[8]) );
  PDDW0204SCDG U_a9 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[9]), .DS(n1), .PE(n1),
       .C(a_w[9]) );
```

```
    PDDW0204SCDG U_a10 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[10]), .DS(n1), .PE(n1)
        , .C(a_w[10]) );
    PDDW0204SCDG U_a11 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[11]), .DS(n1), .PE(n1)
        , .C(a_w[11]) );
    PDDW0204SCDG U_a12 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[12]), .DS(n1), .PE(n1)
        , .C(a_w[12]) );
    PDDW0204SCDG U_a13 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[13]), .DS(n1), .PE(n1)
        , .C(a_w[13]) );
    PDDW0204SCDG U_a14 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[14]), .DS(n1), .PE(n1)
        , .C(a_w[14]) );
    PDDW0204SCDG U_a15 ( .I(n1), .OEN(n2), .IE(n2), .PAD(a[15]), .DS(n1), .PE(n1)
        , .C(a_w[15]) );
    PDDW0204SCDG U_b0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[0]), .DS(n1), .PE(n1),
         .C(b_w[0]) );
    PDDW0204SCDG U_b1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[1]), .DS(n1), .PE(n1),
         .C(b_w[1]) );
    PDDW0204SCDG U_b2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[2]), .DS(n1), .PE(n1),
         .C(b_w[2]) );
    PDDW0204SCDG U_b3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[3]), .DS(n1), .PE(n1),
         .C(b_w[3]) );
    PDDW0204SCDG U_b4 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[4]), .DS(n1), .PE(n1),
         .C(b_w[4]) );
    PDDW0204SCDG U_b5 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[5]), .DS(n1), .PE(n1),
         .C(b_w[5]) );
    PDDW0204SCDG U_b6 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[6]), .DS(n1), .PE(n1),
         .C(b_w[6]) );
    PDDW0204SCDG U_b7 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[7]), .DS(n1), .PE(n1),
         .C(b_w[7]) );
    PDDW0204SCDG U_b8 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[8]), .DS(n1), .PE(n1),
         .C(b_w[8]) );
    PDDW0204SCDG U_b9 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[9]), .DS(n1), .PE(n1),
         .C(b_w[9]) );
    PDDW0204SCDG U_b10 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[10]), .DS(n1), .PE(n1)
        , .C(b_w[10]) );
    PDDW0204SCDG U_b11 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[11]), .DS(n1), .PE(n1)
        , .C(b_w[11]) );
    PDDW0204SCDG U_b12 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[12]), .DS(n1), .PE(n1)
        , .C(b_w[12]) );
    PDDW0204SCDG U_b13 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[13]), .DS(n1), .PE(n1)
        , .C(b_w[13]) );
    PDDW0204SCDG U_b14 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[14]), .DS(n1), .PE(n1)
        , .C(b_w[14]) );
    PDDW0204SCDG U_b15 ( .I(n1), .OEN(n2), .IE(n2), .PAD(b[15]), .DS(n1), .PE(n1)
        , .C(b_w[15]) );
    PDDW0204SCDG U_cin ( .I(n1), .OEN(n2), .IE(n2), .PAD(cin), .DS(n1), .PE(n1),
         .C(cin_w) );
    PDDW0204SCDG U_S0 ( .I(S_w[0]), .OEN(n1), .IE(n1), .PAD(S[0]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S1 ( .I(S_w[1]), .OEN(n1), .IE(n1), .PAD(S[1]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S2 ( .I(S_w[2]), .OEN(n1), .IE(n1), .PAD(S[2]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S3 ( .I(S_w[3]), .OEN(n1), .IE(n1), .PAD(S[3]), .DS(n1), .PE(
        n1) );
```

```
    PDDW0204SCDG U_S4 ( .I(S_w[4]), .OEN(n1), .IE(n1), .PAD(S[4]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S5 ( .I(S_w[5]), .OEN(n1), .IE(n1), .PAD(S[5]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S6 ( .I(S_w[6]), .OEN(n1), .IE(n1), .PAD(S[6]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S7 ( .I(S_w[7]), .OEN(n1), .IE(n1), .PAD(S[7]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S8 ( .I(S_w[8]), .OEN(n1), .IE(n1), .PAD(S[8]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S9 ( .I(S_w[9]), .OEN(n1), .IE(n1), .PAD(S[9]), .DS(n1), .PE(
        n1) );
    PDDW0204SCDG U_S10 ( .I(S_w[10]), .OEN(n1), .IE(n1), .PAD(S[10]), .DS(n1),
        .PE(n1) );
    PDDW0204SCDG U_S11 ( .I(S_w[11]), .OEN(n1), .IE(n1), .PAD(S[11]), .DS(n1),
        .PE(n1) );
    PDDW0204SCDG U_S12 ( .I(S_w[12]), .OEN(n1), .IE(n1), .PAD(S[12]), .DS(n1),
        .PE(n1) );
    PDDW0204SCDG U_S13 ( .I(S_w[13]), .OEN(n1), .IE(n1), .PAD(S[13]), .DS(n1),
        .PE(n1) );
    PDDW0204SCDG U_S14 ( .I(S_w[14]), .OEN(n1), .IE(n1), .PAD(S[14]), .DS(n1),
        .PE(n1) );
    PDDW0204SCDG U_S15 ( .I(S_w[15]), .OEN(n1), .IE(n1), .PAD(S[15]), .DS(n1),
        .PE(n1) );
    PDDW0204SCDG U_cout ( .I(cout_w), .OEN(n1), .IE(n1), .PAD(cout), .DS(n1),
        .PE(n1) );
    TIELBWP7T U6 ( .ZN(n1) );
    TIEHBWP7T U7 ( .Z(n2) );
  endmodule
```

Cuadro 101: Archivo SUB16b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O

13.6.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O

El último paso que se realizó fue la simulación del circuito completamente sintetizado junto con los pines de I/O. Esto se llevó a cabo nuevamente para comprobar que el circuito completamente sintetizado se comporte correctamente.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 102 en una terminal abierta en `Outputs/SUB16b/Outputs_finales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v SUB16b_sintesis_final.v SUB16b_tb.
    v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 102: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en la Figura 144, el comportamiento es idéntico al que se obtuvo anteriormente, lo que confirma que el *netlist* final obtenido del circuito sintetizado con pines de I/O es correcto.

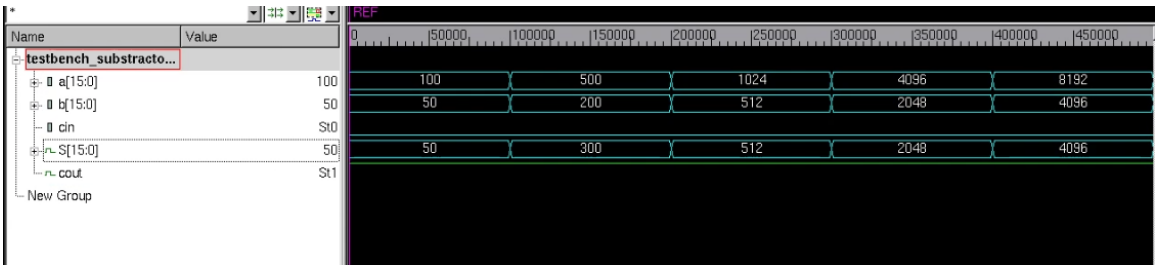


Figura 144: Simulación del *netlist* final

13.7. Proceso para la síntesis lógica de un multiplicador de 32 bits usando un *carry look-ahead adder* de 16 bits

13.7.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench*

El primer paso fue crear una carpeta en el directorio de *inputs*, nombrada según el circuito principal a sintetizar; en este caso, `MULT32b` (tanto la carpeta como el nombre del archivo Verilog que contiene el circuito principal deben tener el mismo nombre). Luego, se diseñaron los archivos necesarios para la simulación y síntesis del circuito.

Siguiendo la estructura explicada en el capítulo y sección 9.2, el archivo que contiene el módulo principal del multiplicador de 32 bits se nombró `MULT32b.v`. El archivo de *testbench*, utilizado para simular y verificar el funcionamiento del circuito, se denominó `MULT32b_tb.v`. Por último, el archivo que define el módulo de entradas y salidas se nombró `IO_MULT32b.v`.

En este caso, se utilizaron otros circuitos para construir el circuito principal. Se emplearon los sumadores de 4 y 16 bits diseñados anteriormente como base para crear el multiplicador de 32 bits. Por lo tanto, los archivos Verilog de los sumadores de 4 y 16 bits también se encuentran en la carpeta `Inputs/MULT32b/`, ya que, para los procesos de simulación y síntesis, es importante hacer referencia a todos los archivos utilizados.

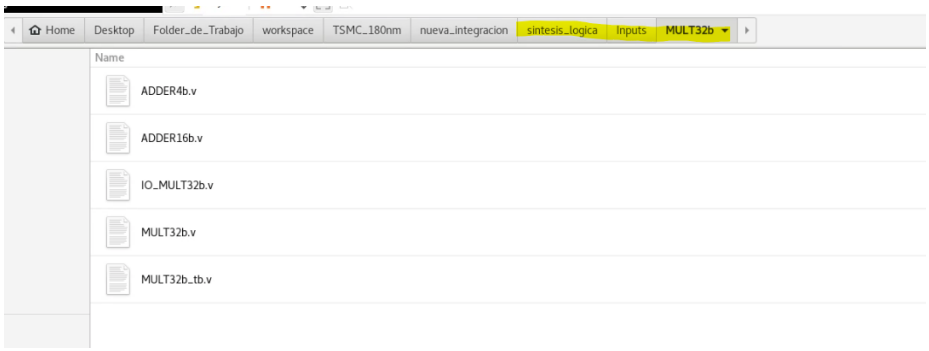


Figura 145: Archivos necesarios en la carpeta del circuito en *inputs*

En el Cuadro 103 se muestra el diseño inicial del circuito, el cual hace referencia al módulo del sumador de 16 bits para su implementación. Por otro lado, en el Cuadro 104 se presenta el diseño final del circuito utilizado. Esto se debió a que, al momento de sintetizar el diseño inicial (Cuadro 103), se obtuvieron errores, los cuales se explicarán posteriormente.

Por otro lado, en el Cuadro 105 se presenta el diseño del módulo de pines de entrada y salida. Además, en el Cuadro 106 se muestra el *testbench* diseñado para la simulación tanto del diseño final del circuito original como del circuito sintetizado.

Dado que el diseño del circuito incluye múltiples entradas y salidas, en el módulo de pines es necesario hacer referencia a cada una de ellas de forma individual. Sin embargo, el proceso de creación de cada pin es el mismo para todas las entradas y salidas (ver Cuadro 105).

```
module multiplier_16bit(Mand, Mer, clk, Sout);
input [15:0] Mand, Mer;
input clk;
output reg [31:0] Sout;
//reg [31:0] S;
wire [15:0] S;
reg [31:0] Sreg;
wire d1;

integer i;
initial
i = 0;

carry_look_ahead_16bit U25(Mand, Sreg[31:16], 1'b0, S, d1);

always@(posedge clk) begin

    //Sreg = Sreg >> 1;
    if(i == 0) begin
        Sreg = {16'b0, Mer};
    end

    if(i>0 & i<16) begin

        if (Sreg[0] == 1'b1) begin
        Sreg[31:16] = S;
```

```
            Sreg = Sreg >> 1;
            Sreg[31] = d1;
            end

            else if (Sreg[0] == 1'b0)begin
            Sreg = Sreg >> 1;
            end
        end

        if(i == 16) begin
            Sreg = Sreg >> 1;
            Sout = Sreg;

        end


        if(i < 17) begin
        i = i+1;
        end

        if (i >= 17) begin
        i=0;
        end
    end
    endmodule
```

Cuadro 103: Archivo MULT32b.v: diseño inicial de un multiplicador de 32 bits en Verilog

```
    module multiplier_16bit(Mand, Mer, clk, reset, Sout);
    input [15:0] Mand, Mer;
    input clk, reset;
    output reg [31:0] Sout;
    wire [15:0] S;
    reg [31:0] Sreg;
    wire d1;

    integer i; // contador

    // Instancia del sumador carry-lookahead
    carry_look_ahead_16bit U25(Mand, Sreg[31:16], 1'b0, S, d1);

    always@(posedge clk or posedge reset) begin
        if (reset) begin
            i = 0; // Inicializa el contador a 0 en un reset
            Sreg = 32'b0; // Inicializa Sreg en el reset
            Sout = 32'b0; // Inicializa el resultado en el reset
        end
        else begin
            if(i == 0) begin
            Sreg = {16'b0, Mer};
        end

        if(i>0 & i<16) begin
```

```
            if (Sreg[0] == 1'b1) begin
            Sreg[31:16] = S;
            Sreg = Sreg >> 1;
            Sreg[31] = d1;
            end

            else if (Sreg[0] == 1'b0)begin
            Sreg = Sreg >> 1;
            end
        end

        if(i == 16) begin
            Sreg = Sreg >> 1;
            Sout = Sreg;

        end


        if(i < 17) begin
        i = i+1;
        end

        if (i >= 17) begin
        i=0;
        end
        end
    end
    endmodule
```

Cuadro 104: Archivo MULT32b.v: diseño final de un multiplicador de 32 bits en Verilog

```
    module multiplier_16bit_IO (
        input [15:0] Mand,
        input [15:0] Mer,
        input clk,
        input reset,
        output [31:0] Sout
    );
        wire [15:0] Mand_w, Mer_w;
        wire clk_w,reset_w;
        wire [31:0] Sout_w;

        // Instancia del mpdulo multiplier_16bit
        multiplier_16bit mult_inst (
            .Mand(Mand_w),
            .Mer(Mer_w),
            .clk(clk_w),
        .reset(reset_w),
            .Sout(Sout_w)
        );

        // Interconexion de pines de entrada (Mand, Mer, clk)
        // Entradas Mand
        PDDW0204SCDG U_Mand0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[0]), .C(
```

```
        Mand_w[0]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[1]), .C(
        Mand_w[1]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[2]), .C(
        Mand_w[2]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[3]), .C(
        Mand_w[3]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[4]), .C(
        Mand_w[4]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[5]), .C(
        Mand_w[5]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[6]), .C(
        Mand_w[6]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[7]), .C(
        Mand_w[7]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[8]), .C(
        Mand_w[8]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[9]), .C(
        Mand_w[9]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[10]), .C(
        Mand_w[10]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[11]), .C(
        Mand_w[11]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[12]), .C(
        Mand_w[12]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[13]), .C(
        Mand_w[13]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[14]), .C(
        Mand_w[14]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mand15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mand[15]), .C(
        Mand_w[15]), .PE(1'b0), .IE(1'b1));

    // Entradas Mer
    PDDW0204SCDG U_Mer0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[0]), .C(
        Mer_w[0]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[1]), .C(
        Mer_w[1]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[2]), .C(
        Mer_w[2]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[3]), .C(
        Mer_w[3]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[4]), .C(
        Mer_w[4]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[5]), .C(
        Mer_w[5]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[6]), .C(
        Mer_w[6]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[7]), .C(
        Mer_w[7]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[8]), .C(
        Mer_w[8]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[9]), .C(
        Mer_w[9]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[10]), .C(
```

```
        Mer_w[10]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[11]), .C(
        Mer_w[11]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[12]), .C(
        Mer_w[12]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[13]), .C(
        Mer_w[13]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[14]), .C(
        Mer_w[14]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Mer15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Mer[15]), .C(
        Mer_w[15]), .PE(1'b0), .IE(1'b1));

    // Entrada del reloj clk y reset
    PDDW0204SCDG U_clk (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(clk), .C(clk_w),
        .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_reset (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(reset), .C(
        reset_w), .PE(1'b0), .IE(1'b1));

    // Conexion de la salida Sout
    PDDW0204SCDG U_Sout0 (.I(Sout_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[0]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout1 (.I(Sout_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[1]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout2 (.I(Sout_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[2]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout3 (.I(Sout_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[3]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout4 (.I(Sout_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[4]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout5 (.I(Sout_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[5]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout6 (.I(Sout_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[6]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout7 (.I(Sout_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[7]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout8 (.I(Sout_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[8]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout9 (.I(Sout_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(Sout[9]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout10 (.I(Sout_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [10]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout11 (.I(Sout_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [11]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout12 (.I(Sout_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [12]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout13 (.I(Sout_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [13]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout14 (.I(Sout_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [14]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout15 (.I(Sout_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [15]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout16 (.I(Sout_w[16]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [16]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout17 (.I(Sout_w[17]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
```

```
        [17]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout18 (.I(Sout_w[18]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [18]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout19 (.I(Sout_w[19]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [19]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout20 (.I(Sout_w[20]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [20]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout21 (.I(Sout_w[21]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [21]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout22 (.I(Sout_w[22]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [22]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout23 (.I(Sout_w[23]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [23]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout24 (.I(Sout_w[24]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [24]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout25 (.I(Sout_w[25]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [25]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout26 (.I(Sout_w[26]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [26]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout27 (.I(Sout_w[27]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [27]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout28 (.I(Sout_w[28]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [28]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout29 (.I(Sout_w[29]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [29]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout30 (.I(Sout_w[30]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [30]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sout31 (.I(Sout_w[31]), .DS(1'b0), .OEN(1'b0), .PAD(Sout
        [31]), .C(1'bx), .PE(1'b0), .IE(1'b0));

//Pines de VDD y VSS
PVDD1CDG PVDD();
PVSS1CDG PVSS();

endmodule
```

Cuadro 105: Archivo IO_MULT32b.v: diseño de módulo de entradas y salidas para un multiplicador de 32 bits en Verilog

```
module MULT32b_tb;
    reg [15:0] Mand, Mer;
    reg clk, reset;
    wire [31:0] Si;


    multiplier_16bit multiply (
    .Mand(Mand),
    .Mer(Mer),
    .clk(clk),
    .reset(reset),
    .Sout(Si)
    );

    // Generar el reloj
```

```
        initial begin
        clk = 1'b0;
        forever #1 clk = ~clk; // Generar un reloj de 1 unidad de tiempo
        end


        initial begin
        // Configurar archivo de volcado para FSDB y VCD
        $fsdbDumpfile("MULT32b.fsdb");
        $fsdbDumpvars(0, MULT32b_tb);
        $dumpfile("MULT32b.vcd");
        $dumpvars(0, MULT32b_tb);


        reset = 1; // Activar el reset
        Mand = 16'd0;
        Mer = 16'd0;

        // Esperar 5 unidades de tiempo para el reset
        #5 reset = 0; // Desactivar el reset


        #100 Mand = 16'd6; Mer = 16'd3;
        #100 Mand = 16'd2; Mer = 16'd4;
        #100 Mand = 16'd5; Mer = 16'd7;
        #100 Mand = 16'd64; Mer = 16'd100;
        #100 Mand = 16'd512; Mer = 16'd7;
        #100 Mand = 16'd0; Mer = 16'd1;
        #100 Mand = 16'd900; Mer = 16'd85;
        #100 Mand = 16'd60000; Mer = 16'd9;

        // Finalizar simulacion despues de un tiempo adicional
        #800 $finish;
        end
endmodule
```

Cuadro 106: Archivo MULT32b_tb.v: diseño del *testbench* para simulación de un multiplicador de 32 bits en Verilog

13.7.2. Simulación del diseño original del circuito

Como siguiente paso, se simuló el diseño original y final del circuito con el objetivo de verificar que no existieran errores de compilación y que el diseño del circuito fuera correcto. Como se explicó anteriormente, si no se realiza este paso, existe el riesgo de encontrar errores debido a posibles fallos en el diseño del circuito, lo que impediría proceder a la ejecución de la síntesis lógica.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 107 en una terminal abierta en `Inputs/MULT32b/`.

```
    vcs -v MULT32b.v ADDER16b.v ADDER4b.v MULT32b_tb.v -o simulacion -full64 -
        debug_access+all
```

```
./simulacion

dve -full64
```

Cuadro 107: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Como se observa en el Cuadro 107, es necesario colocar el nombre de todos los archivos Verilog que se están utilizando. Si no se realiza este paso, se producirá un error, ya que la aplicación no encontrará el módulo al que el diseño original hace referencia.

Siguiendo los pasos descritos en el capítulo y sección 12.1 para la visualización del archivo `.vcd` generado al ejecutar el archivo de simulación, se obtuvo la Figura 146.

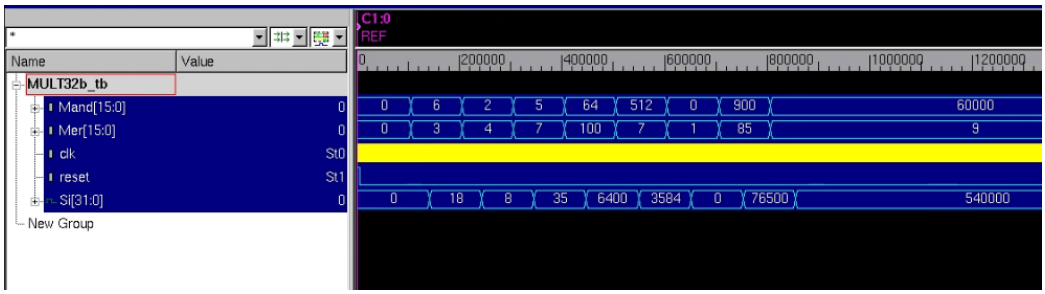


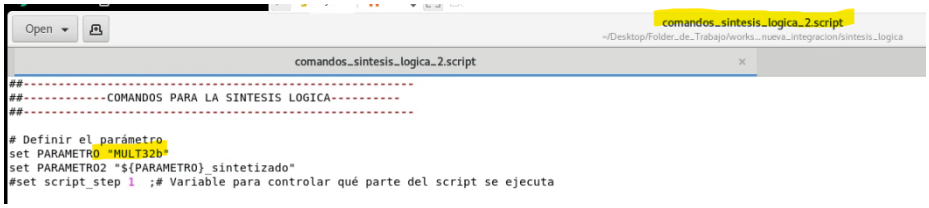
Figura 146: Simulación de sumador de 32 bits en su diseño original

La Figura 146 es la misma tanto para la simulación del circuito inicial como para la del circuito final. Como se observa, el comportamiento de ambos circuitos es el esperado, ya que se realiza la multiplicación entre las entradas `Mand` y `Mer`, y el resultado se obtiene en la salida (S).

Por lo tanto, se procedió a ejecutar el *script* para llevar a cabo la síntesis lógica del circuito.

13.7.3. Síntesis lógica del diseño original del circuito

Antes de proceder a ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, fue necesario abrir tanto este *script* como el archivo `comandos_sintesis_logica_2.script` para definir en la variable `PARAMETRO` el nombre de la carpeta creada en *inputs*, la cual contiene el circuito a sintetizar. En este caso, se definió como `MULT32b`.



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 147: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

De igual manera, fue necesario modificar una parte del *script* `comandos_sintesis_logica_2.script`. Dado que el diseño del multiplicador de 32 bits utiliza los módulos que se encuentran en otros archivos Verilog, es necesario hacer referencia a estos archivos para sintetizar correctamente el diseño.

La modificación debe realizarse en el primer paso de síntesis, en la parte donde se leen los archivos que describen el circuito. En la línea que contiene el comando `read_file`, se deben agregar los archivos adicionales necesarios. En este caso, la modificación quedaría de la siguiente forma:

```
if { $script_step == 1 } {
# Parte 1 - Sintesis Cell original

# Usar el parametro en lugar del nombre del modulo
read_file -format verilog "Inputs/$PARAMETRO/$PARAMETRO.v Inputs/$PARAMETRO/
    ADDER4b.v Inputs/$PARAMETRO/ADDER16b.v"
```

Cuadro 108: Modificación del *script* comandos_sintesis_logica_2.script

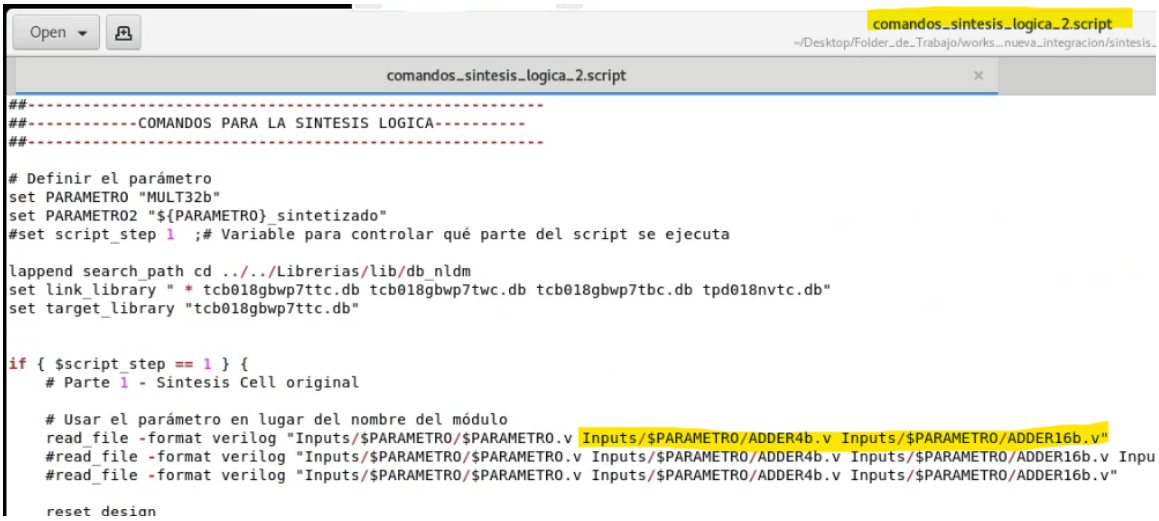


Figura 148: Modificación del *script* comandos_sintesis_logica_2.script

Como se observa en el Cuadro 108 y en la Figura 148, la única modificación realizada fue agregar la lectura de los archivos `ADDER4b.v` y `ADDER16b.v` en el comando `read_file`, ya que los módulos contenidos en esos archivos son utilizados en `MULT32b.v` y, por lo tanto, también deben ser sintetizados. Si no se realiza esta modificación, se producirá un error en el proceso de síntesis.

Luego, se ejecutó el *script* utilizando el comando que se muestra en el Cuadro 109, en una terminal abierta en el directorio `sintesis_logica`.

```
./ejecutar_sintesis_logica_2.bash
```

Cuadro 109: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash

Al ejecutar este comando, se creó una carpeta en *outputs* con el nombre `MULT32b`. Además, se generaron dos subdirectorios dentro de esta, llamados `Outputs_iniciales` y `Outputs_finales`.

Sin embargo, al utilizar el diseño inicial propuesto para el multiplicador surgieron algunas advertencias. Al finalizar el proceso, la terminal mostró una advertencia que indicaba que el generador de síntesis había tenido que añadir una conexión adicional (*net*) en el módulo `multiplier_16bit`. Esta conexión fue prefijada con `SYNOPSYS_UNCONNECTED_`, lo cual indica que la herramienta de síntesis agregó una conexión para asegurar la consistencia del diseño en términos de conectividad (ver Figura 149).

Ante esto, fue necesario revisar si existían otros mensajes de advertencia desde el inicio, ya que si se presenta un problema en la primera síntesis, la segunda síntesis con pines también fallará. En efecto, al comienzo, tras realizar la lectura del diseño original del circuito, se obtuvo otra advertencia (ver Figura 150). Esta advertencia indica que dentro del archivo Verilog `MULT32b.v`, específicamente en la línea 13, se utilizó un bloque *initial*. Para la síntesis lógica, los bloques *initial* son ignorados, ya que están pensados para simulación y no se sintetizan en *hardware* real. Esto podría ser un problema, ya que para la implementación de

circuitos con reloj es importante inicializar variables o registros.

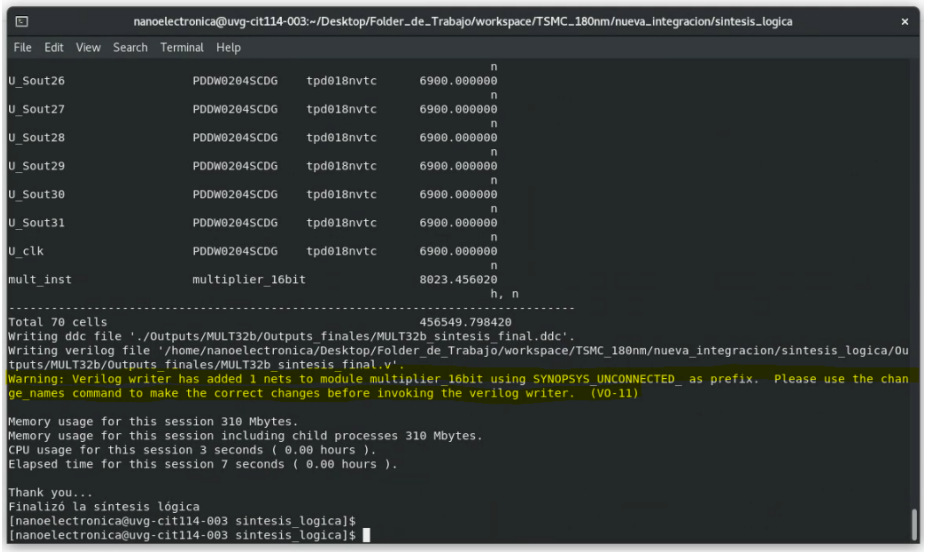


Figura 149: Advertencia #1 al terminar la síntesis

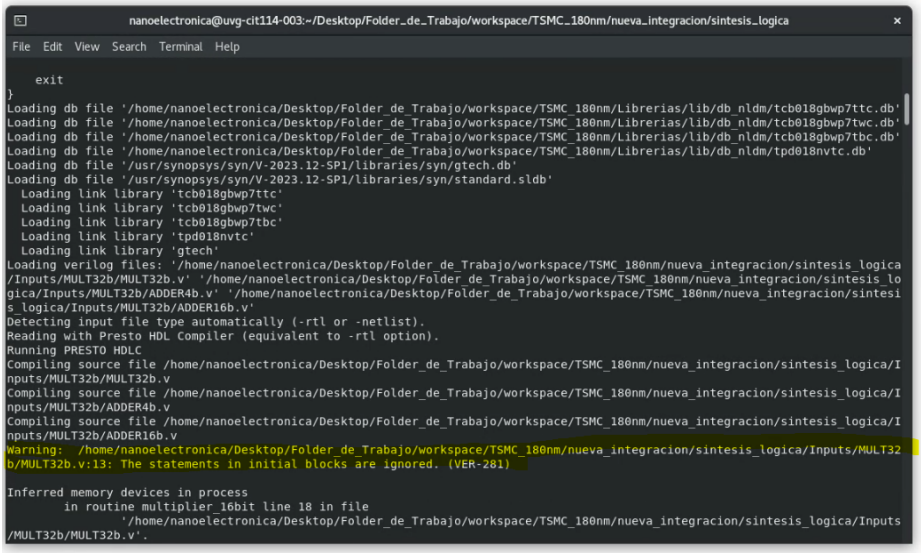


Figura 150: Advertencia #2 en la primera síntesis

A pesar de esto, en la carpeta `Outputs/MULT32b/Outputs_iniciales` se creó un archivo llamado `MULT32b_sintetizado.v`, el cual corresponde al archivo resultante de la primera etapa de la síntesis lógica (ver Figura 151).

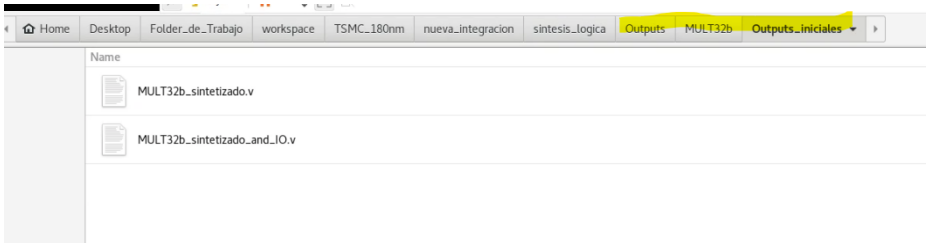


Figura 151: Archivos generados en la carpeta del circuito en Outputs_iniciales

Sin embargo, al simular el circuito sintetizado obtenido a partir del diseño inicial, se obtuvo la Figura 152, la cual demuestra que dicho circuito no se comporta como se esperaba.

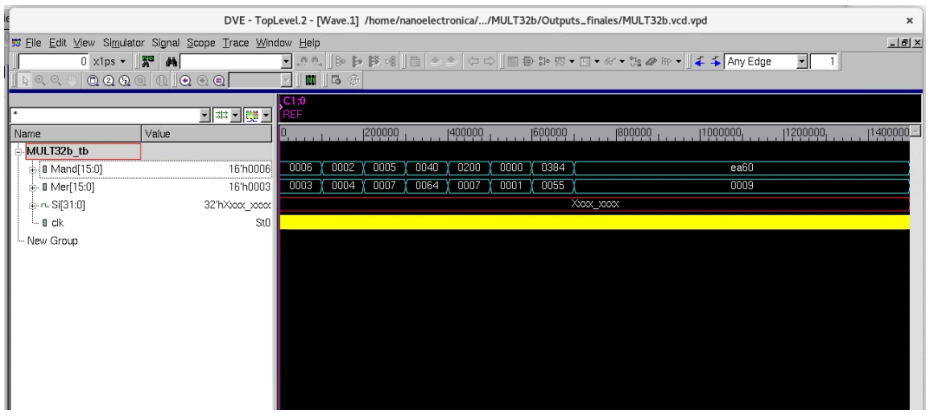


Figura 152: Simulación del circuito sintetizado con errores obtenido a partir del diseño inicial del circuito

Para solucionar este problema, se añadió una entrada externa al circuito llamada *reset*, con el objetivo de implementar la funcionalidad de un bloque *initial*. Esta entrada, al ser inicializada en 1 al inicio del *testbench*, permite establecer los valores iniciales de las variables o registros necesarios. En la implementación física, esto puede lograrse, por ejemplo, mediante un pulsador que inicialice dichas variables.

Al repetir los pasos anteriores utilizando el diseño final del circuito, este se sintetizó sin errores. Aunque todavía apareció la advertencia mostrada en la Figura 153, ya no se presentó la advertencia de la Figura 154, lo que indicó que sería posible inicializar variables y registros. Se generaron reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O. Nuevamente, en la carpeta `Outputs/MULT32b/Outputs_iniciales` se creó un archivo llamado `MULT32b_sintetizado.v`, correspondiente al resultado de la primera etapa de la síntesis lógica.

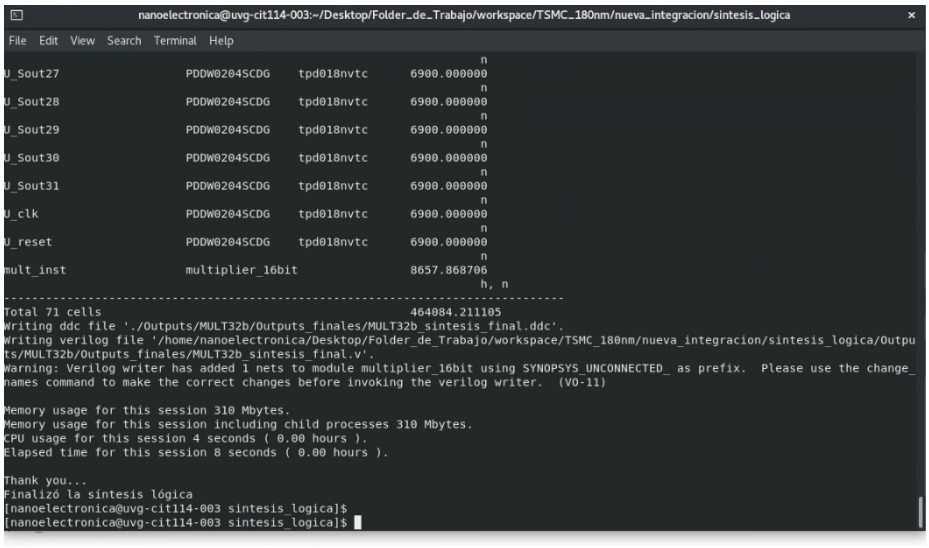


Figura 153: Advertencia al terminar la síntesis

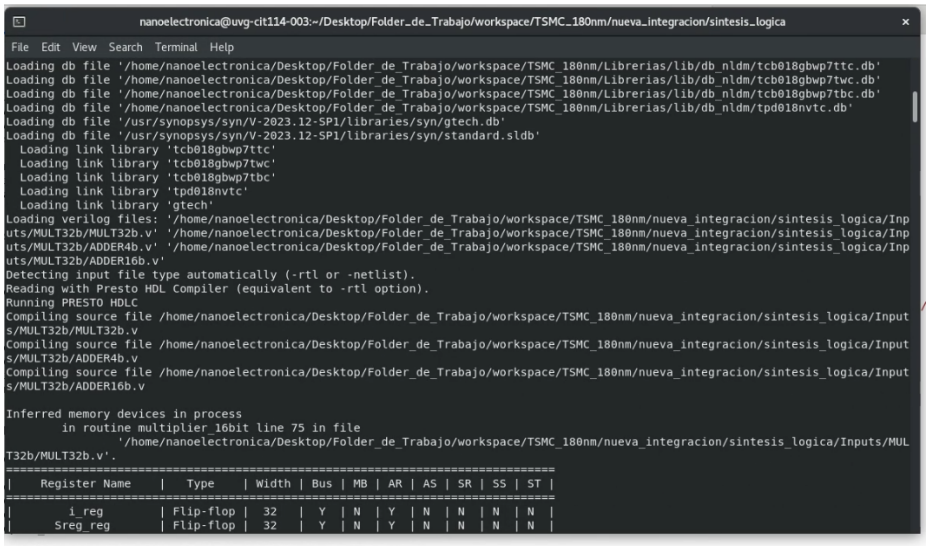


Figura 154: Advertencia en la primera síntesis solucionada

Es importante recalcar que no todos los *warnings* indican problemas en el circuito, ya que algunos pueden ser resueltos por la herramienta Design Compiler, como la creación de conexiones (*nets*), cables (*wires*), entre otros. Sin embargo, existen otros problemas que no se solucionan automáticamente. Por esta razón, siempre es importante revisar los *warnings* o errores en la terminal para asegurar que la síntesis se ejecute correctamente.

El archivo `MULT32b_sintetizado.v`, que corresponde al circuito sintetizado obtenido a partir del diseño final, se muestra en el Cuadro 110.

```
////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
```

```
// Date  : Sat Nov 2 22:03:45 2024
////////////////////////////////////////////////////////////


module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  MOAI22D0BWP7T U2 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  AOI22D0BWP7T U3 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  XNR2D1BWP7T U4 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  CKXOR2D1BWP7T U5 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  AOI22D0BWP7T U6 ( .A1(b[0]), .A2(a[0]), .B1(n7), .B2(cin), .ZN(n5) );
  CKXOR2D1BWP7T U7 ( .A1(cin), .A2(n7), .Z(S[0]) );
  XNR2D1BWP7T U8 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  CKXOR2D1BWP7T U9 ( .A1(n4), .A2(n3), .Z(S[2]) );
  CKXOR2D1BWP7T U10 ( .A1(n1), .A2(n2), .Z(S[3]) );
  CKXOR2D1BWP7T U11 ( .A1(n5), .A2(n6), .Z(S[1]) );
  CKXOR2D1BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n8, n9, n10, n11, n12, n13, n14;

  MOAI22D0BWP7T U1 ( .A1(n14), .A2(n13), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  MOAI22D0BWP7T U2 ( .A1(n10), .A2(n9), .B1(b[1]), .B2(a[1]), .ZN(n12) );
  AOI22D0BWP7T U3 ( .A1(b[0]), .A2(a[0]), .B1(n8), .B2(cin), .ZN(n10) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n12), .B2(n11), .ZN(n14) );
  CKXOR2D1BWP7T U5 ( .A1(cin), .A2(n8), .Z(S[0]) );
  CKXOR2D1BWP7T U6 ( .A1(n10), .A2(n9), .Z(S[1]) );
  CKXOR2D1BWP7T U7 ( .A1(n11), .A2(n12), .Z(S[2]) );
  CKXOR2D1BWP7T U8 ( .A1(n14), .A2(n13), .Z(S[3]) );
  CKXOR2D1BWP7T U9 ( .A1(b[0]), .A2(a[0]), .Z(n8) );
  XNR2D1BWP7T U10 ( .A1(b[1]), .A2(a[1]), .ZN(n9) );
  CKXOR2D1BWP7T U11 ( .A1(b[2]), .A2(a[2]), .Z(n11) );
  XNR2D1BWP7T U12 ( .A1(b[3]), .A2(a[3]), .ZN(n13) );
endmodule


module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
```

```
  input cin;
  output cout;
  wire n8, n9, n10, n11, n12, n13, n14;

  MOAI22D0BWP7T U1 ( .A1(n14), .A2(n13), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  MOAI22D0BWP7T U2 ( .A1(n10), .A2(n9), .B1(b[1]), .B2(a[1]), .ZN(n12) );
  AOI22D0BWP7T U3 ( .A1(b[0]), .A2(a[0]), .B1(n8), .B2(cin), .ZN(n10) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n12), .B2(n11), .ZN(n14) );
  CKXOR2D1BWP7T U5 ( .A1(cin), .A2(n8), .Z(S[0]) );
  CKXOR2D1BWP7T U6 ( .A1(n10), .A2(n9), .Z(S[1]) );
  CKXOR2D1BWP7T U7 ( .A1(n11), .A2(n12), .Z(S[2]) );
  CKXOR2D1BWP7T U8 ( .A1(n14), .A2(n13), .Z(S[3]) );
  CKXOR2D1BWP7T U9 ( .A1(b[0]), .A2(a[0]), .Z(n8) );
  XNR2D1BWP7T U10 ( .A1(b[1]), .A2(a[1]), .ZN(n9) );
  CKXOR2D1BWP7T U11 ( .A1(b[2]), .A2(a[2]), .Z(n11) );
  XNR2D1BWP7T U12 ( .A1(b[3]), .A2(a[3]), .ZN(n13) );
endmodule


module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n8, n9, n10, n11, n12, n13, n14;

  CKXOR2D1BWP7T U1 ( .A1(n14), .A2(n13), .Z(S[3]) );
  MOAI22D0BWP7T U2 ( .A1(n10), .A2(n9), .B1(b[1]), .B2(a[1]), .ZN(n12) );
  AOI22D0BWP7T U3 ( .A1(b[0]), .A2(a[0]), .B1(n8), .B2(cin), .ZN(n10) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n12), .B2(n11), .ZN(n14) );
  CKXOR2D1BWP7T U5 ( .A1(n11), .A2(n12), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n14), .A2(n13), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D1BWP7T U7 ( .A1(n10), .A2(n9), .Z(S[1]) );
  CKXOR2D1BWP7T U8 ( .A1(cin), .A2(n8), .Z(S[0]) );
  CKXOR2D1BWP7T U9 ( .A1(b[0]), .A2(a[0]), .Z(n8) );
  XNR2D1BWP7T U10 ( .A1(b[1]), .A2(a[1]), .ZN(n9) );
  CKXOR2D1BWP7T U11 ( .A1(b[2]), .A2(a[2]), .Z(n11) );
  XNR2D1BWP7T U12 ( .A1(b[3]), .A2(a[3]), .ZN(n13) );
endmodule


module carry_look_ahead_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire \*Logic0* , w1, w2, w3;

  carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(\*Logic0* ), .S(
       S[3:0]), .cout(w1) );
  carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
```

```
  carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(cout) );
  TIELBWP7T U6 ( .ZN(\*Logic0* ) );
endmodule


module multiplier_16bit_DW01_inc_0 ( A, SUM );
  input [31:0] A;
  output [31:0] SUM;

  wire [31:2] carry;

  HA1D0BWP7T U1_1_30 ( .A(A[30]), .B(carry[30]), .CO(carry[31]), .S(SUM[30])
       );
  HA1D0BWP7T U1_1_22 ( .A(A[22]), .B(carry[22]), .CO(carry[23]), .S(SUM[22])
       );
  HA1D0BWP7T U1_1_17 ( .A(A[17]), .B(carry[17]), .CO(carry[18]), .S(SUM[17])
       );
  HA1D0BWP7T U1_1_11 ( .A(A[11]), .B(carry[11]), .CO(carry[12]), .S(SUM[11])
       );
  HA1D0BWP7T U1_1_21 ( .A(A[21]), .B(carry[21]), .CO(carry[22]), .S(SUM[21])
       );
  HA1D0BWP7T U1_1_16 ( .A(A[16]), .B(carry[16]), .CO(carry[17]), .S(SUM[16])
       );
  HA1D0BWP7T U1_1_10 ( .A(A[10]), .B(carry[10]), .CO(carry[11]), .S(SUM[10])
       );
  HA1D0BWP7T U1_1_4 ( .A(A[4]), .B(carry[4]), .CO(carry[5]), .S(SUM[4]) );
  HA1D0BWP7T U1_1_20 ( .A(A[20]), .B(carry[20]), .CO(carry[21]), .S(SUM[20])
       );
  HA1D0BWP7T U1_1_19 ( .A(A[19]), .B(carry[19]), .CO(carry[20]), .S(SUM[19])
       );
  HA1D0BWP7T U1_1_18 ( .A(A[18]), .B(carry[18]), .CO(carry[19]), .S(SUM[18])
       );
  HA1D0BWP7T U1_1_1 ( .A(A[1]), .B(A[0]), .CO(carry[2]), .S(SUM[1]) );
  HA1D0BWP7T U1_1_29 ( .A(A[29]), .B(carry[29]), .CO(carry[30]), .S(SUM[29])
       );
  HA1D0BWP7T U1_1_28 ( .A(A[28]), .B(carry[28]), .CO(carry[29]), .S(SUM[28])
       );
  HA1D0BWP7T U1_1_27 ( .A(A[27]), .B(carry[27]), .CO(carry[28]), .S(SUM[27])
       );
  HA1D0BWP7T U1_1_26 ( .A(A[26]), .B(carry[26]), .CO(carry[27]), .S(SUM[26])
       );
  HA1D0BWP7T U1_1_25 ( .A(A[25]), .B(carry[25]), .CO(carry[26]), .S(SUM[25])
       );
  HA1D0BWP7T U1_1_24 ( .A(A[24]), .B(carry[24]), .CO(carry[25]), .S(SUM[24])
       );
  HA1D0BWP7T U1_1_23 ( .A(A[23]), .B(carry[23]), .CO(carry[24]), .S(SUM[23])
       );
  HA1D0BWP7T U1_1_15 ( .A(A[15]), .B(carry[15]), .CO(carry[16]), .S(SUM[15])
       );
  HA1D0BWP7T U1_1_14 ( .A(A[14]), .B(carry[14]), .CO(carry[15]), .S(SUM[14])
       );
```

```
  HA1D0BWP7T U1_1_13 ( .A(A[13]), .B(carry[13]), .CO(carry[14]), .S(SUM[13])
        );
  HA1D0BWP7T U1_1_12 ( .A(A[12]), .B(carry[12]), .CO(carry[13]), .S(SUM[12])
        );
  HA1D0BWP7T U1_1_9 ( .A(A[9]), .B(carry[9]), .CO(carry[10]), .S(SUM[9]) );
  HA1D0BWP7T U1_1_8 ( .A(A[8]), .B(carry[8]), .CO(carry[9]), .S(SUM[8]) );
  HA1D0BWP7T U1_1_7 ( .A(A[7]), .B(carry[7]), .CO(carry[8]), .S(SUM[7]) );
  HA1D0BWP7T U1_1_6 ( .A(A[6]), .B(carry[6]), .CO(carry[7]), .S(SUM[6]) );
  HA1D0BWP7T U1_1_5 ( .A(A[5]), .B(carry[5]), .CO(carry[6]), .S(SUM[5]) );
  HA1D0BWP7T U1_1_3 ( .A(A[3]), .B(carry[3]), .CO(carry[4]), .S(SUM[3]) );
  HA1D0BWP7T U1_1_2 ( .A(A[2]), .B(carry[2]), .CO(carry[3]), .S(SUM[2]) );
  CKXOR2D0BWP7T U1 ( .A1(carry[31]), .A2(A[31]), .Z(SUM[31]) );
endmodule


module multiplier_16bit ( Mand, Mer, clk, reset, Sout );
  input [15:0] Mand;
  input [15:0] Mer;
  output [31:0] Sout;
  input clk, reset;
  wire d1, N190, N191, N192, N193, N194, N195, N196, N197, N198, N199, N200,
        N201, N202, N203, N204, N205, N206, N207, N208, N209, N210, N211,
        N212, N213, N214, N215, N216, N217, N218, N219, N220, N221, N222,
        N223, N224, N225, N226, N227, N228, N229, N230, N231, N232, N233,
        N234, N235, N236, N237, N238, N239, N240, N241, N242, N243, N244,
        N245, N246, N247, N248, N249, N250, N251, N252, n53, n87, n88, n89,
        n90, n91, n92, n93, n94, n102, n103, n104, n105, n108, n109, n110,
        n111, n112, n114, n115, n235, n236, n237, n238, n239, n240, n241,
        n242, n243, n244, n245, n246, n247, n248, n249, n250, n251, n252,
        n253, n254, n255, n256, n257, n258, n259, n260, n261, n262, n263,
        n264, n265, n266, n267, n268, n269, n270, n271, n272, n273, n274,
        n275, n276, n277, n278, n279, n280, n281, n282, n283, n284, n285,
        n286, n287, n288, n289, n290, n291, n292, n293, n294, n295, n296,
        n297, n298, n299, n300, n301, n302, n303, n304, n305, n306, n307,
        n308, n309, n310, n311, n312, n313, n314, n315, n316, n317, n318,
        n319, n320, n321, n322, n323, n324, n325, n326, n327, n328, n329,
        n330, n331, n332, n333, n334, n335, n336, n337, n338, n339, n340,
        n341, n342, n343, n344, n345, n346, n347, n348, n349, n350, n351,
        n352, n353, n354, n355, n356, n357, n358, n359, n360, n361, n362,
        n363, n364, n365, n366, n367, n368, n369, n370, n371, n372, n373,
        n374, n375, n376, n377, n378, n379, n380, n381, n382, n383, n384,
        n385, n386, n387, n388, n389, n390, n391, n392, n393, n394, n395,
        n396, n397, n398, n399, n400, n401, n402, n403, n404, n405, n406,
        n407, n408, n409, n410, n411, n412, n413, n414, n415, n416, n417,
        n418, n419, n420, n421, n422, n423, n424, n425, n426, n427, n428,
        n429, n430, n431, n432, n433, n434, n435, n436, n437, n438, n439,
        n440, n441, n442, n443, n444, n445, n446, n447, n448, n449, n450,
        n451, n452, n453, n454, n455, n456, n457, n458, n459, n460;
  wire [31:0] Sreg;
  wire [15:0] S;
  wire [31:0] i;
  wire SYNOPSYS_UNCONNECTED__0;

  DFCND0BWP7T \i_reg[31] ( .D(n328), .CP(clk), .CDN(n460), .Q(i[31]), .QN(n53)
```

```
    );
  DFCNDOBWP7T \i_reg[2] ( .D(n326), .CP(clk), .CDN(n332), .Q(i[2]), .QN(n115)
      );
  DFCNDOBWP7T \i_reg[3] ( .D(n325), .CP(clk), .CDN(n460), .Q(i[3]), .QN(n114)
      );
  DFCNDOBWP7T \i_reg[5] ( .D(n323), .CP(clk), .CDN(n332), .Q(i[5]), .QN(n112)
      );
  DFCNDOBWP7T \i_reg[7] ( .D(n321), .CP(clk), .CDN(n332), .Q(i[7]), .QN(n110)
      );
  DFCNDOBWP7T \i_reg[8] ( .D(n320), .CP(clk), .CDN(n460), .Q(i[8]), .QN(n109)
      );
  DFCNDOBWP7T \i_reg[9] ( .D(n319), .CP(clk), .CDN(n332), .Q(i[9]), .QN(n108)
      );
  DFCNDOBWP7T \i_reg[13] ( .D(n315), .CP(clk), .CDN(n332), .Q(i[13]), .QN(
      n104) );
  DFCNDOBWP7T \i_reg[15] ( .D(n313), .CP(clk), .CDN(n460), .Q(i[15]), .QN(
      n102) );
  DFCNDOBWP7T \i_reg[23] ( .D(n305), .CP(clk), .CDN(n460), .Q(i[23]), .QN(n94)
    );
  DFCNDOBWP7T \i_reg[24] ( .D(n304), .CP(clk), .CDN(n460), .Q(i[24]), .QN(n93)
    );
  DFCNDOBWP7T \i_reg[25] ( .D(n303), .CP(clk), .CDN(n460), .Q(i[25]), .QN(n92)
    );
  DFCNDOBWP7T \i_reg[26] ( .D(n302), .CP(clk), .CDN(n460), .Q(i[26]), .QN(n91)
    );
  DFCNDOBWP7T \i_reg[27] ( .D(n301), .CP(clk), .CDN(n460), .Q(i[27]), .QN(n90)
    );
  DFCNDOBWP7T \i_reg[28] ( .D(n300), .CP(clk), .CDN(n460), .Q(i[28]), .QN(n89)
    );
  DFCNDOBWP7T \i_reg[29] ( .D(n299), .CP(clk), .CDN(n460), .Q(i[29]), .QN(n88)
    );
  carry_look_ahead_16bit U25 ( .a(Mand), .b(Sreg[31:16]), .cin(Sout[31]), .S(S)
    , .cout(d1) );
  multiplier_16bit_DW01_inc_0 add_107 ( .A(i), .SUM({N220, N219, N218, N217,
      N216, N215, N214, N213, N212, N211, N210, N209, N208, N207, N206, N205,
      N204, N203, N202, N201, N200, N199, N198, N197, N196, N195, N194, N193,
      N192, N191, N190, SYNOPSYS_UNCONNECTED__0}) );
  DFCNQD1BWP7T \Sout_reg[30] ( .D(n265), .CP(clk), .CDN(n332), .Q(Sout[30])
      );
  DFCNQD1BWP7T \Sout_reg[29] ( .D(n264), .CP(clk), .CDN(n460), .Q(Sout[29])
      );
  DFCNQD1BWP7T \Sout_reg[28] ( .D(n263), .CP(clk), .CDN(n460), .Q(Sout[28])
      );
  DFCNQD1BWP7T \Sout_reg[27] ( .D(n262), .CP(clk), .CDN(n332), .Q(Sout[27])
      );
  DFCNQD1BWP7T \Sout_reg[26] ( .D(n261), .CP(clk), .CDN(n332), .Q(Sout[26])
      );
  DFCNQD1BWP7T \Sout_reg[25] ( .D(n260), .CP(clk), .CDN(n460), .Q(Sout[25])
      );
  DFCNQD1BWP7T \Sout_reg[24] ( .D(n259), .CP(clk), .CDN(n460), .Q(Sout[24])
      );
  DFCNQD1BWP7T \Sout_reg[23] ( .D(n258), .CP(clk), .CDN(n332), .Q(Sout[23])
      );
  DFCNQD1BWP7T \Sout_reg[22] ( .D(n257), .CP(clk), .CDN(n332), .Q(Sout[22])
```

```
      );
    DFCNQD1BWP7T \Sout_reg[21] ( .D(n256), .CP(clk), .CDN(n332), .Q(Sout[21])
      );
    DFCNQD1BWP7T \Sout_reg[20] ( .D(n255), .CP(clk), .CDN(n460), .Q(Sout[20])
      );
    DFCNQD1BWP7T \Sout_reg[19] ( .D(n254), .CP(clk), .CDN(n460), .Q(Sout[19])
      );
    DFCNQD1BWP7T \Sout_reg[18] ( .D(n253), .CP(clk), .CDN(n332), .Q(Sout[18])
      );
    DFCNQD1BWP7T \Sout_reg[17] ( .D(n252), .CP(clk), .CDN(n332), .Q(Sout[17])
      );
    DFCNQD1BWP7T \Sout_reg[16] ( .D(n251), .CP(clk), .CDN(n460), .Q(Sout[16])
      );
    DFCNQD1BWP7T \Sout_reg[15] ( .D(n250), .CP(clk), .CDN(n332), .Q(Sout[15])
      );
    DFCNQD1BWP7T \Sout_reg[14] ( .D(n249), .CP(clk), .CDN(n460), .Q(Sout[14])
      );
    DFCNQD1BWP7T \Sout_reg[13] ( .D(n248), .CP(clk), .CDN(n332), .Q(Sout[13])
      );
    DFCNQD1BWP7T \Sout_reg[12] ( .D(n247), .CP(clk), .CDN(n460), .Q(Sout[12])
      );
    DFCNQD1BWP7T \Sout_reg[11] ( .D(n246), .CP(clk), .CDN(n460), .Q(Sout[11])
      );
    DFCNQD1BWP7T \Sout_reg[10] ( .D(n245), .CP(clk), .CDN(n460), .Q(Sout[10])
      );
    DFCNQD1BWP7T \Sout_reg[9] ( .D(n244), .CP(clk), .CDN(n332), .Q(Sout[9]) );
    DFCNQD1BWP7T \Sout_reg[8] ( .D(n243), .CP(clk), .CDN(n332), .Q(Sout[8]) );
    DFCNQD1BWP7T \Sout_reg[7] ( .D(n242), .CP(clk), .CDN(n332), .Q(Sout[7]) );
    DFCNQD1BWP7T \Sout_reg[6] ( .D(n241), .CP(clk), .CDN(n460), .Q(Sout[6]) );
    DFCNQD1BWP7T \Sout_reg[5] ( .D(n240), .CP(clk), .CDN(n332), .Q(Sout[5]) );
    DFCNQD1BWP7T \Sout_reg[4] ( .D(n239), .CP(clk), .CDN(n460), .Q(Sout[4]) );
    DFCNQD1BWP7T \Sout_reg[3] ( .D(n238), .CP(clk), .CDN(n332), .Q(Sout[3]) );
    DFCNQD1BWP7T \Sout_reg[2] ( .D(n237), .CP(clk), .CDN(n460), .Q(Sout[2]) );
    DFCNQD1BWP7T \Sout_reg[1] ( .D(n236), .CP(clk), .CDN(n460), .Q(Sout[1]) );
    DFCNQD1BWP7T \Sout_reg[0] ( .D(n235), .CP(clk), .CDN(n332), .Q(Sout[0]) );
    DFCNQD1BWP7T \Sreg_reg[15] ( .D(n282), .CP(clk), .CDN(n460), .Q(Sreg[15])
      );
    DFCNQD1BWP7T \Sreg_reg[14] ( .D(n283), .CP(clk), .CDN(n332), .Q(Sreg[14])
      );
    DFCNQD1BWP7T \Sreg_reg[13] ( .D(n284), .CP(clk), .CDN(n332), .Q(Sreg[13])
      );
    DFCNQD1BWP7T \Sreg_reg[12] ( .D(n285), .CP(clk), .CDN(n460), .Q(Sreg[12])
      );
    DFCNQD1BWP7T \Sreg_reg[11] ( .D(n286), .CP(clk), .CDN(n332), .Q(Sreg[11])
      );
    DFCNQD1BWP7T \Sreg_reg[10] ( .D(n287), .CP(clk), .CDN(n460), .Q(Sreg[10])
      );
    DFCNQD1BWP7T \Sreg_reg[9] ( .D(n288), .CP(clk), .CDN(n332), .Q(Sreg[9]) );
    DFCNQD1BWP7T \Sreg_reg[8] ( .D(n289), .CP(clk), .CDN(n332), .Q(Sreg[8]) );
    DFCNQD1BWP7T \Sreg_reg[7] ( .D(n290), .CP(clk), .CDN(n332), .Q(Sreg[7]) );
    DFCNQD1BWP7T \Sreg_reg[6] ( .D(n291), .CP(clk), .CDN(n332), .Q(Sreg[6]) );
    DFCNQD1BWP7T \Sreg_reg[5] ( .D(n292), .CP(clk), .CDN(n460), .Q(Sreg[5]) );
    DFCNQD1BWP7T \Sreg_reg[4] ( .D(n293), .CP(clk), .CDN(n332), .Q(Sreg[4]) );
    DFCNQD1BWP7T \Sreg_reg[3] ( .D(n294), .CP(clk), .CDN(n460), .Q(Sreg[3]) );
```

```
    DFCNQD1BWP7T \Sreg_reg[2] ( .D(n295), .CP(clk), .CDN(n460), .Q(Sreg[2]) );
    DFCNQD1BWP7T \Sreg_reg[1] ( .D(n296), .CP(clk), .CDN(n332), .Q(Sreg[1]) );
    DFCNQD1BWP7T \Sreg_reg[0] ( .D(n297), .CP(clk), .CDN(n332), .Q(Sreg[0]) );
    DFCNQD1BWP7T \Sreg_reg[31] ( .D(n266), .CP(clk), .CDN(n460), .Q(Sreg[31])
          );
    DFCNQD1BWP7T \Sreg_reg[30] ( .D(n267), .CP(clk), .CDN(n460), .Q(Sreg[30])
          );
    DFCNQD1BWP7T \Sreg_reg[29] ( .D(n268), .CP(clk), .CDN(n460), .Q(Sreg[29])
          );
    DFCNQD1BWP7T \Sreg_reg[28] ( .D(n269), .CP(clk), .CDN(n460), .Q(Sreg[28])
          );
    DFCNQD1BWP7T \Sreg_reg[27] ( .D(n270), .CP(clk), .CDN(n332), .Q(Sreg[27])
          );
    DFCNQD1BWP7T \Sreg_reg[26] ( .D(n271), .CP(clk), .CDN(n460), .Q(Sreg[26])
          );
    DFCNQD1BWP7T \Sreg_reg[25] ( .D(n272), .CP(clk), .CDN(n332), .Q(Sreg[25])
          );
    DFCNQD1BWP7T \Sreg_reg[24] ( .D(n273), .CP(clk), .CDN(n332), .Q(Sreg[24])
          );
    DFCNQD1BWP7T \Sreg_reg[23] ( .D(n274), .CP(clk), .CDN(n460), .Q(Sreg[23])
          );
    DFCNQD1BWP7T \Sreg_reg[22] ( .D(n275), .CP(clk), .CDN(n332), .Q(Sreg[22])
          );
    DFCNQD1BWP7T \Sreg_reg[21] ( .D(n276), .CP(clk), .CDN(n332), .Q(Sreg[21])
          );
    DFCNQD1BWP7T \Sreg_reg[20] ( .D(n277), .CP(clk), .CDN(n460), .Q(Sreg[20])
          );
    DFCNQD1BWP7T \Sreg_reg[19] ( .D(n278), .CP(clk), .CDN(n332), .Q(Sreg[19])
          );
    DFCNQD1BWP7T \Sreg_reg[16] ( .D(n281), .CP(clk), .CDN(n332), .Q(Sreg[16])
          );
    DFCNQD1BWP7T \Sreg_reg[18] ( .D(n279), .CP(clk), .CDN(n460), .Q(Sreg[18])
          );
    DFCNQD1BWP7T \Sreg_reg[17] ( .D(n280), .CP(clk), .CDN(n460), .Q(Sreg[17])
          );
    DFCNQD1BWP7T \i_reg[21] ( .D(n307), .CP(clk), .CDN(n332), .Q(i[21]) );
    DFCNQD1BWP7T \i_reg[22] ( .D(n306), .CP(clk), .CDN(n460), .Q(i[22]) );
    DFCNQD1BWP7T \i_reg[20] ( .D(n308), .CP(clk), .CDN(n460), .Q(i[20]) );
    DFCNQD1BWP7T \i_reg[19] ( .D(n309), .CP(clk), .CDN(n332), .Q(i[19]) );
    DFCNQD1BWP7T \i_reg[18] ( .D(n310), .CP(clk), .CDN(n332), .Q(i[18]) );
    DFCNQD1BWP7T \i_reg[17] ( .D(n311), .CP(clk), .CDN(n332), .Q(i[17]) );
    DFCNQD1BWP7T \i_reg[16] ( .D(n312), .CP(clk), .CDN(n332), .Q(i[16]) );
    DFCNQD1BWP7T \i_reg[11] ( .D(n317), .CP(clk), .CDN(n332), .Q(i[11]) );
    DFCNQD1BWP7T \i_reg[10] ( .D(n318), .CP(clk), .CDN(n332), .Q(i[10]) );
    DFCNQD1BWP7T \i_reg[4] ( .D(n324), .CP(clk), .CDN(n332), .Q(i[4]) );
    DFCNQD1BWP7T \i_reg[1] ( .D(n327), .CP(clk), .CDN(n332), .Q(i[1]) );
    DFCNQD1BWP7T \i_reg[0] ( .D(n329), .CP(clk), .CDN(n332), .Q(i[0]) );
    DFCND1BWP7T \i_reg[14] ( .D(n314), .CP(clk), .CDN(n460), .Q(i[14]), .QN(
         n103) );
    DFCND1BWP7T \i_reg[12] ( .D(n316), .CP(clk), .CDN(n460), .Q(i[12]), .QN(
         n105) );
    DFCND1BWP7T \i_reg[6] ( .D(n322), .CP(clk), .CDN(n460), .Q(i[6]), .QN(n111)
          );
    DFCND1BWP7T \i_reg[30] ( .D(n298), .CP(clk), .CDN(n460), .Q(i[30]), .QN(n87)
```

```
    );
  NR3D0BWP7T U339 ( .A1(i[30]), .A2(i[6]), .A3(i[5]), .ZN(n456) );
  NR3D0BWP7T U340 ( .A1(i[12]), .A2(i[14]), .A3(i[13]), .ZN(n452) );
  NR2D1BWP7T U341 ( .A1(i[31]), .A2(n447), .ZN(n391) );
  INR2XD0BWP7T U342 ( .A1(n447), .B1(n446), .ZN(n387) );
  BUFFD1BWP7T U343 ( .I(n387), .Z(n331) );
  ND2D1BWP7T U344 ( .A1(n391), .A2(i[4]), .ZN(n393) );
  INVD1BWP7T U345 ( .I(reset), .ZN(n460) );
  AO31D1BWP7T U346 ( .A1(n342), .A2(n341), .A3(n340), .B(N221), .Z(n330) );
  ND2D1BWP7T U347 ( .A1(n401), .A2(n393), .ZN(n392) );
  INVD1BWP7T U348 ( .I(n390), .ZN(n401) );
  OAI21D0BWP7T U349 ( .A1(n391), .A2(n392), .B(n393), .ZN(n388) );
  NR2D1BWP7T U350 ( .A1(n392), .A2(n397), .ZN(n389) );
  NR2D1BWP7T U351 ( .A1(n397), .A2(n391), .ZN(n390) );
  CKBD1BWP7T U352 ( .I(n460), .Z(n332) );
  NR2D1BWP7T U353 ( .A1(n391), .A2(Sreg[0]), .ZN(n398) );
  CKAN2D1BWP7T U354 ( .A1(n397), .A2(Sreg[0]), .Z(n399) );
  TIELBWP7T U355 ( .ZN(Sout[31]) );
  OR4D0BWP7T U356 ( .A1(N251), .A2(N252), .A3(N249), .A4(N250), .Z(n333) );
  AO211D0BWP7T U357 ( .A1(N248), .A2(n333), .B(N241), .C(N242), .Z(n334) );
  NR4D0BWP7T U358 ( .A1(n334), .A2(N240), .A3(N238), .A4(N239), .ZN(n342) );
  NR4D0BWP7T U359 ( .A1(N231), .A2(N232), .A3(N233), .A4(N234), .ZN(n335) );
  INR4D0BWP7T U360 ( .A1(n335), .B1(N235), .B2(N237), .B3(N236), .ZN(n341) );
  NR3D0BWP7T U361 ( .A1(N230), .A2(N228), .A3(N229), .ZN(n339) );
  NR4D0BWP7T U362 ( .A1(N224), .A2(N225), .A3(N226), .A4(N227), .ZN(n338) );
  NR3D0BWP7T U363 ( .A1(N223), .A2(N247), .A3(N222), .ZN(n337) );
  NR4D0BWP7T U364 ( .A1(N243), .A2(N244), .A3(N245), .A4(N246), .ZN(n336) );
  AN4D0BWP7T U365 ( .A1(n339), .A2(n338), .A3(n337), .A4(n336), .Z(n340) );
  MUX2ND0BWP7T U366 ( .I0(n343), .I1(n344), .S(i[0]), .ZN(n329) );
  NR2D0BWP7T U367 ( .A1(n343), .A2(n345), .ZN(n328) );
  OAI22D0BWP7T U368 ( .A1(n344), .A2(n346), .B1(n343), .B2(n347), .ZN(n327) );
  OAI22D0BWP7T U369 ( .A1(n115), .A2(n344), .B1(n343), .B2(n348), .ZN(n326) );
  OAI22D0BWP7T U370 ( .A1(n114), .A2(n344), .B1(n343), .B2(n349), .ZN(n325) );
  OAI22D0BWP7T U371 ( .A1(n344), .A2(n350), .B1(n343), .B2(n351), .ZN(n324) );
  OAI22D0BWP7T U372 ( .A1(n112), .A2(n344), .B1(n343), .B2(n352), .ZN(n323) );
  OAI22D0BWP7T U373 ( .A1(n111), .A2(n344), .B1(n343), .B2(n353), .ZN(n322) );
  OAI22D0BWP7T U374 ( .A1(n110), .A2(n344), .B1(n343), .B2(n354), .ZN(n321) );
  OAI22D0BWP7T U375 ( .A1(n109), .A2(n344), .B1(n343), .B2(n355), .ZN(n320) );
  OAI22D0BWP7T U376 ( .A1(n108), .A2(n344), .B1(n343), .B2(n356), .ZN(n319) );
  OAI22D0BWP7T U377 ( .A1(n344), .A2(n357), .B1(n343), .B2(n358), .ZN(n318) );
  OAI22D0BWP7T U378 ( .A1(n344), .A2(n359), .B1(n343), .B2(n360), .ZN(n317) );
  OAI22D0BWP7T U379 ( .A1(n105), .A2(n344), .B1(n343), .B2(n361), .ZN(n316) );
  OAI22D0BWP7T U380 ( .A1(n104), .A2(n344), .B1(n343), .B2(n362), .ZN(n315) );
  OAI22D0BWP7T U381 ( .A1(n103), .A2(n344), .B1(n343), .B2(n363), .ZN(n314) );
  OAI22D0BWP7T U382 ( .A1(n102), .A2(n344), .B1(n343), .B2(n364), .ZN(n313) );
  OAI22D0BWP7T U383 ( .A1(n344), .A2(n365), .B1(n343), .B2(n366), .ZN(n312) );
  OAI22D0BWP7T U384 ( .A1(n344), .A2(n367), .B1(n343), .B2(n368), .ZN(n311) );
  OAI22D0BWP7T U385 ( .A1(n344), .A2(n369), .B1(n343), .B2(n370), .ZN(n310) );
  OAI22D0BWP7T U386 ( .A1(n344), .A2(n371), .B1(n343), .B2(n372), .ZN(n309) );
  OAI22D0BWP7T U387 ( .A1(n344), .A2(n373), .B1(n343), .B2(n374), .ZN(n308) );
  OAI22D0BWP7T U388 ( .A1(n344), .A2(n375), .B1(n343), .B2(n376), .ZN(n307) );
  OAI22D0BWP7T U389 ( .A1(n344), .A2(n377), .B1(n343), .B2(n378), .ZN(n306) );
  OAI22D0BWP7T U390 ( .A1(n94), .A2(n344), .B1(n343), .B2(n379), .ZN(n305) );
  OAI22D0BWP7T U391 ( .A1(n93), .A2(n344), .B1(n343), .B2(n380), .ZN(n304) );
```

```
    OAI22D0BWP7T U392 ( .A1(n92), .A2(n344), .B1(n343), .B2(n381), .ZN(n303) );
    OAI22D0BWP7T U393 ( .A1(n91), .A2(n344), .B1(n343), .B2(n382), .ZN(n302) );
    OAI22D0BWP7T U394 ( .A1(n90), .A2(n344), .B1(n343), .B2(n383), .ZN(n301) );
    OAI22D0BWP7T U395 ( .A1(n89), .A2(n344), .B1(n343), .B2(n384), .ZN(n300) );
    OAI22D0BWP7T U396 ( .A1(n88), .A2(n344), .B1(n343), .B2(n385), .ZN(n299) );
    OAI22D0BWP7T U397 ( .A1(n87), .A2(n344), .B1(n343), .B2(n386), .ZN(n298) );
    CKND2D0BWP7T U398 ( .A1(n330), .A2(n344), .ZN(n343) );
    CKND2D0BWP7T U399 ( .A1(n387), .A2(n330), .ZN(n344) );
    AO222D0BWP7T U400 ( .A1(Sreg[1]), .A2(n388), .B1(Mer[0]), .B2(n389), .C1(
         Sreg[0]), .C2(n390), .Z(n297) );
    AO222D0BWP7T U401 ( .A1(Sreg[2]), .A2(n388), .B1(Mer[1]), .B2(n389), .C1(
         n390), .C2(Sreg[1]), .Z(n296) );
    AO222D0BWP7T U402 ( .A1(Sreg[3]), .A2(n388), .B1(Mer[2]), .B2(n389), .C1(
         Sreg[2]), .C2(n390), .Z(n295) );
    AO222D0BWP7T U403 ( .A1(Sreg[4]), .A2(n388), .B1(Mer[3]), .B2(n389), .C1(
         Sreg[3]), .C2(n390), .Z(n294) );
    AO222D0BWP7T U404 ( .A1(Sreg[5]), .A2(n388), .B1(Mer[4]), .B2(n389), .C1(
         Sreg[4]), .C2(n390), .Z(n293) );
    AO222D0BWP7T U405 ( .A1(Sreg[6]), .A2(n388), .B1(Mer[5]), .B2(n389), .C1(
         Sreg[5]), .C2(n390), .Z(n292) );
    AO222D0BWP7T U406 ( .A1(Sreg[7]), .A2(n388), .B1(Mer[6]), .B2(n389), .C1(
         Sreg[6]), .C2(n390), .Z(n291) );
    AO222D0BWP7T U407 ( .A1(Sreg[8]), .A2(n388), .B1(Mer[7]), .B2(n389), .C1(
         Sreg[7]), .C2(n390), .Z(n290) );
    AO222D0BWP7T U408 ( .A1(Sreg[9]), .A2(n388), .B1(Mer[8]), .B2(n389), .C1(
         Sreg[8]), .C2(n390), .Z(n289) );
    AO222D0BWP7T U409 ( .A1(Sreg[10]), .A2(n388), .B1(Mer[9]), .B2(n389), .C1(
         Sreg[9]), .C2(n390), .Z(n288) );
    AO222D0BWP7T U410 ( .A1(Sreg[11]), .A2(n388), .B1(Mer[10]), .B2(n389), .C1(
         Sreg[10]), .C2(n390), .Z(n287) );
    AO222D0BWP7T U411 ( .A1(Sreg[12]), .A2(n388), .B1(Mer[11]), .B2(n389), .C1(
         Sreg[11]), .C2(n390), .Z(n286) );
    AO222D0BWP7T U412 ( .A1(Sreg[13]), .A2(n388), .B1(Mer[12]), .B2(n389), .C1(
         Sreg[12]), .C2(n390), .Z(n285) );
    AO222D0BWP7T U413 ( .A1(Sreg[14]), .A2(n388), .B1(Mer[13]), .B2(n389), .C1(
         Sreg[13]), .C2(n390), .Z(n284) );
    AO222D0BWP7T U414 ( .A1(Sreg[15]), .A2(n388), .B1(Mer[14]), .B2(n389), .C1(
         Sreg[14]), .C2(n390), .Z(n283) );
    OAI211D0BWP7T U415 ( .A1(n394), .A2(n392), .B(n395), .C(n396), .ZN(n282) );
    AOI22D0BWP7T U416 ( .A1(Mer[15]), .A2(n389), .B1(Sreg[15]), .B2(n390), .ZN(
         n396) );
    AOI22D0BWP7T U417 ( .A1(n398), .A2(Sreg[16]), .B1(S[0]), .B2(n399), .ZN(n394)
        );
    OAI221D0BWP7T U418 ( .A1(n400), .A2(n392), .B1(n401), .B2(n402), .C(n403),
         .ZN(n281) );
    CKND0BWP7T U419 ( .I(Sreg[16]), .ZN(n402) );
    AOI22D0BWP7T U420 ( .A1(S[1]), .A2(n399), .B1(Sreg[17]), .B2(n398), .ZN(n400)
        );
    OAI221D0BWP7T U421 ( .A1(n404), .A2(n392), .B1(n401), .B2(n405), .C(n406),
         .ZN(n280) );
    CKND0BWP7T U422 ( .I(Sreg[17]), .ZN(n405) );
    AOI22D0BWP7T U423 ( .A1(S[2]), .A2(n399), .B1(Sreg[18]), .B2(n398), .ZN(n404)
        );
    OAI221D0BWP7T U424 ( .A1(n407), .A2(n392), .B1(n401), .B2(n408), .C(n409),
```

```
      .ZN(n279) );
    CKND0BWP7T U425 ( .I(Sreg[18]), .ZN(n408) );
    AOI22D0BWP7T U426 ( .A1(S[3]), .A2(n399), .B1(Sreg[19]), .B2(n398), .ZN(n407)
      );
    OAI221D0BWP7T U427 ( .A1(n410), .A2(n392), .B1(n401), .B2(n411), .C(n412),
      .ZN(n278) );
    CKND0BWP7T U428 ( .I(Sreg[19]), .ZN(n411) );
    AOI22D0BWP7T U429 ( .A1(S[4]), .A2(n399), .B1(Sreg[20]), .B2(n398), .ZN(n410)
      );
    OAI221D0BWP7T U430 ( .A1(n413), .A2(n392), .B1(n401), .B2(n414), .C(n415),
      .ZN(n277) );
    CKND0BWP7T U431 ( .I(Sreg[20]), .ZN(n414) );
    AOI22D0BWP7T U432 ( .A1(S[5]), .A2(n399), .B1(Sreg[21]), .B2(n398), .ZN(n413)
      );
    OAI221D0BWP7T U433 ( .A1(n416), .A2(n392), .B1(n401), .B2(n417), .C(n418),
      .ZN(n276) );
    CKND0BWP7T U434 ( .I(Sreg[21]), .ZN(n417) );
    AOI22D0BWP7T U435 ( .A1(S[6]), .A2(n399), .B1(Sreg[22]), .B2(n398), .ZN(n416)
      );
    OAI221D0BWP7T U436 ( .A1(n419), .A2(n392), .B1(n401), .B2(n420), .C(n421),
      .ZN(n275) );
    CKND0BWP7T U437 ( .I(Sreg[22]), .ZN(n420) );
    AOI22D0BWP7T U438 ( .A1(S[7]), .A2(n399), .B1(Sreg[23]), .B2(n398), .ZN(n419)
      );
    OAI221D0BWP7T U439 ( .A1(n422), .A2(n392), .B1(n401), .B2(n423), .C(n424),
      .ZN(n274) );
    CKND0BWP7T U440 ( .I(Sreg[23]), .ZN(n423) );
    AOI22D0BWP7T U441 ( .A1(S[8]), .A2(n399), .B1(Sreg[24]), .B2(n398), .ZN(n422)
      );
    OAI221D0BWP7T U442 ( .A1(n425), .A2(n392), .B1(n401), .B2(n426), .C(n427),
      .ZN(n273) );
    CKND0BWP7T U443 ( .I(Sreg[24]), .ZN(n426) );
    AOI22D0BWP7T U444 ( .A1(S[9]), .A2(n399), .B1(Sreg[25]), .B2(n398), .ZN(n425)
      );
    OAI221D0BWP7T U445 ( .A1(n428), .A2(n392), .B1(n401), .B2(n429), .C(n430),
      .ZN(n272) );
    CKND0BWP7T U446 ( .I(Sreg[25]), .ZN(n429) );
    AOI22D0BWP7T U447 ( .A1(S[10]), .A2(n399), .B1(Sreg[26]), .B2(n398), .ZN(
      n428) );
    OAI221D0BWP7T U448 ( .A1(n431), .A2(n392), .B1(n401), .B2(n432), .C(n433),
      .ZN(n271) );
    CKND0BWP7T U449 ( .I(Sreg[26]), .ZN(n432) );
    AOI22D0BWP7T U450 ( .A1(S[11]), .A2(n399), .B1(Sreg[27]), .B2(n398), .ZN(
      n431) );
    OAI221D0BWP7T U451 ( .A1(n434), .A2(n392), .B1(n401), .B2(n435), .C(n436),
      .ZN(n270) );
    CKND0BWP7T U452 ( .I(Sreg[27]), .ZN(n435) );
    AOI22D0BWP7T U453 ( .A1(S[12]), .A2(n399), .B1(Sreg[28]), .B2(n398), .ZN(
      n434) );
    OAI221D0BWP7T U454 ( .A1(n437), .A2(n392), .B1(n401), .B2(n438), .C(n439),
      .ZN(n269) );
    CKND0BWP7T U455 ( .I(Sreg[28]), .ZN(n438) );
    AOI22D0BWP7T U456 ( .A1(S[13]), .A2(n399), .B1(Sreg[29]), .B2(n398), .ZN(
      n437) );
```

```
    OAI221D0BWP7T U457 ( .A1(n440), .A2(n392), .B1(n401), .B2(n441), .C(n442),
         .ZN(n268) );
    CKND0BWP7T U458 ( .I(Sreg[29]), .ZN(n441) );
    AOI22D0BWP7T U459 ( .A1(S[14]), .A2(n399), .B1(Sreg[30]), .B2(n398), .ZN(
         n440) );
    OAI221D0BWP7T U460 ( .A1(n443), .A2(n392), .B1(n401), .B2(n444), .C(n445),
         .ZN(n267) );
    CKND0BWP7T U461 ( .I(Sreg[30]), .ZN(n444) );
    AOI22D0BWP7T U462 ( .A1(S[15]), .A2(n399), .B1(Sreg[31]), .B2(n398), .ZN(
         n443) );
    AO22D0BWP7T U463 ( .A1(n390), .A2(Sreg[31]), .B1(d1), .B2(n399), .Z(n266) );
    AN3D0BWP7T U464 ( .A1(n446), .A2(n447), .A3(n53), .Z(n397) );
    IOA21D0BWP7T U465 ( .A1(Sout[30]), .A2(n393), .B(n445), .ZN(n265) );
    IND2D0BWP7T U466 ( .A1(n393), .B1(Sreg[31]), .ZN(n445) );
    IOA21D0BWP7T U467 ( .A1(Sout[29]), .A2(n393), .B(n442), .ZN(n264) );
    IND2D0BWP7T U468 ( .A1(n393), .B1(Sreg[30]), .ZN(n442) );
    IOA21D0BWP7T U469 ( .A1(Sout[28]), .A2(n393), .B(n439), .ZN(n263) );
    IND2D0BWP7T U470 ( .A1(n393), .B1(Sreg[29]), .ZN(n439) );
    IOA21D0BWP7T U471 ( .A1(Sout[27]), .A2(n393), .B(n436), .ZN(n262) );
    IND2D0BWP7T U472 ( .A1(n393), .B1(Sreg[28]), .ZN(n436) );
    IOA21D0BWP7T U473 ( .A1(Sout[26]), .A2(n393), .B(n433), .ZN(n261) );
    IND2D0BWP7T U474 ( .A1(n393), .B1(Sreg[27]), .ZN(n433) );
    IOA21D0BWP7T U475 ( .A1(Sout[25]), .A2(n393), .B(n430), .ZN(n260) );
    IND2D0BWP7T U476 ( .A1(n393), .B1(Sreg[26]), .ZN(n430) );
    IOA21D0BWP7T U477 ( .A1(Sout[24]), .A2(n393), .B(n427), .ZN(n259) );
    IND2D0BWP7T U478 ( .A1(n393), .B1(Sreg[25]), .ZN(n427) );
    IOA21D0BWP7T U479 ( .A1(Sout[23]), .A2(n393), .B(n424), .ZN(n258) );
    IND2D0BWP7T U480 ( .A1(n393), .B1(Sreg[24]), .ZN(n424) );
    IOA21D0BWP7T U481 ( .A1(Sout[22]), .A2(n393), .B(n421), .ZN(n257) );
    IND2D0BWP7T U482 ( .A1(n393), .B1(Sreg[23]), .ZN(n421) );
    IOA21D0BWP7T U483 ( .A1(Sout[21]), .A2(n393), .B(n418), .ZN(n256) );
    IND2D0BWP7T U484 ( .A1(n393), .B1(Sreg[22]), .ZN(n418) );
    IOA21D0BWP7T U485 ( .A1(Sout[20]), .A2(n393), .B(n415), .ZN(n255) );
    IND2D0BWP7T U486 ( .A1(n393), .B1(Sreg[21]), .ZN(n415) );
    IOA21D0BWP7T U487 ( .A1(Sout[19]), .A2(n393), .B(n412), .ZN(n254) );
    IND2D0BWP7T U488 ( .A1(n393), .B1(Sreg[20]), .ZN(n412) );
    IOA21D0BWP7T U489 ( .A1(Sout[18]), .A2(n393), .B(n409), .ZN(n253) );
    IND2D0BWP7T U490 ( .A1(n393), .B1(Sreg[19]), .ZN(n409) );
    IOA21D0BWP7T U491 ( .A1(Sout[17]), .A2(n393), .B(n406), .ZN(n252) );
    IND2D0BWP7T U492 ( .A1(n393), .B1(Sreg[18]), .ZN(n406) );
    IOA21D0BWP7T U493 ( .A1(Sout[16]), .A2(n393), .B(n403), .ZN(n251) );
    IND2D0BWP7T U494 ( .A1(n393), .B1(Sreg[17]), .ZN(n403) );
    IOA21D0BWP7T U495 ( .A1(Sout[15]), .A2(n393), .B(n395), .ZN(n250) );
    IND2D0BWP7T U496 ( .A1(n393), .B1(Sreg[16]), .ZN(n395) );
    CKMUX2D0BWP7T U497 ( .I0(Sreg[15]), .I1(Sout[14]), .S(n393), .Z(n249) );
    CKMUX2D0BWP7T U498 ( .I0(Sreg[14]), .I1(Sout[13]), .S(n393), .Z(n248) );
    CKMUX2D0BWP7T U499 ( .I0(Sreg[13]), .I1(Sout[12]), .S(n393), .Z(n247) );
    CKMUX2D0BWP7T U500 ( .I0(Sreg[12]), .I1(Sout[11]), .S(n393), .Z(n246) );
    CKMUX2D0BWP7T U501 ( .I0(Sreg[11]), .I1(Sout[10]), .S(n393), .Z(n245) );
    CKMUX2D0BWP7T U502 ( .I0(Sreg[10]), .I1(Sout[9]), .S(n393), .Z(n244) );
    CKMUX2D0BWP7T U503 ( .I0(Sreg[9]), .I1(Sout[8]), .S(n393), .Z(n243) );
    CKMUX2D0BWP7T U504 ( .I0(Sreg[8]), .I1(Sout[7]), .S(n393), .Z(n242) );
    CKMUX2D0BWP7T U505 ( .I0(Sreg[7]), .I1(Sout[6]), .S(n393), .Z(n241) );
    CKMUX2D0BWP7T U506 ( .I0(Sreg[6]), .I1(Sout[5]), .S(n393), .Z(n240) );
```

```
    CKMUX2D0BWP7T U507 ( .I0(Sreg[5]), .I1(Sout[4]), .S(n393), .Z(n239) );
    CKMUX2D0BWP7T U508 ( .I0(Sreg[4]), .I1(Sout[3]), .S(n393), .Z(n238) );
    CKMUX2D0BWP7T U509 ( .I0(Sreg[3]), .I1(Sout[2]), .S(n393), .Z(n237) );
    CKMUX2D0BWP7T U510 ( .I0(Sreg[2]), .I1(Sout[1]), .S(n393), .Z(n236) );
    CKMUX2D0BWP7T U511 ( .I0(Sreg[1]), .I1(Sout[0]), .S(n393), .Z(n235) );
    XNR2D1BWP7T U512 ( .A1(i[0]), .A2(n387), .ZN(N252) );
    MUX2ND0BWP7T U513 ( .I0(n347), .I1(n346), .S(n331), .ZN(N251) );
    CKND0BWP7T U514 ( .I(N190), .ZN(n347) );
    MUX2ND0BWP7T U515 ( .I0(n348), .I1(n115), .S(n331), .ZN(N250) );
    CKND0BWP7T U516 ( .I(N191), .ZN(n348) );
    MUX2ND0BWP7T U517 ( .I0(n349), .I1(n114), .S(n331), .ZN(N249) );
    CKND0BWP7T U518 ( .I(N192), .ZN(n349) );
    MUX2ND0BWP7T U519 ( .I0(n351), .I1(n350), .S(n387), .ZN(N248) );
    CKND0BWP7T U520 ( .I(N193), .ZN(n351) );
    MUX2ND0BWP7T U521 ( .I0(n352), .I1(n112), .S(n387), .ZN(N247) );
    CKND0BWP7T U522 ( .I(N194), .ZN(n352) );
    MUX2ND0BWP7T U523 ( .I0(n353), .I1(n111), .S(n387), .ZN(N246) );
    CKND0BWP7T U524 ( .I(N195), .ZN(n353) );
    MUX2ND0BWP7T U525 ( .I0(n354), .I1(n110), .S(n387), .ZN(N245) );
    CKND0BWP7T U526 ( .I(N196), .ZN(n354) );
    MUX2ND0BWP7T U527 ( .I0(n355), .I1(n109), .S(n387), .ZN(N244) );
    CKND0BWP7T U528 ( .I(N197), .ZN(n355) );
    MUX2ND0BWP7T U529 ( .I0(n356), .I1(n108), .S(n387), .ZN(N243) );
    CKND0BWP7T U530 ( .I(N198), .ZN(n356) );
    MUX2ND0BWP7T U531 ( .I0(n358), .I1(n357), .S(n387), .ZN(N242) );
    CKND0BWP7T U532 ( .I(N199), .ZN(n358) );
    MUX2ND0BWP7T U533 ( .I0(n360), .I1(n359), .S(n387), .ZN(N241) );
    CKND0BWP7T U534 ( .I(N200), .ZN(n360) );
    MUX2ND0BWP7T U535 ( .I0(n361), .I1(n105), .S(n331), .ZN(N240) );
    CKND0BWP7T U536 ( .I(N201), .ZN(n361) );
    MUX2ND0BWP7T U537 ( .I0(n362), .I1(n104), .S(n331), .ZN(N239) );
    CKND0BWP7T U538 ( .I(N202), .ZN(n362) );
    MUX2ND0BWP7T U539 ( .I0(n363), .I1(n103), .S(n387), .ZN(N238) );
    CKND0BWP7T U540 ( .I(N203), .ZN(n363) );
    MUX2ND0BWP7T U541 ( .I0(n364), .I1(n102), .S(n387), .ZN(N237) );
    CKND0BWP7T U542 ( .I(N204), .ZN(n364) );
    MUX2ND0BWP7T U543 ( .I0(n366), .I1(n365), .S(n387), .ZN(N236) );
    CKND0BWP7T U544 ( .I(N205), .ZN(n366) );
    MUX2ND0BWP7T U545 ( .I0(n368), .I1(n367), .S(n387), .ZN(N235) );
    CKND0BWP7T U546 ( .I(N206), .ZN(n368) );
    MUX2ND0BWP7T U547 ( .I0(n370), .I1(n369), .S(n331), .ZN(N234) );
    CKND0BWP7T U548 ( .I(i[18]), .ZN(n369) );
    CKND0BWP7T U549 ( .I(N207), .ZN(n370) );
    MUX2ND0BWP7T U550 ( .I0(n372), .I1(n371), .S(n387), .ZN(N233) );
    CKND0BWP7T U551 ( .I(i[19]), .ZN(n371) );
    CKND0BWP7T U552 ( .I(N208), .ZN(n372) );
    MUX2ND0BWP7T U553 ( .I0(n374), .I1(n373), .S(n387), .ZN(N232) );
    CKND0BWP7T U554 ( .I(i[20]), .ZN(n373) );
    CKND0BWP7T U555 ( .I(N209), .ZN(n374) );
    MUX2ND0BWP7T U556 ( .I0(n376), .I1(n375), .S(n387), .ZN(N231) );
    CKND0BWP7T U557 ( .I(i[21]), .ZN(n375) );
    CKND0BWP7T U558 ( .I(N210), .ZN(n376) );
    MUX2ND0BWP7T U559 ( .I0(n378), .I1(n377), .S(n331), .ZN(N230) );
    CKND0BWP7T U560 ( .I(N211), .ZN(n378) );
```

```
    MUX2ND0BWP7T U561 ( .I0(n379), .I1(n94), .S(n387), .ZN(N229) );
    CKND0BWP7T U562 ( .I(N212), .ZN(n379) );
    MUX2ND0BWP7T U563 ( .I0(n380), .I1(n93), .S(n387), .ZN(N228) );
    CKND0BWP7T U564 ( .I(N213), .ZN(n380) );
    MUX2ND0BWP7T U565 ( .I0(n381), .I1(n92), .S(n387), .ZN(N227) );
    CKND0BWP7T U566 ( .I(N214), .ZN(n381) );
    MUX2ND0BWP7T U567 ( .I0(n382), .I1(n91), .S(n387), .ZN(N226) );
    CKND0BWP7T U568 ( .I(N215), .ZN(n382) );
    MUX2ND0BWP7T U569 ( .I0(n383), .I1(n90), .S(n387), .ZN(N225) );
    CKND0BWP7T U570 ( .I(N216), .ZN(n383) );
    MUX2ND0BWP7T U571 ( .I0(n384), .I1(n89), .S(n387), .ZN(N224) );
    CKND0BWP7T U572 ( .I(N217), .ZN(n384) );
    MUX2ND0BWP7T U573 ( .I0(n385), .I1(n88), .S(n387), .ZN(N223) );
    CKND0BWP7T U574 ( .I(N218), .ZN(n385) );
    MUX2ND0BWP7T U575 ( .I0(n386), .I1(n87), .S(n387), .ZN(N222) );
    CKND0BWP7T U576 ( .I(N219), .ZN(n386) );
    NR2D0BWP7T U577 ( .A1(n387), .A2(n345), .ZN(N221) );
    CKND0BWP7T U578 ( .I(N220), .ZN(n345) );
    OAI21D0BWP7T U579 ( .A1(n448), .A2(n449), .B(n53), .ZN(n446) );
    IND4D0BWP7T U580 ( .A1(n450), .B1(n105), .B2(n104), .B3(n103), .ZN(n449) );
    IND4D0BWP7T U581 ( .A1(n451), .B1(n357), .B2(n359), .B3(n350), .ZN(n448) );
    CKND0BWP7T U582 ( .I(i[4]), .ZN(n350) );
    ND4D0BWP7T U583 ( .A1(n115), .A2(n114), .A3(n452), .A4(n453), .ZN(n447) );
    NR4D0BWP7T U584 ( .A1(n454), .A2(n450), .A3(i[0]), .A4(n451), .ZN(n453) );
    IND3D0BWP7T U585 ( .A1(n455), .B1(n456), .B2(n457), .ZN(n451) );
    AN4D0BWP7T U586 ( .A1(n377), .A2(n108), .A3(n109), .A4(n110), .Z(n457) );
    CKND0BWP7T U587 ( .I(i[22]), .ZN(n377) );
    IND4D0BWP7T U588 ( .A1(n458), .B1(n94), .B2(n92), .B3(n93), .ZN(n455) );
    ND4D0BWP7T U589 ( .A1(n91), .A2(n90), .A3(n89), .A4(n88), .ZN(n458) );
    ND4D0BWP7T U590 ( .A1(n365), .A2(n367), .A3(n102), .A4(n459), .ZN(n450) );
    NR4D0BWP7T U591 ( .A1(i[21]), .A2(i[20]), .A3(i[19]), .A4(i[18]), .ZN(n459)
          );
    CKND0BWP7T U592 ( .I(i[17]), .ZN(n367) );
    CKND0BWP7T U593 ( .I(i[16]), .ZN(n365) );
    ND3D0BWP7T U594 ( .A1(n359), .A2(n346), .A3(n357), .ZN(n454) );
    CKND0BWP7T U595 ( .I(i[10]), .ZN(n357) );
    CKND0BWP7T U596 ( .I(i[1]), .ZN(n346) );
    CKND0BWP7T U597 ( .I(i[11]), .ZN(n359) );
endmodule
```

Cuadro 110: Archivo MULT32b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito

13.7.4. Simulación del circuito sintetizado

El siguiente paso fue la simulación del circuito sintetizado obtenido a partir del diseño final, ya que previamente se mostró la simulación del circuito sintetizado obtenido con el diseño inicial. Sin embargo, los pasos realizados fueron los mismos en ambos casos.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 111 en una terminal abierta en `Outputs/MULT32b/Outputs_iniciales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v MULT32b_sintetizado.v MULT32b_tb.v
     -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 111: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Luego de haber ejecutado los comandos mencionados anteriormente, se obtuvo la Figura 155. Se observó el mismo comportamiento que con el circuito original, lo que confirma que la síntesis del circuito se realizó correctamente.

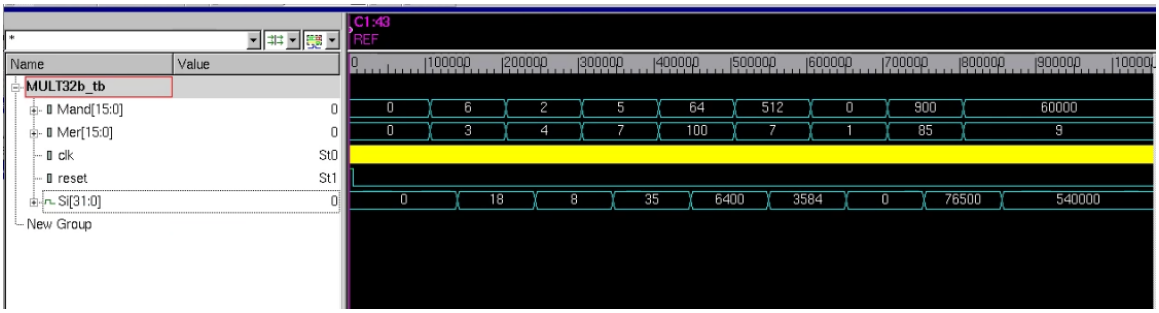


Figura 155: Simulación del multiplicador de 32 bits en su diseño sintetizado

13.7.5. Unión del circuito sintetizado con el módulo de pines de I/O

El archivo denominado MULT32b_sintetizado_and_IO.v se creó uniendo el archivo MULT32b_sintetizado.v con el archivo `IO_MULT32b.v`. Este proceso también lo realiza automáticamente el *script* antes de comenzar el segundo proceso de síntesis lógica.

Por ello, es fundamental tener diseñado desde el principio el archivo `IO_MULT32b.v` y guardarlo en la carpeta `Inputs/MULT32b`. Si este archivo no está presente, el *script* no podrá localizarlo y el nuevo archivo generado contendrá únicamente el contenido de `MULT32b_sintetizado.v`. Esto ocasionará que, en el siguiente paso, se realice nuevamente la síntesis sin incluir el módulo de pines de entrada y salida, obteniéndose un resultado similar al del archivo `MULT32b_sintetizado.v`.

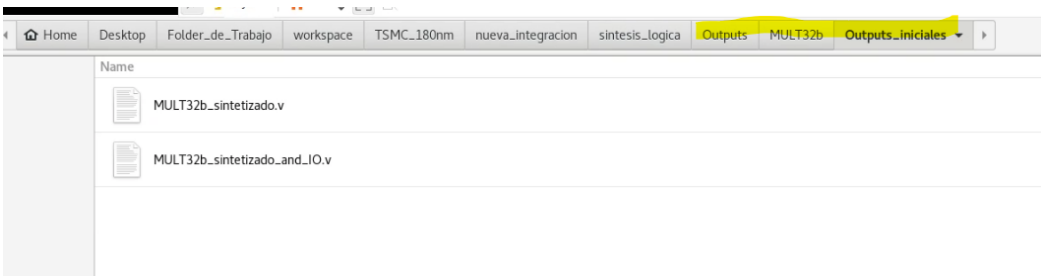


Figura 156: Archivos generados en la carpeta del circuito en Outputs_iniciales

En este caso, no se mostrará el archivo `MULT32b_sintetizado_and_IO.v` en esta sección; sin embargo, dicho archivo consiste únicamente en la combinación entre `MULT32b_sintetizado.v` y `IO_MULT32b.v`.

13.7.6. Síntesis lógica del circuito sintetizado con pines de I/O

A través del archivo `MULT32b_sintetizado_and_IO.v` se obtuvo el circuito completamente sintetizado. Este archivo se generó en el directorio `Outputs/MULT32b/Outputs_finales` con el nombre `MULT32b_sintesis_final.v`. Además, se generaron otros dos archivos que también servirán para el siguiente proceso de síntesis física.

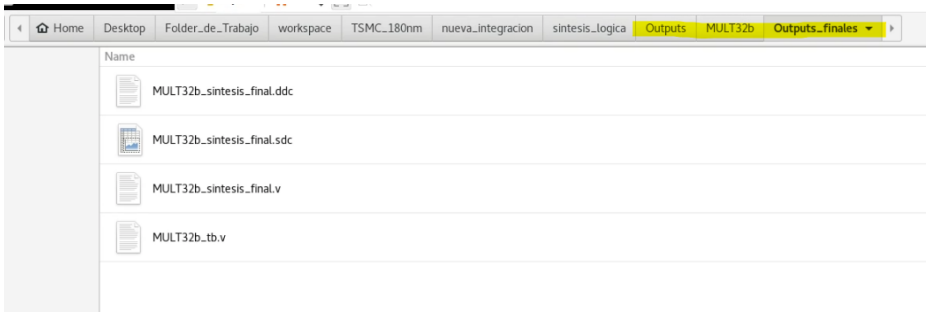


Figura 157: Archivos generados en la carpeta del circuito en Outputs_finales

En el Cuadro 112 se muestra el archivo Verilog obtenido.

```
////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date  : Sat Nov 2 22:03:53 2024
////////////////////////////////////////////////////////////////

module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
```

```
endmodule

module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;
```

```
  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule

module carry_look_ahead_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire w1, w2, w3, n1;

  carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(n1), .S(S[3:0]),
       .cout(w1) );
  carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(cout) );
  TIELBWP7T U5 ( .ZN(n1) );
endmodule

module multiplier_16bit_DW01_inc_0 ( A, SUM );
  input [31:0] A;
  output [31:0] SUM;
  wire n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16,
       n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30;

  XNR2D1BWP7T U1 ( .A1(n1), .A2(A[9]), .ZN(SUM[9]) );
  XNR2D1BWP7T U2 ( .A1(n2), .A2(A[8]), .ZN(SUM[8]) );
  CKND2D0BWP7T U3 ( .A1(A[7]), .A2(n3), .ZN(n2) );
  CKXOR2D0BWP7T U4 ( .A1(A[7]), .A2(n3), .Z(SUM[7]) );
  XNR2D1BWP7T U5 ( .A1(n4), .A2(A[6]), .ZN(SUM[6]) );
  CKND2D0BWP7T U6 ( .A1(n5), .A2(A[5]), .ZN(n4) );
  CKXOR2D0BWP7T U7 ( .A1(n5), .A2(A[5]), .Z(SUM[5]) );
  XNR2D1BWP7T U8 ( .A1(n6), .A2(A[4]), .ZN(SUM[4]) );
  CKND2D0BWP7T U9 ( .A1(A[3]), .A2(n7), .ZN(n6) );
  CKXOR2D0BWP7T U10 ( .A1(A[3]), .A2(n7), .Z(SUM[3]) );
  CKXOR2D0BWP7T U11 ( .A1(A[31]), .A2(n8), .Z(SUM[31]) );
  INR2D0BWP7T U12 ( .A1(A[30]), .B1(n9), .ZN(n8) );
  XNR2D1BWP7T U13 ( .A1(n9), .A2(A[30]), .ZN(SUM[30]) );
  CKND2D0BWP7T U14 ( .A1(A[29]), .A2(n10), .ZN(n9) );
  XNR2D1BWP7T U15 ( .A1(n11), .A2(A[2]), .ZN(SUM[2]) );
```

```
  CKXOR2D0BWP7T U16 ( .A1(A[29]), .A2(n10), .Z(SUM[29]) );
  AN3D0BWP7T U17 ( .A1(A[27]), .A2(n12), .A3(A[28]), .Z(n10) );
  XNR2D1BWP7T U18 ( .A1(n13), .A2(A[28]), .ZN(SUM[28]) );
  CKND2D0BWP7T U19 ( .A1(A[27]), .A2(n12), .ZN(n13) );
  CKXOR2D0BWP7T U20 ( .A1(n12), .A2(A[27]), .Z(SUM[27]) );
  AN2D0BWP7T U21 ( .A1(A[26]), .A2(n14), .Z(n12) );
  CKXOR2D0BWP7T U22 ( .A1(n14), .A2(A[26]), .Z(SUM[26]) );
  INR2D0BWP7T U23 ( .A1(A[25]), .B1(n15), .ZN(n14) );
  XNR2D1BWP7T U24 ( .A1(n15), .A2(A[25]), .ZN(SUM[25]) );
  ND3D0BWP7T U25 ( .A1(A[23]), .A2(n16), .A3(A[24]), .ZN(n15) );
  XNR2D1BWP7T U26 ( .A1(n17), .A2(A[24]), .ZN(SUM[24]) );
  CKND2D0BWP7T U27 ( .A1(A[23]), .A2(n16), .ZN(n17) );
  CKXOR2D0BWP7T U28 ( .A1(A[23]), .A2(n16), .Z(SUM[23]) );
  INR2D0BWP7T U29 ( .A1(A[22]), .B1(n18), .ZN(n16) );
  XNR2D1BWP7T U30 ( .A1(n18), .A2(A[22]), .ZN(SUM[22]) );
  CKND2D0BWP7T U31 ( .A1(A[21]), .A2(n19), .ZN(n18) );
  CKXOR2D0BWP7T U32 ( .A1(A[21]), .A2(n19), .Z(SUM[21]) );
  AN3D0BWP7T U33 ( .A1(A[19]), .A2(n20), .A3(A[20]), .Z(n19) );
  XNR2D1BWP7T U34 ( .A1(n21), .A2(A[20]), .ZN(SUM[20]) );
  CKND2D0BWP7T U35 ( .A1(A[19]), .A2(n20), .ZN(n21) );
  CKXOR2D0BWP7T U36 ( .A1(A[1]), .A2(A[0]), .Z(SUM[1]) );
  CKXOR2D0BWP7T U37 ( .A1(n20), .A2(A[19]), .Z(SUM[19]) );
  AN2D0BWP7T U38 ( .A1(A[18]), .A2(n22), .Z(n20) );
  CKXOR2D0BWP7T U39 ( .A1(n22), .A2(A[18]), .Z(SUM[18]) );
  INR2D0BWP7T U40 ( .A1(A[17]), .B1(n23), .ZN(n22) );
  XNR2D1BWP7T U41 ( .A1(n23), .A2(A[17]), .ZN(SUM[17]) );
  ND3D0BWP7T U42 ( .A1(A[15]), .A2(n24), .A3(A[16]), .ZN(n23) );
  XNR2D1BWP7T U43 ( .A1(n25), .A2(A[16]), .ZN(SUM[16]) );
  CKND2D0BWP7T U44 ( .A1(A[15]), .A2(n24), .ZN(n25) );
  CKXOR2D0BWP7T U45 ( .A1(A[15]), .A2(n24), .Z(SUM[15]) );
  INR2D0BWP7T U46 ( .A1(A[14]), .B1(n26), .ZN(n24) );
  XNR2D1BWP7T U47 ( .A1(n26), .A2(A[14]), .ZN(SUM[14]) );
  CKND2D0BWP7T U48 ( .A1(A[13]), .A2(n27), .ZN(n26) );
  CKXOR2D0BWP7T U49 ( .A1(A[13]), .A2(n27), .Z(SUM[13]) );
  AN3D0BWP7T U50 ( .A1(A[11]), .A2(n28), .A3(A[12]), .Z(n27) );
  XNR2D1BWP7T U51 ( .A1(n29), .A2(A[12]), .ZN(SUM[12]) );
  CKND2D0BWP7T U52 ( .A1(A[11]), .A2(n28), .ZN(n29) );
  CKXOR2D0BWP7T U53 ( .A1(n28), .A2(A[11]), .Z(SUM[11]) );
  AN2D0BWP7T U54 ( .A1(A[10]), .A2(n30), .Z(n28) );
  CKXOR2D0BWP7T U55 ( .A1(n30), .A2(A[10]), .Z(SUM[10]) );
  INR2D0BWP7T U56 ( .A1(A[9]), .B1(n1), .ZN(n30) );
  ND3D0BWP7T U57 ( .A1(A[7]), .A2(n3), .A3(A[8]), .ZN(n1) );
  AN3D0BWP7T U58 ( .A1(n5), .A2(A[5]), .A3(A[6]), .Z(n3) );
  AN3D0BWP7T U59 ( .A1(A[3]), .A2(n7), .A3(A[4]), .Z(n5) );
  INR2D0BWP7T U60 ( .A1(A[2]), .B1(n11), .ZN(n7) );
  CKND2D0BWP7T U61 ( .A1(A[1]), .A2(A[0]), .ZN(n11) );
endmodule

module multiplier_16bit ( Mand, Mer, clk, reset, Sout );
  input [15:0] Mand;
  input [15:0] Mer;
  output [31:0] Sout;
  input clk, reset;
  wire n328, n53, n326, n325, n323, n321, n320, n319, n315, n104, n313, n305,
```

```
      n304, n303, n302, n301, n300, n299, d1, N220, N219, N218, N217, N216,
      N215, N214, N213, N212, N211, N210, N209, N208, N207, N206, N205,
      N204, N203, N202, N201, N200, N199, N198, N197, N196, N195, N194,
      N193, N192, N191, N190, n265, n264, n263, n262, n261, n260, n259,
      n258, n257, n256, n255, n254, n253, n252, n251, n250, n249, n248,
      n247, n246, n245, n244, n243, n242, n241, n240, n239, n238, n237,
      n236, n235, n282, n283, n284, n285, n286, n287, n288, n289, n290,
      n291, n292, n293, n294, n295, n296, n297, n266, n267, n268, n269,
      n270, n271, n272, n273, n274, n275, n276, n277, n278, n281, n279,
      n280, n307, n306, n308, n309, n310, n311, n312, n317, n318, n324,
      n327, n329, n314, n103, n316, n105, n322, n298, n47, n1, n2, n3, n4,
      n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19,
      n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33,
      n34, n35, n36, n37, n38, n39, n40, n41, n42, n43, n44, n45, n46, n48,
      n49, n50, n51, n52, n54, n55, n56, n57, n58;
  wire [31:0] i;
  wire [31:0] Sreg;
  wire [15:0] S;
  wire SYNOPSYS_UNCONNECTED__0;

  DFCNDOBWP7T \i_reg[31] ( .D(n328), .CP(clk), .CDN(n47), .Q(i[31]), .QN(n53)
      );
  DFCNDOBWP7T \i_reg[2] ( .D(n326), .CP(clk), .CDN(n47), .Q(i[2]), .QN(n49)
      );
  DFCNDOBWP7T \i_reg[3] ( .D(n325), .CP(clk), .CDN(n47), .Q(i[3]), .QN(n50)
      );
  DFCNDOBWP7T \i_reg[5] ( .D(n323), .CP(clk), .CDN(n47), .Q(i[5]), .QN(n51)
      );
  DFCNDOBWP7T \i_reg[7] ( .D(n321), .CP(clk), .CDN(n47), .Q(i[7]) );
  DFCNDOBWP7T \i_reg[8] ( .D(n320), .CP(clk), .CDN(n47), .Q(i[8]) );
  DFCNDOBWP7T \i_reg[9] ( .D(n319), .CP(clk), .CDN(n47), .Q(i[9]) );
  DFCNDOBWP7T \i_reg[13] ( .D(n315), .CP(clk), .CDN(n47), .Q(i[13]), .QN(n104)
    );
  DFCNDOBWP7T \i_reg[15] ( .D(n313), .CP(clk), .CDN(n47), .Q(i[15]) );
  DFCNDOBWP7T \i_reg[23] ( .D(n305), .CP(clk), .CDN(n47), .Q(i[23]), .QN(n58)
      );
  DFCNDOBWP7T \i_reg[24] ( .D(n304), .CP(clk), .CDN(n47), .Q(i[24]), .QN(n56)
      );
  DFCNDOBWP7T \i_reg[25] ( .D(n303), .CP(clk), .CDN(n47), .Q(i[25]), .QN(n57)
      );
  DFCNDOBWP7T \i_reg[26] ( .D(n302), .CP(clk), .CDN(n47), .Q(i[26]) );
  DFCNDOBWP7T \i_reg[27] ( .D(n301), .CP(clk), .CDN(n47), .Q(i[27]) );
  DFCNDOBWP7T \i_reg[28] ( .D(n300), .CP(clk), .CDN(n47), .Q(i[28]) );
  DFCNDOBWP7T \i_reg[29] ( .D(n299), .CP(clk), .CDN(n47), .Q(i[29]), .QN(n55)
      );
  carry_look_ahead_16bit U25 ( .a(Mand), .b(Sreg[31:16]), .cin(Sout[31]), .S(S)
    , .cout(d1) );
  multiplier_16bit_DW01_inc_0 add_107 ( .A(i), .SUM({N220, N219, N218, N217,
      N216, N215, N214, N213, N212, N211, N210, N209, N208, N207, N206, N205,
      N204, N203, N202, N201, N200, N199, N198, N197, N196, N195, N194, N193,
      N192, N191, N190, SYNOPSYS_UNCONNECTED__0}) );
  DFCND1BWP7T \i_reg[14] ( .D(n314), .CP(clk), .CDN(n47), .Q(i[14]), .QN(n103)
    );
  DFCND1BWP7T \i_reg[12] ( .D(n316), .CP(clk), .CDN(n47), .Q(i[12]), .QN(n105)
```

```
    );
  DFCND1BWP7T \i_reg[6] ( .D(n322), .CP(clk), .CDN(n47), .Q(i[6]), .QN(n52)
      );
  DFCND1BWP7T \i_reg[30] ( .D(n298), .CP(clk), .CDN(n47), .Q(i[30]), .QN(n54)
      );
  DFCNQD1BWP7T \Sreg_reg[0] ( .D(n297), .CP(clk), .CDN(n47), .Q(Sreg[0]) );
  DFCNQD1BWP7T \Sreg_reg[14] ( .D(n283), .CP(clk), .CDN(n47), .Q(Sreg[14]) );
  DFCNQD1BWP7T \Sreg_reg[13] ( .D(n284), .CP(clk), .CDN(n47), .Q(Sreg[13]) );
  DFCNQD1BWP7T \Sreg_reg[12] ( .D(n285), .CP(clk), .CDN(n47), .Q(Sreg[12]) );
  DFCNQD1BWP7T \Sreg_reg[11] ( .D(n286), .CP(clk), .CDN(n47), .Q(Sreg[11]) );
  DFCNQD1BWP7T \Sreg_reg[10] ( .D(n287), .CP(clk), .CDN(n47), .Q(Sreg[10]) );
  DFCNQD1BWP7T \Sreg_reg[9] ( .D(n288), .CP(clk), .CDN(n47), .Q(Sreg[9]) );
  DFCNQD1BWP7T \Sreg_reg[8] ( .D(n289), .CP(clk), .CDN(n47), .Q(Sreg[8]) );
  DFCNQD1BWP7T \Sreg_reg[7] ( .D(n290), .CP(clk), .CDN(n47), .Q(Sreg[7]) );
  DFCNQD1BWP7T \Sreg_reg[6] ( .D(n291), .CP(clk), .CDN(n47), .Q(Sreg[6]) );
  DFCNQD1BWP7T \Sreg_reg[5] ( .D(n292), .CP(clk), .CDN(n47), .Q(Sreg[5]) );
  DFCNQD1BWP7T \Sreg_reg[4] ( .D(n293), .CP(clk), .CDN(n47), .Q(Sreg[4]) );
  DFCNQD1BWP7T \Sreg_reg[3] ( .D(n294), .CP(clk), .CDN(n47), .Q(Sreg[3]) );
  DFCNQD1BWP7T \Sreg_reg[2] ( .D(n295), .CP(clk), .CDN(n47), .Q(Sreg[2]) );
  DFCNQD1BWP7T \Sreg_reg[15] ( .D(n282), .CP(clk), .CDN(n47), .Q(Sreg[15]) );
  DFCNQD1BWP7T \Sreg_reg[1] ( .D(n296), .CP(clk), .CDN(n47), .Q(Sreg[1]) );
  DFCNQD1BWP7T \i_reg[10] ( .D(n318), .CP(clk), .CDN(n47), .Q(i[10]) );
  DFCNQD1BWP7T \i_reg[18] ( .D(n310), .CP(clk), .CDN(n47), .Q(i[18]) );
  DFCNQD1BWP7T \i_reg[22] ( .D(n306), .CP(clk), .CDN(n47), .Q(i[22]) );
  DFCNQD1BWP7T \i_reg[17] ( .D(n311), .CP(clk), .CDN(n47), .Q(i[17]) );
  DFCNQD1BWP7T \i_reg[1] ( .D(n327), .CP(clk), .CDN(n47), .Q(i[1]) );
  DFCNQD1BWP7T \i_reg[16] ( .D(n312), .CP(clk), .CDN(n47), .Q(i[16]) );
  DFCNQD1BWP7T \i_reg[20] ( .D(n308), .CP(clk), .CDN(n47), .Q(i[20]) );
  DFCNQD1BWP7T \i_reg[19] ( .D(n309), .CP(clk), .CDN(n47), .Q(i[19]) );
  DFCNQD1BWP7T \i_reg[11] ( .D(n317), .CP(clk), .CDN(n47), .Q(i[11]) );
  DFCNQD1BWP7T \i_reg[21] ( .D(n307), .CP(clk), .CDN(n47), .Q(i[21]) );
  DFCNQD1BWP7T \i_reg[4] ( .D(n324), .CP(clk), .CDN(n47), .Q(i[4]) );
  DFCNQD1BWP7T \i_reg[0] ( .D(n329), .CP(clk), .CDN(n47), .Q(i[0]) );
  DFCNQD1BWP7T \Sreg_reg[29] ( .D(n268), .CP(clk), .CDN(n47), .Q(Sreg[29]) );
  DFCNQD1BWP7T \Sreg_reg[27] ( .D(n270), .CP(clk), .CDN(n47), .Q(Sreg[27]) );
  DFCNQD1BWP7T \Sreg_reg[25] ( .D(n272), .CP(clk), .CDN(n47), .Q(Sreg[25]) );
  DFCNQD1BWP7T \Sreg_reg[23] ( .D(n274), .CP(clk), .CDN(n47), .Q(Sreg[23]) );
  DFCNQD1BWP7T \Sreg_reg[21] ( .D(n276), .CP(clk), .CDN(n47), .Q(Sreg[21]) );
  DFCNQD1BWP7T \Sreg_reg[19] ( .D(n278), .CP(clk), .CDN(n47), .Q(Sreg[19]) );
  DFCNQD1BWP7T \Sreg_reg[17] ( .D(n280), .CP(clk), .CDN(n47), .Q(Sreg[17]) );
  DFCNQD1BWP7T \Sreg_reg[31] ( .D(n266), .CP(clk), .CDN(n47), .Q(Sreg[31]) );
  DFCNQD1BWP7T \Sreg_reg[16] ( .D(n281), .CP(clk), .CDN(n47), .Q(Sreg[16]) );
  DFCNQD1BWP7T \Sreg_reg[30] ( .D(n267), .CP(clk), .CDN(n47), .Q(Sreg[30]) );
  DFCNQD1BWP7T \Sreg_reg[26] ( .D(n271), .CP(clk), .CDN(n47), .Q(Sreg[26]) );
  DFCNQD1BWP7T \Sreg_reg[22] ( .D(n275), .CP(clk), .CDN(n47), .Q(Sreg[22]) );
  DFCNQD1BWP7T \Sreg_reg[18] ( .D(n279), .CP(clk), .CDN(n47), .Q(Sreg[18]) );
  DFCNQD1BWP7T \Sreg_reg[28] ( .D(n269), .CP(clk), .CDN(n47), .Q(Sreg[28]) );
  DFCNQD1BWP7T \Sreg_reg[24] ( .D(n273), .CP(clk), .CDN(n47), .Q(Sreg[24]) );
  DFCNQD1BWP7T \Sreg_reg[20] ( .D(n277), .CP(clk), .CDN(n47), .Q(Sreg[20]) );
  DFCNQD1BWP7T \Sout_reg[30] ( .D(n265), .CP(clk), .CDN(n47), .Q(Sout[30]) );
  DFCNQD1BWP7T \Sout_reg[29] ( .D(n264), .CP(clk), .CDN(n47), .Q(Sout[29]) );
  DFCNQD1BWP7T \Sout_reg[28] ( .D(n263), .CP(clk), .CDN(n47), .Q(Sout[28]) );
  DFCNQD1BWP7T \Sout_reg[27] ( .D(n262), .CP(clk), .CDN(n47), .Q(Sout[27]) );
  DFCNQD1BWP7T \Sout_reg[26] ( .D(n261), .CP(clk), .CDN(n47), .Q(Sout[26]) );
```

```
    DFCNQD1BWP7T \Sout_reg[25] ( .D(n260), .CP(clk), .CDN(n47), .Q(Sout[25]) );
    DFCNQD1BWP7T \Sout_reg[24] ( .D(n259), .CP(clk), .CDN(n47), .Q(Sout[24]) );
    DFCNQD1BWP7T \Sout_reg[23] ( .D(n258), .CP(clk), .CDN(n47), .Q(Sout[23]) );
    DFCNQD1BWP7T \Sout_reg[22] ( .D(n257), .CP(clk), .CDN(n47), .Q(Sout[22]) );
    DFCNQD1BWP7T \Sout_reg[21] ( .D(n256), .CP(clk), .CDN(n47), .Q(Sout[21]) );
    DFCNQD1BWP7T \Sout_reg[20] ( .D(n255), .CP(clk), .CDN(n47), .Q(Sout[20]) );
    DFCNQD1BWP7T \Sout_reg[19] ( .D(n254), .CP(clk), .CDN(n47), .Q(Sout[19]) );
    DFCNQD1BWP7T \Sout_reg[18] ( .D(n253), .CP(clk), .CDN(n47), .Q(Sout[18]) );
    DFCNQD1BWP7T \Sout_reg[17] ( .D(n252), .CP(clk), .CDN(n47), .Q(Sout[17]) );
    DFCNQD1BWP7T \Sout_reg[16] ( .D(n251), .CP(clk), .CDN(n47), .Q(Sout[16]) );
    DFCNQD1BWP7T \Sout_reg[15] ( .D(n250), .CP(clk), .CDN(n47), .Q(Sout[15]) );
    DFCNQD1BWP7T \Sout_reg[14] ( .D(n249), .CP(clk), .CDN(n47), .Q(Sout[14]) );
    DFCNQD1BWP7T \Sout_reg[13] ( .D(n248), .CP(clk), .CDN(n47), .Q(Sout[13]) );
    DFCNQD1BWP7T \Sout_reg[12] ( .D(n247), .CP(clk), .CDN(n47), .Q(Sout[12]) );
    DFCNQD1BWP7T \Sout_reg[11] ( .D(n246), .CP(clk), .CDN(n47), .Q(Sout[11]) );
    DFCNQD1BWP7T \Sout_reg[10] ( .D(n245), .CP(clk), .CDN(n47), .Q(Sout[10]) );
    DFCNQD1BWP7T \Sout_reg[9] ( .D(n244), .CP(clk), .CDN(n47), .Q(Sout[9]) );
    DFCNQD1BWP7T \Sout_reg[8] ( .D(n243), .CP(clk), .CDN(n47), .Q(Sout[8]) );
    DFCNQD1BWP7T \Sout_reg[7] ( .D(n242), .CP(clk), .CDN(n47), .Q(Sout[7]) );
    DFCNQD1BWP7T \Sout_reg[6] ( .D(n241), .CP(clk), .CDN(n47), .Q(Sout[6]) );
    DFCNQD1BWP7T \Sout_reg[5] ( .D(n240), .CP(clk), .CDN(n47), .Q(Sout[5]) );
    DFCNQD1BWP7T \Sout_reg[4] ( .D(n239), .CP(clk), .CDN(n47), .Q(Sout[4]) );
    DFCNQD1BWP7T \Sout_reg[3] ( .D(n238), .CP(clk), .CDN(n47), .Q(Sout[3]) );
    DFCNQD1BWP7T \Sout_reg[2] ( .D(n237), .CP(clk), .CDN(n47), .Q(Sout[2]) );
    DFCNQD1BWP7T \Sout_reg[1] ( .D(n236), .CP(clk), .CDN(n47), .Q(Sout[1]) );
    DFCNQD1BWP7T \Sout_reg[0] ( .D(n235), .CP(clk), .CDN(n47), .Q(Sout[0]) );
    OAI31D1BWP7T U1 ( .A1(n31), .A2(Sreg[0]), .A3(n32), .B(n33), .ZN(n34) );
    INVD2BWP7T U2 ( .I(reset), .ZN(n47) );
    ND2D1BWP7T U3 ( .A1(i[4]), .A2(n32), .ZN(n33) );
    INR2XD0BWP7T U4 ( .A1(n53), .B1(n19), .ZN(n32) );
    NR2D1BWP7T U5 ( .A1(n37), .A2(n32), .ZN(n31) );
    TIELBWP7T U6 ( .ZN(Sout[31]) );
    NR2D0BWP7T U7 ( .A1(i[0]), .A2(n1), .ZN(n329) );
    INR2D0BWP7T U8 ( .A1(N220), .B1(n1), .ZN(n328) );
    NR2D0BWP7T U9 ( .A1(n1), .A2(n2), .ZN(n327) );
    NR2D0BWP7T U10 ( .A1(n1), .A2(n3), .ZN(n326) );
    NR2D0BWP7T U11 ( .A1(n1), .A2(n4), .ZN(n325) );
    INR2D0BWP7T U12 ( .A1(N193), .B1(n1), .ZN(n324) );
    INR2D0BWP7T U13 ( .A1(N194), .B1(n1), .ZN(n323) );
    INR2D0BWP7T U14 ( .A1(N195), .B1(n1), .ZN(n322) );
    INR2D0BWP7T U15 ( .A1(N196), .B1(n1), .ZN(n321) );
    INR2D0BWP7T U16 ( .A1(N197), .B1(n1), .ZN(n320) );
    INR2D0BWP7T U17 ( .A1(N198), .B1(n1), .ZN(n319) );
    INR2D0BWP7T U18 ( .A1(N199), .B1(n1), .ZN(n318) );
    INR2D0BWP7T U19 ( .A1(N200), .B1(n1), .ZN(n317) );
    INR2D0BWP7T U20 ( .A1(N201), .B1(n1), .ZN(n316) );
    NR2D0BWP7T U21 ( .A1(n1), .A2(n5), .ZN(n315) );
    NR2D0BWP7T U22 ( .A1(n1), .A2(n6), .ZN(n314) );
    NR2D0BWP7T U23 ( .A1(n1), .A2(n7), .ZN(n313) );
    NR2D0BWP7T U24 ( .A1(n1), .A2(n8), .ZN(n312) );
    NR2D0BWP7T U26 ( .A1(n1), .A2(n9), .ZN(n311) );
    NR2D0BWP7T U27 ( .A1(n1), .A2(n10), .ZN(n310) );
    NR2D0BWP7T U28 ( .A1(n1), .A2(n11), .ZN(n309) );
    INR2D0BWP7T U29 ( .A1(N209), .B1(n1), .ZN(n308) );
```

```
    INR2D0BWP7T U30 ( .A1(N210), .B1(n1), .ZN(n307) );
    INR2D0BWP7T U31 ( .A1(N211), .B1(n1), .ZN(n306) );
    INR2D0BWP7T U32 ( .A1(N212), .B1(n1), .ZN(n305) );
    NR2D0BWP7T U33 ( .A1(n1), .A2(n12), .ZN(n304) );
    NR2D0BWP7T U34 ( .A1(n1), .A2(n13), .ZN(n303) );
    NR2D0BWP7T U35 ( .A1(n1), .A2(n14), .ZN(n302) );
    INR2D0BWP7T U36 ( .A1(N216), .B1(n1), .ZN(n301) );
    INR2D0BWP7T U37 ( .A1(N217), .B1(n1), .ZN(n300) );
    INR2D0BWP7T U38 ( .A1(N218), .B1(n1), .ZN(n299) );
    INR2D0BWP7T U39 ( .A1(N219), .B1(n1), .ZN(n298) );
    OA31D0BWP7T U40 ( .A1(n15), .A2(n16), .A3(n17), .B(n18), .Z(n1) );
    IOA21D0BWP7T U41 ( .A1(n19), .A2(n20), .B(N220), .ZN(n18) );
    ND4D0BWP7T U42 ( .A1(n10), .A2(n11), .A3(n9), .A4(n21), .ZN(n17) );
    NR4D0BWP7T U43 ( .A1(N212), .A2(N211), .A3(N210), .A4(N209), .ZN(n21) );
    CKND0BWP7T U44 ( .I(N206), .ZN(n9) );
    CKND0BWP7T U45 ( .I(N208), .ZN(n11) );
    CKND0BWP7T U46 ( .I(N207), .ZN(n10) );
    ND4D0BWP7T U47 ( .A1(n13), .A2(n14), .A3(n12), .A4(n22), .ZN(n16) );
    NR4D0BWP7T U48 ( .A1(N219), .A2(N218), .A3(N217), .A4(N216), .ZN(n22) );
    CKND0BWP7T U49 ( .I(N213), .ZN(n12) );
    CKND0BWP7T U50 ( .I(N215), .ZN(n14) );
    CKND0BWP7T U51 ( .I(N214), .ZN(n13) );
    ND3D0BWP7T U52 ( .A1(n23), .A2(n24), .A3(n25), .ZN(n15) );
    NR4D0BWP7T U53 ( .A1(n26), .A2(N199), .A3(N201), .A4(N200), .ZN(n25) );
    ND4D0BWP7T U54 ( .A1(n5), .A2(n6), .A3(n7), .A4(n8), .ZN(n26) );
    CKND0BWP7T U55 ( .I(N205), .ZN(n8) );
    CKND0BWP7T U56 ( .I(N204), .ZN(n7) );
    CKND0BWP7T U57 ( .I(N203), .ZN(n6) );
    CKND0BWP7T U58 ( .I(N202), .ZN(n5) );
    NR4D0BWP7T U59 ( .A1(N198), .A2(N197), .A3(N196), .A4(N195), .ZN(n24) );
    AOI221D0BWP7T U60 ( .A1(n20), .A2(n19), .B1(N193), .B2(n27), .C(N194), .ZN(
        n23) );
    ND4D0BWP7T U61 ( .A1(i[0]), .A2(n2), .A3(n3), .A4(n4), .ZN(n27) );
    CKND0BWP7T U62 ( .I(N192), .ZN(n4) );
    CKND0BWP7T U63 ( .I(N191), .ZN(n3) );
    CKND0BWP7T U64 ( .I(N190), .ZN(n2) );
    CKND0BWP7T U65 ( .I(n28), .ZN(n20) );
    AO222D0BWP7T U66 ( .A1(Mer[0]), .A2(n29), .B1(Sreg[1]), .B2(n30), .C1(
        Sreg[0]), .C2(n31), .Z(n297) );
    AO222D0BWP7T U67 ( .A1(Mer[1]), .A2(n29), .B1(Sreg[2]), .B2(n30), .C1(n31),
        .C2(Sreg[1]), .Z(n296) );
    AO222D0BWP7T U68 ( .A1(Mer[2]), .A2(n29), .B1(Sreg[3]), .B2(n30), .C1(
        Sreg[2]), .C2(n31), .Z(n295) );
    AO222D0BWP7T U69 ( .A1(Mer[3]), .A2(n29), .B1(Sreg[4]), .B2(n30), .C1(
        Sreg[3]), .C2(n31), .Z(n294) );
    AO222D0BWP7T U70 ( .A1(Mer[4]), .A2(n29), .B1(Sreg[5]), .B2(n30), .C1(
        Sreg[4]), .C2(n31), .Z(n293) );
    AO222D0BWP7T U71 ( .A1(Mer[5]), .A2(n29), .B1(Sreg[6]), .B2(n30), .C1(
        Sreg[5]), .C2(n31), .Z(n292) );
    AO222D0BWP7T U72 ( .A1(Mer[6]), .A2(n29), .B1(Sreg[7]), .B2(n30), .C1(
        Sreg[6]), .C2(n31), .Z(n291) );
    AO222D0BWP7T U73 ( .A1(Mer[7]), .A2(n29), .B1(Sreg[8]), .B2(n30), .C1(
        Sreg[7]), .C2(n31), .Z(n290) );
    AO222D0BWP7T U74 ( .A1(Mer[8]), .A2(n29), .B1(Sreg[9]), .B2(n30), .C1(
```

```
      Sreg[8]), .C2(n31), .Z(n289) );
    AO222D0BWP7T U75 ( .A1(Mer[9]), .A2(n29), .B1(Sreg[10]), .B2(n30), .C1(
      Sreg[9]), .C2(n31), .Z(n288) );
    AO222D0BWP7T U76 ( .A1(Mer[10]), .A2(n29), .B1(Sreg[11]), .B2(n30), .C1(
      Sreg[10]), .C2(n31), .Z(n287) );
    AO222D0BWP7T U77 ( .A1(Mer[11]), .A2(n29), .B1(Sreg[12]), .B2(n30), .C1(
      Sreg[11]), .C2(n31), .Z(n286) );
    AO222D0BWP7T U78 ( .A1(Mer[12]), .A2(n29), .B1(Sreg[13]), .B2(n30), .C1(
      Sreg[12]), .C2(n31), .Z(n285) );
    AO222D0BWP7T U79 ( .A1(Mer[13]), .A2(n29), .B1(Sreg[14]), .B2(n30), .C1(
      Sreg[13]), .C2(n31), .Z(n284) );
    AO222D0BWP7T U80 ( .A1(Mer[14]), .A2(n29), .B1(Sreg[15]), .B2(n30), .C1(
      Sreg[14]), .C2(n31), .Z(n283) );
    OAI21D0BWP7T U81 ( .A1(n32), .A2(n31), .B(n33), .ZN(n30) );
    AO221D0BWP7T U82 ( .A1(Mer[15]), .A2(n29), .B1(Sreg[16]), .B2(n34), .C(n35),
      .Z(n282) );
    AO22D0BWP7T U83 ( .A1(n31), .A2(Sreg[15]), .B1(S[0]), .B2(n36), .Z(n35) );
    INR3D0BWP7T U84 ( .A1(n33), .B1(n31), .B2(n37), .ZN(n29) );
    AO222D0BWP7T U85 ( .A1(S[1]), .A2(n36), .B1(Sreg[17]), .B2(n34), .C1(
      Sreg[16]), .C2(n31), .Z(n281) );
    AO222D0BWP7T U86 ( .A1(S[2]), .A2(n36), .B1(Sreg[18]), .B2(n34), .C1(
      Sreg[17]), .C2(n31), .Z(n280) );
    AO222D0BWP7T U87 ( .A1(S[3]), .A2(n36), .B1(Sreg[19]), .B2(n34), .C1(
      Sreg[18]), .C2(n31), .Z(n279) );
    AO222D0BWP7T U88 ( .A1(S[4]), .A2(n36), .B1(Sreg[20]), .B2(n34), .C1(
      Sreg[19]), .C2(n31), .Z(n278) );
    AO222D0BWP7T U89 ( .A1(S[5]), .A2(n36), .B1(Sreg[21]), .B2(n34), .C1(
      Sreg[20]), .C2(n31), .Z(n277) );
    AO222D0BWP7T U90 ( .A1(S[6]), .A2(n36), .B1(Sreg[22]), .B2(n34), .C1(
      Sreg[21]), .C2(n31), .Z(n276) );
    AO222D0BWP7T U91 ( .A1(S[7]), .A2(n36), .B1(Sreg[23]), .B2(n34), .C1(
      Sreg[22]), .C2(n31), .Z(n275) );
    AO222D0BWP7T U92 ( .A1(S[8]), .A2(n36), .B1(Sreg[24]), .B2(n34), .C1(
      Sreg[23]), .C2(n31), .Z(n274) );
    AO222D0BWP7T U93 ( .A1(S[9]), .A2(n36), .B1(Sreg[25]), .B2(n34), .C1(
      Sreg[24]), .C2(n31), .Z(n273) );
    AO222D0BWP7T U94 ( .A1(S[10]), .A2(n36), .B1(Sreg[26]), .B2(n34), .C1(
      Sreg[25]), .C2(n31), .Z(n272) );
    AO222D0BWP7T U95 ( .A1(S[11]), .A2(n36), .B1(Sreg[27]), .B2(n34), .C1(
      Sreg[26]), .C2(n31), .Z(n271) );
    AO222D0BWP7T U96 ( .A1(S[12]), .A2(n36), .B1(Sreg[28]), .B2(n34), .C1(
      Sreg[27]), .C2(n31), .Z(n270) );
    AO222D0BWP7T U97 ( .A1(S[13]), .A2(n36), .B1(Sreg[29]), .B2(n34), .C1(
      Sreg[28]), .C2(n31), .Z(n269) );
    AO222D0BWP7T U98 ( .A1(S[14]), .A2(n36), .B1(Sreg[30]), .B2(n34), .C1(
      Sreg[29]), .C2(n31), .Z(n268) );
    AO222D0BWP7T U99 ( .A1(S[15]), .A2(n36), .B1(Sreg[31]), .B2(n34), .C1(
      Sreg[30]), .C2(n31), .Z(n267) );
    AO22D0BWP7T U100 ( .A1(n31), .A2(Sreg[31]), .B1(d1), .B2(n36), .Z(n266) );
    AN2D0BWP7T U101 ( .A1(n37), .A2(Sreg[0]), .Z(n36) );
    AN3D0BWP7T U102 ( .A1(n28), .A2(n19), .A3(n53), .Z(n37) );
    OAI21D0BWP7T U103 ( .A1(i[4]), .A2(n38), .B(n53), .ZN(n28) );
    CKMUX2D0BWP7T U104 ( .I0(Sreg[31]), .I1(Sout[30]), .S(n33), .Z(n265) );
    CKMUX2D0BWP7T U105 ( .I0(Sreg[30]), .I1(Sout[29]), .S(n33), .Z(n264) );
```

```
  CKMUX2D0BWP7T U106 ( .I0(Sreg[29]), .I1(Sout[28]), .S(n33), .Z(n263) );
  CKMUX2D0BWP7T U107 ( .I0(Sreg[28]), .I1(Sout[27]), .S(n33), .Z(n262) );
  CKMUX2D0BWP7T U108 ( .I0(Sreg[27]), .I1(Sout[26]), .S(n33), .Z(n261) );
  CKMUX2D0BWP7T U109 ( .I0(Sreg[26]), .I1(Sout[25]), .S(n33), .Z(n260) );
  CKMUX2D0BWP7T U110 ( .I0(Sreg[25]), .I1(Sout[24]), .S(n33), .Z(n259) );
  CKMUX2D0BWP7T U111 ( .I0(Sreg[24]), .I1(Sout[23]), .S(n33), .Z(n258) );
  CKMUX2D0BWP7T U112 ( .I0(Sreg[23]), .I1(Sout[22]), .S(n33), .Z(n257) );
  CKMUX2D0BWP7T U113 ( .I0(Sreg[22]), .I1(Sout[21]), .S(n33), .Z(n256) );
  CKMUX2D0BWP7T U114 ( .I0(Sreg[21]), .I1(Sout[20]), .S(n33), .Z(n255) );
  CKMUX2D0BWP7T U115 ( .I0(Sreg[20]), .I1(Sout[19]), .S(n33), .Z(n254) );
  CKMUX2D0BWP7T U116 ( .I0(Sreg[19]), .I1(Sout[18]), .S(n33), .Z(n253) );
  CKMUX2D0BWP7T U117 ( .I0(Sreg[18]), .I1(Sout[17]), .S(n33), .Z(n252) );
  CKMUX2D0BWP7T U118 ( .I0(Sreg[17]), .I1(Sout[16]), .S(n33), .Z(n251) );
  CKMUX2D0BWP7T U119 ( .I0(Sreg[16]), .I1(Sout[15]), .S(n33), .Z(n250) );
  CKMUX2D0BWP7T U120 ( .I0(Sreg[15]), .I1(Sout[14]), .S(n33), .Z(n249) );
  CKMUX2D0BWP7T U121 ( .I0(Sreg[14]), .I1(Sout[13]), .S(n33), .Z(n248) );
  CKMUX2D0BWP7T U122 ( .I0(Sreg[13]), .I1(Sout[12]), .S(n33), .Z(n247) );
  CKMUX2D0BWP7T U123 ( .I0(Sreg[12]), .I1(Sout[11]), .S(n33), .Z(n246) );
  CKMUX2D0BWP7T U124 ( .I0(Sreg[11]), .I1(Sout[10]), .S(n33), .Z(n245) );
  CKMUX2D0BWP7T U125 ( .I0(Sreg[10]), .I1(Sout[9]), .S(n33), .Z(n244) );
  CKMUX2D0BWP7T U126 ( .I0(Sreg[9]), .I1(Sout[8]), .S(n33), .Z(n243) );
  CKMUX2D0BWP7T U127 ( .I0(Sreg[8]), .I1(Sout[7]), .S(n33), .Z(n242) );
  CKMUX2D0BWP7T U128 ( .I0(Sreg[7]), .I1(Sout[6]), .S(n33), .Z(n241) );
  CKMUX2D0BWP7T U129 ( .I0(Sreg[6]), .I1(Sout[5]), .S(n33), .Z(n240) );
  CKMUX2D0BWP7T U130 ( .I0(Sreg[5]), .I1(Sout[4]), .S(n33), .Z(n239) );
  CKMUX2D0BWP7T U131 ( .I0(Sreg[4]), .I1(Sout[3]), .S(n33), .Z(n238) );
  CKMUX2D0BWP7T U132 ( .I0(Sreg[3]), .I1(Sout[2]), .S(n33), .Z(n237) );
  CKMUX2D0BWP7T U133 ( .I0(Sreg[2]), .I1(Sout[1]), .S(n33), .Z(n236) );
  CKMUX2D0BWP7T U134 ( .I0(Sreg[1]), .I1(Sout[0]), .S(n33), .Z(n235) );
  IND4D0BWP7T U135 ( .A1(n38), .B1(n49), .B2(n50), .B3(n39), .ZN(n19) );
  NR2D0BWP7T U136 ( .A1(i[1]), .A2(i[0]), .ZN(n39) );
  ND4D0BWP7T U137 ( .A1(n40), .A2(n41), .A3(n42), .A4(n43), .ZN(n38) );
  INR4D0BWP7T U138 ( .A1(n44), .B1(i[18]), .B2(i[16]), .B3(i[17]), .ZN(n43) );
  NR4D0BWP7T U139 ( .A1(i[22]), .A2(i[21]), .A3(i[20]), .A4(i[19]), .ZN(n44)
        );
  NR4D0BWP7T U140 ( .A1(n45), .A2(i[15]), .A3(i[11]), .A4(i[10]), .ZN(n42) );
  ND3D0BWP7T U141 ( .A1(n104), .A2(n103), .A3(n105), .ZN(n45) );
  NR4D0BWP7T U142 ( .A1(n46), .A2(i[26]), .A3(i[28]), .A4(i[27]), .ZN(n41) );
  ND4D0BWP7T U143 ( .A1(n55), .A2(n56), .A3(n57), .A4(n58), .ZN(n46) );
  NR4D0BWP7T U144 ( .A1(n48), .A2(i[7]), .A3(i[9]), .A4(i[8]), .ZN(n40) );
  ND3D0BWP7T U145 ( .A1(n52), .A2(n54), .A3(n51), .ZN(n48) );
endmodule

module multiplier_16bit_IO ( Mand, Mer, clk, reset, Sout );
  input [15:0] Mand;
  input [15:0] Mer;
  output [31:0] Sout;
  input clk, reset;
  wire clk_w, reset_w, net1687, n2, n3, n4;
  wire [15:0] Mand_w;
  wire [15:0] Mer_w;
  wire [31:0] Sout_w;
  tri [15:0] Mand;
  tri [15:0] Mer;
```

```
    tri clk;
    tri reset;
    tri [31:0] Sout;

    multiplier_16bit mult_inst ( .Mand(Mand_w), .Mer(Mer_w), .clk(n4), .reset(
        reset_w), .Sout(Sout_w) );
    PDDW0204SCDG U_Mand0 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[0]), .DS(
        net1687), .PE(net1687), .C(Mand_w[0]) );
    PDDW0204SCDG U_Mand1 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[1]), .DS(
        net1687), .PE(net1687), .C(Mand_w[1]) );
    PDDW0204SCDG U_Mand2 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[2]), .DS(
        net1687), .PE(net1687), .C(Mand_w[2]) );
    PDDW0204SCDG U_Mand3 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[3]), .DS(
        net1687), .PE(net1687), .C(Mand_w[3]) );
    PDDW0204SCDG U_Mand4 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[4]), .DS(
        net1687), .PE(net1687), .C(Mand_w[4]) );
    PDDW0204SCDG U_Mand5 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[5]), .DS(
        net1687), .PE(net1687), .C(Mand_w[5]) );
    PDDW0204SCDG U_Mand6 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[6]), .DS(
        net1687), .PE(net1687), .C(Mand_w[6]) );
    PDDW0204SCDG U_Mand7 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[7]), .DS(
        net1687), .PE(net1687), .C(Mand_w[7]) );
    PDDW0204SCDG U_Mand8 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[8]), .DS(
        net1687), .PE(net1687), .C(Mand_w[8]) );
    PDDW0204SCDG U_Mand9 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[9]), .DS(
        net1687), .PE(net1687), .C(Mand_w[9]) );
    PDDW0204SCDG U_Mand10 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[10]), .DS(
        net1687), .PE(net1687), .C(Mand_w[10]) );
    PDDW0204SCDG U_Mand11 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[11]), .DS(
        net1687), .PE(net1687), .C(Mand_w[11]) );
    PDDW0204SCDG U_Mand12 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[12]), .DS(
        net1687), .PE(net1687), .C(Mand_w[12]) );
    PDDW0204SCDG U_Mand13 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[13]), .DS(
        net1687), .PE(net1687), .C(Mand_w[13]) );
    PDDW0204SCDG U_Mand14 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[14]), .DS(
        net1687), .PE(net1687), .C(Mand_w[14]) );
    PDDW0204SCDG U_Mand15 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mand[15]), .DS(
        net1687), .PE(net1687), .C(Mand_w[15]) );
    PDDW0204SCDG U_Mer0 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[0]), .DS(
        net1687), .PE(net1687), .C(Mer_w[0]) );
    PDDW0204SCDG U_Mer1 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[1]), .DS(
        net1687), .PE(net1687), .C(Mer_w[1]) );
    PDDW0204SCDG U_Mer2 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[2]), .DS(
        net1687), .PE(net1687), .C(Mer_w[2]) );
    PDDW0204SCDG U_Mer3 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[3]), .DS(
        net1687), .PE(net1687), .C(Mer_w[3]) );
    PDDW0204SCDG U_Mer4 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[4]), .DS(
        net1687), .PE(net1687), .C(Mer_w[4]) );
    PDDW0204SCDG U_Mer5 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[5]), .DS(
        net1687), .PE(net1687), .C(Mer_w[5]) );
    PDDW0204SCDG U_Mer6 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[6]), .DS(
        net1687), .PE(net1687), .C(Mer_w[6]) );
    PDDW0204SCDG U_Mer7 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[7]), .DS(
        net1687), .PE(net1687), .C(Mer_w[7]) );
```

```
    PDDW0204SCDG U_Mer8 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[8]), .DS(
        net1687), .PE(net1687), .C(Mer_w[8]) );
    PDDW0204SCDG U_Mer9 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[9]), .DS(
        net1687), .PE(net1687), .C(Mer_w[9]) );
    PDDW0204SCDG U_Mer10 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[10]), .DS(
        net1687), .PE(net1687), .C(Mer_w[10]) );
    PDDW0204SCDG U_Mer11 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[11]), .DS(
        net1687), .PE(net1687), .C(Mer_w[11]) );
    PDDW0204SCDG U_Mer12 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[12]), .DS(
        net1687), .PE(net1687), .C(Mer_w[12]) );
    PDDW0204SCDG U_Mer13 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[13]), .DS(
        net1687), .PE(net1687), .C(Mer_w[13]) );
    PDDW0204SCDG U_Mer14 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[14]), .DS(
        net1687), .PE(net1687), .C(Mer_w[14]) );
    PDDW0204SCDG U_Mer15 ( .I(net1687), .OEN(n2), .IE(n2), .PAD(Mer[15]), .DS(
        net1687), .PE(net1687), .C(Mer_w[15]) );
    PDDW0204SCDG U_clk ( .I(net1687), .OEN(n2), .IE(n2), .PAD(clk), .DS(net1687),
        .PE(net1687), .C(clk_w) );
    PDDW0204SCDG U_reset ( .I(net1687), .OEN(n2), .IE(n2), .PAD(reset), .DS(
        net1687), .PE(net1687), .C(reset_w) );
    PDDW0204SCDG U_Sout0 ( .I(Sout_w[0]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[0]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout1 ( .I(Sout_w[1]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[1]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout2 ( .I(Sout_w[2]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[2]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout3 ( .I(Sout_w[3]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[3]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout4 ( .I(Sout_w[4]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[4]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout5 ( .I(Sout_w[5]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[5]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout6 ( .I(Sout_w[6]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[6]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout7 ( .I(Sout_w[7]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[7]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout8 ( .I(Sout_w[8]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[8]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout9 ( .I(Sout_w[9]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[9]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout10 ( .I(Sout_w[10]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[10]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout11 ( .I(Sout_w[11]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[11]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout12 ( .I(Sout_w[12]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[12]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout13 ( .I(Sout_w[13]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[13]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout14 ( .I(Sout_w[14]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[14]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout15 ( .I(Sout_w[15]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[15]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout16 ( .I(Sout_w[16]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[16]), .DS(net1687), .PE(net1687) );
```

```
    PDDW0204SCDG U_Sout17 ( .I(Sout_w[17]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[17]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout18 ( .I(Sout_w[18]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[18]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout19 ( .I(Sout_w[19]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[19]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout20 ( .I(Sout_w[20]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[20]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout21 ( .I(Sout_w[21]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[21]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout22 ( .I(Sout_w[22]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[22]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout23 ( .I(Sout_w[23]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[23]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout24 ( .I(Sout_w[24]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[24]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout25 ( .I(Sout_w[25]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[25]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout26 ( .I(Sout_w[26]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[26]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout27 ( .I(Sout_w[27]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[27]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout28 ( .I(Sout_w[28]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[28]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout29 ( .I(Sout_w[29]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[29]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout30 ( .I(Sout_w[30]), .OEN(net1687), .IE(net1687), .PAD(
        Sout[30]), .DS(net1687), .PE(net1687) );
    PDDW0204SCDG U_Sout31 ( .I(net1687), .OEN(net1687), .IE(net1687), .PAD(
        Sout[31]), .DS(net1687), .PE(net1687) );
    CKND1BWP7T U6 ( .I(clk_w), .ZN(n3) );
    INVD1BWP7T U7 ( .I(n3), .ZN(n4) );
    TIELBWP7T U8 ( .ZN(net1687) );
    TIEHBWP7T U9 ( .Z(n2) );
  endmodule
```

Cuadro 112: Archivo MULT32b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O

13.7.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O

El último paso que se realizó fue la simulación del circuito completamente sintetizado junto con los pines de I/O. Esto se llevó a cabo nuevamente para comprobar que el circuito completamente sintetizado se comporte correctamente.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 113 en una terminal abierta en `Outputs/MULT32b/Outputs_finales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v MULT32b_sintesis_final.v
    MULT32b_tb.v -o simulacion -full64 -debug_access+all
```

```
./simulacion

dve -full64
```

Cuadro 113: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en la Figura 158, el comportamiento es idéntico al que se obtuvo anteriormente, lo que confirma que el *netlist* final obtenido del circuito sintetizado con pines de I/O es correcto.

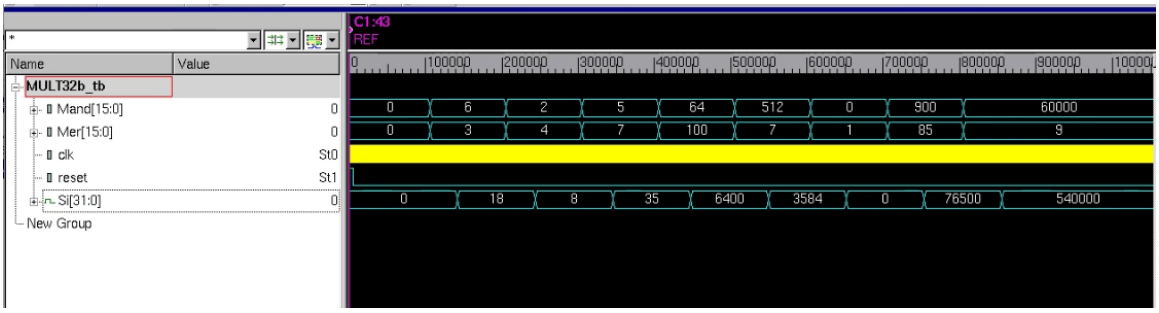


Figura 158: Simulación del *netlist* final

13.8. Proceso para la síntesis lógica de un divisor de 32 bit usando un *carry look-ahead subtractor* de 16 bits

13.8.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench*

El primer paso fue crear una carpeta en el directorio de *inputs*, nombrada según el circuito principal a sintetizar; en este caso, `DIV32b` (tanto la carpeta como el nombre del archivo Verilog que contiene el circuito principal deben tener el mismo nombre). Luego, se diseñaron los archivos necesarios para la simulación y síntesis del circuito.

Siguiendo la estructura explicada en el capítulo y sección 9.2, el archivo que contiene el módulo principal del divisor de 32 bits se nombró `DIV32b.v`. El archivo de *testbench*, utilizado para simular y verificar el funcionamiento del circuito, se denominó `DIV32b_tb.v`. Por último, el archivo que define el módulo de entradas y salidas se nombró `IO_DIV32b.v`.

En este caso, se utilizaron otros circuitos para construir el circuito principal. Se emplearon los sumadores de 4 y 16 bits y el restador de 16 bits diseñados anteriormente como base para crear el divisor de 32 bits. Por lo tanto, los archivos Verilog de los sumadores de 4 y 16 bits, así como el del restador de 16 bits, también se encuentran en la carpeta `Inputs/DIV32b/`, ya que, para los procesos de simulación y síntesis, es importante hacer referencia a todos los archivos utilizados.

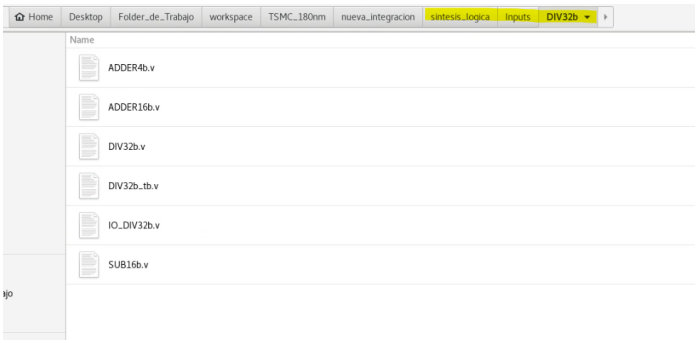


Figura 159: Archivos necesarios en la carpeta del circuito en *inputs*

En el Cuadro 114 se muestra el diseño inicial del circuito, el cual hace referencia al módulo del restador de 16 bits para su implementación. Por otro lado, en el Cuadro 115 se presenta el diseño final del circuito utilizado. Esto se debió a que, al momento de sintetizar el diseño inicial (Cuadro 114), se obtuvieron errores, los cuales se explicarán posteriormente.

Por otro lado, en el Cuadro 116 se presenta el diseño del módulo de pines de entrada y salida. Además, en el Cuadro 117 se muestra el *testbench* diseñado para la simulación tanto del diseño final del circuito original como del circuito sintetizado.

Dado que el diseño del circuito incluye múltiples entradas y salidas, en el módulo de pines es necesario hacer referencia a cada una de ellas de forma individual. Sin embargo, el proceso de creación de cada pin es el mismo para todas las entradas y salidas (ver Cuadro 116).

```
module divisor_16bit(Divisor, Dividend, clk, Quotient, Remainder,Sreg);

input [15:0] Divisor, Dividend;
input clk;
output reg [15:0] Quotient;
output reg [15:0] Remainder;
//reg [31:0] S;
wire [15:0] S;
reg clk2;
reg [15:0] Divisor_temp;
output reg [31:0] Sreg;
reg [15:0] quo;
//wire [31:0] Sreg;
wire d1;
reg F;
integer i;
initial begin
i = 0;
F=clk;

Sreg = {16'b0, Dividend};
end


substractor_16bit U36(Sreg[31:16],Divisor_temp, 1'b0, S, d1);
```

```
always@(posedge clk or negedge clk) begin

        //Sreg = Sreg << 1;
        F=~F;
        if(i == 0&F==1) begin
            Sreg = {16'b0, Dividend};
            Divisor_temp = Divisor;
            //Sreg = Sreg << 1;


        end

        if(i>0 & i<18&F==1) begin

            Sreg = Sreg << 1;
        end


        if(i>0 & i<18&F==0) begin
            if (S[15] == 1'b1) begin

                quo = quo << 1;

                end

                if (S[15] == 1'b0)begin

                quo = quo << 1;
                quo[0] = 1'b1;
                Sreg[31:16] = S;
                end
        end

        if(i == 18 &F==0) begin
            Quotient = quo;
            //Remainder = Sreg[31:16];

        end

        if(i == 17 &F==0) begin
            //Quotient = quo;
            Remainder = Sreg[31:16];

        end


        if(i < 19 &F==1) begin
        i = i+1;
        end

        if (i >= 19 &F==1) begin
```

```
                i=0;
            end

    end
    endmodule
```

Cuadro 114: Archivo DIV32b.v: diseño inicial de un divisor de 32 bits en Verilog

```
    module divisor_16bit(Divisor, Dividend, clk, reset, Quotient, Remainder, Sreg);

    input [15:0] Divisor, Dividend;
    input clk, reset;
    output reg [15:0] Quotient;
    output reg [15:0] Remainder;
    output reg [31:0] Sreg;

    reg [15:0] Divisor_temp;
    reg [15:0] quo;
    reg F;
    wire [15:0] S;
    wire d1;
    integer i;

substractor_16bit U36(Sreg[31:16],Divisor_temp, 1'b0, S, d1);

always @(posedge clk or posedge reset) begin
        if (reset) begin
                i = 0; // Inicializa el contador a 0 en un reset
                //Sreg = {16'b0, Dividend};
                F = 1'b0;
        end
        else begin
                F=~F;
                if(i == 0&F==1) begin
                    Sreg = {16'b0, Dividend};
                    Divisor_temp = Divisor;
                    //Sreg = Sreg << 1;


                end

                if(i>0 & i<18&F==1) begin

                    Sreg = Sreg << 1;
                end


                if(i>0 & i<18&F==0) begin
                    if (S[15] == 1'b1) begin

                        quo = quo << 1;

                        end
```

```
                    if (S[15] == 1'b0)begin

                    quo = quo << 1;
                    quo[0] = 1'b1;
                    Sreg[31:16] = S;
                    end
            end

            if(i == 18 &F==0) begin
                Quotient = quo;
                //Remainder = Sreg[31:16];

            end

            if(i == 17 &F==0) begin
                //Quotient = quo;
                Remainder = Sreg[31:16];

            end


            if(i < 19 &F==1) begin
            i = i+1;
            end

            if (i >= 19 &F==1) begin
                i=0;
            end
        end

end
endmodule
```

Cuadro 115: Archivo DIV32b.v: diseño final de un divisor de 32 bits en Verilog

```
    module divisor_16bit_IO (
        input [15:0] Divisor,
        input [15:0] Dividend,
        input clk,
        input reset,
        output [15:0] Quotient,
        output [15:0] Remainder,
        output [31:0] Sreg
    );
        wire [15:0] Divisor_w, Dividend_w, Quotient_w, Remainder_w;
        wire [31:0] Sreg_w;
        wire clk_w, reset_w;

        // Instancia del modulo divisor_16bit
        divisor_16bit div_inst (
            .Divisor(Divisor_w),
            .Dividend(Dividend_w),
            .clk(clk_w),
            .reset(reset_w),
```

```
        .Quotient(Quotient_w),
        .Remainder(Remainder_w),
        .Sreg(Sreg_w)
    );

    // Interconexion de pines de entrada (Divisor, Dividend, clk, reset)
    // Conexiones de los pines del Dividend
    PDDW0204SCDG U_Divd0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[0]), .
        C(Dividend_w[0]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[1]), .
        C(Dividend_w[1]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[2]), .
        C(Dividend_w[2]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[3]), .
        C(Dividend_w[3]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[4]), .
        C(Dividend_w[4]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[5]), .
        C(Dividend_w[5]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[6]), .
        C(Dividend_w[6]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[7]), .
        C(Dividend_w[7]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[8]), .
        C(Dividend_w[8]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[9]), .
        C(Dividend_w[9]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[10]),
         .C(Dividend_w[10]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[11]),
         .C(Dividend_w[11]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[12]),
         .C(Dividend_w[12]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[13]),
         .C(Dividend_w[13]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[14]),
         .C(Dividend_w[14]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Divd15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Dividend[15]),
         .C(Dividend_w[15]), .PE(1'b0), .IE(1'b1));

    // Conexiones de los pines del Divisor
    PDDW0204SCDG U_Div0 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[0]), .C(
        Divisor_w[0]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Div1 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[1]), .C(
        Divisor_w[1]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Div2 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[2]), .C(
        Divisor_w[2]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Div3 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[3]), .C(
        Divisor_w[3]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Div4 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[4]), .C(
        Divisor_w[4]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Div5 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[5]), .C(
        Divisor_w[5]), .PE(1'b0), .IE(1'b1));
    PDDW0204SCDG U_Div6 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[6]), .C(
```

```
    Divisor_w[6]), .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_Div7 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[7]), .C(
    Divisor_w[7]), .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_Div8 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[8]), .C(
    Divisor_w[8]), .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_Div9 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[9]), .C(
    Divisor_w[9]), .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_Div10 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[10]), .
    C(Divisor_w[10]), .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_Div11 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[11]), .
    C(Divisor_w[11]), .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_Div12 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[12]), .
    C(Divisor_w[12]), .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_Div13 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[13]), .
    C(Divisor_w[13]), .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_Div14 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[14]), .
    C(Divisor_w[14]), .PE(1'b0), .IE(1'b1));
PDDW0204SCDG U_Div15 (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(Divisor[15]), .
    C(Divisor_w[15]), .PE(1'b0), .IE(1'b1));


// Entrada clk
PDDW0204SCDG U_clk (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(clk), .C(clk_w),
    .PE(1'b0), .IE(1'b1));

// Entrada reset
PDDW0204SCDG U_reset (.I(1'b0), .DS(1'b0), .OEN(1'b1), .PAD(reset), .C(
    reset_w), .PE(1'b0), .IE(1'b1));

// Interconexion de pines de salida (Quotient, Remainder, Sreg)
// Conexiones de los pines del Quotient
PDDW0204SCDG U_Quot0 (.I(Quotient_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[0]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot1 (.I(Quotient_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[1]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot2 (.I(Quotient_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[2]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot3 (.I(Quotient_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[3]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot4 (.I(Quotient_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[4]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot5 (.I(Quotient_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[5]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot6 (.I(Quotient_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[6]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot7 (.I(Quotient_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[7]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot8 (.I(Quotient_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[8]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot9 (.I(Quotient_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[9]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot10 (.I(Quotient_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(
    Quotient[10]), .C(1'bx), .PE(1'b0), .IE(1'b0));
PDDW0204SCDG U_Quot11 (.I(Quotient_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(
```

```
        Quotient[11]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Quot12 (.I(Quotient_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(
        Quotient[12]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Quot13 (.I(Quotient_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(
        Quotient[13]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Quot14 (.I(Quotient_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(
        Quotient[14]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Quot15 (.I(Quotient_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(
        Quotient[15]), .C(1'bx), .PE(1'b0), .IE(1'b0));

    // Conexiones de los pines del Remainder
    PDDW0204SCDG U_Rem0 (.I(Remainder_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[0]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem1 (.I(Remainder_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[1]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem2 (.I(Remainder_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[2]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem3 (.I(Remainder_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[3]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem4 (.I(Remainder_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[4]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem5 (.I(Remainder_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[5]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem6 (.I(Remainder_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[6]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem7 (.I(Remainder_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[7]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem8 (.I(Remainder_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[8]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem9 (.I(Remainder_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[9]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem10 (.I(Remainder_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[10]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem11 (.I(Remainder_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[11]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem12 (.I(Remainder_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[12]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem13 (.I(Remainder_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[13]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem14 (.I(Remainder_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[14]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Rem15 (.I(Remainder_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(
        Remainder[15]), .C(1'bx), .PE(1'b0), .IE(1'b0));

    // Salida Sreg (32 bits)
    PDDW0204SCDG U_Sreg0 (.I(Sreg_w[0]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[0]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg1 (.I(Sreg_w[1]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[1]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg2 (.I(Sreg_w[2]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[2]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg3 (.I(Sreg_w[3]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[3]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg4 (.I(Sreg_w[4]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[4]),
```

```
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg5 (.I(Sreg_w[5]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[5]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg6 (.I(Sreg_w[6]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[6]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg7 (.I(Sreg_w[7]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[7]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg8 (.I(Sreg_w[8]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[8]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg9 (.I(Sreg_w[9]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg[9]),
        .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg10 (.I(Sreg_w[10]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [10]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg11 (.I(Sreg_w[11]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [11]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg12 (.I(Sreg_w[12]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [12]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg13 (.I(Sreg_w[13]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [13]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg14 (.I(Sreg_w[14]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [14]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg15 (.I(Sreg_w[15]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [15]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg16 (.I(Sreg_w[16]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [16]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg17 (.I(Sreg_w[17]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [17]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg18 (.I(Sreg_w[18]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [18]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg19 (.I(Sreg_w[19]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [19]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg20 (.I(Sreg_w[20]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [20]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg21 (.I(Sreg_w[21]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [21]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg22 (.I(Sreg_w[22]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [22]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg23 (.I(Sreg_w[23]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [23]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg24 (.I(Sreg_w[24]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [24]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg25 (.I(Sreg_w[25]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [25]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg26 (.I(Sreg_w[26]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [26]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg27 (.I(Sreg_w[27]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [27]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg28 (.I(Sreg_w[28]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [28]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg29 (.I(Sreg_w[29]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [29]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg30 (.I(Sreg_w[30]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
        [30]), .C(1'bx), .PE(1'b0), .IE(1'b0));
    PDDW0204SCDG U_Sreg31 (.I(Sreg_w[31]), .DS(1'b0), .OEN(1'b0), .PAD(Sreg
```

```
            [31]), .C(1'bx), .PE(1'b0), .IE(1'b0));


        // Pines de VDD y VSS
        PVDD1CDG PVDD();
        PVSS1CDG PVSS();

    endmodule
```

Cuadro 116: Archivo IO_DIV32b.v: diseño de módulo de entradas y salidas para un divisor de 32 bits en Verilog

```
module DIV32b_tb;
reg [15:0] Divisor, Dividend;

wire [15:0] Quotient, Remainder;
wire [31:0] Sreg;
reg clk,reset;

// instantiate the design block
divisor_16bit divisor (Divisor, Dividend, clk,reset, Quotient, Remainder,Sreg);
//ivisor_16bit(Divisor, Dividend, clk, Quotient, Remainder,Sreg);

// Generar el reloj
    initial begin
        clk = 1'b0;
        forever #1 clk = ~clk; // Generar un reloj de 1 unidad de tiempo
    end

initial begin
// Configurar archivo de volcado para FSDB y VCD
        $fsdbDumpfile("DIV32b.fsdb");
        $fsdbDumpvars(0, DIV32b_tb);
        $dumpfile("DIV32b.vcd");
        $dumpvars(0, DIV32b_tb);

    reset = 1;
    #5 reset = 0; // Desactivar el reset


    Dividend = 16'd12; Divisor = 16'd4; clk = 0;
    #100
    Divisor = 16'd2;
    Dividend = 16'd4;

    #100
    Dividend = 16'd512;
    Divisor = 16'd7;

    #100
    Divisor = 16'd6;
    Dividend = 16'd20464;

    #1000 $finish;
    end
endmodule
```

Cuadro 117: Archivo DIV32b_tb.v: diseño del *testbench* para simulación de un divisor de 32 bits en Verilog

13.8.2. Simulación del diseño original del circuito

Como siguiente paso, se simuló el diseño original y final del circuito con el objetivo de verificar que no existieran errores de compilación y que el diseño del circuito fuera correcto. Como se explicó anteriormente, si no se realiza este paso, existe el riesgo de encontrar errores debido a posibles fallos en el diseño del circuito, lo que impediría proceder a la ejecución de la síntesis lógica.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 118 en una terminal abierta en `Inputs/DIV32b/`.

```
vcs -v DIV32b.v SUB16b.v ADDER16b.v ADDER4b.v DIV32b_tb.v -o simulacion -full64
    -debug_access+all

./simulacion

dve -full64
```

Cuadro 118: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Como se observa en el Cuadro 118, es necesario colocar el nombre de todos los archivos Verilog que se están utilizando. Si no se realiza este paso, se producirá un error, ya que la aplicación no encontrará el módulo al que el diseño original hace referencia.

Siguiendo los pasos descritos en el capítulo y sección 12.1 para la visualización del archivo `.vcd` generado al ejecutar el archivo de simulación, se obtuvo la Figura 160.

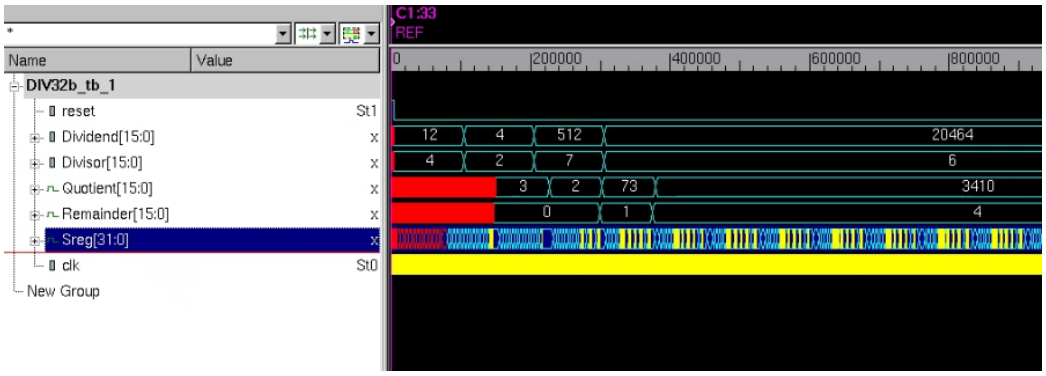


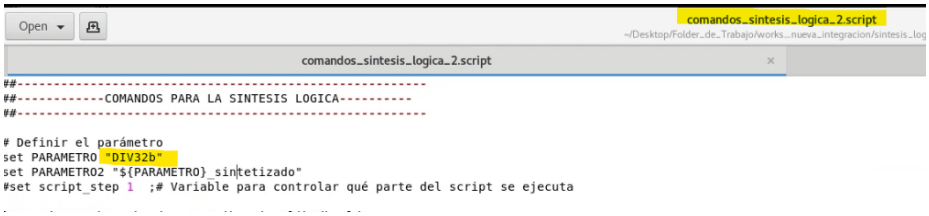
Figura 160: Simulación de sumador de 32 bits en su diseño original

La Figura 160 es la misma tanto para la simulación del diseño inicial como para la del diseño final del circuito. Como se observa, el comportamiento de ambos circuitos es el esperado, ya que se realiza la división entre las entradas *Dividend* y *Divisor*, y el resultado se obtiene correctamente en las salidas *quotient* (cociente) y *remainder* (residuo).

Por lo tanto, se procedió a ejecutar el *script* para llevar a cabo la síntesis lógica del circuito.

13.8.3. Síntesis lógica del diseño original del circuito

Antes de proceder a ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, fue necesario abrir tanto este *script* como el archivo `comandos_sintesis_logica_2.script` para definir en la variable `PARAMETRO` el nombre de la carpeta que se creó en *inputs*, la cual contiene el circuito a sintetizar. En este caso, se definió como `DIV32b`.



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 161: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

De igual manera, fue necesario modificar una parte del *script* `comandos_sintesis_logica_2.script`. Dado que el diseño del divisor de 32 bits utiliza los módulos que se encuentran en otros archivos Verilog, es necesario hacer referencia a estos archivos para sintetizar correctamente el diseño.

La modificación debe realizarse en el primer paso de síntesis, en la parte donde se leen los archivos que describen el circuito. En la línea que contiene el comando `read_file`, se deben agregar los archivos adicionales necesarios. En este caso, la modificación quedaría de la siguiente forma:

```
if { $script_step == 1 } {
# Parte 1 - Sintesis Cell original

# Usar el parametro en lugar del nombre del modulo
read_file -format verilog "Inputs/$PARAMETRO/$PARAMETRO.v Inputs/$PARAMETRO/
    ADDER4b.v Inputs/$PARAMETRO/ADDER16b.v Inputs/$PARAMETRO/SUB16b.v"
```

Cuadro 119: Modificación del *script* comandos_sintesis_logica_2.script

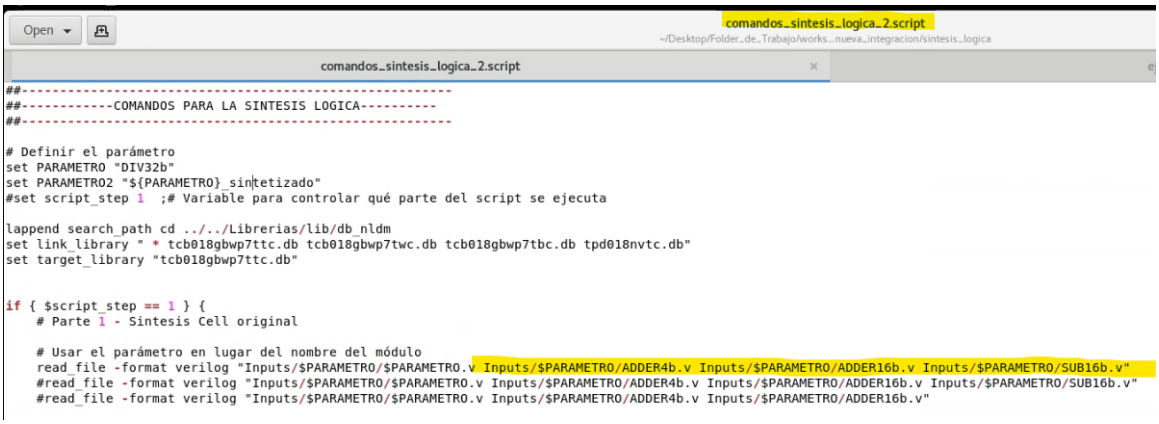


Figura 162: Modificación del *script* comandos_sintesis_logica_2.script

Como se observa en el Cuadro 119 y en la Figura 162, la única modificación realizada fue agregar la lectura de los archivos `ADDER4b.v`, `ADDER16b.v` y `SUB16b.v` en el comando `read_file`, ya que los módulos contenidos en esos archivos son utilizados en `DIV32b.v` y, por lo tanto, también deben ser sintetizados. Si no se realiza esta modificación, se producirá un error en el proceso de síntesis.

Luego, se ejecutó el *script* utilizando el comando que se muestra en el Cuadro 120, en una terminal abierta en el directorio `sintesis_logica`.

```
./ejecutar_sintesis_logica_2.bash
```

Cuadro 120: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash

Al ejecutar este comando, se creó una carpeta en *outputs* con el nombre `DIV32b`. Además, se generaron dos subdirectorios dentro de esta, llamados `Outputs_iniciales` y `Outputs_finales`.

Sin embargo, al utilizar el diseño original propuesto para el divisor surgieron algunas advertencias. Al finalizar el proceso, la terminal mostró una advertencia que indicaba que el generador de síntesis había tenido que añadir una conexión adicional (*net*) en el módulo `divisor_16bit`. Esta conexión fue prefijada con `SYNOPSYS_UNCONNECTED_`, lo cual indica que la herramienta de síntesis agregó dicha conexión para asegurar la consistencia del diseño en términos de conectividad (ver Figura 163).

Ante esto, fue necesario revisar si existían otros mensajes de advertencia desde el inicio, ya que si se presenta un problema en la primera síntesis, la segunda síntesis con pines también fallará. En efecto, al comienzo, tras realizar la lectura del diseño original del circuito, se obtuvo un error (ver Figura 164). Este error ocurre en la línea 30 del archivo `DIV32b.v`, donde se indica que se están utilizando eventos de disparo `posedge` y `negedge` en la misma señal `clk` dentro de un mismo bloque, lo cual no es compatible con la síntesis lógica.

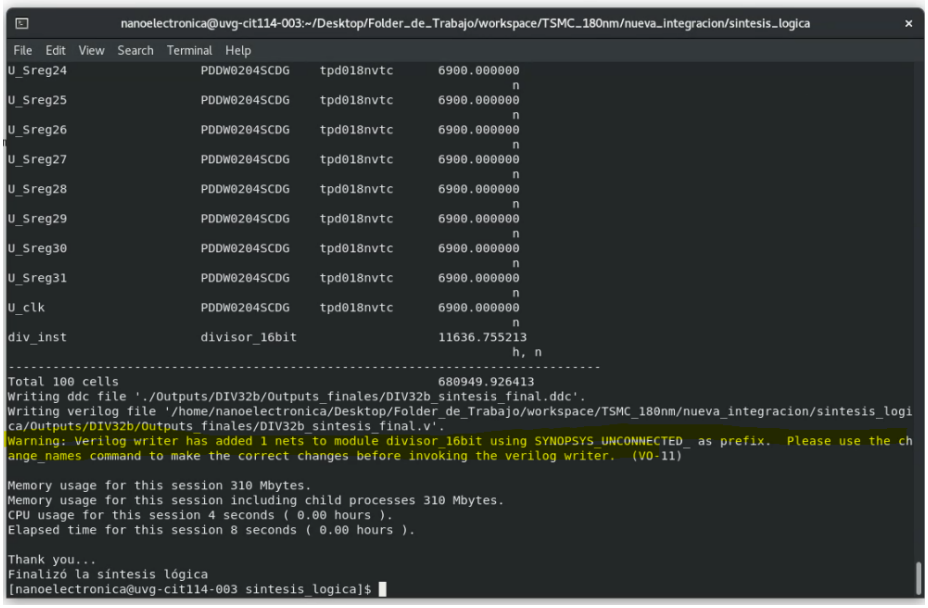


Figura 163: Advertencia al terminar la síntesis

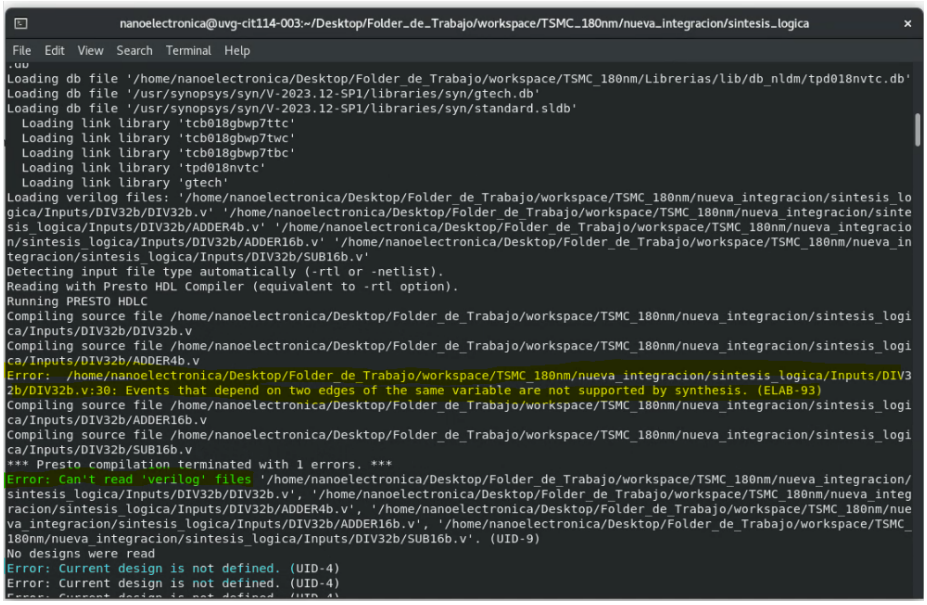


Figura 164: Error en la primera síntesis

Debido a este error, surgieron otros errores que indicaban que no se podían leer los archivos Verilog a causa del problema inicial. Si la síntesis se detiene por un error en uno de los archivos, los demás no pueden ser procesados. Por esta razón, en la carpeta `Outputs/DIV32b/Outputs_iniciales` no se creó el archivo `DIV32b_sintetizado.v`.

Para solucionar este problema, se intentó utilizar dos bloques *always* para manejar por separado los eventos de flanco positivo (`posedge`) y flanco negativo (`negedge`), lo cual es permitido en síntesis. Sin embargo, también surgieron errores al intentar manipular los mis-

mos registros y variables en ambos bloques *always*. Para resolver esto, se rediseñó el divisor para que utilizara únicamente el flanco positivo (`posedge`). Además, se añadió una entrada externa al circuito llamada *reset*, con el objetivo de implementar la funcionalidad de un bloque *initial*. Esto fue necesario porque en el diseño original se estaban utilizando bloques *initial*, los cuales no son compatibles con la síntesis, tal como se observó anteriormente en el caso del multiplicador.

Al repetir los pasos anteriores utilizando el diseño final del circuito, este se sintetizó sin errores. Aunque todavía apareció la advertencia mostrada en la Figura 165, ya no se presentó la advertencia de la Figura 166, lo que permitió sintetizar correctamente el diseño. También se generaron reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O. Asimismo, en la carpeta `Outputs/DIV32b/Outputs_iniciales` se creó un archivo llamado `DIV32b_sintetizado.v`, que corresponde al resultado de la primera etapa de la síntesis lógica.

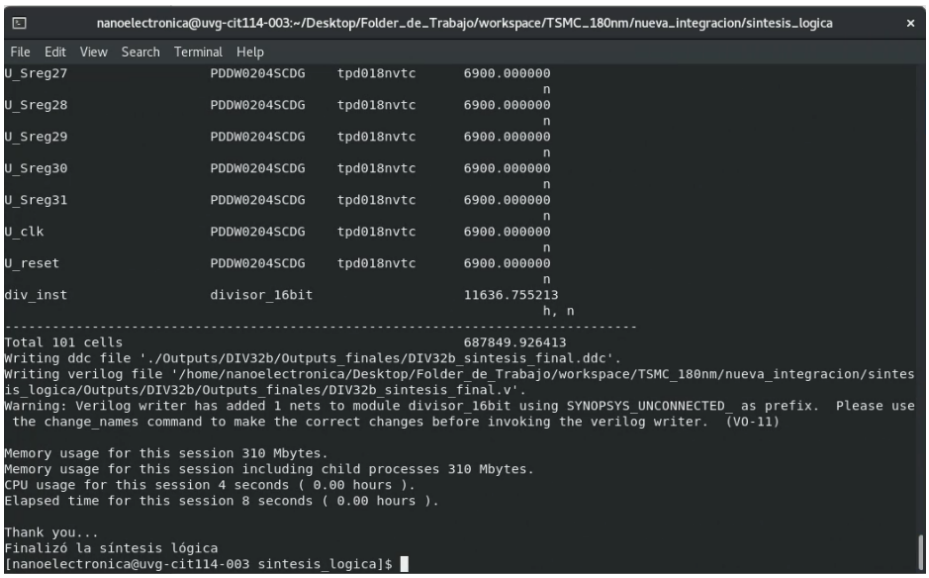


Figura 165: Advertencia al terminar la síntesis

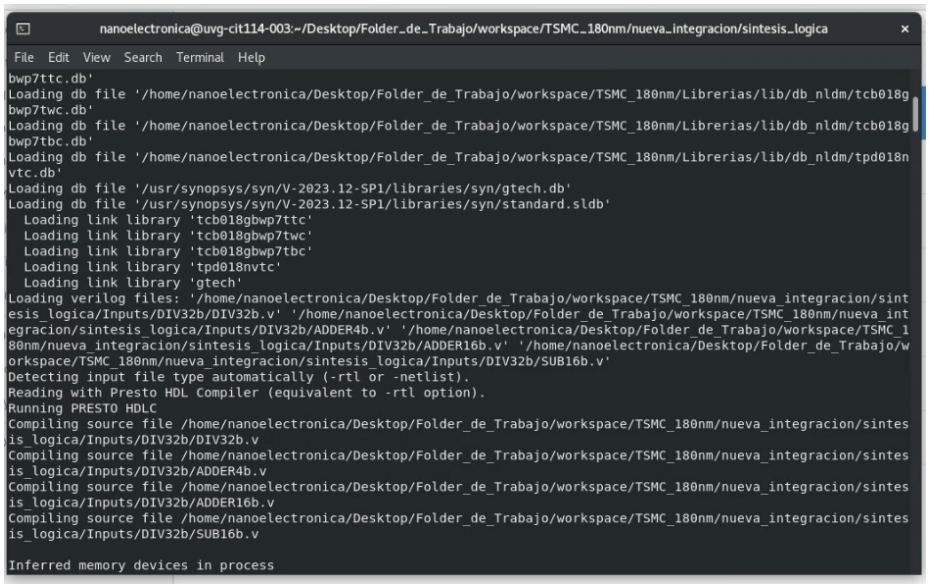


Figura 166: Error solucionado de la primera síntesis

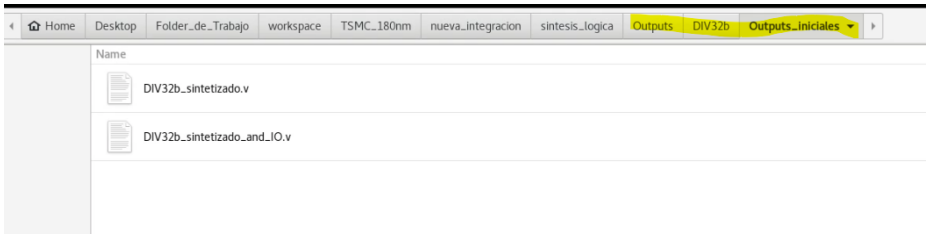


Figura 167: Archivos generados en la carpeta del circuito en Outputs_iniciales

Es importante recalcar que siempre es necesario revisar los *warnings* o errores en la terminal para asegurar que la síntesis se ejecute correctamente.

El archivo `DIV32b_sintetizado.v`, que corresponde al circuito sintetizado obtenido a partir del diseño final, se muestra en el Cuadro 121.

```
////////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date  : Sun Nov 3 00:39:21 2024
////////////////////////////////////////////////////////////////


module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;
```

```
  CKXOR2D1BWP7T U1 ( .A1(n1), .A2(n2), .Z(S[3]) );
  CKXOR2D1BWP7T U2 ( .A1(n4), .A2(n3), .Z(S[2]) );
  CKXOR2D1BWP7T U3 ( .A1(n5), .A2(n6), .Z(S[1]) );
  MOAI22D0BWP7T U4 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  AOI22D0BWP7T U5 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D1BWP7T U6 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  MOAI22D0BWP7T U7 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  XNR2D1BWP7T U8 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  XNR2D1BWP7T U9 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U10 ( .A1(b[0]), .A2(a[0]), .B1(n7), .B2(cin), .ZN(n5) );
  CKXOR2D1BWP7T U11 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
  CKXOR2D1BWP7T U12 ( .A1(cin), .A2(n7), .Z(S[0]) );
endmodule


module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n8, n9, n10, n11, n12, n13, n14;

  CKXOR2D1BWP7T U1 ( .A1(n14), .A2(n13), .Z(S[3]) );
  CKXOR2D1BWP7T U2 ( .A1(n11), .A2(n12), .Z(S[2]) );
  CKXOR2D1BWP7T U3 ( .A1(n10), .A2(n9), .Z(S[1]) );
  CKXOR2D1BWP7T U4 ( .A1(cin), .A2(n8), .Z(S[0]) );
  MOAI22D0BWP7T U5 ( .A1(n14), .A2(n13), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  MOAI22D0BWP7T U6 ( .A1(n10), .A2(n9), .B1(b[1]), .B2(a[1]), .ZN(n12) );
  AOI22D0BWP7T U7 ( .A1(b[0]), .A2(a[0]), .B1(n8), .B2(cin), .ZN(n10) );
  AOI22D0BWP7T U8 ( .A1(b[2]), .A2(a[2]), .B1(n12), .B2(n11), .ZN(n14) );
  CKXOR2D1BWP7T U9 ( .A1(b[2]), .A2(a[2]), .Z(n11) );
  CKXOR2D1BWP7T U10 ( .A1(b[0]), .A2(a[0]), .Z(n8) );
  XNR2D1BWP7T U11 ( .A1(b[1]), .A2(a[1]), .ZN(n9) );
  XNR2D1BWP7T U12 ( .A1(b[3]), .A2(a[3]), .ZN(n13) );
endmodule


module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n8, n9, n10, n11, n12, n13, n14;

  CKXOR2D1BWP7T U1 ( .A1(n14), .A2(n13), .Z(S[3]) );
  CKXOR2D1BWP7T U2 ( .A1(n11), .A2(n12), .Z(S[2]) );
  CKXOR2D1BWP7T U3 ( .A1(n10), .A2(n9), .Z(S[1]) );
  CKXOR2D1BWP7T U4 ( .A1(cin), .A2(n8), .Z(S[0]) );
  MOAI22D0BWP7T U5 ( .A1(n14), .A2(n13), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  MOAI22D0BWP7T U6 ( .A1(n10), .A2(n9), .B1(b[1]), .B2(a[1]), .ZN(n12) );
  AOI22D0BWP7T U7 ( .A1(b[0]), .A2(a[0]), .B1(n8), .B2(cin), .ZN(n10) );
  AOI22D0BWP7T U8 ( .A1(b[2]), .A2(a[2]), .B1(n12), .B2(n11), .ZN(n14) );
```

```
  CKXOR2D1BWP7T U9 ( .A1(b[2]), .A2(a[2]), .Z(n11) );
  CKXOR2D1BWP7T U10 ( .A1(b[0]), .A2(a[0]), .Z(n8) );
  XNR2D1BWP7T U11 ( .A1(b[1]), .A2(a[1]), .ZN(n9) );
  XNR2D1BWP7T U12 ( .A1(b[3]), .A2(a[3]), .ZN(n13) );
endmodule

module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n8, n9, n10, n11, n12, n13, n14;

  CKXOR2D1BWP7T U1 ( .A1(n14), .A2(n13), .Z(S[3]) );
  CKXOR2D1BWP7T U2 ( .A1(n11), .A2(n12), .Z(S[2]) );
  CKXOR2D1BWP7T U3 ( .A1(n10), .A2(n9), .Z(S[1]) );
  CKXOR2D1BWP7T U4 ( .A1(cin), .A2(n8), .Z(S[0]) );
  MOAI22D0BWP7T U5 ( .A1(n14), .A2(n13), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  MOAI22D0BWP7T U6 ( .A1(n10), .A2(n9), .B1(b[1]), .B2(a[1]), .ZN(n12) );
  AOI22D0BWP7T U7 ( .A1(b[0]), .A2(a[0]), .B1(n8), .B2(cin), .ZN(n10) );
  AOI22D0BWP7T U8 ( .A1(b[2]), .A2(a[2]), .B1(n12), .B2(n11), .ZN(n14) );
  CKXOR2D1BWP7T U9 ( .A1(b[2]), .A2(a[2]), .Z(n11) );
  CKXOR2D1BWP7T U10 ( .A1(b[0]), .A2(a[0]), .Z(n8) );
  XNR2D1BWP7T U11 ( .A1(b[3]), .A2(a[3]), .ZN(n13) );
  XNR2D1BWP7T U12 ( .A1(b[1]), .A2(a[1]), .ZN(n9) );
endmodule

module carry_look_ahead_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire \*Logic0* , w1, w2, w3;

  carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(\*Logic0* ), .S(
       S[3:0]), .cout(w1) );
  carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(cout) );
  TIELBWP7T U6 ( .ZN(\*Logic0* ) );
endmodule

module substractor_16bit_DW01_inc_0_DW01_inc_1 ( A, SUM );
  input [15:0] A;
  output [15:0] SUM;

  wire [15:2] carry;

  HA1D0BWP7T U1_1_13 ( .A(A[13]), .B(carry[13]), .CO(carry[14]), .S(SUM[13])
```

```
        );
  HA1D0BWP7T U1_1_12 ( .A(A[12]), .B(carry[12]), .CO(carry[13]), .S(SUM[12])
        );
  HA1D0BWP7T U1_1_11 ( .A(A[11]), .B(carry[11]), .CO(carry[12]), .S(SUM[11])
        );
  HA1D0BWP7T U1_1_14 ( .A(A[14]), .B(carry[14]), .CO(carry[15]), .S(SUM[14])
        );
  HA1D0BWP7T U1_1_9 ( .A(A[9]), .B(carry[9]), .CO(carry[10]), .S(SUM[9]) );
  HA1D0BWP7T U1_1_8 ( .A(A[8]), .B(carry[8]), .CO(carry[9]), .S(SUM[8]) );
  HA1D0BWP7T U1_1_7 ( .A(A[7]), .B(carry[7]), .CO(carry[8]), .S(SUM[7]) );
  HA1D0BWP7T U1_1_10 ( .A(A[10]), .B(carry[10]), .CO(carry[11]), .S(SUM[10])
        );
  HA1D0BWP7T U1_1_5 ( .A(A[5]), .B(carry[5]), .CO(carry[6]), .S(SUM[5]) );
  HA1D0BWP7T U1_1_4 ( .A(A[4]), .B(carry[4]), .CO(carry[5]), .S(SUM[4]) );
  HA1D0BWP7T U1_1_3 ( .A(A[3]), .B(carry[3]), .CO(carry[4]), .S(SUM[3]) );
  HA1D0BWP7T U1_1_6 ( .A(A[6]), .B(carry[6]), .CO(carry[7]), .S(SUM[6]) );
  HA1D0BWP7T U1_1_1 ( .A(A[1]), .B(A[0]), .CO(carry[2]), .S(SUM[1]) );
  HA1D0BWP7T U1_1_2 ( .A(A[2]), .B(carry[2]), .CO(carry[3]), .S(SUM[2]) );
  INVD1BWP7T U1 ( .I(A[0]), .ZN(SUM[0]) );
  CKXOR2D0BWP7T U2 ( .A1(carry[15]), .A2(A[15]), .Z(SUM[15]) );
endmodule

module substractor_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire \*Logic0* , n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13,
       n14, n15, n16;
  wire [15:0] b_comp;

  carry_look_ahead_16bit U35 ( .a(a), .b(b_comp), .cin(\*Logic0* ), .S(S),
       .cout(cout) );
  substractor_16bit_DW01_inc_0_DW01_inc_1 add_29 ( .A({n16, n15, n14, n13, n12,
       n11, n10, n9, n8, n7, n6, n5, n4, n3, n2, n1}), .SUM(b_comp) );
  INVD1BWP7T U3 ( .I(b[2]), .ZN(n3) );
  INVD1BWP7T U4 ( .I(b[1]), .ZN(n2) );
  INVD1BWP7T U5 ( .I(b[0]), .ZN(n1) );
  INVD1BWP7T U6 ( .I(b[6]), .ZN(n7) );
  INVD1BWP7T U7 ( .I(b[3]), .ZN(n4) );
  INVD1BWP7T U8 ( .I(b[4]), .ZN(n5) );
  INVD1BWP7T U9 ( .I(b[5]), .ZN(n6) );
  INVD1BWP7T U10 ( .I(b[10]), .ZN(n11) );
  INVD1BWP7T U11 ( .I(b[7]), .ZN(n8) );
  INVD1BWP7T U12 ( .I(b[8]), .ZN(n9) );
  INVD1BWP7T U13 ( .I(b[9]), .ZN(n10) );
  INVD1BWP7T U14 ( .I(b[14]), .ZN(n15) );
  INVD1BWP7T U15 ( .I(b[11]), .ZN(n12) );
  INVD1BWP7T U16 ( .I(b[12]), .ZN(n13) );
  INVD1BWP7T U17 ( .I(b[13]), .ZN(n14) );
  INVD1BWP7T U18 ( .I(b[15]), .ZN(n16) );
  TIELBWP7T U19 ( .ZN(\*Logic0* ) );
endmodule
```

```
module divisor_16bit_DW01_inc_0 ( A, SUM );
  input [31:0] A;
  output [31:0] SUM;

  wire [31:2] carry;

  HA1D0BWP7T U1_1_30 ( .A(A[30]), .B(carry[30]), .CO(carry[31]), .S(SUM[30])
        );
  HA1D0BWP7T U1_1_19 ( .A(A[19]), .B(carry[19]), .CO(carry[20]), .S(SUM[19])
        );
  HA1D0BWP7T U1_1_18 ( .A(A[18]), .B(carry[18]), .CO(carry[19]), .S(SUM[18])
        );
  HA1D0BWP7T U1_1_17 ( .A(A[17]), .B(carry[17]), .CO(carry[18]), .S(SUM[17])
        );
  HA1D0BWP7T U1_1_26 ( .A(A[26]), .B(carry[26]), .CO(carry[27]), .S(SUM[26])
        );
  HA1D0BWP7T U1_1_25 ( .A(A[25]), .B(carry[25]), .CO(carry[26]), .S(SUM[25])
        );
  HA1D0BWP7T U1_1_24 ( .A(A[24]), .B(carry[24]), .CO(carry[25]), .S(SUM[24])
        );
  HA1D0BWP7T U1_1_4 ( .A(A[4]), .B(carry[4]), .CO(carry[5]), .S(SUM[4]) );
  HA1D0BWP7T U1_1_12 ( .A(A[12]), .B(carry[12]), .CO(carry[13]), .S(SUM[12])
        );
  HA1D0BWP7T U1_1_10 ( .A(A[10]), .B(carry[10]), .CO(carry[11]), .S(SUM[10])
        );
  HA1D0BWP7T U1_1_11 ( .A(A[11]), .B(carry[11]), .CO(carry[12]), .S(SUM[11])
        );
  HA1D0BWP7T U1_1_1 ( .A(A[1]), .B(A[0]), .CO(carry[2]), .S(SUM[1]) );
  HA1D0BWP7T U1_1_29 ( .A(A[29]), .B(carry[29]), .CO(carry[30]), .S(SUM[29])
        );
  HA1D0BWP7T U1_1_28 ( .A(A[28]), .B(carry[28]), .CO(carry[29]), .S(SUM[28])
        );
  HA1D0BWP7T U1_1_27 ( .A(A[27]), .B(carry[27]), .CO(carry[28]), .S(SUM[27])
        );
  HA1D0BWP7T U1_1_23 ( .A(A[23]), .B(carry[23]), .CO(carry[24]), .S(SUM[23])
        );
  HA1D0BWP7T U1_1_22 ( .A(A[22]), .B(carry[22]), .CO(carry[23]), .S(SUM[22])
        );
  HA1D0BWP7T U1_1_21 ( .A(A[21]), .B(carry[21]), .CO(carry[22]), .S(SUM[21])
        );
  HA1D0BWP7T U1_1_20 ( .A(A[20]), .B(carry[20]), .CO(carry[21]), .S(SUM[20])
        );
  HA1D0BWP7T U1_1_16 ( .A(A[16]), .B(carry[16]), .CO(carry[17]), .S(SUM[16])
        );
  HA1D0BWP7T U1_1_15 ( .A(A[15]), .B(carry[15]), .CO(carry[16]), .S(SUM[15])
        );
  HA1D0BWP7T U1_1_14 ( .A(A[14]), .B(carry[14]), .CO(carry[15]), .S(SUM[14])
        );
  HA1D0BWP7T U1_1_13 ( .A(A[13]), .B(carry[13]), .CO(carry[14]), .S(SUM[13])
        );
  HA1D0BWP7T U1_1_9 ( .A(A[9]), .B(carry[9]), .CO(carry[10]), .S(SUM[9]) );
  HA1D0BWP7T U1_1_8 ( .A(A[8]), .B(carry[8]), .CO(carry[9]), .S(SUM[8]) );
  HA1D0BWP7T U1_1_7 ( .A(A[7]), .B(carry[7]), .CO(carry[8]), .S(SUM[7]) );
```

```
  HA1D0BWP7T U1_1_6 ( .A(A[6]), .B(carry[6]), .CO(carry[7]), .S(SUM[6]) );
  HA1D0BWP7T U1_1_5 ( .A(A[5]), .B(carry[5]), .CO(carry[6]), .S(SUM[5]) );
  HA1D0BWP7T U1_1_3 ( .A(A[3]), .B(carry[3]), .CO(carry[4]), .S(SUM[3]) );
  HA1D0BWP7T U1_1_2 ( .A(A[2]), .B(carry[2]), .CO(carry[3]), .S(SUM[2]) );
  CKXOR2D0BWP7T U1 ( .A1(carry[31]), .A2(A[31]), .Z(SUM[31]) );
endmodule

module divisor_16bit ( Divisor, Dividend, clk, reset, Quotient, Remainder,
       Sreg );
  input [15:0] Divisor;
  input [15:0] Dividend;
  output [15:0] Quotient;
  output [15:0] Remainder;
  output [31:0] Sreg;
  input clk, reset;
  wire \*Logic0* , F, N184, N185, N186, N187, N188, N189, N190, N191, N192,
        N193, N194, N195, N196, N197, N198, N199, N200, N201, N202, N203,
        N204, N205, N206, N207, N208, N209, N210, N211, N212, N213, N214,
        N215, N216, N217, N218, N219, N220, N221, N222, N223, N224, N225,
        N226, N227, N228, N229, N230, N231, N232, N233, N234, N235, N236,
        N237, N238, N239, N240, N241, N242, N243, N244, N245, N246, n179,
        n180, n181, n182, n183, n184, n185, n186, n187, n188, n189, n190,
        n191, n192, n193, n194, n196, n197, n198, n199, n200, n201, n202,
        n203, n204, n205, n206, n207, n208, n209, n210, n211, n212, n213,
        n214, n215, n216, n217, n218, n219, n220, n221, n222, n223, n224,
        n225, n226, n227, n228, n229, n230, n231, n233, n234, n235, n236,
        n237, n238, n239, n240, n241, n242, n243, n244, n245, n246, n247,
        n248, n249, n250, n251, n252, n253, n254, n255, n256, n257, n258,
        n259, n260, n261, n262, n263, n264, n265, n266, n267, n268, n269,
        n270, n271, n272, n273, n274, n275, n276, n277, n278, n279, n280,
        n281, n282, n283, n284, n285, n286, n287, n288, n289, n290, n291,
        n292, n293, n294, n295, n296, n297, n298, n299, n300, n301, n302,
        n303, n304, n305, n306, n307, n308, n309, n310, n311, n312, n313,
        n314, n315, n316, n317, n318, n319, n320, n321, n322, n323, n324,
        n325, n326, n327, n328, n329, n330, n331, n332, n333, n334, n335,
        n336, n337, n338, n339, n340, n341, n342, n343, n344, n345, n346,
        n347, n348, n349, n350, n351, n352, n353, n354, n355, n356, n357,
        n358, n359, n360, n361, n362, n363, n364, n365, n366, n367, n368,
        n369, n370, n371, n372, n373, n374, n375, n376, n377, n378, n379,
        n380, n381, n382, n383, n384, n385, n386, n387, n388, n389, n390,
        n391, n392, n393, n394, n395, n396, n397, n398, n399, n400, n401,
        n402, n403, n404, n405, n406, n407, n408, n409, n410, n411, n412,
        n413, n414, n415, n416, n417, n418, n419, n420, n421, n422, n423,
        n424, n425, n426, n427, n428, n429, n430, n431, n432, n433, n434,
        n435, n436, n437, n438, n439, n440, n441, n442, n443, n444, n445,
        n446, n447, n448, n449, n450, n451, n452, n453, n454, n455, n456,
        n457, n458, n459, n460, n461, n462, n463, n464, n465, n466, n467,
        n468, n469, n470, n471, n472, n473, n474, n475, n476, n477, n478,
        n479, n480, n481, n482, n483, n484, n485;
  wire [15:0] Divisor_temp;
  wire [15:0] S;
  wire [31:0] i;
  wire [15:0] quo;
  wire SYNOPSYS_UNCONNECTED__0;
```

```
    DFQD1BWP7T \Divisor_temp_reg[0] ( .D(n194), .CP(clk), .Q(Divisor_temp[0])
          );
    DFQD1BWP7T \Divisor_temp_reg[1] ( .D(n193), .CP(clk), .Q(Divisor_temp[1])
          );
    DFQD1BWP7T \Divisor_temp_reg[2] ( .D(n192), .CP(clk), .Q(Divisor_temp[2])
          );
    DFQD1BWP7T \Divisor_temp_reg[3] ( .D(n191), .CP(clk), .Q(Divisor_temp[3])
          );
    DFQD1BWP7T \Divisor_temp_reg[4] ( .D(n190), .CP(clk), .Q(Divisor_temp[4])
          );
    DFQD1BWP7T \Divisor_temp_reg[5] ( .D(n189), .CP(clk), .Q(Divisor_temp[5])
          );
    DFQD1BWP7T \Divisor_temp_reg[6] ( .D(n188), .CP(clk), .Q(Divisor_temp[6])
          );
    DFQD1BWP7T \Divisor_temp_reg[7] ( .D(n187), .CP(clk), .Q(Divisor_temp[7])
          );
    DFQD1BWP7T \Divisor_temp_reg[8] ( .D(n186), .CP(clk), .Q(Divisor_temp[8])
          );
    DFQD1BWP7T \Divisor_temp_reg[9] ( .D(n185), .CP(clk), .Q(Divisor_temp[9])
          );
    DFQD1BWP7T \Divisor_temp_reg[10] ( .D(n184), .CP(clk), .Q(Divisor_temp[10])
          );
    DFQD1BWP7T \Divisor_temp_reg[11] ( .D(n183), .CP(clk), .Q(Divisor_temp[11])
          );
    DFQD1BWP7T \Divisor_temp_reg[12] ( .D(n182), .CP(clk), .Q(Divisor_temp[12])
          );
    DFQD1BWP7T \Divisor_temp_reg[13] ( .D(n181), .CP(clk), .Q(Divisor_temp[13])
          );
    DFQD1BWP7T \Divisor_temp_reg[14] ( .D(n180), .CP(clk), .Q(Divisor_temp[14])
          );
    DFQD1BWP7T \Divisor_temp_reg[15] ( .D(n179), .CP(clk), .Q(Divisor_temp[15])
          );
    DFQD1BWP7T \Sreg_reg[1] ( .D(n295), .CP(clk), .Q(Sreg[1]) );
    DFQD1BWP7T \Sreg_reg[2] ( .D(n294), .CP(clk), .Q(Sreg[2]) );
    DFQD1BWP7T \Sreg_reg[3] ( .D(n293), .CP(clk), .Q(Sreg[3]) );
    DFQD1BWP7T \Sreg_reg[4] ( .D(n292), .CP(clk), .Q(Sreg[4]) );
    DFQD1BWP7T \Sreg_reg[5] ( .D(n291), .CP(clk), .Q(Sreg[5]) );
    DFQD1BWP7T \Sreg_reg[6] ( .D(n290), .CP(clk), .Q(Sreg[6]) );
    DFQD1BWP7T \Sreg_reg[7] ( .D(n289), .CP(clk), .Q(Sreg[7]) );
    DFQD1BWP7T \Sreg_reg[8] ( .D(n288), .CP(clk), .Q(Sreg[8]) );
    DFQD1BWP7T \Sreg_reg[9] ( .D(n287), .CP(clk), .Q(Sreg[9]) );
    DFQD1BWP7T \Sreg_reg[10] ( .D(n286), .CP(clk), .Q(Sreg[10]) );
    DFQD1BWP7T \Sreg_reg[11] ( .D(n285), .CP(clk), .Q(Sreg[11]) );
    DFQD1BWP7T \Sreg_reg[12] ( .D(n284), .CP(clk), .Q(Sreg[12]) );
    DFQD1BWP7T \Sreg_reg[13] ( .D(n283), .CP(clk), .Q(Sreg[13]) );
    DFQD1BWP7T \Sreg_reg[14] ( .D(n282), .CP(clk), .Q(Sreg[14]) );
    DFQD1BWP7T \Sreg_reg[15] ( .D(n281), .CP(clk), .Q(Sreg[15]) );
    DFQD1BWP7T \Sreg_reg[16] ( .D(n311), .CP(clk), .Q(Sreg[16]) );
    DFQD1BWP7T \Sreg_reg[17] ( .D(n297), .CP(clk), .Q(Sreg[17]) );
    DFQD1BWP7T \Sreg_reg[18] ( .D(n298), .CP(clk), .Q(Sreg[18]) );
    DFQD1BWP7T \Sreg_reg[19] ( .D(n299), .CP(clk), .Q(Sreg[19]) );
    DFQD1BWP7T \Sreg_reg[20] ( .D(n300), .CP(clk), .Q(Sreg[20]) );
    DFQD1BWP7T \Sreg_reg[21] ( .D(n301), .CP(clk), .Q(Sreg[21]) );
```

```
    DFQD1BWP7T \Sreg_reg[22] ( .D(n302), .CP(clk), .Q(Sreg[22]) );
    DFQD1BWP7T \Sreg_reg[23] ( .D(n303), .CP(clk), .Q(Sreg[23]) );
    DFQD1BWP7T \Sreg_reg[24] ( .D(n304), .CP(clk), .Q(Sreg[24]) );
    DFQD1BWP7T \Sreg_reg[25] ( .D(n305), .CP(clk), .Q(Sreg[25]) );
    DFQD1BWP7T \Sreg_reg[26] ( .D(n306), .CP(clk), .Q(Sreg[26]) );
    DFQD1BWP7T \Sreg_reg[27] ( .D(n307), .CP(clk), .Q(Sreg[27]) );
    DFQD1BWP7T \Sreg_reg[28] ( .D(n308), .CP(clk), .Q(Sreg[28]) );
    DFQD1BWP7T \Sreg_reg[29] ( .D(n309), .CP(clk), .Q(Sreg[29]) );
    DFQD1BWP7T \Sreg_reg[30] ( .D(n310), .CP(clk), .Q(Sreg[30]) );
    DFQD1BWP7T \Sreg_reg[31] ( .D(n312), .CP(clk), .Q(Sreg[31]) );
    DFQD1BWP7T \quo_reg[0] ( .D(n280), .CP(clk), .Q(quo[0]) );
    DFQD1BWP7T \quo_reg[1] ( .D(n278), .CP(clk), .Q(quo[1]) );
    DFQD1BWP7T \quo_reg[2] ( .D(n276), .CP(clk), .Q(quo[2]) );
    DFQD1BWP7T \quo_reg[3] ( .D(n274), .CP(clk), .Q(quo[3]) );
    DFQD1BWP7T \quo_reg[4] ( .D(n272), .CP(clk), .Q(quo[4]) );
    DFQD1BWP7T \quo_reg[5] ( .D(n270), .CP(clk), .Q(quo[5]) );
    DFQD1BWP7T \quo_reg[6] ( .D(n268), .CP(clk), .Q(quo[6]) );
    DFQD1BWP7T \quo_reg[7] ( .D(n266), .CP(clk), .Q(quo[7]) );
    DFQD1BWP7T \quo_reg[8] ( .D(n264), .CP(clk), .Q(quo[8]) );
    DFQD1BWP7T \quo_reg[9] ( .D(n262), .CP(clk), .Q(quo[9]) );
    DFQD1BWP7T \quo_reg[10] ( .D(n260), .CP(clk), .Q(quo[10]) );
    DFQD1BWP7T \quo_reg[11] ( .D(n258), .CP(clk), .Q(quo[11]) );
    DFQD1BWP7T \quo_reg[12] ( .D(n256), .CP(clk), .Q(quo[12]) );
    DFQD1BWP7T \quo_reg[13] ( .D(n254), .CP(clk), .Q(quo[13]) );
    DFQD1BWP7T \quo_reg[14] ( .D(n252), .CP(clk), .Q(quo[14]) );
    DFQD1BWP7T \quo_reg[15] ( .D(n250), .CP(clk), .Q(quo[15]) );
    DFQD1BWP7T \Quotient_reg[15] ( .D(n249), .CP(clk), .Q(Quotient[15]) );
    DFQD1BWP7T \Quotient_reg[14] ( .D(n251), .CP(clk), .Q(Quotient[14]) );
    DFQD1BWP7T \Quotient_reg[13] ( .D(n253), .CP(clk), .Q(Quotient[13]) );
    DFQD1BWP7T \Quotient_reg[12] ( .D(n255), .CP(clk), .Q(Quotient[12]) );
    DFQD1BWP7T \Quotient_reg[11] ( .D(n257), .CP(clk), .Q(Quotient[11]) );
    DFQD1BWP7T \Quotient_reg[10] ( .D(n259), .CP(clk), .Q(Quotient[10]) );
    DFQD1BWP7T \Quotient_reg[9] ( .D(n261), .CP(clk), .Q(Quotient[9]) );
    DFQD1BWP7T \Quotient_reg[8] ( .D(n263), .CP(clk), .Q(Quotient[8]) );
    DFQD1BWP7T \Quotient_reg[7] ( .D(n265), .CP(clk), .Q(Quotient[7]) );
    DFQD1BWP7T \Quotient_reg[6] ( .D(n267), .CP(clk), .Q(Quotient[6]) );
    DFQD1BWP7T \Quotient_reg[5] ( .D(n269), .CP(clk), .Q(Quotient[5]) );
    DFQD1BWP7T \Quotient_reg[4] ( .D(n271), .CP(clk), .Q(Quotient[4]) );
    DFQD1BWP7T \Quotient_reg[3] ( .D(n273), .CP(clk), .Q(Quotient[3]) );
    DFQD1BWP7T \Quotient_reg[2] ( .D(n275), .CP(clk), .Q(Quotient[2]) );
    DFQD1BWP7T \Quotient_reg[1] ( .D(n277), .CP(clk), .Q(Quotient[1]) );
    DFQD1BWP7T \Quotient_reg[0] ( .D(n279), .CP(clk), .Q(Quotient[0]) );
    DFQD1BWP7T \Remainder_reg[15] ( .D(n233), .CP(clk), .Q(Remainder[15]) );
    substractor_16bit U36 ( .a(Sreg[31:16]), .b(Divisor_temp), .cin(\*Logic0* ),
         .S(S) );
    divisor_16bit_DW01_inc_0 add_169 ( .A(i), .SUM({N214, N213, N212, N211, N210,
         N209, N208, N207, N206, N205, N204, N203, N202, N201, N200, N199, N198,
         N197, N196, N195, N194, N193, N192, N191, N190, N189, N188, N187, N186,
         N185, N184, SYNOPSYS_UNCONNECTED__0}) );
    DFD1BWP7T \Remainder_reg[9] ( .D(n239), .CP(clk), .Q(Remainder[9]), .QN(
         n202) );
    DFD1BWP7T \Remainder_reg[8] ( .D(n240), .CP(clk), .Q(Remainder[8]), .QN(
         n203) );
    DFD1BWP7T \Remainder_reg[7] ( .D(n241), .CP(clk), .Q(Remainder[7]), .QN(
```

```
      n204) );
    DFD1BWP7T \Remainder_reg[6] ( .D(n242), .CP(clk), .Q(Remainder[6]), .QN(
      n205) );
    DFD1BWP7T \Remainder_reg[5] ( .D(n243), .CP(clk), .Q(Remainder[5]), .QN(
      n206) );
    DFD1BWP7T \Remainder_reg[4] ( .D(n244), .CP(clk), .Q(Remainder[4]), .QN(
      n207) );
    DFD1BWP7T \Remainder_reg[3] ( .D(n245), .CP(clk), .Q(Remainder[3]), .QN(
      n208) );
    DFD1BWP7T \Remainder_reg[2] ( .D(n246), .CP(clk), .Q(Remainder[2]), .QN(
      n209) );
    DFD1BWP7T \Remainder_reg[1] ( .D(n247), .CP(clk), .Q(Remainder[1]), .QN(
      n210) );
    DFD1BWP7T \Remainder_reg[0] ( .D(n248), .CP(clk), .Q(Remainder[0]), .QN(
      n211) );
    DFD1BWP7T \Remainder_reg[14] ( .D(n234), .CP(clk), .Q(Remainder[14]), .QN(
      n197) );
    DFD1BWP7T \Remainder_reg[13] ( .D(n235), .CP(clk), .Q(Remainder[13]), .QN(
      n198) );
    DFD1BWP7T \Remainder_reg[12] ( .D(n236), .CP(clk), .Q(Remainder[12]), .QN(
      n199) );
    DFD1BWP7T \Remainder_reg[11] ( .D(n237), .CP(clk), .Q(Remainder[11]), .QN(
      n200) );
    DFD1BWP7T \Remainder_reg[10] ( .D(n238), .CP(clk), .Q(Remainder[10]), .QN(
      n201) );
    DFCNQD1BWP7T F_reg ( .D(n485), .CP(clk), .CDN(n484), .Q(F) );
    DFCNQD1BWP7T \i_reg[25] ( .D(n318), .CP(clk), .CDN(n484), .Q(i[25]) );
    DFCNQD1BWP7T \i_reg[26] ( .D(n317), .CP(clk), .CDN(n484), .Q(i[26]) );
    DFCNQD1BWP7T \i_reg[24] ( .D(n319), .CP(clk), .CDN(n484), .Q(i[24]) );
    DFCNQD1BWP7T \i_reg[19] ( .D(n324), .CP(clk), .CDN(n484), .Q(i[19]) );
    DFCNQD1BWP7T \i_reg[18] ( .D(n325), .CP(clk), .CDN(n484), .Q(i[18]) );
    DFCNQD1BWP7T \i_reg[17] ( .D(n326), .CP(clk), .CDN(n484), .Q(i[17]) );
    DFCNQD1BWP7T \i_reg[12] ( .D(n331), .CP(clk), .CDN(n484), .Q(i[12]) );
    DFCNQD1BWP7T \i_reg[11] ( .D(n332), .CP(clk), .CDN(n484), .Q(i[11]) );
    DFCNQD1BWP7T \i_reg[10] ( .D(n333), .CP(clk), .CDN(n484), .Q(i[10]) );
    DFCNQD1BWP7T \i_reg[4] ( .D(n339), .CP(clk), .CDN(n484), .Q(i[4]) );
    DFCNQD1BWP7T \i_reg[0] ( .D(n344), .CP(clk), .CDN(n484), .Q(i[0]) );
    DFCNQD1BWP7T \i_reg[1] ( .D(n342), .CP(clk), .CDN(n484), .Q(i[1]) );
    DFD0BWP7T \Sreg_reg[0] ( .D(n296), .CP(clk), .Q(Sreg[0]), .QN(n196) );
    DFCND0BWP7T \i_reg[30] ( .D(n313), .CP(clk), .CDN(n484), .Q(i[30]), .QN(
      n213) );
    DFCND0BWP7T \i_reg[29] ( .D(n314), .CP(clk), .CDN(n484), .Q(i[29]), .QN(
      n214) );
    DFCND0BWP7T \i_reg[28] ( .D(n315), .CP(clk), .CDN(n484), .Q(i[28]), .QN(
      n215) );
    DFCND0BWP7T \i_reg[27] ( .D(n316), .CP(clk), .CDN(n484), .Q(i[27]), .QN(
      n216) );
    DFCND0BWP7T \i_reg[23] ( .D(n320), .CP(clk), .CDN(n484), .Q(i[23]), .QN(
      n217) );
    DFCND0BWP7T \i_reg[22] ( .D(n321), .CP(clk), .CDN(n484), .Q(i[22]), .QN(
      n218) );
    DFCND0BWP7T \i_reg[21] ( .D(n322), .CP(clk), .CDN(n484), .Q(i[21]), .QN(
      n219) );
    DFCND0BWP7T \i_reg[20] ( .D(n323), .CP(clk), .CDN(n484), .Q(i[20]), .QN(
```

```
      n220) );
    DFCND0BWP7T \i_reg[16] ( .D(n327), .CP(clk), .CDN(n484), .Q(i[16]), .QN(
      n221) );
    DFCND0BWP7T \i_reg[15] ( .D(n328), .CP(clk), .CDN(n484), .Q(i[15]), .QN(
      n222) );
    DFCND0BWP7T \i_reg[31] ( .D(n343), .CP(clk), .CDN(n484), .Q(i[31]), .QN(
      n212) );
    DFCND0BWP7T \i_reg[14] ( .D(n329), .CP(clk), .CDN(n484), .Q(i[14]), .QN(
      n223) );
    DFCND0BWP7T \i_reg[13] ( .D(n330), .CP(clk), .CDN(n484), .Q(i[13]), .QN(
      n224) );
    DFCND0BWP7T \i_reg[9] ( .D(n334), .CP(clk), .CDN(n484), .Q(i[9]), .QN(n225)
      );
    DFCND0BWP7T \i_reg[8] ( .D(n335), .CP(clk), .CDN(n484), .Q(i[8]), .QN(n226)
      );
    DFCND0BWP7T \i_reg[7] ( .D(n336), .CP(clk), .CDN(n484), .Q(i[7]), .QN(n227)
      );
    DFCND0BWP7T \i_reg[6] ( .D(n337), .CP(clk), .CDN(n484), .Q(i[6]), .QN(n228)
      );
    DFCND0BWP7T \i_reg[5] ( .D(n338), .CP(clk), .CDN(n484), .Q(i[5]), .QN(n229)
      );
    DFCND0BWP7T \i_reg[3] ( .D(n340), .CP(clk), .CDN(n484), .Q(i[3]), .QN(n230)
      );
    DFCND0BWP7T \i_reg[2] ( .D(n341), .CP(clk), .CDN(n484), .Q(i[2]), .QN(n231)
      );
    NR4D0BWP7T U333 ( .A1(i[9]), .A2(n480), .A3(n481), .A4(i[13]), .ZN(n479) );
    NR4D0BWP7T U334 ( .A1(i[14]), .A2(i[22]), .A3(n470), .A4(i[13]), .ZN(n469)
      );
    NR4D0BWP7T U335 ( .A1(i[21]), .A2(i[19]), .A3(i[18]), .A4(i[17]), .ZN(n483)
      );
    NR4D0BWP7T U336 ( .A1(i[6]), .A2(i[5]), .A3(n474), .A4(n463), .ZN(n473) );
    BUFFD1BWP7T U337 ( .I(n403), .Z(n347) );
    INVD1BWP7T U338 ( .I(reset), .ZN(n484) );
    AO31D1BWP7T U339 ( .A1(n357), .A2(n356), .A3(n355), .B(N215), .Z(n345) );
    ND2D1BWP7T U340 ( .A1(n456), .A2(n441), .ZN(n454) );
    NR2D1BWP7T U341 ( .A1(n446), .A2(n442), .ZN(n445) );
    NR2D1BWP7T U342 ( .A1(n440), .A2(n442), .ZN(n444) );
    ND2D1BWP7T U343 ( .A1(n439), .A2(n404), .ZN(n409) );
    ND2D1BWP7T U344 ( .A1(n440), .A2(n404), .ZN(n406) );
    CKAN2D1BWP7T U345 ( .A1(n452), .A2(n484), .Z(n450) );
    INVD1BWP7T U346 ( .I(n457), .ZN(n456) );
    CKBD1BWP7T U347 ( .I(n358), .Z(n346) );
    OAI21D0BWP7T U348 ( .A1(reset), .A2(n441), .B(n442), .ZN(n404) );
    ND3D0BWP7T U349 ( .A1(n460), .A2(n461), .A3(n467), .ZN(n447) );
    AN3D1BWP7T U350 ( .A1(i[1]), .A2(n360), .A3(n453), .Z(n451) );
    TIELBWP7T U351 ( .ZN(\*Logic0* ) );
    INVD1BWP7T U352 ( .I(n443), .ZN(n442) );
    ND2D1BWP7T U353 ( .A1(n456), .A2(n439), .ZN(n455) );
    AO211D0BWP7T U354 ( .A1(N245), .A2(N246), .B(N243), .C(N244), .Z(n348) );
    AO211D0BWP7T U355 ( .A1(N242), .A2(n348), .B(N235), .C(N236), .Z(n349) );
    NR4D0BWP7T U356 ( .A1(n349), .A2(N234), .A3(N232), .A4(N233), .ZN(n357) );
    NR4D0BWP7T U357 ( .A1(N225), .A2(N226), .A3(N227), .A4(N228), .ZN(n350) );
    INR4D0BWP7T U358 ( .A1(n350), .B1(N229), .B2(N231), .B3(N230), .ZN(n356) );
    NR3D0BWP7T U359 ( .A1(N224), .A2(N222), .A3(N223), .ZN(n354) );
```

```
    NR4D0BWP7T U360 ( .A1(N218), .A2(N219), .A3(N220), .A4(N221), .ZN(n353) );
    NR3D0BWP7T U361 ( .A1(N217), .A2(N241), .A3(N216), .ZN(n352) );
    NR4D0BWP7T U362 ( .A1(N237), .A2(N238), .A3(N239), .A4(N240), .ZN(n351) );
    AN4D0BWP7T U363 ( .A1(n354), .A2(n353), .A3(n352), .A4(n351), .Z(n355) );
    MUX2ND0BWP7T U364 ( .I0(n346), .I1(n359), .S(n360), .ZN(n344) );
    OAI22D0BWP7T U365 ( .A1(n212), .A2(n358), .B1(n359), .B2(n361), .ZN(n343) );
    OAI22D0BWP7T U366 ( .A1(n362), .A2(n358), .B1(n359), .B2(n363), .ZN(n342) );
    OAI22D0BWP7T U367 ( .A1(n231), .A2(n346), .B1(n359), .B2(n364), .ZN(n341) );
    OAI22D0BWP7T U368 ( .A1(n230), .A2(n358), .B1(n359), .B2(n365), .ZN(n340) );
    OAI22D0BWP7T U369 ( .A1(n366), .A2(n358), .B1(n359), .B2(n367), .ZN(n339) );
    OAI22D0BWP7T U370 ( .A1(n229), .A2(n358), .B1(n359), .B2(n368), .ZN(n338) );
    OAI22D0BWP7T U371 ( .A1(n228), .A2(n358), .B1(n359), .B2(n369), .ZN(n337) );
    OAI22D0BWP7T U372 ( .A1(n227), .A2(n346), .B1(n359), .B2(n370), .ZN(n336) );
    OAI22D0BWP7T U373 ( .A1(n226), .A2(n358), .B1(n359), .B2(n371), .ZN(n335) );
    OAI22D0BWP7T U374 ( .A1(n225), .A2(n346), .B1(n359), .B2(n372), .ZN(n334) );
    OAI22D0BWP7T U375 ( .A1(n346), .A2(n373), .B1(n359), .B2(n374), .ZN(n333) );
    OAI22D0BWP7T U376 ( .A1(n346), .A2(n375), .B1(n359), .B2(n376), .ZN(n332) );
    OAI22D0BWP7T U377 ( .A1(n358), .A2(n377), .B1(n359), .B2(n378), .ZN(n331) );
    OAI22D0BWP7T U378 ( .A1(n224), .A2(n346), .B1(n359), .B2(n379), .ZN(n330) );
    OAI22D0BWP7T U379 ( .A1(n223), .A2(n346), .B1(n359), .B2(n380), .ZN(n329) );
    OAI22D0BWP7T U380 ( .A1(n222), .A2(n346), .B1(n359), .B2(n381), .ZN(n328) );
    OAI22D0BWP7T U381 ( .A1(n221), .A2(n346), .B1(n359), .B2(n382), .ZN(n327) );
    OAI22D0BWP7T U382 ( .A1(n346), .A2(n383), .B1(n359), .B2(n384), .ZN(n326) );
    OAI22D0BWP7T U383 ( .A1(n358), .A2(n385), .B1(n359), .B2(n386), .ZN(n325) );
    OAI22D0BWP7T U384 ( .A1(n358), .A2(n387), .B1(n359), .B2(n388), .ZN(n324) );
    OAI22D0BWP7T U385 ( .A1(n220), .A2(n346), .B1(n359), .B2(n389), .ZN(n323) );
    OAI22D0BWP7T U386 ( .A1(n219), .A2(n346), .B1(n359), .B2(n390), .ZN(n322) );
    OAI22D0BWP7T U387 ( .A1(n218), .A2(n346), .B1(n359), .B2(n391), .ZN(n321) );
    OAI22D0BWP7T U388 ( .A1(n217), .A2(n346), .B1(n359), .B2(n392), .ZN(n320) );
    OAI22D0BWP7T U389 ( .A1(n358), .A2(n393), .B1(n359), .B2(n394), .ZN(n319) );
    OAI22D0BWP7T U390 ( .A1(n358), .A2(n395), .B1(n359), .B2(n396), .ZN(n318) );
    OAI22D0BWP7T U391 ( .A1(n358), .A2(n397), .B1(n359), .B2(n398), .ZN(n317) );
    OAI22D0BWP7T U392 ( .A1(n216), .A2(n346), .B1(n359), .B2(n399), .ZN(n316) );
    OAI22D0BWP7T U393 ( .A1(n215), .A2(n346), .B1(n359), .B2(n400), .ZN(n315) );
    OAI22D0BWP7T U394 ( .A1(n214), .A2(n346), .B1(n359), .B2(n401), .ZN(n314) );
    OAI22D0BWP7T U395 ( .A1(n213), .A2(n346), .B1(n359), .B2(n402), .ZN(n313) );
    CKND2D0BWP7T U396 ( .A1(n345), .A2(n358), .ZN(n359) );
    OAI21D0BWP7T U397 ( .A1(F), .A2(n345), .B(n347), .ZN(n358) );
    OAI22D0BWP7T U398 ( .A1(n404), .A2(n405), .B1(n406), .B2(n407), .ZN(n312) );
    OAI222D0BWP7T U399 ( .A1(n406), .A2(n408), .B1(n409), .B2(n410), .C1(n404),
         .C2(n411), .ZN(n311) );
    CKND0BWP7T U400 ( .I(Sreg[15]), .ZN(n408) );
    OAI222D0BWP7T U401 ( .A1(n406), .A2(n412), .B1(n409), .B2(n413), .C1(n407),
         .C2(n404), .ZN(n310) );
    OAI222D0BWP7T U402 ( .A1(n406), .A2(n414), .B1(n409), .B2(n415), .C1(n404),
         .C2(n412), .ZN(n309) );
    OAI222D0BWP7T U403 ( .A1(n406), .A2(n416), .B1(n409), .B2(n417), .C1(n404),
         .C2(n414), .ZN(n308) );
    OAI222D0BWP7T U404 ( .A1(n406), .A2(n418), .B1(n409), .B2(n419), .C1(n404),
         .C2(n416), .ZN(n307) );
    OAI222D0BWP7T U405 ( .A1(n406), .A2(n420), .B1(n409), .B2(n421), .C1(n404),
         .C2(n418), .ZN(n306) );
    OAI222D0BWP7T U406 ( .A1(n406), .A2(n422), .B1(n409), .B2(n423), .C1(n404),
         .C2(n420), .ZN(n305) );
```

```
    OAI222D0BWP7T U407 ( .A1(n406), .A2(n424), .B1(n409), .B2(n425), .C1(n404),
         .C2(n422), .ZN(n304) );
    OAI222D0BWP7T U408 ( .A1(n406), .A2(n426), .B1(n409), .B2(n427), .C1(n404),
         .C2(n424), .ZN(n303) );
    OAI222D0BWP7T U409 ( .A1(n406), .A2(n428), .B1(n409), .B2(n429), .C1(n404),
         .C2(n426), .ZN(n302) );
    OAI222D0BWP7T U410 ( .A1(n406), .A2(n430), .B1(n409), .B2(n431), .C1(n404),
         .C2(n428), .ZN(n301) );
    OAI222D0BWP7T U411 ( .A1(n406), .A2(n432), .B1(n409), .B2(n433), .C1(n404),
         .C2(n430), .ZN(n300) );
    OAI222D0BWP7T U412 ( .A1(n406), .A2(n434), .B1(n409), .B2(n435), .C1(n404),
         .C2(n432), .ZN(n299) );
    OAI222D0BWP7T U413 ( .A1(n406), .A2(n436), .B1(n409), .B2(n437), .C1(n404),
         .C2(n434), .ZN(n298) );
    OAI222D0BWP7T U414 ( .A1(n406), .A2(n411), .B1(n409), .B2(n438), .C1(n404),
         .C2(n436), .ZN(n297) );
    MOAI22D0BWP7T U415 ( .A1(n196), .A2(n443), .B1(Dividend[0]), .B2(n444), .ZN(
        n296) );
    AO222D0BWP7T U416 ( .A1(Dividend[1]), .A2(n444), .B1(n445), .B2(Sreg[0]),
         .C1(Sreg[1]), .C2(n442), .Z(n295) );
    AO222D0BWP7T U417 ( .A1(Dividend[2]), .A2(n444), .B1(Sreg[1]), .B2(n445),
         .C1(Sreg[2]), .C2(n442), .Z(n294) );
    AO222D0BWP7T U418 ( .A1(Dividend[3]), .A2(n444), .B1(Sreg[2]), .B2(n445),
         .C1(Sreg[3]), .C2(n442), .Z(n293) );
    AO222D0BWP7T U419 ( .A1(Dividend[4]), .A2(n444), .B1(Sreg[3]), .B2(n445),
         .C1(Sreg[4]), .C2(n442), .Z(n292) );
    AO222D0BWP7T U420 ( .A1(Dividend[5]), .A2(n444), .B1(Sreg[4]), .B2(n445),
         .C1(Sreg[5]), .C2(n442), .Z(n291) );
    AO222D0BWP7T U421 ( .A1(Dividend[6]), .A2(n444), .B1(Sreg[5]), .B2(n445),
         .C1(Sreg[6]), .C2(n442), .Z(n290) );
    AO222D0BWP7T U422 ( .A1(Dividend[7]), .A2(n444), .B1(Sreg[6]), .B2(n445),
         .C1(Sreg[7]), .C2(n442), .Z(n289) );
    AO222D0BWP7T U423 ( .A1(Dividend[8]), .A2(n444), .B1(Sreg[7]), .B2(n445),
         .C1(Sreg[8]), .C2(n442), .Z(n288) );
    AO222D0BWP7T U424 ( .A1(Dividend[9]), .A2(n444), .B1(Sreg[8]), .B2(n445),
         .C1(Sreg[9]), .C2(n442), .Z(n287) );
    AO222D0BWP7T U425 ( .A1(Dividend[10]), .A2(n444), .B1(Sreg[9]), .B2(n445),
         .C1(Sreg[10]), .C2(n442), .Z(n286) );
    AO222D0BWP7T U426 ( .A1(Dividend[11]), .A2(n444), .B1(Sreg[10]), .B2(n445),
         .C1(Sreg[11]), .C2(n442), .Z(n285) );
    AO222D0BWP7T U427 ( .A1(Dividend[12]), .A2(n444), .B1(Sreg[11]), .B2(n445),
         .C1(Sreg[12]), .C2(n442), .Z(n284) );
    AO222D0BWP7T U428 ( .A1(Dividend[13]), .A2(n444), .B1(Sreg[12]), .B2(n445),
         .C1(Sreg[13]), .C2(n442), .Z(n283) );
    AO222D0BWP7T U429 ( .A1(Dividend[14]), .A2(n444), .B1(Sreg[13]), .B2(n445),
         .C1(Sreg[14]), .C2(n442), .Z(n282) );
    AO222D0BWP7T U430 ( .A1(Dividend[15]), .A2(n444), .B1(Sreg[14]), .B2(n445),
         .C1(Sreg[15]), .C2(n442), .Z(n281) );
    OAI21D0BWP7T U431 ( .A1(reset), .A2(n446), .B(n447), .ZN(n443) );
    CKND0BWP7T U432 ( .I(n446), .ZN(n440) );
    CKND2D0BWP7T U433 ( .A1(n448), .A2(n485), .ZN(n446) );
    MUX2ND0BWP7T U434 ( .I0(n449), .I1(S[15]), .S(n450), .ZN(n280) );
    CKND0BWP7T U435 ( .I(quo[0]), .ZN(n449) );
    CKMUX2D0BWP7T U436 ( .I0(Quotient[0]), .I1(quo[0]), .S(n451), .Z(n279) );
```

```
    CKMUX2D0BWP7T U437 ( .I0(quo[1]), .I1(quo[0]), .S(n450), .Z(n278) );
    CKMUX2D0BWP7T U438 ( .I0(Quotient[1]), .I1(quo[1]), .S(n451), .Z(n277) );
    CKMUX2D0BWP7T U439 ( .I0(quo[2]), .I1(quo[1]), .S(n450), .Z(n276) );
    CKMUX2D0BWP7T U440 ( .I0(Quotient[2]), .I1(quo[2]), .S(n451), .Z(n275) );
    CKMUX2D0BWP7T U441 ( .I0(quo[3]), .I1(quo[2]), .S(n450), .Z(n274) );
    CKMUX2D0BWP7T U442 ( .I0(Quotient[3]), .I1(quo[3]), .S(n451), .Z(n273) );
    CKMUX2D0BWP7T U443 ( .I0(quo[4]), .I1(quo[3]), .S(n450), .Z(n272) );
    CKMUX2D0BWP7T U444 ( .I0(Quotient[4]), .I1(quo[4]), .S(n451), .Z(n271) );
    CKMUX2D0BWP7T U445 ( .I0(quo[5]), .I1(quo[4]), .S(n450), .Z(n270) );
    CKMUX2D0BWP7T U446 ( .I0(Quotient[5]), .I1(quo[5]), .S(n451), .Z(n269) );
    CKMUX2D0BWP7T U447 ( .I0(quo[6]), .I1(quo[5]), .S(n450), .Z(n268) );
    CKMUX2D0BWP7T U448 ( .I0(Quotient[6]), .I1(quo[6]), .S(n451), .Z(n267) );
    CKMUX2D0BWP7T U449 ( .I0(quo[7]), .I1(quo[6]), .S(n450), .Z(n266) );
    CKMUX2D0BWP7T U450 ( .I0(Quotient[7]), .I1(quo[7]), .S(n451), .Z(n265) );
    CKMUX2D0BWP7T U451 ( .I0(quo[8]), .I1(quo[7]), .S(n450), .Z(n264) );
    CKMUX2D0BWP7T U452 ( .I0(Quotient[8]), .I1(quo[8]), .S(n451), .Z(n263) );
    CKMUX2D0BWP7T U453 ( .I0(quo[9]), .I1(quo[8]), .S(n450), .Z(n262) );
    CKMUX2D0BWP7T U454 ( .I0(Quotient[9]), .I1(quo[9]), .S(n451), .Z(n261) );
    CKMUX2D0BWP7T U455 ( .I0(quo[10]), .I1(quo[9]), .S(n450), .Z(n260) );
    CKMUX2D0BWP7T U456 ( .I0(Quotient[10]), .I1(quo[10]), .S(n451), .Z(n259) );
    CKMUX2D0BWP7T U457 ( .I0(quo[11]), .I1(quo[10]), .S(n450), .Z(n258) );
    CKMUX2D0BWP7T U458 ( .I0(Quotient[11]), .I1(quo[11]), .S(n451), .Z(n257) );
    CKMUX2D0BWP7T U459 ( .I0(quo[12]), .I1(quo[11]), .S(n450), .Z(n256) );
    CKMUX2D0BWP7T U460 ( .I0(Quotient[12]), .I1(quo[12]), .S(n451), .Z(n255) );
    CKMUX2D0BWP7T U461 ( .I0(quo[13]), .I1(quo[12]), .S(n450), .Z(n254) );
    CKMUX2D0BWP7T U462 ( .I0(Quotient[13]), .I1(quo[13]), .S(n451), .Z(n253) );
    CKMUX2D0BWP7T U463 ( .I0(quo[14]), .I1(quo[13]), .S(n450), .Z(n252) );
    CKMUX2D0BWP7T U464 ( .I0(Quotient[14]), .I1(quo[14]), .S(n451), .Z(n251) );
    CKMUX2D0BWP7T U465 ( .I0(quo[15]), .I1(quo[14]), .S(n450), .Z(n250) );
    CKMUX2D0BWP7T U466 ( .I0(Quotient[15]), .I1(quo[15]), .S(n451), .Z(n249) );
    OAI222D0BWP7T U467 ( .A1(n411), .A2(n454), .B1(n410), .B2(n455), .C1(n211),
         .C2(n456), .ZN(n248) );
    CKND0BWP7T U468 ( .I(S[0]), .ZN(n410) );
    CKND0BWP7T U469 ( .I(Sreg[16]), .ZN(n411) );
    OAI222D0BWP7T U470 ( .A1(n436), .A2(n454), .B1(n438), .B2(n455), .C1(n210),
         .C2(n456), .ZN(n247) );
    CKND0BWP7T U471 ( .I(S[1]), .ZN(n438) );
    CKND0BWP7T U472 ( .I(Sreg[17]), .ZN(n436) );
    OAI222D0BWP7T U473 ( .A1(n434), .A2(n454), .B1(n437), .B2(n455), .C1(n209),
         .C2(n456), .ZN(n246) );
    CKND0BWP7T U474 ( .I(S[2]), .ZN(n437) );
    CKND0BWP7T U475 ( .I(Sreg[18]), .ZN(n434) );
    OAI222D0BWP7T U476 ( .A1(n432), .A2(n454), .B1(n435), .B2(n455), .C1(n208),
         .C2(n456), .ZN(n245) );
    CKND0BWP7T U477 ( .I(S[3]), .ZN(n435) );
    CKND0BWP7T U478 ( .I(Sreg[19]), .ZN(n432) );
    OAI222D0BWP7T U479 ( .A1(n430), .A2(n454), .B1(n433), .B2(n455), .C1(n207),
         .C2(n456), .ZN(n244) );
    CKND0BWP7T U480 ( .I(S[4]), .ZN(n433) );
    CKND0BWP7T U481 ( .I(Sreg[20]), .ZN(n430) );
    OAI222D0BWP7T U482 ( .A1(n428), .A2(n454), .B1(n431), .B2(n455), .C1(n206),
         .C2(n456), .ZN(n243) );
    CKND0BWP7T U483 ( .I(S[5]), .ZN(n431) );
    CKND0BWP7T U484 ( .I(Sreg[21]), .ZN(n428) );
```

```
    OAI222D0BWP7T U485 ( .A1(n426), .A2(n454), .B1(n429), .B2(n455), .C1(n205),
         .C2(n456), .ZN(n242) );
    CKND0BWP7T U486 ( .I(S[6]), .ZN(n429) );
    CKND0BWP7T U487 ( .I(Sreg[22]), .ZN(n426) );
    OAI222D0BWP7T U488 ( .A1(n424), .A2(n454), .B1(n427), .B2(n455), .C1(n204),
         .C2(n456), .ZN(n241) );
    CKND0BWP7T U489 ( .I(S[7]), .ZN(n427) );
    CKND0BWP7T U490 ( .I(Sreg[23]), .ZN(n424) );
    OAI222D0BWP7T U491 ( .A1(n422), .A2(n454), .B1(n425), .B2(n455), .C1(n203),
         .C2(n456), .ZN(n240) );
    CKND0BWP7T U492 ( .I(S[8]), .ZN(n425) );
    CKND0BWP7T U493 ( .I(Sreg[24]), .ZN(n422) );
    OAI222D0BWP7T U494 ( .A1(n420), .A2(n454), .B1(n423), .B2(n455), .C1(n202),
         .C2(n456), .ZN(n239) );
    CKND0BWP7T U495 ( .I(S[9]), .ZN(n423) );
    CKND0BWP7T U496 ( .I(Sreg[25]), .ZN(n420) );
    OAI222D0BWP7T U497 ( .A1(n418), .A2(n454), .B1(n421), .B2(n455), .C1(n201),
         .C2(n456), .ZN(n238) );
    CKND0BWP7T U498 ( .I(S[10]), .ZN(n421) );
    CKND0BWP7T U499 ( .I(Sreg[26]), .ZN(n418) );
    OAI222D0BWP7T U500 ( .A1(n416), .A2(n454), .B1(n419), .B2(n455), .C1(n200),
         .C2(n456), .ZN(n237) );
    CKND0BWP7T U501 ( .I(S[11]), .ZN(n419) );
    CKND0BWP7T U502 ( .I(Sreg[27]), .ZN(n416) );
    OAI222D0BWP7T U503 ( .A1(n414), .A2(n454), .B1(n417), .B2(n455), .C1(n199),
         .C2(n456), .ZN(n236) );
    CKND0BWP7T U504 ( .I(S[12]), .ZN(n417) );
    CKND0BWP7T U505 ( .I(Sreg[28]), .ZN(n414) );
    OAI222D0BWP7T U506 ( .A1(n412), .A2(n454), .B1(n415), .B2(n455), .C1(n198),
         .C2(n456), .ZN(n235) );
    CKND0BWP7T U507 ( .I(S[13]), .ZN(n415) );
    CKND0BWP7T U508 ( .I(Sreg[29]), .ZN(n412) );
    OAI222D0BWP7T U509 ( .A1(n407), .A2(n454), .B1(n413), .B2(n455), .C1(n197),
         .C2(n456), .ZN(n234) );
    CKND0BWP7T U510 ( .I(S[14]), .ZN(n413) );
    CKND0BWP7T U511 ( .I(Sreg[30]), .ZN(n407) );
    MOAI22D0BWP7T U512 ( .A1(n405), .A2(n454), .B1(Remainder[15]), .B2(n457),
         .ZN(n233) );
    CKND0BWP7T U513 ( .I(n439), .ZN(n441) );
    INR2D0BWP7T U514 ( .A1(n452), .B1(S[15]), .ZN(n439) );
    AN2D0BWP7T U515 ( .A1(n448), .A2(F), .Z(n452) );
    AN4D0BWP7T U516 ( .A1(n373), .A2(n375), .A3(n458), .A4(n459), .Z(n448) );
    AOI211D0BWP7T U517 ( .A1(n460), .A2(n461), .B(i[31]), .C(n462), .ZN(n459) );
    OAI21D0BWP7T U518 ( .A1(i[1]), .A2(n463), .B(i[4]), .ZN(n458) );
    ND3D0BWP7T U519 ( .A1(i[0]), .A2(n362), .A3(n453), .ZN(n457) );
    AN4D0BWP7T U520 ( .A1(n212), .A2(n464), .A3(n465), .A4(n466), .Z(n453) );
    NR4D0BWP7T U521 ( .A1(reset), .A2(i[11]), .A3(i[10]), .A4(n485), .ZN(n466)
         );
    NR2D0BWP7T U522 ( .A1(n463), .A2(n366), .ZN(n465) );
    CKND0BWP7T U523 ( .I(Sreg[31]), .ZN(n405) );
    CKMUX2D0BWP7T U524 ( .I0(Divisor[0]), .I1(Divisor_temp[0]), .S(n447), .Z(
         n194) );
    CKMUX2D0BWP7T U525 ( .I0(Divisor[1]), .I1(Divisor_temp[1]), .S(n447), .Z(
         n193) );
```

```
    CKMUX2D0BWP7T U526 ( .I0(Divisor[2]), .I1(Divisor_temp[2]), .S(n447), .Z(
        n192) );
    CKMUX2D0BWP7T U527 ( .I0(Divisor[3]), .I1(Divisor_temp[3]), .S(n447), .Z(
        n191) );
    CKMUX2D0BWP7T U528 ( .I0(Divisor[4]), .I1(Divisor_temp[4]), .S(n447), .Z(
        n190) );
    CKMUX2D0BWP7T U529 ( .I0(Divisor[5]), .I1(Divisor_temp[5]), .S(n447), .Z(
        n189) );
    CKMUX2D0BWP7T U530 ( .I0(Divisor[6]), .I1(Divisor_temp[6]), .S(n447), .Z(
        n188) );
    CKMUX2D0BWP7T U531 ( .I0(Divisor[7]), .I1(Divisor_temp[7]), .S(n447), .Z(
        n187) );
    CKMUX2D0BWP7T U532 ( .I0(Divisor[8]), .I1(Divisor_temp[8]), .S(n447), .Z(
        n186) );
    CKMUX2D0BWP7T U533 ( .I0(Divisor[9]), .I1(Divisor_temp[9]), .S(n447), .Z(
        n185) );
    CKMUX2D0BWP7T U534 ( .I0(Divisor[10]), .I1(Divisor_temp[10]), .S(n447), .Z(
        n184) );
    CKMUX2D0BWP7T U535 ( .I0(Divisor[11]), .I1(Divisor_temp[11]), .S(n447), .Z(
        n183) );
    CKMUX2D0BWP7T U536 ( .I0(Divisor[12]), .I1(Divisor_temp[12]), .S(n447), .Z(
        n182) );
    CKMUX2D0BWP7T U537 ( .I0(Divisor[13]), .I1(Divisor_temp[13]), .S(n447), .Z(
        n181) );
    CKMUX2D0BWP7T U538 ( .I0(Divisor[14]), .I1(Divisor_temp[14]), .S(n447), .Z(
        n180) );
    CKMUX2D0BWP7T U539 ( .I0(Divisor[15]), .I1(Divisor_temp[15]), .S(n447), .Z(
        n179) );
    NR3D0BWP7T U540 ( .A1(i[31]), .A2(reset), .A3(F), .ZN(n467) );
    AN4D0BWP7T U541 ( .A1(n360), .A2(n373), .A3(n468), .A4(n469), .Z(n461) );
    NR3D0BWP7T U542 ( .A1(i[11]), .A2(i[1]), .A3(i[12]), .ZN(n468) );
    CKND0BWP7T U543 ( .I(i[0]), .ZN(n360) );
    AN3D0BWP7T U544 ( .A1(n471), .A2(n472), .A3(n473), .Z(n460) );
    ND4D0BWP7T U545 ( .A1(n227), .A2(n226), .A3(n225), .A4(n217), .ZN(n474) );
    NR4D0BWP7T U546 ( .A1(i[4]), .A2(i[26]), .A3(i[25]), .A4(i[24]), .ZN(n472)
         );
    XNR2D1BWP7T U547 ( .A1(n347), .A2(i[0]), .ZN(N246) );
    MUX2ND0BWP7T U548 ( .I0(n363), .I1(n362), .S(n347), .ZN(N245) );
    CKND0BWP7T U549 ( .I(i[1]), .ZN(n362) );
    CKND0BWP7T U550 ( .I(N184), .ZN(n363) );
    MUX2ND0BWP7T U551 ( .I0(n364), .I1(n231), .S(n347), .ZN(N244) );
    CKND0BWP7T U552 ( .I(N185), .ZN(n364) );
    MUX2ND0BWP7T U553 ( .I0(n365), .I1(n230), .S(n347), .ZN(N243) );
    CKND0BWP7T U554 ( .I(N186), .ZN(n365) );
    MUX2ND0BWP7T U555 ( .I0(n367), .I1(n366), .S(n347), .ZN(N242) );
    CKND0BWP7T U556 ( .I(N187), .ZN(n367) );
    MUX2ND0BWP7T U557 ( .I0(n368), .I1(n229), .S(n347), .ZN(N241) );
    CKND0BWP7T U558 ( .I(N188), .ZN(n368) );
    MUX2ND0BWP7T U559 ( .I0(n369), .I1(n228), .S(n347), .ZN(N240) );
    CKND0BWP7T U560 ( .I(N189), .ZN(n369) );
    MUX2ND0BWP7T U561 ( .I0(n370), .I1(n227), .S(n347), .ZN(N239) );
    CKND0BWP7T U562 ( .I(N190), .ZN(n370) );
    MUX2ND0BWP7T U563 ( .I0(n371), .I1(n226), .S(n347), .ZN(N238) );
    CKND0BWP7T U564 ( .I(N191), .ZN(n371) );
```

```
    MUX2ND0BWP7T U565 ( .I0(n372), .I1(n225), .S(n403), .ZN(N237) );
    CKND0BWP7T U566 ( .I(N192), .ZN(n372) );
    MUX2ND0BWP7T U567 ( .I0(n374), .I1(n373), .S(n403), .ZN(N236) );
    CKND0BWP7T U568 ( .I(i[10]), .ZN(n373) );
    CKND0BWP7T U569 ( .I(N193), .ZN(n374) );
    MUX2ND0BWP7T U570 ( .I0(n376), .I1(n375), .S(n403), .ZN(N235) );
    CKND0BWP7T U571 ( .I(i[11]), .ZN(n375) );
    CKND0BWP7T U572 ( .I(N194), .ZN(n376) );
    MUX2ND0BWP7T U573 ( .I0(n378), .I1(n377), .S(n347), .ZN(N234) );
    CKND0BWP7T U574 ( .I(i[12]), .ZN(n377) );
    CKND0BWP7T U575 ( .I(N195), .ZN(n378) );
    MUX2ND0BWP7T U576 ( .I0(n379), .I1(n224), .S(n347), .ZN(N233) );
    CKND0BWP7T U577 ( .I(N196), .ZN(n379) );
    MUX2ND0BWP7T U578 ( .I0(n380), .I1(n223), .S(n347), .ZN(N232) );
    CKND0BWP7T U579 ( .I(N197), .ZN(n380) );
    MUX2ND0BWP7T U580 ( .I0(n381), .I1(n222), .S(n403), .ZN(N231) );
    CKND0BWP7T U581 ( .I(N198), .ZN(n381) );
    MUX2ND0BWP7T U582 ( .I0(n382), .I1(n221), .S(n347), .ZN(N230) );
    CKND0BWP7T U583 ( .I(N199), .ZN(n382) );
    MUX2ND0BWP7T U584 ( .I0(n384), .I1(n383), .S(n403), .ZN(N229) );
    CKND0BWP7T U585 ( .I(N200), .ZN(n384) );
    MUX2ND0BWP7T U586 ( .I0(n386), .I1(n385), .S(n347), .ZN(N228) );
    CKND0BWP7T U587 ( .I(N201), .ZN(n386) );
    MUX2ND0BWP7T U588 ( .I0(n388), .I1(n387), .S(n347), .ZN(N227) );
    CKND0BWP7T U589 ( .I(N202), .ZN(n388) );
    MUX2ND0BWP7T U590 ( .I0(n389), .I1(n220), .S(n347), .ZN(N226) );
    CKND0BWP7T U591 ( .I(N203), .ZN(n389) );
    MUX2ND0BWP7T U592 ( .I0(n390), .I1(n219), .S(n347), .ZN(N225) );
    CKND0BWP7T U593 ( .I(N204), .ZN(n390) );
    MUX2ND0BWP7T U594 ( .I0(n391), .I1(n218), .S(n403), .ZN(N224) );
    CKND0BWP7T U595 ( .I(N205), .ZN(n391) );
    MUX2ND0BWP7T U596 ( .I0(n392), .I1(n217), .S(n403), .ZN(N223) );
    CKND0BWP7T U597 ( .I(N206), .ZN(n392) );
    MUX2ND0BWP7T U598 ( .I0(n394), .I1(n393), .S(n403), .ZN(N222) );
    CKND0BWP7T U599 ( .I(N207), .ZN(n394) );
    MUX2ND0BWP7T U600 ( .I0(n396), .I1(n395), .S(n347), .ZN(N221) );
    CKND0BWP7T U601 ( .I(N208), .ZN(n396) );
    MUX2ND0BWP7T U602 ( .I0(n398), .I1(n397), .S(n403), .ZN(N220) );
    CKND0BWP7T U603 ( .I(N209), .ZN(n398) );
    MUX2ND0BWP7T U604 ( .I0(n399), .I1(n216), .S(n403), .ZN(N219) );
    CKND0BWP7T U605 ( .I(N210), .ZN(n399) );
    MUX2ND0BWP7T U606 ( .I0(n400), .I1(n215), .S(n403), .ZN(N218) );
    CKND0BWP7T U607 ( .I(N211), .ZN(n400) );
    MUX2ND0BWP7T U608 ( .I0(n401), .I1(n214), .S(n403), .ZN(N217) );
    CKND0BWP7T U609 ( .I(N212), .ZN(n401) );
    MUX2ND0BWP7T U610 ( .I0(n402), .I1(n213), .S(n347), .ZN(N216) );
    CKND0BWP7T U611 ( .I(N213), .ZN(n402) );
    MUX2ND0BWP7T U612 ( .I0(n361), .I1(n212), .S(n347), .ZN(N215) );
    OAI21D0BWP7T U613 ( .A1(n475), .A2(i[31]), .B(n485), .ZN(n403) );
    CKND0BWP7T U614 ( .I(F), .ZN(n485) );
    NR3D0BWP7T U615 ( .A1(n476), .A2(i[11]), .A3(i[10]), .ZN(n475) );
    OAI21D0BWP7T U616 ( .A1(n477), .A2(n366), .B(n464), .ZN(n476) );
    CKND0BWP7T U617 ( .I(n462), .ZN(n464) );
    ND3D0BWP7T U618 ( .A1(n471), .A2(n478), .A3(n479), .ZN(n462) );
```

```
    ND3D0BWP7T U619 ( .A1(n218), .A2(n217), .A3(n223), .ZN(n481) );
    ND4D0BWP7T U620 ( .A1(n229), .A2(n228), .A3(n227), .A4(n226), .ZN(n480) );
    NR3D0BWP7T U621 ( .A1(n482), .A2(i[12]), .A3(n470), .ZN(n478) );
    ND4D0BWP7T U622 ( .A1(n221), .A2(n220), .A3(n222), .A4(n483), .ZN(n470) );
    CKND0BWP7T U623 ( .I(i[17]), .ZN(n383) );
    CKND0BWP7T U624 ( .I(i[18]), .ZN(n385) );
    CKND0BWP7T U625 ( .I(i[19]), .ZN(n387) );
    ND3D0BWP7T U626 ( .A1(n395), .A2(n397), .A3(n393), .ZN(n482) );
    CKND0BWP7T U627 ( .I(i[24]), .ZN(n393) );
    CKND0BWP7T U628 ( .I(i[26]), .ZN(n397) );
    CKND0BWP7T U629 ( .I(i[25]), .ZN(n395) );
    NR4D0BWP7T U630 ( .A1(i[30]), .A2(i[29]), .A3(i[28]), .A4(i[27]), .ZN(n471)
          );
    CKND0BWP7T U631 ( .I(i[4]), .ZN(n366) );
    AOI21D0BWP7T U632 ( .A1(i[1]), .A2(i[0]), .B(n463), .ZN(n477) );
    CKND2D0BWP7T U633 ( .A1(n231), .A2(n230), .ZN(n463) );
    CKND0BWP7T U634 ( .I(N214), .ZN(n361) );
endmodule
```

Cuadro 121: Archivo DIV32b_sintetizado.v: *netlist* obtenido de la síntesis lógica del diseño original del circuito

13.8.4. Simulación del circuito sintetizado

El siguiente paso fue la simulación del circuito sintetizado obtenido a partir del diseño final.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 122 en una terminal abierta en `Outputs/DIV32b/Outputs_iniciales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v DIV32b_sintetizado.v DIV32b_tb.v -
    o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 122: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Luego de haber ejecutado los comandos mencionados anteriormente, se obtuvo la Figura 168. Se observó el mismo comportamiento que con el circuito original, lo que confirma que la síntesis del circuito se realizó correctamente. Sin embargo, al eliminar el flanco de bajada, la señal de salida presenta un ligero retraso. No obstante, este retraso no afecta el funcionamiento del circuito.

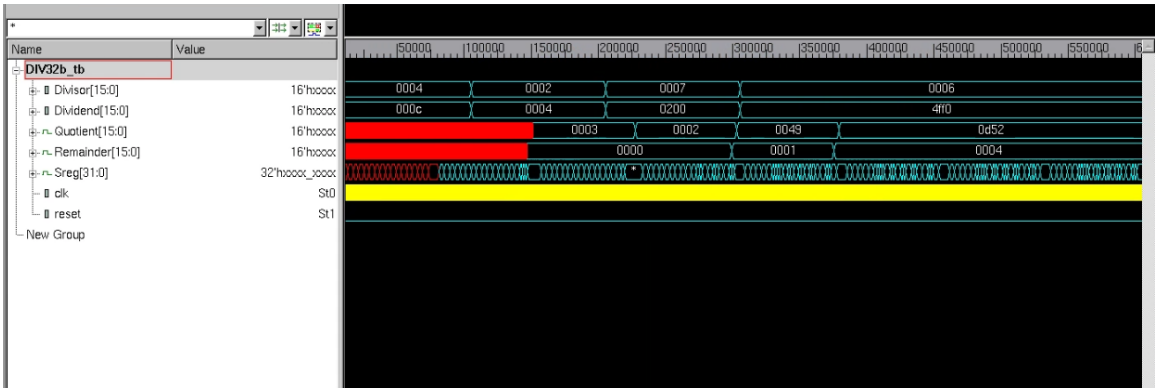


Figura 168: Simulación del divisor de 32 bits en su diseño sintetizado

13.8.5. Unión del circuito sintetizado con el módulo de pines de I/O

El archivo denominado `DIV32b_sintetizado_and_IO.v` se creó uniendo el archivo `DIV32b_sintetizado.v` con el archivo `IO_DIV32b.v`. Este proceso también lo realiza automáticamente el *script* antes de comenzar el segundo proceso de síntesis lógica.

Por ello, es fundamental tener diseñado desde el principio el archivo `IO_DIV32b.v` y guardarlo en la carpeta `Inputs/DIV32b`. Si este archivo no está presente, el *script* no podrá localizarlo y el nuevo archivo generado contendrá únicamente el contenido de `DIV32b_sintetizado.v`. Esto ocasionará que, en el siguiente paso, se realice nuevamente la síntesis sin incluir el módulo de pines de entrada y salida, obteniéndose un resultado similar al del archivo `DIV32b_sintetizado.v`.

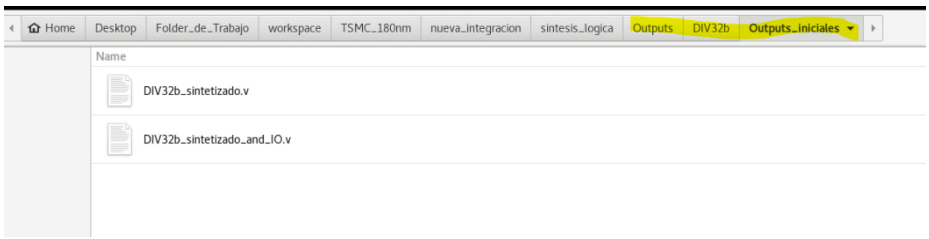


Figura 169: Archivos generados en la carpeta del circuito en Outputs_iniciales

Para este caso, no se mostrará el archivo `DIV32b_sintetizado_and_IO.v` debido a su gran tamaño; sin embargo, dicho archivo consiste únicamente en la combinación entre `DIV32b_sintetizado.v` y `IO_DIV32b.v`.

13.8.6. Síntesis lógica del circuito sintetizado con pines de I/O

A través del archivo `DIV32b_sintetizado_and_IO.v` se obtuvo el circuito completamente sintetizado. Este archivo se generó en el directorio `Outputs/DIV32b/Outputs_finales` con el nombre `DIV32b_sintesis_final.v`. Además, se generaron otros dos archivos que también servirán para el siguiente proceso de síntesis física.

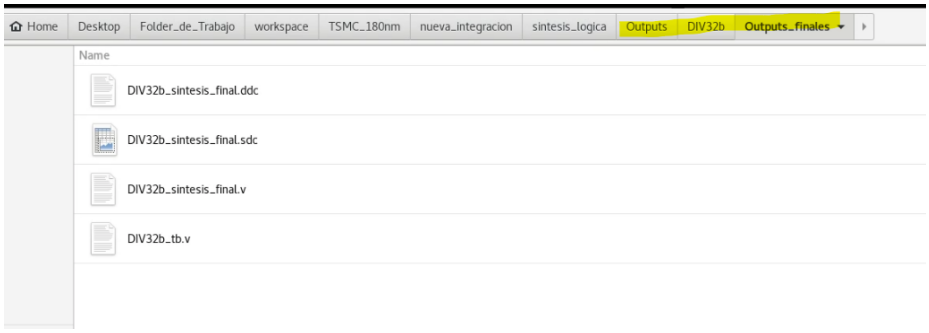


Figura 170: Archivos generados en la carpeta del circuito en Outputs_finales

En el Cuadro 123 se muestra el archivo Verilog obtenido.

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date  : Sun Nov 3 00:39:29 2024
////////////////////////////////////////////////////////////


module carry_look_ahead_4bit_3 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_2 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
```

```
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_1 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_4bit_0 ( a, b, cin, S, cout );
  input [3:0] a;
  input [3:0] b;
  output [3:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7;

  MOAI22D0BWP7T U1 ( .A1(n1), .A2(n2), .B1(b[3]), .B2(a[3]), .ZN(cout) );
  CKXOR2D0BWP7T U2 ( .A1(n1), .A2(n2), .Z(S[3]) );
  XNR2D1BWP7T U3 ( .A1(b[3]), .A2(a[3]), .ZN(n2) );
  AOI22D0BWP7T U4 ( .A1(b[2]), .A2(a[2]), .B1(n3), .B2(n4), .ZN(n1) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(n3), .Z(S[2]) );
  MOAI22D0BWP7T U6 ( .A1(n5), .A2(n6), .B1(b[1]), .B2(a[1]), .ZN(n3) );
  CKXOR2D0BWP7T U7 ( .A1(b[2]), .A2(a[2]), .Z(n4) );
  CKXOR2D0BWP7T U8 ( .A1(n5), .A2(n6), .Z(S[1]) );
  XNR2D1BWP7T U9 ( .A1(b[1]), .A2(a[1]), .ZN(n6) );
```

```
  AOI22D0BWP7T U10 ( .A1(n7), .A2(cin), .B1(a[0]), .B2(b[0]), .ZN(n5) );
  CKXOR2D0BWP7T U11 ( .A1(cin), .A2(n7), .Z(S[0]) );
  CKXOR2D0BWP7T U12 ( .A1(b[0]), .A2(a[0]), .Z(n7) );
endmodule


module carry_look_ahead_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire w1, w2, w3, n1;

  carry_look_ahead_4bit_3 U1 ( .a(a[3:0]), .b(b[3:0]), .cin(n1), .S(S[3:0]),
       .cout(w1) );
  carry_look_ahead_4bit_2 U2 ( .a(a[7:4]), .b(b[7:4]), .cin(w1), .S(S[7:4]),
       .cout(w2) );
  carry_look_ahead_4bit_1 U3 ( .a(a[11:8]), .b(b[11:8]), .cin(w2), .S(S[11:8]),
       .cout(w3) );
  carry_look_ahead_4bit_0 U4 ( .a(a[15:12]), .b(b[15:12]), .cin(w3), .S(
       S[15:12]), .cout(cout) );
  TIELBWP7T U5 ( .ZN(n1) );
endmodule


module substractor_16bit_DW01_inc_0_DW01_inc_1 ( A, SUM );
  input [15:0] A;
  output [15:0] SUM;
  wire n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14;

  CKXOR2D0BWP7T U1 ( .A1(A[9]), .A2(n1), .Z(SUM[9]) );
  XNR2D1BWP7T U2 ( .A1(n2), .A2(A[8]), .ZN(SUM[8]) );
  CKND2D0BWP7T U3 ( .A1(A[7]), .A2(n3), .ZN(n2) );
  CKXOR2D0BWP7T U4 ( .A1(n3), .A2(A[7]), .Z(SUM[7]) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(A[6]), .Z(SUM[6]) );
  XNR2D1BWP7T U6 ( .A1(n5), .A2(A[5]), .ZN(SUM[5]) );
  XNR2D1BWP7T U7 ( .A1(n6), .A2(A[4]), .ZN(SUM[4]) );
  CKND2D0BWP7T U8 ( .A1(A[3]), .A2(n7), .ZN(n6) );
  CKXOR2D0BWP7T U9 ( .A1(A[3]), .A2(n7), .Z(SUM[3]) );
  XNR2D1BWP7T U10 ( .A1(n8), .A2(A[2]), .ZN(SUM[2]) );
  XNR2D1BWP7T U11 ( .A1(A[1]), .A2(SUM[0]), .ZN(SUM[1]) );
  CKXOR2D0BWP7T U12 ( .A1(A[15]), .A2(n9), .Z(SUM[15]) );
  INR2D0BWP7T U13 ( .A1(A[14]), .B1(n10), .ZN(n9) );
  XNR2D1BWP7T U14 ( .A1(n10), .A2(A[14]), .ZN(SUM[14]) );
  IND2D0BWP7T U15 ( .A1(n11), .B1(A[13]), .ZN(n10) );
  XNR2D1BWP7T U16 ( .A1(n11), .A2(A[13]), .ZN(SUM[13]) );
  ND3D0BWP7T U17 ( .A1(A[11]), .A2(n12), .A3(A[12]), .ZN(n11) );
  XNR2D1BWP7T U18 ( .A1(n13), .A2(A[12]), .ZN(SUM[12]) );
  CKND2D0BWP7T U19 ( .A1(A[11]), .A2(n12), .ZN(n13) );
  CKXOR2D0BWP7T U20 ( .A1(A[11]), .A2(n12), .Z(SUM[11]) );
  INR2D0BWP7T U21 ( .A1(A[10]), .B1(n14), .ZN(n12) );
  XNR2D1BWP7T U22 ( .A1(n14), .A2(A[10]), .ZN(SUM[10]) );
  CKND2D0BWP7T U23 ( .A1(n1), .A2(A[9]), .ZN(n14) );
```

```
  AN3D0BWP7T U24 ( .A1(A[7]), .A2(n3), .A3(A[8]), .Z(n1) );
  AN2D0BWP7T U25 ( .A1(A[6]), .A2(n4), .Z(n3) );
  INR2D0BWP7T U26 ( .A1(A[5]), .B1(n5), .ZN(n4) );
  ND3D0BWP7T U27 ( .A1(A[3]), .A2(n7), .A3(A[4]), .ZN(n5) );
  INR2D0BWP7T U28 ( .A1(A[2]), .B1(n8), .ZN(n7) );
  CKND2D0BWP7T U29 ( .A1(A[1]), .A2(A[0]), .ZN(n8) );
  CKND0BWP7T U30 ( .I(A[0]), .ZN(SUM[0]) );
endmodule


module substractor_16bit ( a, b, cin, S, cout );
  input [15:0] a;
  input [15:0] b;
  output [15:0] S;
  input cin;
  output cout;
  wire n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16,
       n17;
  wire [15:0] b_comp;

  carry_look_ahead_16bit U35 ( .a(a), .b(b_comp), .cin(n17), .S(S), .cout(cout)
      );
  substractor_16bit_DW01_inc_0_DW01_inc_1 add_29 ( .A({n1, n2, n3, n4, n5, n6,
       n7, n8, n9, n10, n11, n12, n13, n14, n15, n16}), .SUM(b_comp) );
  TIELBWP7T U1 ( .ZN(n17) );
  CKND0BWP7T U2 ( .I(b[7]), .ZN(n9) );
  CKND0BWP7T U3 ( .I(b[8]), .ZN(n8) );
  CKND0BWP7T U4 ( .I(b[9]), .ZN(n7) );
  CKND0BWP7T U5 ( .I(b[10]), .ZN(n6) );
  CKND0BWP7T U6 ( .I(b[11]), .ZN(n5) );
  CKND0BWP7T U7 ( .I(b[12]), .ZN(n4) );
  CKND0BWP7T U8 ( .I(b[13]), .ZN(n3) );
  CKND0BWP7T U9 ( .I(b[14]), .ZN(n2) );
  CKND0BWP7T U10 ( .I(b[0]), .ZN(n16) );
  CKND0BWP7T U11 ( .I(b[1]), .ZN(n15) );
  CKND0BWP7T U12 ( .I(b[2]), .ZN(n14) );
  CKND0BWP7T U13 ( .I(b[3]), .ZN(n13) );
  CKND0BWP7T U14 ( .I(b[4]), .ZN(n12) );
  CKND0BWP7T U15 ( .I(b[5]), .ZN(n11) );
  CKND0BWP7T U16 ( .I(b[6]), .ZN(n10) );
  CKND0BWP7T U17 ( .I(b[15]), .ZN(n1) );
endmodule


module divisor_16bit_DW01_inc_0 ( A, SUM );
  input [31:0] A;
  output [31:0] SUM;
  wire n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16,
       n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30;

  XNR2D1BWP7T U1 ( .A1(n1), .A2(A[9]), .ZN(SUM[9]) );
  XNR2D1BWP7T U2 ( .A1(n2), .A2(A[8]), .ZN(SUM[8]) );
  CKND2D0BWP7T U3 ( .A1(A[7]), .A2(n3), .ZN(n2) );
  CKXOR2D0BWP7T U4 ( .A1(A[7]), .A2(n3), .Z(SUM[7]) );
```

```
    XNR2D1BWP7T U5 ( .A1(n4), .A2(A[6]), .ZN(SUM[6]) );
    CKND2D0BWP7T U6 ( .A1(n5), .A2(A[5]), .ZN(n4) );
    CKXOR2D0BWP7T U7 ( .A1(n5), .A2(A[5]), .Z(SUM[5]) );
    XNR2D1BWP7T U8 ( .A1(n6), .A2(A[4]), .ZN(SUM[4]) );
    CKND2D0BWP7T U9 ( .A1(A[3]), .A2(n7), .ZN(n6) );
    CKXOR2D0BWP7T U10 ( .A1(A[3]), .A2(n7), .Z(SUM[3]) );
    CKXOR2D0BWP7T U11 ( .A1(A[31]), .A2(n8), .Z(SUM[31]) );
    INR2D0BWP7T U12 ( .A1(A[30]), .B1(n9), .ZN(n8) );
    XNR2D1BWP7T U13 ( .A1(n9), .A2(A[30]), .ZN(SUM[30]) );
    CKND2D0BWP7T U14 ( .A1(A[29]), .A2(n10), .ZN(n9) );
    XNR2D1BWP7T U15 ( .A1(n11), .A2(A[2]), .ZN(SUM[2]) );
    CKXOR2D0BWP7T U16 ( .A1(A[29]), .A2(n10), .Z(SUM[29]) );
    AN3D0BWP7T U17 ( .A1(A[27]), .A2(n12), .A3(A[28]), .Z(n10) );
    XNR2D1BWP7T U18 ( .A1(n13), .A2(A[28]), .ZN(SUM[28]) );
    CKND2D0BWP7T U19 ( .A1(A[27]), .A2(n12), .ZN(n13) );
    CKXOR2D0BWP7T U20 ( .A1(n12), .A2(A[27]), .Z(SUM[27]) );
    AN2D0BWP7T U21 ( .A1(A[26]), .A2(n14), .Z(n12) );
    CKXOR2D0BWP7T U22 ( .A1(n14), .A2(A[26]), .Z(SUM[26]) );
    INR2D0BWP7T U23 ( .A1(A[25]), .B1(n15), .ZN(n14) );
    XNR2D1BWP7T U24 ( .A1(n15), .A2(A[25]), .ZN(SUM[25]) );
    ND3D0BWP7T U25 ( .A1(A[23]), .A2(n16), .A3(A[24]), .ZN(n15) );
    XNR2D1BWP7T U26 ( .A1(n17), .A2(A[24]), .ZN(SUM[24]) );
    CKND2D0BWP7T U27 ( .A1(A[23]), .A2(n16), .ZN(n17) );
    CKXOR2D0BWP7T U28 ( .A1(A[23]), .A2(n16), .Z(SUM[23]) );
    INR2D0BWP7T U29 ( .A1(A[22]), .B1(n18), .ZN(n16) );
    XNR2D1BWP7T U30 ( .A1(n18), .A2(A[22]), .ZN(SUM[22]) );
    CKND2D0BWP7T U31 ( .A1(A[21]), .A2(n19), .ZN(n18) );
    CKXOR2D0BWP7T U32 ( .A1(A[21]), .A2(n19), .Z(SUM[21]) );
    AN3D0BWP7T U33 ( .A1(A[19]), .A2(n20), .A3(A[20]), .Z(n19) );
    XNR2D1BWP7T U34 ( .A1(n21), .A2(A[20]), .ZN(SUM[20]) );
    CKND2D0BWP7T U35 ( .A1(A[19]), .A2(n20), .ZN(n21) );
    CKXOR2D0BWP7T U36 ( .A1(A[1]), .A2(A[0]), .Z(SUM[1]) );
    CKXOR2D0BWP7T U37 ( .A1(n20), .A2(A[19]), .Z(SUM[19]) );
    AN2D0BWP7T U38 ( .A1(A[18]), .A2(n22), .Z(n20) );
    CKXOR2D0BWP7T U39 ( .A1(n22), .A2(A[18]), .Z(SUM[18]) );
    INR2D0BWP7T U40 ( .A1(A[17]), .B1(n23), .ZN(n22) );
    XNR2D1BWP7T U41 ( .A1(n23), .A2(A[17]), .ZN(SUM[17]) );
    ND3D0BWP7T U42 ( .A1(A[15]), .A2(n24), .A3(A[16]), .ZN(n23) );
    XNR2D1BWP7T U43 ( .A1(n25), .A2(A[16]), .ZN(SUM[16]) );
    CKND2D0BWP7T U44 ( .A1(A[15]), .A2(n24), .ZN(n25) );
    CKXOR2D0BWP7T U45 ( .A1(A[15]), .A2(n24), .Z(SUM[15]) );
    INR2D0BWP7T U46 ( .A1(A[14]), .B1(n26), .ZN(n24) );
    XNR2D1BWP7T U47 ( .A1(n26), .A2(A[14]), .ZN(SUM[14]) );
    CKND2D0BWP7T U48 ( .A1(A[13]), .A2(n27), .ZN(n26) );
    CKXOR2D0BWP7T U49 ( .A1(A[13]), .A2(n27), .Z(SUM[13]) );
    AN3D0BWP7T U50 ( .A1(A[11]), .A2(n28), .A3(A[12]), .Z(n27) );
    XNR2D1BWP7T U51 ( .A1(n29), .A2(A[12]), .ZN(SUM[12]) );
    CKND2D0BWP7T U52 ( .A1(A[11]), .A2(n28), .ZN(n29) );
    CKXOR2D0BWP7T U53 ( .A1(n28), .A2(A[11]), .Z(SUM[11]) );
    AN2D0BWP7T U54 ( .A1(A[10]), .A2(n30), .Z(n28) );
    CKXOR2D0BWP7T U55 ( .A1(n30), .A2(A[10]), .Z(SUM[10]) );
    INR2D0BWP7T U56 ( .A1(A[9]), .B1(n1), .ZN(n30) );
    ND3D0BWP7T U57 ( .A1(A[7]), .A2(n3), .A3(A[8]), .ZN(n1) );
    AN3D0BWP7T U58 ( .A1(n5), .A2(A[5]), .A3(A[6]), .Z(n3) );
```

```
  AN3D0BWP7T U59 ( .A1(A[3]), .A2(n7), .A3(A[4]), .Z(n5) );
  INR2D0BWP7T U60 ( .A1(A[2]), .B1(n11), .ZN(n7) );
  CKND2D0BWP7T U61 ( .A1(A[1]), .A2(A[0]), .ZN(n11) );
endmodule


module divisor_16bit ( Divisor, Dividend, clk, reset, Quotient, Remainder,
      Sreg );
  input [15:0] Divisor;
  input [15:0] Dividend;
  output [15:0] Quotient;
  output [15:0] Remainder;
  output [31:0] Sreg;
  input clk, reset;
  wire n194, n193, n192, n191, n190, n189, n188, n187, n186, n185, n184,
       n183, n182, n181, n180, n179, n295, n294, n293, n292, n291, n290,
       n289, n288, n287, n286, n285, n284, n283, n282, n281, n311, n297,
       n298, n299, n300, n301, n302, n303, n304, n305, n306, n307, n308,
       n309, n310, n312, n280, n278, n276, n274, n272, n270, n268, n266,
       n264, n262, n260, n258, n256, n254, n252, n250, n249, n251, n253,
       n255, n257, n259, n261, n263, n265, n267, n269, n271, n273, n275,
       n277, n279, n233, N214, N213, N212, N211, N210, N209, N208, N207,
       N206, N205, N204, N203, N202, N201, N200, N199, N198, N197, N196,
       N195, N194, N193, N192, N191, N190, N189, N188, N187, N186, N185,
       N184, n239, n202, n240, n203, n241, n204, n242, n205, n243, n206,
       n244, n207, n245, n208, n246, n209, n247, n210, n248, n211, n234,
       n197, n235, n198, n236, n199, n237, n200, n238, n201, F, n318, n317,
       n319, n324, n325, n326, n331, n332, n333, n339, n344, n342, n296,
       n313, n213, n314, n214, n315, n215, n316, n216, n320, n217, n321,
       n218, n322, n219, n323, n220, n327, n221, n328, n222, n343, n212,
       n329, n223, n330, n224, n334, n225, n335, n226, n336, n227, n337,
       n228, n338, n229, n340, n230, n341, n231, n100, n117, n118, n1, n2,
       n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16, n17,
       n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30, n31,
       n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42, n43, n44, n45,
       n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58, n59,
       n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72, n73,
       n74, n75, n76, n77, n78, n79, n80, n81, n82, n83, n84, n85, n86, n87,
       n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n101,
       n102, n103, n104, n105, n106, n107, n108, n109, n110, n111, n112,
       n113, n114, n115, n116, n119, n120, n121, n122, n123, n124, n125,
       n126, n127, n128, n129, n130, n131, n132, n133, n134, n135, n136,
       n137, n138, n139;
  wire [15:0] Divisor_temp;
  wire [15:0] quo;
  wire [15:0] S;
  wire [31:0] i;
  wire SYNOPSYS_UNCONNECTED__0;

  DFQD1BWP7T \Divisor_temp_reg[0] ( .D(n194), .CP(n12), .Q(Divisor_temp[0])
      );
  DFQD1BWP7T \Divisor_temp_reg[1] ( .D(n193), .CP(n12), .Q(Divisor_temp[1])
      );
  DFQD1BWP7T \Divisor_temp_reg[2] ( .D(n192), .CP(n12), .Q(Divisor_temp[2])
```

```
      );
    DFQD1BWP7T \Divisor_temp_reg[3] ( .D(n191), .CP(n12), .Q(Divisor_temp[3])
      );
    DFQD1BWP7T \Divisor_temp_reg[4] ( .D(n190), .CP(n12), .Q(Divisor_temp[4])
      );
    DFQD1BWP7T \Divisor_temp_reg[5] ( .D(n189), .CP(n12), .Q(Divisor_temp[5])
      );
    DFQD1BWP7T \Divisor_temp_reg[6] ( .D(n188), .CP(n12), .Q(Divisor_temp[6])
      );
    DFQD1BWP7T \Divisor_temp_reg[7] ( .D(n187), .CP(n12), .Q(Divisor_temp[7])
      );
    DFQD1BWP7T \Divisor_temp_reg[8] ( .D(n186), .CP(n13), .Q(Divisor_temp[8])
      );
    DFQD1BWP7T \Divisor_temp_reg[9] ( .D(n185), .CP(n13), .Q(Divisor_temp[9])
      );
    DFQD1BWP7T \Divisor_temp_reg[10] ( .D(n184), .CP(n13), .Q(Divisor_temp[10])
      );
    DFQD1BWP7T \Divisor_temp_reg[11] ( .D(n183), .CP(n13), .Q(Divisor_temp[11])
      );
    DFQD1BWP7T \Divisor_temp_reg[12] ( .D(n182), .CP(n13), .Q(Divisor_temp[12])
      );
    DFQD1BWP7T \Divisor_temp_reg[13] ( .D(n181), .CP(n13), .Q(Divisor_temp[13])
      );
    DFQD1BWP7T \Divisor_temp_reg[14] ( .D(n180), .CP(n13), .Q(Divisor_temp[14])
      );
    DFQD1BWP7T \Divisor_temp_reg[15] ( .D(n179), .CP(n13), .Q(Divisor_temp[15])
      );
    DFQD1BWP7T \Sreg_reg[1] ( .D(n295), .CP(n14), .Q(Sreg[1]) );
    DFQD1BWP7T \Sreg_reg[2] ( .D(n294), .CP(n14), .Q(Sreg[2]) );
    DFQD1BWP7T \Sreg_reg[3] ( .D(n293), .CP(n14), .Q(Sreg[3]) );
    DFQD1BWP7T \Sreg_reg[4] ( .D(n292), .CP(n14), .Q(Sreg[4]) );
    DFQD1BWP7T \Sreg_reg[5] ( .D(n291), .CP(n14), .Q(Sreg[5]) );
    DFQD1BWP7T \Sreg_reg[6] ( .D(n290), .CP(n14), .Q(Sreg[6]) );
    DFQD1BWP7T \Sreg_reg[7] ( .D(n289), .CP(n14), .Q(Sreg[7]) );
    DFQD1BWP7T \Sreg_reg[8] ( .D(n288), .CP(n14), .Q(Sreg[8]) );
    DFQD1BWP7T \Sreg_reg[9] ( .D(n287), .CP(n14), .Q(Sreg[9]) );
    DFQD1BWP7T \Sreg_reg[10] ( .D(n286), .CP(n14), .Q(Sreg[10]) );
    DFQD1BWP7T \Sreg_reg[11] ( .D(n285), .CP(n14), .Q(Sreg[11]) );
    DFQD1BWP7T \Sreg_reg[12] ( .D(n284), .CP(n13), .Q(Sreg[12]) );
    DFQD1BWP7T \Sreg_reg[13] ( .D(n283), .CP(n13), .Q(Sreg[13]) );
    DFQD1BWP7T \Sreg_reg[14] ( .D(n282), .CP(n13), .Q(Sreg[14]) );
    DFQD1BWP7T \Sreg_reg[15] ( .D(n281), .CP(n13), .Q(Sreg[15]) );
    DFQD1BWP7T \Sreg_reg[16] ( .D(n311), .CP(n7), .Q(Sreg[16]) );
    DFQD1BWP7T \Sreg_reg[17] ( .D(n297), .CP(n7), .Q(Sreg[17]) );
    DFQD1BWP7T \Sreg_reg[18] ( .D(n298), .CP(n7), .Q(Sreg[18]) );
    DFQD1BWP7T \Sreg_reg[19] ( .D(n299), .CP(n7), .Q(Sreg[19]) );
    DFQD1BWP7T \Sreg_reg[20] ( .D(n300), .CP(n12), .Q(Sreg[20]) );
    DFQD1BWP7T \Sreg_reg[21] ( .D(n301), .CP(n12), .Q(Sreg[21]) );
    DFQD1BWP7T \Sreg_reg[22] ( .D(n302), .CP(n12), .Q(Sreg[22]) );
    DFQD1BWP7T \Sreg_reg[23] ( .D(n303), .CP(n12), .Q(Sreg[23]) );
    DFQD1BWP7T \Sreg_reg[24] ( .D(n304), .CP(n11), .Q(Sreg[24]) );
    DFQD1BWP7T \Sreg_reg[25] ( .D(n305), .CP(n11), .Q(Sreg[25]) );
    DFQD1BWP7T \Sreg_reg[26] ( .D(n306), .CP(n11), .Q(Sreg[26]) );
    DFQD1BWP7T \Sreg_reg[27] ( .D(n307), .CP(n11), .Q(Sreg[27]) );
```

```
    DFQD1BWP7T \Sreg_reg[28] ( .D(n308), .CP(n11), .Q(Sreg[28]) );
    DFQD1BWP7T \Sreg_reg[29] ( .D(n309), .CP(n11), .Q(Sreg[29]) );
    DFQD1BWP7T \Sreg_reg[30] ( .D(n310), .CP(n11), .Q(Sreg[30]) );
    DFQD1BWP7T \Sreg_reg[31] ( .D(n312), .CP(n11), .Q(Sreg[31]) );
    DFQD1BWP7T \quo_reg[0] ( .D(n280), .CP(n11), .Q(quo[0]) );
    DFQD1BWP7T \quo_reg[1] ( .D(n278), .CP(n11), .Q(quo[1]) );
    DFQD1BWP7T \quo_reg[2] ( .D(n276), .CP(n10), .Q(quo[2]) );
    DFQD1BWP7T \quo_reg[3] ( .D(n274), .CP(n10), .Q(quo[3]) );
    DFQD1BWP7T \quo_reg[4] ( .D(n272), .CP(n10), .Q(quo[4]) );
    DFQD1BWP7T \quo_reg[5] ( .D(n270), .CP(n10), .Q(quo[5]) );
    DFQD1BWP7T \quo_reg[6] ( .D(n268), .CP(n10), .Q(quo[6]) );
    DFQD1BWP7T \quo_reg[7] ( .D(n266), .CP(n10), .Q(quo[7]) );
    DFQD1BWP7T \quo_reg[8] ( .D(n264), .CP(n9), .Q(quo[8]) );
    DFQD1BWP7T \quo_reg[9] ( .D(n262), .CP(n9), .Q(quo[9]) );
    DFQD1BWP7T \quo_reg[10] ( .D(n260), .CP(n9), .Q(quo[10]) );
    DFQD1BWP7T \quo_reg[11] ( .D(n258), .CP(n9), .Q(quo[11]) );
    DFQD1BWP7T \quo_reg[12] ( .D(n256), .CP(n9), .Q(quo[12]) );
    DFQD1BWP7T \quo_reg[13] ( .D(n254), .CP(n9), .Q(quo[13]) );
    DFQD1BWP7T \quo_reg[14] ( .D(n252), .CP(n8), .Q(quo[14]) );
    DFQD1BWP7T \quo_reg[15] ( .D(n250), .CP(n8), .Q(quo[15]) );
    DFQD1BWP7T \Quotient_reg[15] ( .D(n249), .CP(n8), .Q(Quotient[15]) );
    DFQD1BWP7T \Quotient_reg[14] ( .D(n251), .CP(n8), .Q(Quotient[14]) );
    DFQD1BWP7T \Quotient_reg[13] ( .D(n253), .CP(n9), .Q(Quotient[13]) );
    DFQD1BWP7T \Quotient_reg[12] ( .D(n255), .CP(n9), .Q(Quotient[12]) );
    DFQD1BWP7T \Quotient_reg[11] ( .D(n257), .CP(n9), .Q(Quotient[11]) );
    DFQD1BWP7T \Quotient_reg[10] ( .D(n259), .CP(n9), .Q(Quotient[10]) );
    DFQD1BWP7T \Quotient_reg[9] ( .D(n261), .CP(n9), .Q(Quotient[9]) );
    DFQD1BWP7T \Quotient_reg[8] ( .D(n263), .CP(n9), .Q(Quotient[8]) );
    DFQD1BWP7T \Quotient_reg[7] ( .D(n265), .CP(n10), .Q(Quotient[7]) );
    DFQD1BWP7T \Quotient_reg[6] ( .D(n267), .CP(n10), .Q(Quotient[6]) );
    DFQD1BWP7T \Quotient_reg[5] ( .D(n269), .CP(n10), .Q(Quotient[5]) );
    DFQD1BWP7T \Quotient_reg[4] ( .D(n271), .CP(n10), .Q(Quotient[4]) );
    DFQD1BWP7T \Quotient_reg[3] ( .D(n273), .CP(n10), .Q(Quotient[3]) );
    DFQD1BWP7T \Quotient_reg[2] ( .D(n275), .CP(n10), .Q(Quotient[2]) );
    DFQD1BWP7T \Quotient_reg[1] ( .D(n277), .CP(n11), .Q(Quotient[1]) );
    DFQD1BWP7T \Quotient_reg[0] ( .D(n279), .CP(n11), .Q(Quotient[0]) );
    DFQD1BWP7T \Remainder_reg[15] ( .D(n233), .CP(n7), .Q(Remainder[15]) );
    substractor_16bit U36 ( .a(Sreg[31:16]), .b(Divisor_temp), .cin(n118), .S(S)
          );
    divisor_16bit_DW01_inc_0 add_169 ( .A(i), .SUM({N214, N213, N212, N211, N210,
         N209, N208, N207, N206, N205, N204, N203, N202, N201, N200, N199, N198,
         N197, N196, N195, N194, N193, N192, N191, N190, N189, N188, N187, N186,
         N185, N184, SYNOPSYS_UNCONNECTED__0}) );
    DFCND0BWP7T \i_reg[30] ( .D(n313), .CP(n17), .CDN(n117), .Q(i[30]), .QN(
         n213) );
    DFCND0BWP7T \i_reg[29] ( .D(n314), .CP(n15), .CDN(n117), .Q(i[29]), .QN(
         n214) );
    DFCND0BWP7T \i_reg[28] ( .D(n315), .CP(n15), .CDN(n117), .Q(i[28]), .QN(
         n215) );
    DFCND0BWP7T \i_reg[27] ( .D(n316), .CP(n15), .CDN(n117), .Q(i[27]), .QN(
         n216) );
    DFCND0BWP7T \i_reg[23] ( .D(n320), .CP(n15), .CDN(n117), .Q(i[23]), .QN(
         n217) );
    DFCND0BWP7T \i_reg[22] ( .D(n321), .CP(n15), .CDN(n117), .Q(i[22]), .QN(
```

```
      n218) );
  DFCND0BWP7T \i_reg[21] ( .D(n322), .CP(n15), .CDN(n117), .Q(i[21]), .QN(
      n219) );
  DFCND0BWP7T \i_reg[20] ( .D(n323), .CP(n15), .CDN(n117), .Q(i[20]), .QN(
      n220) );
  DFCND0BWP7T \i_reg[16] ( .D(n327), .CP(n16), .CDN(n117), .Q(i[16]), .QN(
      n221) );
  DFCND0BWP7T \i_reg[15] ( .D(n328), .CP(n16), .CDN(n117), .Q(i[15]), .QN(
      n222) );
  DFCND0BWP7T \i_reg[31] ( .D(n343), .CP(n15), .CDN(n117), .Q(i[31]), .QN(
      n212) );
  DFCND0BWP7T \i_reg[14] ( .D(n329), .CP(n16), .CDN(n117), .Q(i[14]), .QN(
      n223) );
  DFCND0BWP7T \i_reg[13] ( .D(n330), .CP(n16), .CDN(n117), .Q(i[13]), .QN(
      n224) );
  DFCND0BWP7T \i_reg[9] ( .D(n334), .CP(n16), .CDN(n117), .Q(i[9]), .QN(n225)
       );
  DFCND0BWP7T \i_reg[8] ( .D(n335), .CP(n16), .CDN(n117), .Q(i[8]), .QN(n226)
       );
  DFCND0BWP7T \i_reg[7] ( .D(n336), .CP(n16), .CDN(n117), .Q(i[7]), .QN(n227)
       );
  DFCND0BWP7T \i_reg[6] ( .D(n337), .CP(n16), .CDN(n117), .Q(i[6]), .QN(n228)
       );
  DFCND0BWP7T \i_reg[5] ( .D(n338), .CP(n16), .CDN(n117), .Q(i[5]), .QN(n229)
       );
  DFCND0BWP7T \i_reg[3] ( .D(n340), .CP(n17), .CDN(n117), .Q(i[3]), .QN(n230)
       );
  DFCND0BWP7T \i_reg[2] ( .D(n341), .CP(n17), .CDN(n117), .Q(i[2]), .QN(n231)
       );
  DFCNQD1BWP7T \i_reg[26] ( .D(n317), .CP(n15), .CDN(n117), .Q(i[26]) );
  DFCNQD1BWP7T \i_reg[25] ( .D(n318), .CP(n15), .CDN(n117), .Q(i[25]) );
  DFCNQD1BWP7T \i_reg[24] ( .D(n319), .CP(n15), .CDN(n117), .Q(i[24]) );
  DFCNQD1BWP7T \i_reg[19] ( .D(n324), .CP(n15), .CDN(n117), .Q(i[19]) );
  DFCNQD1BWP7T \i_reg[10] ( .D(n333), .CP(n16), .CDN(n117), .Q(i[10]) );
  DFCNQD1BWP7T \i_reg[18] ( .D(n325), .CP(n15), .CDN(n117), .Q(i[18]) );
  DFCNQD1BWP7T \i_reg[0] ( .D(n344), .CP(n17), .CDN(n117), .Q(i[0]) );
  DFCNQD1BWP7T \i_reg[17] ( .D(n326), .CP(n16), .CDN(n117), .Q(i[17]) );
  DFCNQD1BWP7T \i_reg[12] ( .D(n331), .CP(n16), .CDN(n117), .Q(i[12]) );
  DFCNQD1BWP7T \i_reg[11] ( .D(n332), .CP(n16), .CDN(n117), .Q(i[11]) );
  DFCNQD1BWP7T \i_reg[4] ( .D(n339), .CP(n17), .CDN(n117), .Q(i[4]) );
  DFCNQD1BWP7T \i_reg[1] ( .D(n342), .CP(n17), .CDN(n117), .Q(i[1]) );
  DFCNQD1BWP7T F_reg ( .D(n100), .CP(n17), .CDN(n117), .Q(F) );
  DFD1BWP7T \Remainder_reg[9] ( .D(n239), .CP(n8), .Q(Remainder[9]), .QN(n202)
     );
  DFD1BWP7T \Remainder_reg[8] ( .D(n240), .CP(n8), .Q(Remainder[8]), .QN(n203)
     );
  DFD1BWP7T \Remainder_reg[7] ( .D(n241), .CP(n8), .Q(Remainder[7]), .QN(n204)
     );
  DFD1BWP7T \Remainder_reg[6] ( .D(n242), .CP(n7), .Q(Remainder[6]), .QN(n205)
     );
  DFD1BWP7T \Remainder_reg[5] ( .D(n243), .CP(n7), .Q(Remainder[5]), .QN(n206)
     );
  DFD1BWP7T \Remainder_reg[4] ( .D(n244), .CP(n7), .Q(Remainder[4]), .QN(n207)
     );
```

```
    DFD1BWP7T \Remainder_reg[3] ( .D(n245), .CP(n7), .Q(Remainder[3]), .QN(n208)
        );
    DFD1BWP7T \Remainder_reg[2] ( .D(n246), .CP(n7), .Q(Remainder[2]), .QN(n209)
        );
    DFD1BWP7T \Remainder_reg[1] ( .D(n247), .CP(n7), .Q(Remainder[1]), .QN(n210)
        );
    DFD1BWP7T \Remainder_reg[0] ( .D(n248), .CP(n7), .Q(Remainder[0]), .QN(n211)
        );
    DFD1BWP7T \Remainder_reg[14] ( .D(n234), .CP(n8), .Q(Remainder[14]), .QN(
        n197) );
    DFD1BWP7T \Remainder_reg[13] ( .D(n235), .CP(n8), .Q(Remainder[13]), .QN(
        n198) );
    DFD1BWP7T \Remainder_reg[12] ( .D(n236), .CP(n8), .Q(Remainder[12]), .QN(
        n199) );
    DFD1BWP7T \Remainder_reg[11] ( .D(n237), .CP(n8), .Q(Remainder[11]), .QN(
        n200) );
    DFD1BWP7T \Remainder_reg[10] ( .D(n238), .CP(n8), .Q(Remainder[10]), .QN(
        n201) );
    DFD1BWP7T \Sreg_reg[0] ( .D(n296), .CP(n14), .Q(Sreg[0]) );
    NR4D1BWP7T U1 ( .A1(n128), .A2(i[31]), .A3(F), .A4(reset), .ZN(n115) );
    INVD1BWP7T U2 ( .I(reset), .ZN(n117) );
    CKBD1BWP7T U3 ( .I(n3), .Z(n15) );
    CKBD1BWP7T U4 ( .I(n4), .Z(n16) );
    CKBD1BWP7T U5 ( .I(n1), .Z(n8) );
    CKBD1BWP7T U6 ( .I(n1), .Z(n9) );
    CKBD1BWP7T U7 ( .I(n2), .Z(n10) );
    CKBD1BWP7T U8 ( .I(n2), .Z(n11) );
    CKBD1BWP7T U9 ( .I(n1), .Z(n7) );
    CKBD1BWP7T U10 ( .I(n3), .Z(n14) );
    CKBD1BWP7T U11 ( .I(n3), .Z(n13) );
    CKBD1BWP7T U12 ( .I(n2), .Z(n12) );
    CKBD1BWP7T U13 ( .I(n4), .Z(n17) );
    CKBD1BWP7T U14 ( .I(n6), .Z(n1) );
    CKBD1BWP7T U15 ( .I(n5), .Z(n3) );
    CKBD1BWP7T U16 ( .I(n6), .Z(n2) );
    CKBD1BWP7T U17 ( .I(n5), .Z(n4) );
    CKBD1BWP7T U18 ( .I(clk), .Z(n5) );
    CKBD1BWP7T U19 ( .I(clk), .Z(n6) );
    TIELBWP7T U20 ( .ZN(n118) );
    MUX2ND0BWP7T U21 ( .I0(n100), .I1(n18), .S(n19), .ZN(n344) );
    OAI22D0BWP7T U22 ( .A1(n212), .A2(n100), .B1(n18), .B2(n20), .ZN(n343) );
    OAI22D0BWP7T U23 ( .A1(n100), .A2(n21), .B1(n18), .B2(n22), .ZN(n342) );
    OAI22D0BWP7T U24 ( .A1(n231), .A2(n100), .B1(n18), .B2(n23), .ZN(n341) );
    OAI22D0BWP7T U25 ( .A1(n230), .A2(n100), .B1(n18), .B2(n24), .ZN(n340) );
    OAI22D0BWP7T U26 ( .A1(n100), .A2(n25), .B1(n18), .B2(n26), .ZN(n339) );
    CKND0BWP7T U27 ( .I(N187), .ZN(n26) );
    OAI22D0BWP7T U28 ( .A1(n229), .A2(n100), .B1(n18), .B2(n27), .ZN(n338) );
    CKND0BWP7T U29 ( .I(N188), .ZN(n27) );
    OAI22D0BWP7T U30 ( .A1(n228), .A2(n100), .B1(n18), .B2(n28), .ZN(n337) );
    CKND0BWP7T U31 ( .I(N189), .ZN(n28) );
    OAI22D0BWP7T U32 ( .A1(n227), .A2(n100), .B1(n18), .B2(n29), .ZN(n336) );
    CKND0BWP7T U33 ( .I(N190), .ZN(n29) );
    OAI22D0BWP7T U34 ( .A1(n226), .A2(n100), .B1(n18), .B2(n30), .ZN(n335) );
    CKND0BWP7T U35 ( .I(N191), .ZN(n30) );
```

```
    OAI22D0BWP7T U37 ( .A1(n225), .A2(n100), .B1(n18), .B2(n31), .ZN(n334) );
    CKND0BWP7T U38 ( .I(N192), .ZN(n31) );
    MOAI22D0BWP7T U39 ( .A1(n18), .A2(n32), .B1(F), .B2(i[10]), .ZN(n333) );
    CKND0BWP7T U40 ( .I(N193), .ZN(n32) );
    MOAI22D0BWP7T U41 ( .A1(n18), .A2(n33), .B1(F), .B2(i[11]), .ZN(n332) );
    CKND0BWP7T U42 ( .I(N194), .ZN(n33) );
    MOAI22D0BWP7T U43 ( .A1(n18), .A2(n34), .B1(F), .B2(i[12]), .ZN(n331) );
    CKND0BWP7T U44 ( .I(N195), .ZN(n34) );
    OAI22D0BWP7T U45 ( .A1(n224), .A2(n100), .B1(n18), .B2(n35), .ZN(n330) );
    OAI22D0BWP7T U46 ( .A1(n223), .A2(n100), .B1(n18), .B2(n36), .ZN(n329) );
    OAI22D0BWP7T U47 ( .A1(n222), .A2(n100), .B1(n18), .B2(n37), .ZN(n328) );
    OAI22D0BWP7T U48 ( .A1(n221), .A2(n100), .B1(n18), .B2(n38), .ZN(n327) );
    MOAI22D0BWP7T U49 ( .A1(n18), .A2(n39), .B1(F), .B2(i[17]), .ZN(n326) );
    MOAI22D0BWP7T U50 ( .A1(n18), .A2(n40), .B1(F), .B2(i[18]), .ZN(n325) );
    OAI22D0BWP7T U51 ( .A1(n100), .A2(n41), .B1(n18), .B2(n42), .ZN(n324) );
    OAI22D0BWP7T U52 ( .A1(n220), .A2(n100), .B1(n18), .B2(n43), .ZN(n323) );
    CKND0BWP7T U53 ( .I(N203), .ZN(n43) );
    OAI22D0BWP7T U54 ( .A1(n219), .A2(n100), .B1(n18), .B2(n44), .ZN(n322) );
    CKND0BWP7T U55 ( .I(N204), .ZN(n44) );
    OAI22D0BWP7T U56 ( .A1(n218), .A2(n100), .B1(n18), .B2(n45), .ZN(n321) );
    CKND0BWP7T U57 ( .I(N205), .ZN(n45) );
    OAI22D0BWP7T U58 ( .A1(n217), .A2(n100), .B1(n18), .B2(n46), .ZN(n320) );
    CKND0BWP7T U59 ( .I(N206), .ZN(n46) );
    OAI22D0BWP7T U60 ( .A1(n100), .A2(n47), .B1(n18), .B2(n48), .ZN(n319) );
    OAI22D0BWP7T U61 ( .A1(n100), .A2(n49), .B1(n18), .B2(n50), .ZN(n318) );
    OAI22D0BWP7T U62 ( .A1(n100), .A2(n51), .B1(n18), .B2(n52), .ZN(n317) );
    OAI22D0BWP7T U63 ( .A1(n216), .A2(n100), .B1(n18), .B2(n53), .ZN(n316) );
    CKND0BWP7T U64 ( .I(N210), .ZN(n53) );
    OAI22D0BWP7T U65 ( .A1(n215), .A2(n100), .B1(n18), .B2(n54), .ZN(n315) );
    CKND0BWP7T U66 ( .I(N211), .ZN(n54) );
    OAI22D0BWP7T U67 ( .A1(n214), .A2(n100), .B1(n18), .B2(n55), .ZN(n314) );
    CKND0BWP7T U68 ( .I(N212), .ZN(n55) );
    OAI22D0BWP7T U69 ( .A1(n213), .A2(n100), .B1(n18), .B2(n56), .ZN(n313) );
    CKND0BWP7T U70 ( .I(N213), .ZN(n56) );
    OA32D0BWP7T U71 ( .A1(n57), .A2(n58), .A3(n59), .B1(n60), .B2(n20), .Z(n18)
          );
    CKND0BWP7T U72 ( .I(N214), .ZN(n20) );
    ND4D0BWP7T U73 ( .A1(n40), .A2(n42), .A3(n39), .A4(n61), .ZN(n59) );
    NR4D0BWP7T U74 ( .A1(N206), .A2(N205), .A3(N204), .A4(N203), .ZN(n61) );
    CKND0BWP7T U75 ( .I(N200), .ZN(n39) );
    CKND0BWP7T U76 ( .I(N202), .ZN(n42) );
    CKND0BWP7T U77 ( .I(N201), .ZN(n40) );
    ND4D0BWP7T U78 ( .A1(n50), .A2(n52), .A3(n48), .A4(n62), .ZN(n58) );
    NR4D0BWP7T U79 ( .A1(N213), .A2(N212), .A3(N211), .A4(N210), .ZN(n62) );
    CKND0BWP7T U80 ( .I(N207), .ZN(n48) );
    CKND0BWP7T U81 ( .I(N209), .ZN(n52) );
    CKND0BWP7T U82 ( .I(N208), .ZN(n50) );
    IND4D0BWP7T U83 ( .A1(n60), .B1(n63), .B2(n64), .B3(n65), .ZN(n57) );
    NR4D0BWP7T U84 ( .A1(n66), .A2(N193), .A3(N195), .A4(N194), .ZN(n65) );
    ND4D0BWP7T U85 ( .A1(n35), .A2(n36), .A3(n37), .A4(n38), .ZN(n66) );
    CKND0BWP7T U86 ( .I(N199), .ZN(n38) );
    CKND0BWP7T U87 ( .I(N198), .ZN(n37) );
    CKND0BWP7T U88 ( .I(N197), .ZN(n36) );
    CKND0BWP7T U89 ( .I(N196), .ZN(n35) );
```

```
    NR4D0BWP7T U90 ( .A1(N192), .A2(N191), .A3(N190), .A4(N189), .ZN(n64) );
    AOI21D0BWP7T U91 ( .A1(N187), .A2(n67), .B(N188), .ZN(n63) );
    OAI211D0BWP7T U92 ( .A1(i[0]), .A2(n22), .B(n23), .C(n24), .ZN(n67) );
    CKND0BWP7T U93 ( .I(N186), .ZN(n24) );
    CKND0BWP7T U94 ( .I(N185), .ZN(n23) );
    CKND0BWP7T U95 ( .I(N184), .ZN(n22) );
    OAI21D0BWP7T U96 ( .A1(n68), .A2(i[31]), .B(n100), .ZN(n60) );
    AOI21D0BWP7T U97 ( .A1(i[4]), .A2(n69), .B(n70), .ZN(n68) );
    OAI21D0BWP7T U98 ( .A1(n19), .A2(n21), .B(n71), .ZN(n69) );
    OAI22D0BWP7T U99 ( .A1(n72), .A2(n73), .B1(n74), .B2(n75), .ZN(n312) );
    OAI222D0BWP7T U100 ( .A1(n74), .A2(n76), .B1(n77), .B2(n78), .C1(n72), .C2(
         n79), .ZN(n311) );
    CKND0BWP7T U101 ( .I(Sreg[15]), .ZN(n76) );
    OAI222D0BWP7T U102 ( .A1(n74), .A2(n80), .B1(n77), .B2(n81), .C1(n75), .C2(
         n72), .ZN(n310) );
    OAI222D0BWP7T U103 ( .A1(n74), .A2(n82), .B1(n77), .B2(n83), .C1(n72), .C2(
         n80), .ZN(n309) );
    OAI222D0BWP7T U104 ( .A1(n74), .A2(n84), .B1(n77), .B2(n85), .C1(n72), .C2(
         n82), .ZN(n308) );
    OAI222D0BWP7T U105 ( .A1(n74), .A2(n86), .B1(n77), .B2(n87), .C1(n72), .C2(
         n84), .ZN(n307) );
    OAI222D0BWP7T U106 ( .A1(n74), .A2(n88), .B1(n77), .B2(n89), .C1(n72), .C2(
         n86), .ZN(n306) );
    OAI222D0BWP7T U107 ( .A1(n74), .A2(n90), .B1(n77), .B2(n91), .C1(n72), .C2(
         n88), .ZN(n305) );
    OAI222D0BWP7T U108 ( .A1(n74), .A2(n92), .B1(n77), .B2(n93), .C1(n72), .C2(
         n90), .ZN(n304) );
    OAI222D0BWP7T U109 ( .A1(n74), .A2(n94), .B1(n77), .B2(n95), .C1(n72), .C2(
         n92), .ZN(n303) );
    OAI222D0BWP7T U110 ( .A1(n74), .A2(n96), .B1(n77), .B2(n97), .C1(n72), .C2(
         n94), .ZN(n302) );
    OAI222D0BWP7T U111 ( .A1(n74), .A2(n98), .B1(n77), .B2(n99), .C1(n72), .C2(
         n96), .ZN(n301) );
    OAI222D0BWP7T U112 ( .A1(n74), .A2(n101), .B1(n77), .B2(n102), .C1(n72),
         .C2(n98), .ZN(n300) );
    OAI222D0BWP7T U113 ( .A1(n74), .A2(n103), .B1(n77), .B2(n104), .C1(n72),
         .C2(n101), .ZN(n299) );
    OAI222D0BWP7T U114 ( .A1(n74), .A2(n105), .B1(n77), .B2(n106), .C1(n72),
         .C2(n103), .ZN(n298) );
    OAI222D0BWP7T U115 ( .A1(n74), .A2(n79), .B1(n77), .B2(n107), .C1(n72), .C2(
         n105), .ZN(n297) );
    CKND2D0BWP7T U116 ( .A1(n108), .A2(n72), .ZN(n77) );
    CKND2D0BWP7T U117 ( .A1(n109), .A2(n72), .ZN(n74) );
    OAI21D0BWP7T U118 ( .A1(reset), .A2(n110), .B(n111), .ZN(n72) );
    AO22D0BWP7T U119 ( .A1(Sreg[0]), .A2(n111), .B1(Dividend[0]), .B2(n112), .Z(
         n296) );
    AO222D0BWP7T U120 ( .A1(Dividend[1]), .A2(n112), .B1(n113), .B2(Sreg[0]),
         .C1(Sreg[1]), .C2(n111), .Z(n295) );
    AO222D0BWP7T U121 ( .A1(Dividend[2]), .A2(n112), .B1(Sreg[1]), .B2(n113),
         .C1(Sreg[2]), .C2(n111), .Z(n294) );
    AO222D0BWP7T U122 ( .A1(Dividend[3]), .A2(n112), .B1(Sreg[2]), .B2(n113),
         .C1(Sreg[3]), .C2(n111), .Z(n293) );
    AO222D0BWP7T U123 ( .A1(Dividend[4]), .A2(n112), .B1(Sreg[3]), .B2(n113),
         .C1(Sreg[4]), .C2(n111), .Z(n292) );
```

```
    AO222D0BWP7T U124 ( .A1(Dividend[5]), .A2(n112), .B1(Sreg[4]), .B2(n113),
         .C1(Sreg[5]), .C2(n111), .Z(n291) );
    AO222D0BWP7T U125 ( .A1(Dividend[6]), .A2(n112), .B1(Sreg[5]), .B2(n113),
         .C1(Sreg[6]), .C2(n111), .Z(n290) );
    AO222D0BWP7T U126 ( .A1(Dividend[7]), .A2(n112), .B1(Sreg[6]), .B2(n113),
         .C1(Sreg[7]), .C2(n111), .Z(n289) );
    AO222D0BWP7T U127 ( .A1(Dividend[8]), .A2(n112), .B1(Sreg[7]), .B2(n113),
         .C1(Sreg[8]), .C2(n111), .Z(n288) );
    AO222D0BWP7T U128 ( .A1(Dividend[9]), .A2(n112), .B1(Sreg[8]), .B2(n113),
         .C1(Sreg[9]), .C2(n111), .Z(n287) );
    AO222D0BWP7T U129 ( .A1(Dividend[10]), .A2(n112), .B1(Sreg[9]), .B2(n113),
         .C1(Sreg[10]), .C2(n111), .Z(n286) );
    AO222D0BWP7T U130 ( .A1(Dividend[11]), .A2(n112), .B1(Sreg[10]), .B2(n113),
         .C1(Sreg[11]), .C2(n111), .Z(n285) );
    AO222D0BWP7T U131 ( .A1(Dividend[12]), .A2(n112), .B1(Sreg[11]), .B2(n113),
         .C1(Sreg[12]), .C2(n111), .Z(n284) );
    AO222D0BWP7T U132 ( .A1(Dividend[13]), .A2(n112), .B1(Sreg[12]), .B2(n113),
         .C1(Sreg[13]), .C2(n111), .Z(n283) );
    AO222D0BWP7T U133 ( .A1(Dividend[14]), .A2(n112), .B1(Sreg[13]), .B2(n113),
         .C1(Sreg[14]), .C2(n111), .Z(n282) );
    AO222D0BWP7T U134 ( .A1(Dividend[15]), .A2(n112), .B1(Sreg[14]), .B2(n113),
         .C1(Sreg[15]), .C2(n111), .Z(n281) );
    NR2D0BWP7T U135 ( .A1(n114), .A2(n111), .ZN(n113) );
    NR2D0BWP7T U136 ( .A1(n109), .A2(n111), .ZN(n112) );
    AOI21D0BWP7T U137 ( .A1(n117), .A2(n109), .B(n115), .ZN(n111) );
    CKND0BWP7T U138 ( .I(n114), .ZN(n109) );
    CKND2D0BWP7T U139 ( .A1(n116), .A2(n100), .ZN(n114) );
    MUX2ND0BWP7T U140 ( .I0(n119), .I1(S[15]), .S(n120), .ZN(n280) );
    CKND0BWP7T U141 ( .I(quo[0]), .ZN(n119) );
    CKMUX2D0BWP7T U142 ( .I0(Quotient[0]), .I1(quo[0]), .S(n121), .Z(n279) );
    CKMUX2D0BWP7T U143 ( .I0(quo[1]), .I1(quo[0]), .S(n120), .Z(n278) );
    CKMUX2D0BWP7T U144 ( .I0(Quotient[1]), .I1(quo[1]), .S(n121), .Z(n277) );
    CKMUX2D0BWP7T U145 ( .I0(quo[2]), .I1(quo[1]), .S(n120), .Z(n276) );
    CKMUX2D0BWP7T U146 ( .I0(Quotient[2]), .I1(quo[2]), .S(n121), .Z(n275) );
    CKMUX2D0BWP7T U147 ( .I0(quo[3]), .I1(quo[2]), .S(n120), .Z(n274) );
    CKMUX2D0BWP7T U148 ( .I0(Quotient[3]), .I1(quo[3]), .S(n121), .Z(n273) );
    CKMUX2D0BWP7T U149 ( .I0(quo[4]), .I1(quo[3]), .S(n120), .Z(n272) );
    CKMUX2D0BWP7T U150 ( .I0(Quotient[4]), .I1(quo[4]), .S(n121), .Z(n271) );
    CKMUX2D0BWP7T U151 ( .I0(quo[5]), .I1(quo[4]), .S(n120), .Z(n270) );
    CKMUX2D0BWP7T U152 ( .I0(Quotient[5]), .I1(quo[5]), .S(n121), .Z(n269) );
    CKMUX2D0BWP7T U153 ( .I0(quo[6]), .I1(quo[5]), .S(n120), .Z(n268) );
    CKMUX2D0BWP7T U154 ( .I0(Quotient[6]), .I1(quo[6]), .S(n121), .Z(n267) );
    CKMUX2D0BWP7T U155 ( .I0(quo[7]), .I1(quo[6]), .S(n120), .Z(n266) );
    CKMUX2D0BWP7T U156 ( .I0(Quotient[7]), .I1(quo[7]), .S(n121), .Z(n265) );
    CKMUX2D0BWP7T U157 ( .I0(quo[8]), .I1(quo[7]), .S(n120), .Z(n264) );
    CKMUX2D0BWP7T U158 ( .I0(Quotient[8]), .I1(quo[8]), .S(n121), .Z(n263) );
    CKMUX2D0BWP7T U159 ( .I0(quo[9]), .I1(quo[8]), .S(n120), .Z(n262) );
    CKMUX2D0BWP7T U160 ( .I0(Quotient[9]), .I1(quo[9]), .S(n121), .Z(n261) );
    CKMUX2D0BWP7T U161 ( .I0(quo[10]), .I1(quo[9]), .S(n120), .Z(n260) );
    CKMUX2D0BWP7T U162 ( .I0(Quotient[10]), .I1(quo[10]), .S(n121), .Z(n259) );
    CKMUX2D0BWP7T U163 ( .I0(quo[11]), .I1(quo[10]), .S(n120), .Z(n258) );
    CKMUX2D0BWP7T U164 ( .I0(Quotient[11]), .I1(quo[11]), .S(n121), .Z(n257) );
    CKMUX2D0BWP7T U165 ( .I0(quo[12]), .I1(quo[11]), .S(n120), .Z(n256) );
    CKMUX2D0BWP7T U166 ( .I0(Quotient[12]), .I1(quo[12]), .S(n121), .Z(n255) );
```

```
    CKMUX2D0BWP7T U167 ( .I0(quo[13]), .I1(quo[12]), .S(n120), .Z(n254) );
    CKMUX2D0BWP7T U168 ( .I0(Quotient[13]), .I1(quo[13]), .S(n121), .Z(n253) );
    CKMUX2D0BWP7T U169 ( .I0(quo[14]), .I1(quo[13]), .S(n120), .Z(n252) );
    CKMUX2D0BWP7T U170 ( .I0(Quotient[14]), .I1(quo[14]), .S(n121), .Z(n251) );
    CKMUX2D0BWP7T U171 ( .I0(quo[15]), .I1(quo[14]), .S(n120), .Z(n250) );
    AN3D0BWP7T U172 ( .A1(F), .A2(n117), .A3(n116), .Z(n120) );
    CKMUX2D0BWP7T U173 ( .I0(Quotient[15]), .I1(quo[15]), .S(n121), .Z(n249) );
    AN3D0BWP7T U174 ( .A1(i[1]), .A2(n19), .A3(n122), .Z(n121) );
    OAI222D0BWP7T U175 ( .A1(n79), .A2(n123), .B1(n78), .B2(n124), .C1(n211),
         .C2(n125), .ZN(n248) );
    CKND0BWP7T U176 ( .I(S[0]), .ZN(n78) );
    CKND0BWP7T U177 ( .I(Sreg[16]), .ZN(n79) );
    OAI222D0BWP7T U178 ( .A1(n105), .A2(n123), .B1(n107), .B2(n124), .C1(n210),
         .C2(n125), .ZN(n247) );
    CKND0BWP7T U179 ( .I(S[1]), .ZN(n107) );
    CKND0BWP7T U180 ( .I(Sreg[17]), .ZN(n105) );
    OAI222D0BWP7T U181 ( .A1(n103), .A2(n123), .B1(n106), .B2(n124), .C1(n209),
         .C2(n125), .ZN(n246) );
    CKND0BWP7T U182 ( .I(S[2]), .ZN(n106) );
    CKND0BWP7T U183 ( .I(Sreg[18]), .ZN(n103) );
    OAI222D0BWP7T U184 ( .A1(n101), .A2(n123), .B1(n104), .B2(n124), .C1(n208),
         .C2(n125), .ZN(n245) );
    CKND0BWP7T U185 ( .I(S[3]), .ZN(n104) );
    CKND0BWP7T U186 ( .I(Sreg[19]), .ZN(n101) );
    OAI222D0BWP7T U187 ( .A1(n98), .A2(n123), .B1(n102), .B2(n124), .C1(n207),
         .C2(n125), .ZN(n244) );
    CKND0BWP7T U188 ( .I(S[4]), .ZN(n102) );
    CKND0BWP7T U189 ( .I(Sreg[20]), .ZN(n98) );
    OAI222D0BWP7T U190 ( .A1(n96), .A2(n123), .B1(n99), .B2(n124), .C1(n206),
         .C2(n125), .ZN(n243) );
    CKND0BWP7T U191 ( .I(S[5]), .ZN(n99) );
    CKND0BWP7T U192 ( .I(Sreg[21]), .ZN(n96) );
    OAI222D0BWP7T U193 ( .A1(n94), .A2(n123), .B1(n97), .B2(n124), .C1(n205),
         .C2(n125), .ZN(n242) );
    CKND0BWP7T U194 ( .I(S[6]), .ZN(n97) );
    CKND0BWP7T U195 ( .I(Sreg[22]), .ZN(n94) );
    OAI222D0BWP7T U196 ( .A1(n92), .A2(n123), .B1(n95), .B2(n124), .C1(n204),
         .C2(n125), .ZN(n241) );
    CKND0BWP7T U197 ( .I(S[7]), .ZN(n95) );
    CKND0BWP7T U198 ( .I(Sreg[23]), .ZN(n92) );
    OAI222D0BWP7T U199 ( .A1(n90), .A2(n123), .B1(n93), .B2(n124), .C1(n203),
         .C2(n125), .ZN(n240) );
    CKND0BWP7T U200 ( .I(S[8]), .ZN(n93) );
    CKND0BWP7T U201 ( .I(Sreg[24]), .ZN(n90) );
    OAI222D0BWP7T U202 ( .A1(n88), .A2(n123), .B1(n91), .B2(n124), .C1(n202),
         .C2(n125), .ZN(n239) );
    CKND0BWP7T U203 ( .I(S[9]), .ZN(n91) );
    CKND0BWP7T U204 ( .I(Sreg[25]), .ZN(n88) );
    OAI222D0BWP7T U205 ( .A1(n86), .A2(n123), .B1(n89), .B2(n124), .C1(n201),
         .C2(n125), .ZN(n238) );
    CKND0BWP7T U206 ( .I(S[10]), .ZN(n89) );
    CKND0BWP7T U207 ( .I(Sreg[26]), .ZN(n86) );
    OAI222D0BWP7T U208 ( .A1(n84), .A2(n123), .B1(n87), .B2(n124), .C1(n200),
         .C2(n125), .ZN(n237) );
```

```
    CKND0BWP7T U209 ( .I(S[11]), .ZN(n87) );
    CKND0BWP7T U210 ( .I(Sreg[27]), .ZN(n84) );
    OAI222D0BWP7T U211 ( .A1(n82), .A2(n123), .B1(n85), .B2(n124), .C1(n199),
         .C2(n125), .ZN(n236) );
    CKND0BWP7T U212 ( .I(S[12]), .ZN(n85) );
    CKND0BWP7T U213 ( .I(Sreg[28]), .ZN(n82) );
    OAI222D0BWP7T U214 ( .A1(n80), .A2(n123), .B1(n83), .B2(n124), .C1(n198),
         .C2(n125), .ZN(n235) );
    CKND0BWP7T U215 ( .I(S[13]), .ZN(n83) );
    CKND0BWP7T U216 ( .I(Sreg[29]), .ZN(n80) );
    OAI222D0BWP7T U217 ( .A1(n75), .A2(n123), .B1(n81), .B2(n124), .C1(n197),
         .C2(n125), .ZN(n234) );
    CKND2D0BWP7T U218 ( .A1(n125), .A2(n108), .ZN(n124) );
    CKND0BWP7T U219 ( .I(n110), .ZN(n108) );
    CKND0BWP7T U220 ( .I(S[14]), .ZN(n81) );
    CKND0BWP7T U221 ( .I(Sreg[30]), .ZN(n75) );
    MOAI22D0BWP7T U222 ( .A1(n73), .A2(n123), .B1(Remainder[15]), .B2(n126),
         .ZN(n233) );
    CKND2D0BWP7T U223 ( .A1(n125), .A2(n110), .ZN(n123) );
    IND3D0BWP7T U224 ( .A1(S[15]), .B1(F), .B2(n116), .ZN(n110) );
    IINR4D0BWP7T U225 ( .A1(n127), .A2(n128), .B1(i[31]), .B2(n70), .ZN(n116) );
    OAI21D0BWP7T U226 ( .A1(i[1]), .A2(n129), .B(i[4]), .ZN(n127) );
    CKND0BWP7T U227 ( .I(n126), .ZN(n125) );
    ND3D0BWP7T U228 ( .A1(i[0]), .A2(n21), .A3(n122), .ZN(n126) );
    AN4D0BWP7T U229 ( .A1(F), .A2(n117), .A3(n71), .A4(n130), .Z(n122) );
    NR3D0BWP7T U230 ( .A1(i[31]), .A2(n25), .A3(n70), .ZN(n130) );
    CKND0BWP7T U231 ( .I(n129), .ZN(n71) );
    CKND0BWP7T U232 ( .I(Sreg[31]), .ZN(n73) );
    CKMUX2D0BWP7T U233 ( .I0(Divisor_temp[0]), .I1(Divisor[0]), .S(n115), .Z(
         n194) );
    CKMUX2D0BWP7T U234 ( .I0(Divisor_temp[1]), .I1(Divisor[1]), .S(n115), .Z(
         n193) );
    CKMUX2D0BWP7T U235 ( .I0(Divisor_temp[2]), .I1(Divisor[2]), .S(n115), .Z(
         n192) );
    CKMUX2D0BWP7T U236 ( .I0(Divisor_temp[3]), .I1(Divisor[3]), .S(n115), .Z(
         n191) );
    CKMUX2D0BWP7T U237 ( .I0(Divisor_temp[4]), .I1(Divisor[4]), .S(n115), .Z(
         n190) );
    CKMUX2D0BWP7T U238 ( .I0(Divisor_temp[5]), .I1(Divisor[5]), .S(n115), .Z(
         n189) );
    CKMUX2D0BWP7T U239 ( .I0(Divisor_temp[6]), .I1(Divisor[6]), .S(n115), .Z(
         n188) );
    CKMUX2D0BWP7T U240 ( .I0(Divisor_temp[7]), .I1(Divisor[7]), .S(n115), .Z(
         n187) );
    CKMUX2D0BWP7T U241 ( .I0(Divisor_temp[8]), .I1(Divisor[8]), .S(n115), .Z(
         n186) );
    CKMUX2D0BWP7T U242 ( .I0(Divisor_temp[9]), .I1(Divisor[9]), .S(n115), .Z(
         n185) );
    CKMUX2D0BWP7T U243 ( .I0(Divisor_temp[10]), .I1(Divisor[10]), .S(n115), .Z(
         n184) );
    CKMUX2D0BWP7T U244 ( .I0(Divisor_temp[11]), .I1(Divisor[11]), .S(n115), .Z(
         n183) );
    CKMUX2D0BWP7T U245 ( .I0(Divisor_temp[12]), .I1(Divisor[12]), .S(n115), .Z(
         n182) );
```

```
  CKMUX2D0BWP7T U246 ( .I0(Divisor_temp[13]), .I1(Divisor[13]), .S(n115), .Z(
       n181) );
  CKMUX2D0BWP7T U247 ( .I0(Divisor_temp[14]), .I1(Divisor[14]), .S(n115), .Z(
       n180) );
  CKMUX2D0BWP7T U248 ( .I0(Divisor_temp[15]), .I1(Divisor[15]), .S(n115), .Z(
       n179) );
  ND4D0BWP7T U249 ( .A1(n21), .A2(n25), .A3(n19), .A4(n131), .ZN(n128) );
  NR2D0BWP7T U250 ( .A1(n129), .A2(n70), .ZN(n131) );
  IIND4D0BWP7T U251 ( .A1(n132), .A2(n133), .B1(n134), .B2(n135), .ZN(n70) );
  NR4D0BWP7T U252 ( .A1(n136), .A2(i[12]), .A3(i[18]), .A4(i[17]), .ZN(n135)
       );
  ND4D0BWP7T U253 ( .A1(n41), .A2(n47), .A3(n49), .A4(n51), .ZN(n136) );
  CKND0BWP7T U254 ( .I(i[26]), .ZN(n51) );
  CKND0BWP7T U255 ( .I(i[25]), .ZN(n49) );
  CKND0BWP7T U256 ( .I(i[24]), .ZN(n47) );
  CKND0BWP7T U257 ( .I(i[19]), .ZN(n41) );
  INR4D0BWP7T U258 ( .A1(n213), .B1(n137), .B2(i[11]), .B3(i[10]), .ZN(n134)
       );
  ND3D0BWP7T U259 ( .A1(n215), .A2(n214), .A3(n216), .ZN(n137) );
  IND4D0BWP7T U260 ( .A1(n138), .B1(n223), .B2(n221), .B3(n222), .ZN(n133) );
  ND4D0BWP7T U261 ( .A1(n220), .A2(n219), .A3(n218), .A4(n217), .ZN(n138) );
  IND4D0BWP7T U262 ( .A1(n139), .B1(n229), .B2(n227), .B3(n228), .ZN(n132) );
  ND3D0BWP7T U263 ( .A1(n225), .A2(n224), .A3(n226), .ZN(n139) );
  CKND2D0BWP7T U264 ( .A1(n231), .A2(n230), .ZN(n129) );
  CKND0BWP7T U265 ( .I(i[0]), .ZN(n19) );
  CKND0BWP7T U266 ( .I(i[4]), .ZN(n25) );
  CKND0BWP7T U267 ( .I(i[1]), .ZN(n21) );
  CKND0BWP7T U268 ( .I(F), .ZN(n100) );
endmodule


module divisor_16bit_IO ( Divisor, Dividend, clk, reset, Quotient, Remainder,
       Sreg );
  input [15:0] Divisor;
  input [15:0] Dividend;
  output [15:0] Quotient;
  output [15:0] Remainder;
  output [31:0] Sreg;
  input clk, reset;
  wire clk_w, reset_w, n1, n2;
  wire [15:0] Divisor_w;
  wire [15:0] Dividend_w;
  wire [15:0] Quotient_w;
  wire [15:0] Remainder_w;
  wire [31:0] Sreg_w;
  tri [15:0] Divisor;
  tri [15:0] Dividend;
  tri clk;
  tri reset;
  tri [15:0] Quotient;
  tri [15:0] Remainder;
  tri [31:0] Sreg;

  divisor_16bit div_inst ( .Divisor(Divisor_w), .Dividend(Dividend_w), .clk(
```

```
        clk_w), .reset(reset_w), .Quotient(Quotient_w), .Remainder(Remainder_w)
            , .Sreg(Sreg_w) );
    PDDW0204SCDG U_Divd0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[0]), .DS(n1),
        .PE(n1), .C(Dividend_w[0]) );
    PDDW0204SCDG U_Divd1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[1]), .DS(n1),
        .PE(n1), .C(Dividend_w[1]) );
    PDDW0204SCDG U_Divd2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[2]), .DS(n1),
        .PE(n1), .C(Dividend_w[2]) );
    PDDW0204SCDG U_Divd3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[3]), .DS(n1),
        .PE(n1), .C(Dividend_w[3]) );
    PDDW0204SCDG U_Divd4 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[4]), .DS(n1),
        .PE(n1), .C(Dividend_w[4]) );
    PDDW0204SCDG U_Divd5 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[5]), .DS(n1),
        .PE(n1), .C(Dividend_w[5]) );
    PDDW0204SCDG U_Divd6 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[6]), .DS(n1),
        .PE(n1), .C(Dividend_w[6]) );
    PDDW0204SCDG U_Divd7 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[7]), .DS(n1),
        .PE(n1), .C(Dividend_w[7]) );
    PDDW0204SCDG U_Divd8 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[8]), .DS(n1),
        .PE(n1), .C(Dividend_w[8]) );
    PDDW0204SCDG U_Divd9 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[9]), .DS(n1),
        .PE(n1), .C(Dividend_w[9]) );
    PDDW0204SCDG U_Divd10 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[10]), .DS(
        n1), .PE(n1), .C(Dividend_w[10]) );
    PDDW0204SCDG U_Divd11 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[11]), .DS(
        n1), .PE(n1), .C(Dividend_w[11]) );
    PDDW0204SCDG U_Divd12 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[12]), .DS(
        n1), .PE(n1), .C(Dividend_w[12]) );
    PDDW0204SCDG U_Divd13 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[13]), .DS(
        n1), .PE(n1), .C(Dividend_w[13]) );
    PDDW0204SCDG U_Divd14 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[14]), .DS(
        n1), .PE(n1), .C(Dividend_w[14]) );
    PDDW0204SCDG U_Divd15 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Dividend[15]), .DS(
        n1), .PE(n1), .C(Dividend_w[15]) );
    PDDW0204SCDG U_Div0 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[0]), .DS(n1),
        .PE(n1), .C(Divisor_w[0]) );
    PDDW0204SCDG U_Div1 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[1]), .DS(n1),
        .PE(n1), .C(Divisor_w[1]) );
    PDDW0204SCDG U_Div2 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[2]), .DS(n1),
        .PE(n1), .C(Divisor_w[2]) );
    PDDW0204SCDG U_Div3 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[3]), .DS(n1),
        .PE(n1), .C(Divisor_w[3]) );
    PDDW0204SCDG U_Div4 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[4]), .DS(n1),
        .PE(n1), .C(Divisor_w[4]) );
    PDDW0204SCDG U_Div5 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[5]), .DS(n1),
        .PE(n1), .C(Divisor_w[5]) );
    PDDW0204SCDG U_Div6 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[6]), .DS(n1),
        .PE(n1), .C(Divisor_w[6]) );
    PDDW0204SCDG U_Div7 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[7]), .DS(n1),
        .PE(n1), .C(Divisor_w[7]) );
    PDDW0204SCDG U_Div8 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[8]), .DS(n1),
        .PE(n1), .C(Divisor_w[8]) );
    PDDW0204SCDG U_Div9 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[9]), .DS(n1),
        .PE(n1), .C(Divisor_w[9]) );
```

```
    PDDW0204SCDG U_Div10 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[10]), .DS(n1),
        .PE(n1), .C(Divisor_w[10]) );
    PDDW0204SCDG U_Div11 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[11]), .DS(n1),
        .PE(n1), .C(Divisor_w[11]) );
    PDDW0204SCDG U_Div12 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[12]), .DS(n1),
        .PE(n1), .C(Divisor_w[12]) );
    PDDW0204SCDG U_Div13 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[13]), .DS(n1),
        .PE(n1), .C(Divisor_w[13]) );
    PDDW0204SCDG U_Div14 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[14]), .DS(n1),
        .PE(n1), .C(Divisor_w[14]) );
    PDDW0204SCDG U_Div15 ( .I(n1), .OEN(n2), .IE(n2), .PAD(Divisor[15]), .DS(n1),
        .PE(n1), .C(Divisor_w[15]) );
    PDDW0204SCDG U_clk ( .I(n1), .OEN(n2), .IE(n2), .PAD(clk), .DS(n1), .PE(n1),
        .C(clk_w) );
    PDDW0204SCDG U_reset ( .I(n1), .OEN(n2), .IE(n2), .PAD(reset), .DS(n1), .PE(
        n1), .C(reset_w) );
    PDDW0204SCDG U_Quot0 ( .I(Quotient_w[0]), .OEN(n1), .IE(n1), .PAD(
        Quotient[0]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot1 ( .I(Quotient_w[1]), .OEN(n1), .IE(n1), .PAD(
        Quotient[1]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot2 ( .I(Quotient_w[2]), .OEN(n1), .IE(n1), .PAD(
        Quotient[2]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot3 ( .I(Quotient_w[3]), .OEN(n1), .IE(n1), .PAD(
        Quotient[3]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot4 ( .I(Quotient_w[4]), .OEN(n1), .IE(n1), .PAD(
        Quotient[4]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot5 ( .I(Quotient_w[5]), .OEN(n1), .IE(n1), .PAD(
        Quotient[5]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot6 ( .I(Quotient_w[6]), .OEN(n1), .IE(n1), .PAD(
        Quotient[6]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot7 ( .I(Quotient_w[7]), .OEN(n1), .IE(n1), .PAD(
        Quotient[7]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot8 ( .I(Quotient_w[8]), .OEN(n1), .IE(n1), .PAD(
        Quotient[8]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot9 ( .I(Quotient_w[9]), .OEN(n1), .IE(n1), .PAD(
        Quotient[9]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot10 ( .I(Quotient_w[10]), .OEN(n1), .IE(n1), .PAD(
        Quotient[10]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot11 ( .I(Quotient_w[11]), .OEN(n1), .IE(n1), .PAD(
        Quotient[11]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot12 ( .I(Quotient_w[12]), .OEN(n1), .IE(n1), .PAD(
        Quotient[12]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot13 ( .I(Quotient_w[13]), .OEN(n1), .IE(n1), .PAD(
        Quotient[13]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot14 ( .I(Quotient_w[14]), .OEN(n1), .IE(n1), .PAD(
        Quotient[14]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Quot15 ( .I(Quotient_w[15]), .OEN(n1), .IE(n1), .PAD(
        Quotient[15]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem0 ( .I(Remainder_w[0]), .OEN(n1), .IE(n1), .PAD(
        Remainder[0]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem1 ( .I(Remainder_w[1]), .OEN(n1), .IE(n1), .PAD(
        Remainder[1]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem2 ( .I(Remainder_w[2]), .OEN(n1), .IE(n1), .PAD(
        Remainder[2]), .DS(n1), .PE(n1) );
```

```
    PDDW0204SCDG U_Rem3 ( .I(Remainder_w[3]), .OEN(n1), .IE(n1), .PAD(
         Remainder[3]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem4 ( .I(Remainder_w[4]), .OEN(n1), .IE(n1), .PAD(
         Remainder[4]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem5 ( .I(Remainder_w[5]), .OEN(n1), .IE(n1), .PAD(
         Remainder[5]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem6 ( .I(Remainder_w[6]), .OEN(n1), .IE(n1), .PAD(
         Remainder[6]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem7 ( .I(Remainder_w[7]), .OEN(n1), .IE(n1), .PAD(
         Remainder[7]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem8 ( .I(Remainder_w[8]), .OEN(n1), .IE(n1), .PAD(
         Remainder[8]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem9 ( .I(Remainder_w[9]), .OEN(n1), .IE(n1), .PAD(
         Remainder[9]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem10 ( .I(Remainder_w[10]), .OEN(n1), .IE(n1), .PAD(
         Remainder[10]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem11 ( .I(Remainder_w[11]), .OEN(n1), .IE(n1), .PAD(
         Remainder[11]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem12 ( .I(Remainder_w[12]), .OEN(n1), .IE(n1), .PAD(
         Remainder[12]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem13 ( .I(Remainder_w[13]), .OEN(n1), .IE(n1), .PAD(
         Remainder[13]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem14 ( .I(Remainder_w[14]), .OEN(n1), .IE(n1), .PAD(
         Remainder[14]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Rem15 ( .I(Remainder_w[15]), .OEN(n1), .IE(n1), .PAD(
         Remainder[15]), .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg0 ( .I(Sreg_w[0]), .OEN(n1), .IE(n1), .PAD(Sreg[0]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg1 ( .I(Sreg_w[1]), .OEN(n1), .IE(n1), .PAD(Sreg[1]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg2 ( .I(Sreg_w[2]), .OEN(n1), .IE(n1), .PAD(Sreg[2]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg3 ( .I(Sreg_w[3]), .OEN(n1), .IE(n1), .PAD(Sreg[3]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg4 ( .I(Sreg_w[4]), .OEN(n1), .IE(n1), .PAD(Sreg[4]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg5 ( .I(Sreg_w[5]), .OEN(n1), .IE(n1), .PAD(Sreg[5]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg6 ( .I(Sreg_w[6]), .OEN(n1), .IE(n1), .PAD(Sreg[6]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg7 ( .I(Sreg_w[7]), .OEN(n1), .IE(n1), .PAD(Sreg[7]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg8 ( .I(Sreg_w[8]), .OEN(n1), .IE(n1), .PAD(Sreg[8]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg9 ( .I(Sreg_w[9]), .OEN(n1), .IE(n1), .PAD(Sreg[9]), .DS(
         n1), .PE(n1) );
    PDDW0204SCDG U_Sreg10 ( .I(Sreg_w[10]), .OEN(n1), .IE(n1), .PAD(Sreg[10]),
         .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg11 ( .I(Sreg_w[11]), .OEN(n1), .IE(n1), .PAD(Sreg[11]),
         .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg12 ( .I(Sreg_w[12]), .OEN(n1), .IE(n1), .PAD(Sreg[12]),
         .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg13 ( .I(Sreg_w[13]), .OEN(n1), .IE(n1), .PAD(Sreg[13]),
         .DS(n1), .PE(n1) );
```

```
    PDDW0204SCDG U_Sreg14 ( .I(Sreg_w[14]), .OEN(n1), .IE(n1), .PAD(Sreg[14]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg15 ( .I(Sreg_w[15]), .OEN(n1), .IE(n1), .PAD(Sreg[15]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg16 ( .I(Sreg_w[16]), .OEN(n1), .IE(n1), .PAD(Sreg[16]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg17 ( .I(Sreg_w[17]), .OEN(n1), .IE(n1), .PAD(Sreg[17]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg18 ( .I(Sreg_w[18]), .OEN(n1), .IE(n1), .PAD(Sreg[18]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg19 ( .I(Sreg_w[19]), .OEN(n1), .IE(n1), .PAD(Sreg[19]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg20 ( .I(Sreg_w[20]), .OEN(n1), .IE(n1), .PAD(Sreg[20]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg21 ( .I(Sreg_w[21]), .OEN(n1), .IE(n1), .PAD(Sreg[21]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg22 ( .I(Sreg_w[22]), .OEN(n1), .IE(n1), .PAD(Sreg[22]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg23 ( .I(Sreg_w[23]), .OEN(n1), .IE(n1), .PAD(Sreg[23]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg24 ( .I(Sreg_w[24]), .OEN(n1), .IE(n1), .PAD(Sreg[24]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg25 ( .I(Sreg_w[25]), .OEN(n1), .IE(n1), .PAD(Sreg[25]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg26 ( .I(Sreg_w[26]), .OEN(n1), .IE(n1), .PAD(Sreg[26]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg27 ( .I(Sreg_w[27]), .OEN(n1), .IE(n1), .PAD(Sreg[27]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg28 ( .I(Sreg_w[28]), .OEN(n1), .IE(n1), .PAD(Sreg[28]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg29 ( .I(Sreg_w[29]), .OEN(n1), .IE(n1), .PAD(Sreg[29]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg30 ( .I(Sreg_w[30]), .OEN(n1), .IE(n1), .PAD(Sreg[30]),
          .DS(n1), .PE(n1) );
    PDDW0204SCDG U_Sreg31 ( .I(Sreg_w[31]), .OEN(n1), .IE(n1), .PAD(Sreg[31]),
          .DS(n1), .PE(n1) );
    TIELBWP7T U6 ( .ZN(n1) );
    TIEHBWP7T U7 ( .Z(n2) );
  endmodule
```

Cuadro 123: Archivo DIV32b_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O

13.8.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O

El último paso que se realizó fue la simulación del circuito completamente sintetizado junto con los pines de I/O. Esto se llevó a cabo nuevamente para comprobar que el circuito completamente sintetizado se comportara correctamente.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 124 en una terminal

abierta en `Outputs/DIV32b/Outputs_finales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v DIV32b_sintesis_final.v DIV32b_tb.
    v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 124: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en la Figura 171, el comportamiento es idéntico al que se obtuvo anteriormente, lo que confirma que el *netlist* final obtenido del circuito sintetizado con pines de I/O es correcto.

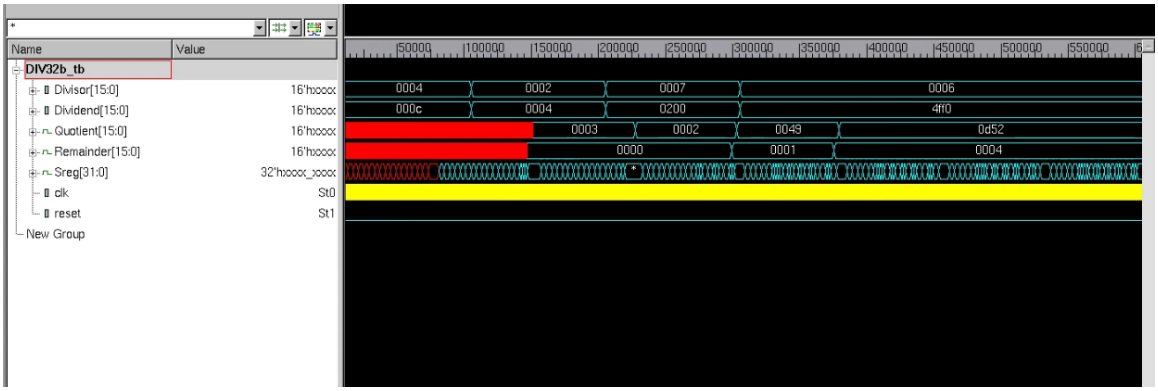


Figura 171: Simulación del *netlist* final

13.9. Proceso para la síntesis lógica de el nanochip El Gran Jaguar

13.9.1. Diseño del módulo del circuito, módulo de pines de entrada y salida y *testbench*

En esta sección se sintetizó el circuito nanochip El Gran Jaguar. El primer paso fue crear una carpeta en el directorio de *inputs*, nombrada según el circuito principal a sintetizar; en este caso, `nanochip` (tanto la carpeta como el nombre del archivo Verilog que contiene el circuito principal deben tener el mismo nombre). Luego, se diseñaron los archivos necesarios para la simulación y síntesis del circuito.

Siguiendo la estructura explicada en el capítulo y sección 9.2, el archivo que contiene el módulo principal del nanochip se nombró `nanochip.v`. El archivo de *testbench*, utilizado para simular y verificar el funcionamiento del circuito, se denominó `nanochip_tb.v`. Por último, el archivo que define el módulo de entradas y salidas se nombró `IO_nanochip.v`.

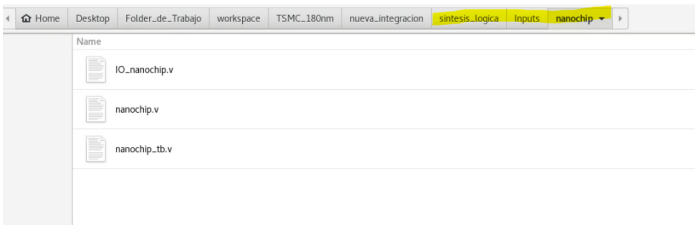


Figura 172: Archivos necesarios en la carpeta del circuito en *inputs*

Para crear el archivo `nanochip.v`, se utilizó el *script* de Python llamado escriturafinal_vp3.py, desarrollado por estudiantes de años anteriores, el cual automatiza la creación del circuito. Este archivo puede ubicarse en el directorio `nueva_integracion/verilog_generation`. Solo fue necesario modificar las variables `texto` y `texto2` con el nuevo mensaje para obtener el nuevo circuito y ejecutar el *script*.

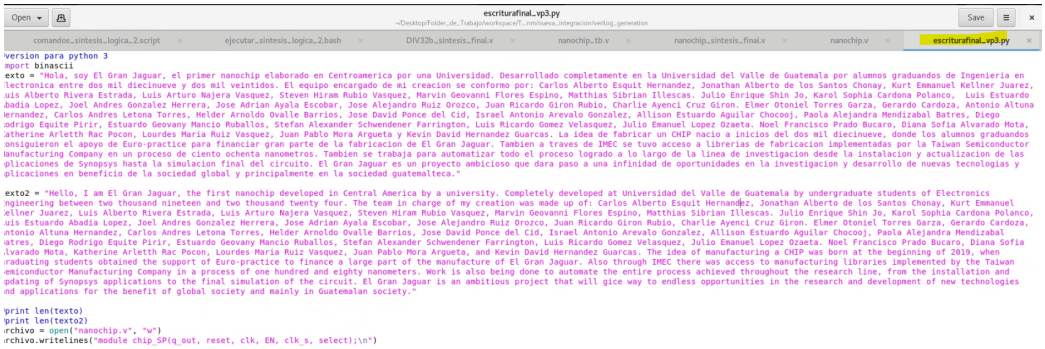


Figura 173: Creación del circuito nanochip El Gran Jaguar

En la parte de anexos se muestra el diseño del circuito, mientras que en el Cuadro 125 se presenta el diseño del módulo de pines de entrada y salida. Además, en el Cuadro 126 se muestra el *testbench* diseñado para la simulación del circuito original y del circuito sintetizado.

Dado que el diseño del circuito incluye múltiples entradas y salidas, en el módulo de pines es necesario hacer referencia a cada una de ellas de forma individual. Sin embargo, el proceso de creación de cada pin es el mismo para todas las entradas y salidas (ver Cuadro 125).

```
module circuit(reset,clk,select,EN, q_out,clk_s);

output [7:0] q_out;
output clk_s;
input reset;
input clk;
input [1:0]select;
input EN;
wire reset_w;
wire clk_w;
wire [1:0]select_w;
wire EN_w;
wire [7:0] q_out_w;
```

```
    wire clk_s_w;

    chip_SP circuit_new (. q_out( q_out_w), .clk_s(clk_s_w), .reset(reset_w), .clk(
        clk_w), .select(select_w), .EN(EN_w));

    //Entradas (C):
    PDDW0204SCDG IN0 (1'b0,1'b0,1'b1,reset,reset_w,1'b0,1'b1);
    PDDW0204SCDG IN1 (1'b0,1'b0,1'b1,clk,clk_w,1'b0,1'b1);
    PDDW0204SCDG IN2 (1'b0,1'b0,1'b1,EN,EN_w,1'b0,1'b1);
    PDDW0204SCDG IN3 (1'b0,1'b0,1'b1,select[0],select_w[0],1'b0,1'b1);
    PDDW0204SCDG IN4 (1'b0,1'b0,1'b1,select[1],select_w[1],1'b0,1'b1);
    //Salidas (I):
    PDDW0204SCDG OUT0 (clk_s_w,1'b0,1'b0,clk_s,1'bx,1'b0,1'b0);
    PDDW0204SCDG OUT1 ( q_out_w[0],1'b0,1'b0, q_out[0],1'bx,1'b0,1'b0);
    PDDW0204SCDG OUT2 ( q_out_w[1],1'b0,1'b0, q_out[1],1'bx,1'b0,1'b0);
    PDDW0204SCDG OUT3 ( q_out_w[2],1'b0,1'b0, q_out[2],1'bx,1'b0,1'b0);
    PDDW0204SCDG OUT4 ( q_out_w[3],1'b0,1'b0, q_out[3],1'bx,1'b0,1'b0);
    PDDW0204SCDG OUT5 ( q_out_w[4],1'b0,1'b0, q_out[4],1'bx,1'b0,1'b0);
    PDDW0204SCDG OUT6 ( q_out_w[5],1'b0,1'b0, q_out[5],1'bx,1'b0,1'b0);
    PDDW0204SCDG OUT7 ( q_out_w[6],1'b0,1'b0, q_out[6],1'bx,1'b0,1'b0);
    PDDW0204SCDG OUT8 ( q_out_w[7],1'b0,1'b0, q_out[7],1'bx,1'b0,1'b0);
    //Pines de VDD y VSS
    PVDD1CDG PVDD();
    PVSS1CDG PVSS();
    endmodule
```

Cuadro 125: Archivo IO_nanochip.v: diseño de módulo de entradas y salidas para el nanochip en Verilog

```
    //Test_Bench
    module nanochip_tb;
    reg clk;
    reg reset;
    reg EN;
    reg [1:0] select;
    wire [7:0] q_out;
    wire clk_s;

    chip_SP chip1(q_out, reset, clk, EN, clk_s,select);
    initial
    clk = 1'b0;
    always
    #1 clk=~clk;
    initial begin
    $fsdbDumpfile("nanochip.fsdb");
    $fsdbDumpvars(0, nanochip_tb);
    $dumpfile("nanochip.vcd");
    $dumpvars(0, nanochip_tb);
    EN = 1'b0;
    reset = 1'b1;
    select = 2'b10;
    #15 reset =1'b0;
    #15 reset =1'b1;
    #15 reset =1'b0;
```

```
    #8000 $finish;
    end
    endmodule
```

Cuadro 126: Archivo nanochip_tb.v: diseño del *testbench* para simulación del nanochip en Verilog

13.9.2. Simulación del diseño original del circuito

Como siguiente paso, se simuló el circuito original con el objetivo de verificar que no existieran errores de compilación y que el diseño del circuito fuera correcto. Como se explicó anteriormente, si no se realiza este paso, existe el riesgo de encontrar errores debido a posibles fallos en el diseño del circuito, lo que impediría proceder a la ejecución de la síntesis lógica.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 127 en una terminal abierta en `Inputs/nanochip/`.

```
vcs -v nanochip.v nanochip_tb.v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 127: Comandos para la compilación y simulación del diseño del circuito con visualización en DVE

Siguiendo los pasos descritos en el capítulo y sección 12.1 para la visualización del archivo `.vcd` generado al ejecutar el archivo de simulación, se obtuvo la Figura 174.

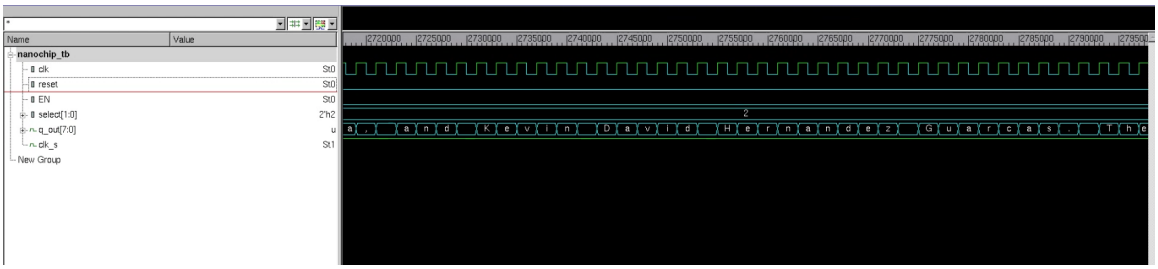
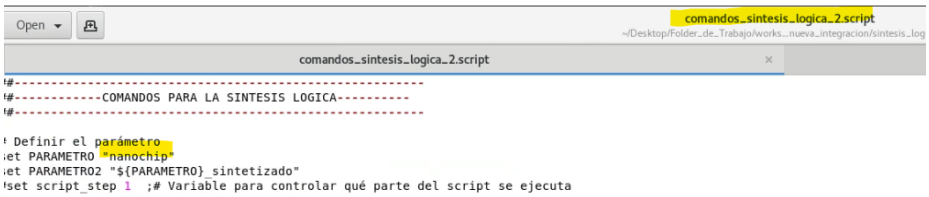


Figura 174: Simulación del nanochip en su diseño original

Como se observa en la Figura 174, el comportamiento del circuito es el esperado, ya que se está mostrando el mensaje programado en el diseño.

13.9.3. Síntesis lógica del diseño original del circuito

Antes de proceder a ejecutar el *script* `ejecutar_sintesis_logica_2.bash`, fue necesario abrir tanto este *script* como el archivo `comandos_sintesis_logica_2.script` para definir en la variable `PARAMETRO` el nombre de la carpeta que se creó en *inputs*, la cual contiene el circuito a sintetizar. En este caso, se definió como `nanochip`.



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 175: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

Luego, se ejecutó el *script* utilizando el comando que se muestra en el Cuadro 128, en una terminal abierta en el directorio `sintesis_logica`.

```
./ejecutar_sintesis_logica_2.bash
```

Cuadro 128: Comando para ejecutar la síntesis lógica del circuito con el *script* ejecutar_sintesis_logica_2.bash

Al ejecutar este comando, se creó una carpeta en *outputs* con el nombre `nanochip`. Además, se generaron dos subdirectorios dentro de esta, llamados `Outputs_iniciales` y `Outputs_finales`. En la terminal no apareció ningún error y se generaron los reportes individuales tanto del circuito sintetizado como del circuito sintetizado con los pines de I/O. Asimismo, en la carpeta `Outputs/nanochip/Outputs_iniciales` se creó un archivo llamado `nanochip_sintetizado.v`, el cual corresponde al archivo resultante de la primera etapa de la síntesis lógica (ver Figuras 176 y 177).

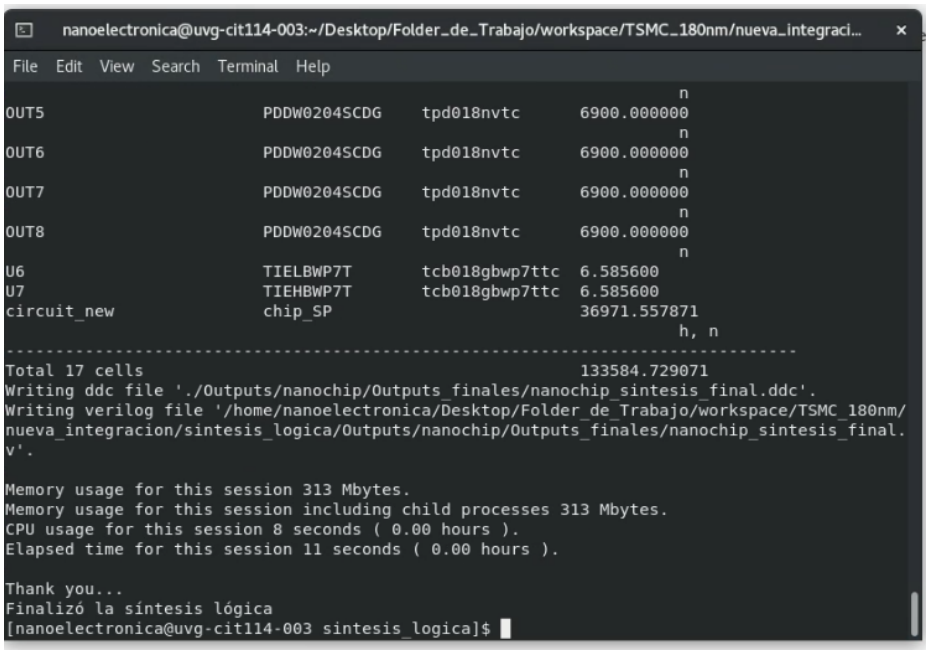


Figura 176: Ejecución del *script* ejecutar_sintesis_logica_2.bash sin errores

Figura 177: Archivos generados en la carpeta del circuito en Outputs_iniciales

El archivo `nanochip_sintetizado.v`, que corresponde al circuito sintetizado, se muestra en la parte de anexos.

13.9.4. Simulación del circuito sintetizado

El siguiente paso que se llevó a cabo, y que es fundamental para corroborar que el circuito sintetizado se comportara correctamente, fue la simulación del circuito sintetizado.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 129 en una terminal abierta en `Outputs/nanochip/Outputs_iniciales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v nanochip_sintetizado.v nanochip_tb
    .v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 129: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en el Cuadro 129, para simular el circuito sintetizado es importante hacer referencia a los archivos `tcb018gbwp7t.v` y `tpd018nv.v`, ya que estos contienen las celdas a las cuales los módulos del archivo `nanochip_sintetizado.v` hacen referencia. Dado que dichos archivos se encuentran ubicados en el directorio de *outputs* y se está ejecutando el comando en una terminal abierta en el directorio `Outputs/nanochip/Outputs_iniciales`, es necesario subir dos niveles en la jerarquía de directorios para poder encontrarlos; por esta razón, se utiliza la ruta `../../`.

Luego de haber ejecutado los comandos mencionados anteriormente, se obtuvo la Figura 178. Se observó el mismo comportamiento que con el circuito original, lo que confirma que la síntesis del circuito se realizó correctamente.

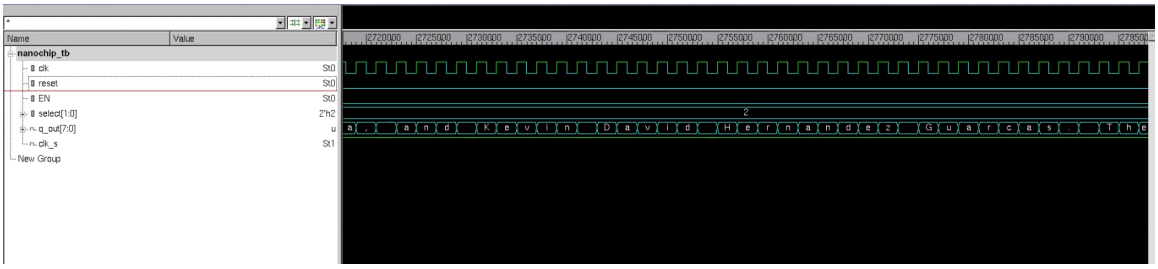


Figura 178: nanochip en su diseño sintetizado

13.9.5. Unión del circuito sintetizado con el módulo de pines de I/O

El archivo denominado `nanochip_sintetizado_and_IO.v` se creó uniendo el archivo `nanochip_sintetizado.v` con el archivo `IO_nanochip.v`. Este proceso también lo realiza automáticamente el *script* antes de comenzar el segundo proceso de síntesis lógica.

Por ello, es fundamental tener diseñado desde el principio el archivo `IO_nanochip.v` y guardarlo en la carpeta `Inputs/nanochip`. Si este archivo no está presente, el *script* no podrá localizarlo y el nuevo archivo generado contendrá únicamente el contenido de `nanochip_sintetizado.v`. Esto ocasionará que, en el siguiente paso, se realice nuevamente la síntesis sin incluir el módulo de pines de entrada y salida, obteniéndose un resultado similar al del archivo `nanochip_sintetizado.v`.

Figura 179: Archivos generados en la carpeta del circuito en Outputs_iniciales

En este caso, el archivo `nanochip_sintetizado_and_IO.v` no se mostrará; sin embargo, dicho archivo consiste únicamente en la combinación entre `nanochip_sintetizado.v` y `IO_nanochip.v`.

13.9.6. Síntesis lógica del circuito sintetizado con pines de I/O

A través del archivo `nanochip_sintetizado_and_IO.v` se obtuvo el circuito completamente sintetizado. Este archivo se generó en el directorio `Outputs/nanochip/Outputs_finales` con el nombre `nanochip_sintesis_final.v`. Además, se generaron otros dos archivos que también servirán para el siguiente proceso de síntesis física.

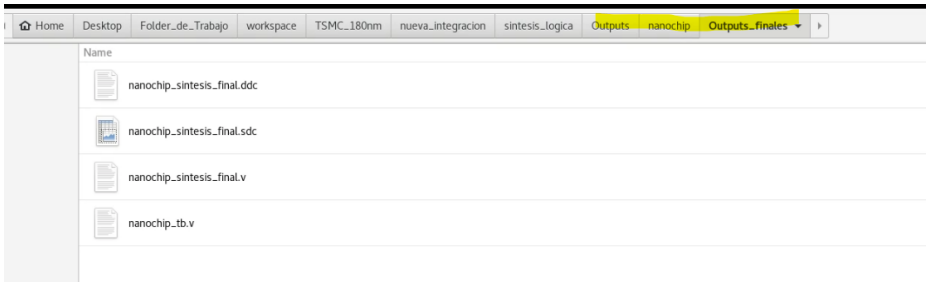


Figura 180: Archivos generados en la carpeta del circuito en Outputs_finales

En la parte de anexos se muestra el *netlist* final.

13.9.7. Simulación del *netlist* final del circuito sintetizado con pines de I/O

El último paso que se realizó fue la simulación del circuito completamente sintetizado junto con los pines de I/O. Esto se llevó a cabo nuevamente para comprobar que el circuito completamente sintetizado se comportara correctamente.

Para ello, se ejecutaron los comandos que se muestran en el Cuadro 130 en una terminal abierta en `Outputs/nanochip/Outputs_finales`.

```
vcs -v ../../tcb018gbwp7t.v ../../tpd018nv.v nanochip_sintesis_final.v
    nanochip_tb.v -o simulacion -full64 -debug_access+all

./simulacion

dve -full64
```

Cuadro 130: Comandos para la compilación y simulación del diseño del circuito sintetizado con visualización en DVE

Como se puede observar en la Figura 181, el comportamiento es idéntico al que se obtuvo anteriormente, lo que confirma que el *netlist* final obtenido del circuito sintetizado con pines de I/O es correcto.

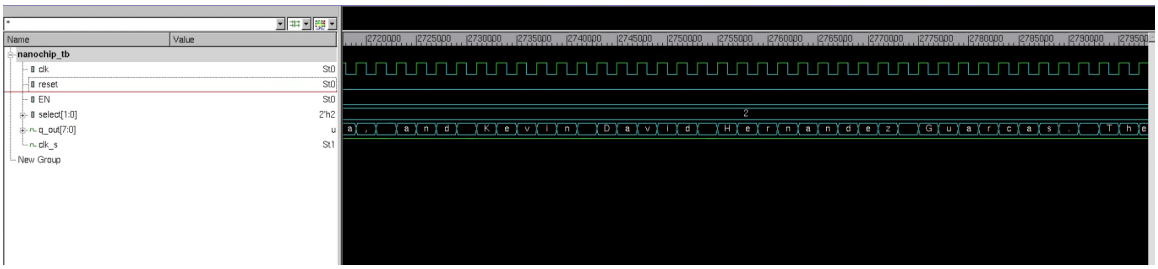


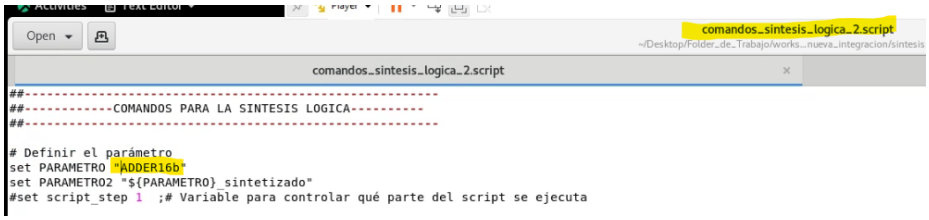
Figura 181: Simulación del *netlist* final

CAPÍTULO 14

Mejoras realizadas en el *script* `ejecutar_sintesis_logica_2.bash`

Como se mencionó anteriormente, se desarrollaron dos *scripts* para automatizar todo el proceso de síntesis lógica de un circuito tomando en cuenta los pines de I/O: uno denominado `ejecutar_sintesis_logica_2.bash` y otro denominado `comandos_sintesis_logica_2.script`. Sin embargo, en ambos *scripts* era necesario modificar manualmente la variable `PARAMETRO`, en la cual se debía ingresar el nombre del circuito que se deseaba sintetizar.

Además, si el circuito a sintetizar hacía referencia a otros módulos ubicados en archivos Verilog distintos, también era necesario agregar dichos archivos Verilog de forma manual (ver Figuras 182 y 183).



(a) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* comandos_sintesis_logica_2.script

(b) Definición de PARAMETRO con el nombre del circuito a sintetizar en el *script* ejecutar_sintesis_logica_2.bash

Figura 182: Definición de PARAMETRO con el nombre de la carpeta y el archivo del circuito a sintetizar

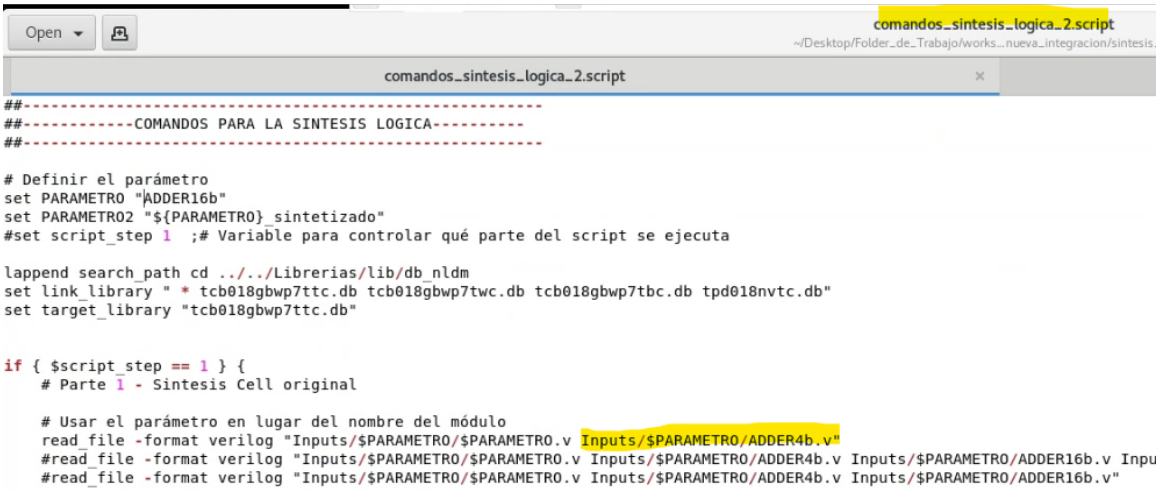


Figura 183: Modificación del *script* comandos_sintesis_logica_2.script

Para evitar realizar estos pasos manualmente, se desarrolló un nuevo *script*, denominado `ejecutar_sintesis_logica_3.bash`, que incluye las siguientes mejoras:

1. Al ejecutarse, el *script* permite ingresar el nombre de la carpeta que contiene el circuito a sintetizar y modifica automáticamente la variable `PARAMETRO` en los *scripts* `ejecutar_sintesis_logica_3.bash` y `comandos_sintesis_logica_2.script`, eliminando así la necesidad de cambiar manualmente dicha variable en ambos *scripts*.

2. El *script* solicita al usuario indicar si el módulo del circuito a sintetizar depende de otros módulos ubicados en archivos Verilog distintos. Si no es el caso, el usuario debe

ingresar `n/N`; si sí depende de otros módulos, debe ingresar `s/S` y proporcionar los nombres de los archivos Verilog que contienen los módulos a los que hace referencia el circuito principal.

3. El *script* realiza automáticamente todo el proceso descrito anteriormente para el archivo `ejecutar_sintesis_logica_2.bash`.

4. Si se ingresa un nombre incorrecto para un circuito que no se encuentra en `sintesis_logica/Inputs/`, el sistema mostrará los mensajes de error correspondientes.

Este nuevo *script* facilita la síntesis de circuitos sin la necesidad de modificar directamente los *scripts* mencionados anteriormente.

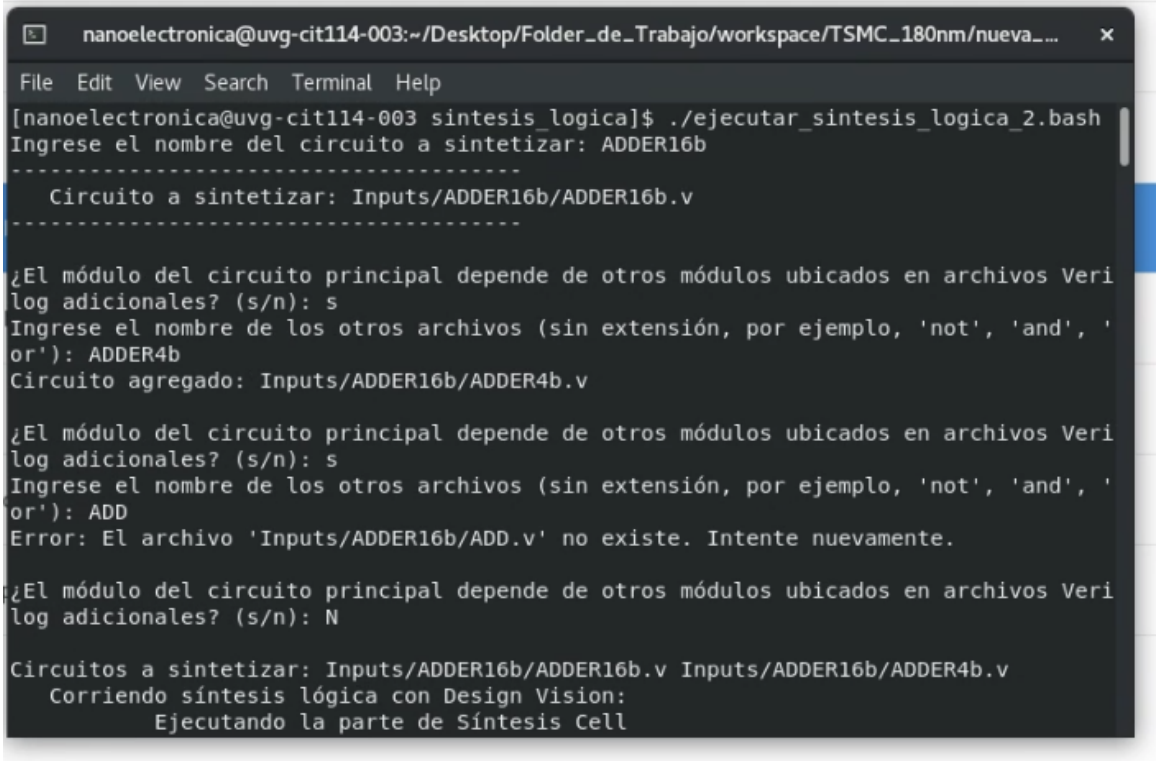


Figura 184: Ejecución del *script* comandos_sintesis_logica_3.script para sintetizar un sumador de 16 bits

```
#!/bin/bash
#PARAMETRO="xor"

#--------------------------------------------
# CODIGO REALIZADO PARA INGRESAR EL NOMBRE
# DEL CIRCUITO A SINTETIZAR
#--------------------------------------------


# Solicitar el valor de PARAMETRO
read -p "Ingrese el nombre del circuito a sintetizar: " PARAMETRO
```

```
# Inicializa la variable para almacenar los circuitos a sintetizar
circuitos="Inputs/$PARAMETRO/$PARAMETRO.v"

# Validar que el directorio y archivo principal existen
INPUT_DIR=./Inputs/$PARAMETRO
MAIN_FILE="$INPUT_DIR/$PARAMETRO.v"

if [[ ! -d "$INPUT_DIR" || ! -f "$MAIN_FILE" ]]; then
    echo "Error: El directorio o el archivo principal del circuito '$PARAMETRO'
         no existen en sintesis_logic/\textit{inputs}."
    exit 1
fi

echo "----------------------------------------"
echo " Circuito a sintetizar: $circuitos "
echo "----------------------------------------"
echo ""

# Ruta del \textit{script} que contiene los comandos
SCRIPT_COMANDOS="comandos_sintesis_logica_2.script"

# Modificar el valor de PARAMETRO en el \textit{script} de comandos
sed -i "s/set PARAMETRO .*/set PARAMETRO \"$PARAMETRO\"/" "$SCRIPT_COMANDOS"

# Mostrar el valor actualizado
#echo "El valor de PARAMETRO se ha actualizado a: $PARAMETRO"


# Solicitar archivos Verilog adicionales si es necesario
#echo "Colocar s/S si el módulo del circuito principal depende de otros módulos
     ubicados en archivos Verilog adicionales"
#echo "Colocar n/N si el módulo del circuito principal no depende de otros mó
    dulos ubicados en archivos Verilog adicionales"

# Preguntar si se desea sintetizar otros circuitos
while true; do
    read -p "¿El módulo del circuito principal depende de otros módulos
        ubicados en archivos Verilog adicionales? (s/n): " respuesta
    if [[ "$respuesta" == "s" || "$respuesta" == "S" ]]; then
        read -p "Ingrese el nombre de los otros archivos (sin extensión, por
            ejemplo, 'not', 'and', 'or'): " circuito
        circuito_path="Inputs/$PARAMETRO/$circuito.v"

        # Verifica si el archivo existe
        if [[ -f "$circuito_path" ]]; then
            # Verifica si el circuito ya está en la lista
            if [[ "$circuitos" == *"$circuito_path"* ]]; then
                echo "Error: El circuito '$circuito_path' ya está seleccionado."
        echo ""
            else
                circuitos="$circuitos $circuito_path"
                echo "Circuito agregado: $circuito_path"
        echo ""
```

```
            fi
        else
            echo "Error: El archivo '$circuito_path' no existe. Intente
                nuevamente."
        echo ""
        fi
    elif [[ "$respuesta" == "n" || "$respuesta" == "N" ]]; then
        break
    else
        echo "Por favor, ingrese 's' para sí o 'n' para no."
    echo ""
    fi
done
echo ""

# Modificar comando para la lectura de los archivos
sed -i "s|read_file -format verilog \"Inputs/[^\"]*|read_file -format verilog \
    "${circuitos}|" comandos_sintesis_logica_2.script

# Muestra la lista final de circuitos a sintetizar
echo "Circuitos a sintetizar: $circuitos"


#-------------------------------------------
# CODIGO REALIZADO PARA EMPEZAAR LA
# SINTESIS DE LOS CIRCUITOS INGRESADOS
#-------------------------------------------

# paso 1: Creacion de carpetas de OUtputs
mkdir -p ./Outputs/$PARAMETRO/Outputs_iniciales ./Outputs/$PARAMETRO/
    Outputs_finales

cp ./Inputs/$PARAMETRO/${PARAMETRO}_tb.v ./Outputs/$PARAMETRO/Outputs_finales

echo " Corriendo síntesis lógica con Design Vision:"
echo " Ejecutando la parte de Síntesis Cell"

# Paso 2: Ejecutar la parte 1 del \textit{script}
echo "------------------------------------------------------------"
echo "-------- EJECUTANDO SINTESIS LOGICA SOLO CON MODULO -------"
echo "--------------------- DEL CIRCUITO ORIGINAL ---------------"
#dc_shell -f comandos_sintesis_logica_2.script -x "set script_step 1"
dc_shell -f comandos_sintesis_logica_2.script -x "set PARAMETRO \"$PARAMETRO\";
     set script_step 1"

# Paso 3: Mover los archivos de log a la carpeta correspondiente
mv default.svf logs/
mv command.log logs/

# Paso 4: Combinar el archivo generado $PARAMETRO2.v con el archivo
    PINES_IO_$PARAMETRO.v
PARAMETRO2="${PARAMETRO}_sintetizado"
COMBINED_FILE="Outputs/$PARAMETRO/Outputs_iniciales/${PARAMETRO2}_and_IO.v"
cat "Inputs/$PARAMETRO/IO_$PARAMETRO.v" "Outputs/$PARAMETRO/Outputs_iniciales/
```

```
    $PARAMETRO2.v" > "$COMBINED_FILE"

# Paso 5: Ejecutar la parte 2 del \textit{script}
echo "------------------------------------------------------------"
echo "--------- EJECUTANDO SINTESIS LOGICA CON EL MODULO -------"
echo "---------------------- DE PINES I/O ----------------------"
dc_shell -f comandos_sintesis_logica_2.script -x "set script_step 2"

# Paso 6: Mover los archivos de log a la carpeta correspondiente
mv default.svf logs/
mv command.log logs/

# Finalizar
echo "Finalizó la síntesis lógica"

exit
```

Cuadro 131: *Script* ejecutar_sintesis_logica_3.bash

CAPÍTULO 15

Script realizado para mejora del proceso de las aplicaciones de Synopsys

En este capítulo se presenta el *script* en Bash diseñado para optimizar el proceso de descarga de las aplicaciones de Synopsys. Este *script* crea las dependencias iniciales necesarias para ejecutar correctamente el instalador, lo que incluye la creación de directorios, la asignación de permisos, la instalación de librerías y aplicaciones, así como la generación de claves de seguridad RSA para la conexión al servidor de Synopsys.

Además, el *script* realiza la descarga del instalador de las aplicaciones de Synopsys y ejecuta su instalación, pasos que son esenciales antes de proceder con la descarga completa de las aplicaciones.

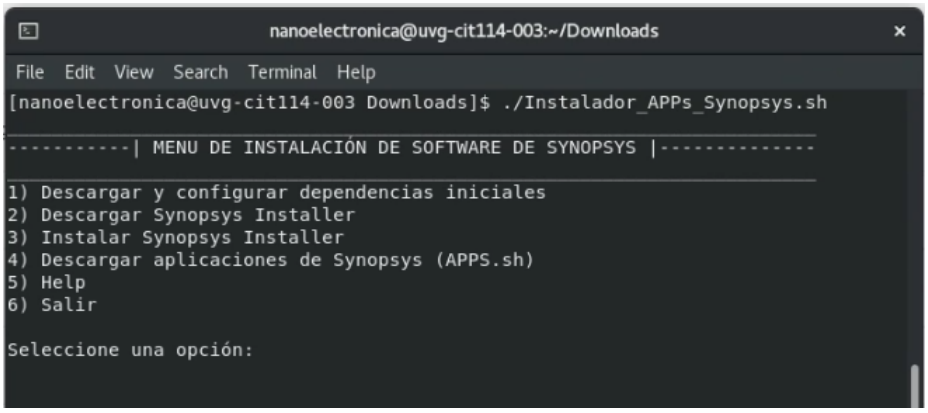


Figura 185: Interfaz del *script* realizado

A continuación, se muestra el *script* completo realizado:

```
#!/bin/bash
# Autor > Kevin David Hernández Guarcas
```

```
# Objetivo > Mejora del proceso de Descarga e Instalacion de Aplicaciones y
    Librerias de Synopsys.
# Fecha > septiembre de 2024

#_____________________________________________________________________#

# ------------------------| FUNCIONES | ---------------------------
#_____________________________________________________________________

# ++++++++++++++++ Función Dependencias Iniciales ++++++++++++++++++
#OBJETIVO:

Dependencias_Iniciales() {
# Detectar el usuario actual
current_user=$USER

# Detectar el grupo principal del usuario actual
current_group=$(id -gn $USER)

#Archivo de estado de descarga de dependencias
ESTADO_FILE="/home/$current_user/Downloads/Installer_Flags/dependencias_state"

echo "Ejecutando con el usuario: $current_user y grupo: $current_group"

# 1) ___________|Agregar Servidor de Licencias al BASHRC|___________

echo -e "Agregrando servidor de licencias al BASHRC..."
# Ruta al archivo .bashrc
BASHRC_FILE="$HOME/.bashrc"

# Servidor de licencias UVG
IP="192.168.74.1" # IP Del servidor
LINE_SERVER_LICENSE="LM_LICENSE_FILE=\"27020@$IP\"; export LM_LICENSE_FILE"

# Verificar si la línea de servidor de licencias ya existe en el archivo .bashrc
if grep -q "LM_LICENSE_FILE" "$BASHRC_FILE"; then
    # Si existe, reemplazar la línea con la nueva IP
    sed -i "s|LM_LICENSE_FILE=\"[^\"]*\"; export LM_LICENSE_FILE|
        $LINE_SERVER_LICENSE|" "$BASHRC_FILE"
    echo -e "La línea del servidor UVG ha sido actualizada en el archivo .bashrc\n"
else
    # Si no existe, agregarla antes de la línea "export PATH"
    sed -i "/export PATH/i $LINE_SERVER_LICENSE" "$BASHRC_FILE"
    echo -e "Línea del servidor UVG agregada al archivo .bashrc\n"
fi

# Verificar si la línea del PATH INSTALLER ya existe en el archivo .bashrc
LINE_INSTALLER='PATH=/usr/synopsys/installer:$PATH'
if ! grep -q "$LINE_INSTALLER" "$BASHRC_FILE"; then
    # Si no existe, agregarla antes de la línea "export PATH"
    sed -i "/^export PATH/i $LINE_INSTALLER" "$BASHRC_FILE"
    echo -e "Línea del PATH INSTALLER agregada al archivo .bashrc \n"
else
    echo -e "La línea del PATH INSTALLER ya existe en el archivo .bashrc \n"
```

```
fi

# 2) __________| Creación de Carpetas |__________________________

# Crear la carpeta 'Folder_de_Trabajo' en HOME/Desktop
#echo "Creando carpeta de Folder_de_Trabajo en la ruta /Desktop/Folder_de_Trabajo"
mkdir -p ~/Desktop/Folder_de_Trabajo

# Configurar permisos para ~/Desktop/Folder_de_Trabajo
chmod 775 ~/Desktop/Folder_de_Trabajo

mkdir -p ~/Downloads/Installer_Flags

# Configurar permisos para ~/Desktop/Folder_de_Trabajo
chmod 775 ~/Downloads/Installer_Flags

# 3) __________|Conexión al servidor de Synopsys|_____________

# Ruta del archivo known_hosts
#KNOWN_HOSTS_FILE="/home/nanoelectronica/.ssh/known_hosts"
KNOWN_HOSTS_FILE="/home/$current_user/.ssh/known_hosts"

# Verificar si la carpeta .ssh existe, si no, crearla
#if [ ! -d "/home/nanoelectronica/.ssh" ]; then
if [ ! -d "/home/$current_user/.ssh" ]; then
    mkdir -p /home/$current_user/.ssh
    chmod 700 /home/$current_user/.ssh
fi

# Verificar si el archivo known_hosts no existe
if [ ! -f "$KNOWN_HOSTS_FILE" ]; then
    # Conexión al servidor por SSH
    #con StrictHostKeyChecking=no desactivo la verificacion de la clave del
        servidor
    if ssh -o StrictHostKeyChecking=no eft.synopsys.com; then
    echo "Conexión exitosa con eft.synopsys.com."
    else
    echo "Creación de llaves de seguridad RSA exitosa."
    echo "Verificar llaves de seguridad RSA en /home/$current_user/.ssh/known_hosts
        "
    fi
else
    echo "El archivo known_hosts ya existe. Saltando la conexión SSH."
    echo "Verificar llaves de seguridad RSA en /home/$current_user/.ssh/known_hosts
        "
fi




# 4) ________| Creación de Carpetas Con Permisos de Root |____________

# ___________|Instalación de Librerias y Aplicaciones|__________
```

```
# 1) Solicitar nombre de usuario y verificar permisos
    echo ""
    echo "Ingrese su nombre de usuario con permisos de root para crear carpetas en
        usr/"
    echo "Y descargar las librerias y aplicaciones necesarias: "

    read input_user

    # Verificar si el usuario tiene permisos de root (pertenece al grupo wheel o
        sudo)
    if id -nG "$input_user" | grep -qw "wheel\|sudo"; then
    echo ""
        echo "$input_user tiene permisos de root, se ejecutarán los comandos con
            sudo."
        can_run_sudo=true
    else
    echo ""
        echo "$input_user NO tiene permisos de root, no se ejecutarán los comandos
            con sudo."
        can_run_sudo=false
    fi

    if [ "$can_run_sudo" = true ]; then
    # Crear la carpeta 'synopsys' en /usr
    echo -e "Creando directorio /usr/synopsys...\n"
    sudo mkdir -p /usr/synopsys

    # Crear la carpeta 'iPDK' dentro de 'synopsys'
    echo "Creando carpeta de iPDK en la ruta /usr/synopsys/iPDK"
    sudo mkdir -p /usr/synopsys/iPDK

    sudo mkdir -p /usr/synopsys/installer

    # Cambiar la propiedad de /usr/synopsys al usuario y grupo actual
    sudo chown -R $current_user:$current_group /usr/synopsys/

    # Configurar permisos para /usr/synopsys
    sudo chmod -R +x /usr/synopsys/

        echo -e "Instalando aplicaciones y librerías..."
    sudo dnf update -y
    sudo dnf install -y lftp
    sudo dnf install -y libXScrnSaver #installer
    sudo dnf install -y libnsl #waveview
    sudo dnf install -y tcsh #cdesigner
    #sudo dnf install -y rsh

    touch "$ESTADO_FILE"
    else
        #echo "El usuario no existe o no tiene permisos de root ..."
    echo "saliendo..."
    exit 1
    fi
```

```
}


# ++++++++++++ Función Instalación automatica de Installer ++++++++++++
#OBJETIVO:

Synopsys_Installer() {
# Detectar el usuario actual
current_user=$USER
# Detectar el grupo principal del usuario actual
current_group=$(id -gn $USER)

# Directorio donde se encuentra el archivo de SynopsysInstaller
#DIRECTORIO="/home/nanoelectronica/Downloads"
DIRECTORIO="/home/$current_user/Downloads"

# Directorio de instalación de SynopsysInstaller
INSTALACION_DIR="/usr/synopsys/installer"

ESTADO_FILE3="/home/$current_user/Downloads/Installer_Flags/
    synopsys_installer_state2"

# Comprobar si hay instalacion previa
if [ -d "$INSTALACION_DIR" ] && [ "$(ls -A "$INSTALACION_DIR")" ]; then
    echo "Synopsys ya parece estar instalado en $INSTALACION_DIR."
    read -p "¿Desea limpiar el directorio de instalación y proceder con una nueva
        instalación? (s/n): " RESPUESTA

    if [[ "$RESPUESTA" =~ ^[Ss]$ ]]; then
    echo "Limpiando el directorio de instalación..."
    rm -rf "$INSTALACION_DIR"/*
    rm "$ESTADO_FILE3"
    echo "Directorio limpiado."

    else
    echo "Instalación abortada."
    exit 0
    fi
fi

# Encontrar los archivos SynopsysInstaller_*.run disponibles
ARCHIVOS=($(find "$DIRECTORIO" -maxdepth 1 -name "SynopsysInstaller_*.run" | sort
    -V))

if [ ${#ARCHIVOS[@]} -eq 0 ]; then
    echo "No se encontró ningún archivo que coincida con SynopsysInstaller_*.run"
    exit 1
fi

# Mostrar menú de versiones disponibles
echo ""
echo "Seleccione la versión de SynopsysInstaller que desea instalar:"
for i in "${!ARCHIVOS[@]}"; do
    echo "$((i+1)). ${ARCHIVOS[$i]}"
```

```
done

# Leer la selección del usuario
read -p "Ingrese el número de la versión que desea instalar: " SELECCION

if ! [[ "$SELECCION" =~ ^[0-9]+$ ]] || [ "$SELECCION" -lt 1 ] || [ "$SELECCION" -
    gt "${#ARCHIVOS[@]}" ]; then
    echo "Selección inválida. Saliendo..."
    exit 1
fi

# Obtener el archivo seleccionado
ARCHIVO="${ARCHIVOS[$((SELECCION-1))]}"
echo "Has seleccionado el archivo: $ARCHIVO"

# Hacer el archivo ejecutable
chmod +x "$ARCHIVO"
echo "El archivo '$ARCHIVO' ahora es ejecutable."

# Crear el directorio de instalación si no existe
if [ ! -d "$INSTALACION_DIR" ]; then
    sudo mkdir -p "$INSTALACION_DIR"
fi

# Ejecutar el archivo con el directorio de instalación especificado
sudo "$ARCHIVO" -dir "$INSTALACION_DIR"

# Cambiar la propiedad de /usr/synopsys y sus contenidos
#sudo chown -R nanoelectronica:nanoelectronica "$INSTALACION_DIR"
sudo chown -R $current_user:$current_group "$INSTALACION_DIR"

# Configurar permisos para /usr/synopsys
sudo chmod -R +x "$INSTALACION_DIR"

echo "El archivo '$ARCHIVO' ha sido ejecutado con el directorio de instalación '
    $INSTALACION_DIR'."
touch "$ESTADO_FILE3"
}


#____________________________________________________________#

# ------------| CONFIGURACIONES INICIALES | --------------
#____________________________________________________________#

# Archivo de estado
current_user=$USER
ESTADO_FILE3="/home/$current_user/Downloads/Installer_Flags/
    synopsys_installer_state2"
ESTADO_FILE2="/home/$current_user/Downloads/Installer_Flags/
    synopsys_installer_state"
ESTADO_FILE="/home/$current_user/Downloads/Installer_Flags/dependencias_state"

#____________________________________________________________#
```

```
# ------------------| MAIN PRINCIPAL | -------------------
#_____________________________________________________#


while true; do
    echo "_____________________________________________________________________________"
    echo "-----------| MENU DE INSTALACIÓN DE SOFTWARE DE SYNOPSYS |--------------"
    echo "_____________________________________________________________________________"
    echo "1) Descargar y configurar dependencias iniciales"
    echo "2) Descargar Synopsys Installer"
    echo "3) Instalar Synopsys Installer"
    echo "4) Descargar aplicaciones de Synopsys (APPS.sh)"
    echo "5) Help "
    echo "6) Salir"
    echo " "
    read -p "Seleccione una opción: " SELECCION
    echo ""

    case $SELECCION in

    1)
    # +++++++++++++++++++ Dependencias Iniciales +++++++++++++++++++++
    if [ ! -f "$ESTADO_FILE" ]; then
        echo "Ejecutando Función de Dependencias_Iniciales..."
        echo " "
        Dependencias_Iniciales

        # Verificar si dependencias_iniciales.sh se ejecutó correctamente
        if [ $? -eq 0 ]; then

        echo "Dependencias_Iniciales se ejecutó correctamente."
        echo ""
        else
        echo "Error al ejecutar Dependencias_Iniciales. Saliendo..."
        exit 1
        fi

    else
        echo "Las dependencias ya se han instalado anteriormente."
        #secho " "
    fi
    ;;

    2)

    # ++++++++++++ Descarga Inicial de Installer_Synopsys ++++++++++++
    if [ -f "$ESTADO_FILE" ] && [ ! -f "$ESTADO_FILE2" ]; then

        # Verificar si el directorio de instalación está vacío
        if [ -z "$(ls -A /usr/synopsys/installer)" ]; then
        # Ejecutar APPS.sh para descargar inicialmente el instalador de synopsys
        echo "El directorio /usr/synopsys/installer está vacío. Ejecutando APPS.sh
            para descargar inicialmente el instalador de Synopsys..."
```

```
        echo "Ejecutando APPS.sh..."
        echo " "
        ./APPS.sh

        # Verificar si APPS.sh se ejecutó correctamente
        if [ $? -eq 0 ]; then
            echo "APPS.sh se ejecutó correctamente."
        else
            echo "Error al ejecutar APPS.sh. Saliendo..."
            exit 1
        fi
        fi

        # Verificar si existen archivos SynopsysInstaller_*.run en la carpeta
            Downloads
        # Esta condición servira para crear un archivo de estado y asi ejecutar una
             unica vez el \textit{script} de APPS.sh al descargar por primera vez
            el Installer
        if find /home/$current_user/Downloads/Installer_Flags -name "
            SynopsysInstaller_*.run" | grep -q .; then
        # Crear el archivo de estado
        touch "$ESTADO_FILE2"
        echo "Archivo de estado creado."
        else
        echo "No se encontró ningún archivo SynopsysInstaller_*.run en la carpeta
            Downloads. Saliendo..."
        #exit 1
        fi
    elif [ -f "$ESTADO_FILE" ] && [ -f "$ESTADO_FILE2" ]; then
        echo "Ya tiene descargado un SynopsysInstaller*.run."
    else
        echo "¡ERROR! No se ha ejecutado la Opción 1 correctamente (Instalación de
            dependencias)"
        echo "Intente nuevamente una vez que haya ejecutado correctamente la opción
             1."
        echo " "
    fi
    ;;

        3)
            # ++++++++++++++ Ejecutar Synopsys_Installer para realizar la
                instalacion del instalador ++++++++++++
    if [ -f "$ESTADO_FILE2" ]; then
            echo "Ejecutando Función de Synopsys_Installer..."
            Synopsys_Installer

            # Verificar si synopsys_installer.sh se ejecutó correctamente
            if [ $? -eq 0 ]; then
                echo "Synopsys_Installer se ejecutó correctamente."
        echo ""
            else
                echo "Error al ejecutar Synopsys_Installer. Saliendo..."
                exit 1
            fi
```

```
    else
        echo "¡ERROR! No se ha realizado una descarga del SynopsysIstaller*.run"
        echo "Intente nuevamente una vez que haya ejecutado correctamente la opción
             2"
        echo " "
    fi
            ;;

        4)
        # +++++++++++++++ Descargar las aplicaciones ++++++++++++++++++++
    if [ -f "$ESTADO_FILE3" ]; then
            # Verificar si el directorio está vacío
            if [ -z "$(ls -A /usr/synopsys/installer)" ]; then
        echo "¡ADVERTENCIA! -> No se puede instalar las aplicaciones de Synopsys."
                echo "Es necesario seleccionar la Opción 1 del menú para instalar el
                     instalador de Synopsys."
        echo "Ejecute la opción 1 y vuelva a intentar."

            else
                # +++++++++++++++ Ejecutar APPS.sh para descargar las aplicaciones
                    de synopsys +++++++++++++
                echo "Ejecutando APPS.sh..."
                ./APPS.sh

                # Verificar si APPS.sh se ejecutó correctamente
                if [ $? -eq 0 ]; then
                    echo "APPS.sh se ejecutó correctamente."
                    echo ""
                else
                    echo "Error al ejecutar APPS.sh. Saliendo..."
                    exit 1
                fi
            fi
    else
        echo "¡ERROR! No se ha realizado la instalación del SynopsysInstaller*run"
        echo "Intente nuevamente una vez que haya ejecutado correctamente la opción
             3"
        echo " "
    fi
            ;;

    5)
    echo "Para lograr instalar las aplicaciones correctamente se debe ejecutar"
    echo "cada opción en orden (de la opción 1 a la 4)"
    echo ""
    echo "Ayuda para la opción 1)"
    echo "Para lograr ejecutar la opción con éxito se debe tener un usuario con"
    echo "permisos de Root. Para tener un usuario con permisos de Root"
    echo "se debe realizar lo siguiente: "
    echo " 1) Verificar si un usuario pertenece al grupo "wheel" con:"
    echo " groups <nombre_usuario>"
    echo " 2) Si no se tiene ningún usuario con permisos de root crearlo con:"
    echo " su -"
    echo " useradd -m <nombre_usuario_a_crear>"
```

```
    echo " passwd <nombre_usuario_a_crear> e ingresar la contraseña"
    echo " 3) Agregar al nuevo usuario al grupo "wheel" con:"
    echo " su -"
    echo " usermod -aG wheel <nombre_usuario_creado>"
    echo " 4) Verificar si el usuario creado pertenece al grupo "wheel" con:"
    echo " groups <nombre_usuario_creado>"
        ;;

        6)
            echo "Saliendo..."
            exit 0
            ;;

        *)
            echo "Opción inválida. Intente de nuevo."
            echo ""
            ;;
    esac
done
```

Cuadro 132: Archivo Instalador_APPs_Synopsys.sh: *script* realizado para mejorar el proceso de descarga de las aplicaciones de Synopsys

15.1. Pasos previos a realizar antes de ejecutar el *script*

15.1.1. Verificación de *scripts* con permisos de ejecución

1. En una terminal, verificar si los *scripts* tienen permisos de ejecución con el siguiente comando:

```
ls -l
```

2. Si los *scripts* `instalador_apps_synopsys.sh` o `apps.sh` no tienen permisos de ejecución, ejecutar el siguiente comando:

```
chmod +x nombre_del_script
```

15.1.2. Creación de un usuario con permisos de *root*

Es importante crear un usuario con permisos de *root*, ya que para descargar y configurar las dependencias iniciales el *script* solicita un usuario con privilegios de superusuario para poder ejecutar ciertos comandos que requieren permisos elevados. Esto es necesario para descargar algunas librerías y crear directorios específicos en `/usr`, los cuales solo pueden configurarse con privilegios de *root*.

Para ello, se realiza lo siguiente:

1. En una terminal, ejecutar el comando `useradd` para crear un nuevo usuario y el comando `passwd` para asignarle una contraseña:

```
su -
useradd -m <nombre_usuario>
passwd <nombre_usuario>
```

2. Para permitir que el nuevo usuario ejecute comandos como *root* usando `sudo`, se debe agregar al grupo `wheel` mediante el siguiente comando:

```
su -
usermod -aG wheel <nombre_usuario>
```

3. Verificar que el nuevo usuario pertenezca al grupo `wheel` utilizando el siguiente comando:

```
groups <nombre_usuario>
```

15.2. Explicación general del *script*

El *script* presenta un menú de opciones para el usuario, permitiéndole seleccionar una de las siguientes acciones, las cuales deben ejecutarse en el orden indicado (de la opción 1 a la 4). Estas opciones se explican a continuación:

1. Descargar y configurar dependencias iniciales: llama a la función `Dependencias_Iniciales` para preparar el entorno y descargar todas las dependencias necesarias antes de ejecutar el *script* que permite la descarga de las aplicaciones de Synopsys (`APPS.sh`).

2. Descargar Synopsys Installer: verifica si las dependencias ya están instaladas. Si es así, se habilita la opción para continuar con la descarga del instalador de Synopsys mediante el *script* `APPS.sh`.

3. Instalar Synopsys Installer: si las dependencias están correctamente configuradas y el instalador de Synopsys (`SynopsysInstaller_*.run`) ya está descargado, el *script* permite ejecutar la instalación mediante la función `Synopsys_Installer`.

4. Descargar aplicaciones de Synopsys (`APPS.sh`): si el instalador de Synopsys ya está instalado, se habilita esta opción para ejecutar el *script* `APPS.sh` y descargar las aplicaciones de Synopsys.

5. *Help*: proporciona una descripción de las opciones disponibles en el menú.

6. Salir: finaliza la ejecución del *script*.

15.2.1. Función `Dependencias_Iniciales`

Esta función configura el entorno de trabajo, asegurando que las dependencias y configuraciones iniciales estén listas para la instalación de las aplicaciones de Synopsys.

1. Detecta el usuario actual y su grupo principal (`current_user` y `current_group`).
2. Añade la línea del servidor de licencias al archivo `.bashrc` si no está presente, o la actualiza con la nueva dirección del servidor de licencias.
3. Agrega el `PATH` del instalador de Synopsys al archivo `.bashrc`, si aún no está configurado.
4. Crea directorios importantes, como `Folder_de_Trabajo` e `Installer_Flags`, con los permisos adecuados para su uso posterior.
5. Verifica o crea la carpeta `.ssh` y se conecta al servidor de Synopsys mediante SSH, agregando su llave RSA al archivo `known_hosts` para permitir el acceso al servidor.
6. Solicita y verifica si existe un usuario con permisos de *root* para proceder con la creación de directorios en `/usr` y con la instalación de las librerías necesarias para el uso de las aplicaciones de Synopsys, entre otras tareas.

15.2.2. Función `Synopsys_Installer`

Esta función realiza la instalación automática del instalador de Synopsys, incluyendo la limpieza de instalaciones previas cuando se desee descargar una nueva versión.

1. Verifica si el instalador ya está presente en el directorio `/usr/synopsys/installer`. Si ya existe, ofrece la opción de limpiar este directorio para realizar una nueva instalación.
2. Busca archivos de instaladores en el directorio *downloads* que coincidan con el patrón `SynopsysInstaller_*.run` y muestra la lista para que el usuario seleccione cuál desea instalar.
3. Hace ejecutable el archivo seleccionado, crea el directorio del instalador si no existe y ejecuta el instalador.

CAPÍTULO 16

Conclusiones

- Se desarrollaron tres nuevos *scripts* para automatizar la síntesis lógica, permitiendo la síntesis tanto del circuito original como del circuito ya sintetizado, junto con el módulo de pines de entrada y salida, todo de manera automática. Esto elimina la necesidad de realizar manualmente dos procesos de síntesis.

  En los *scripts* `ejecutar_sintesis_logica_2.bash` y `comandos_sintesis_logica_2.script`, es necesario modificar manualmente el valor de la variable `PARAMETRO` para ingresar el nombre del circuito a sintetizar. Además, se debe ajustar la línea correspondiente a la lectura de archivos Verilog cuando el diseño depende de módulos ubicados en otros archivos. En cambio, con el *script* `ejecutar_sintesis_logica_3.bash` se solicita al usuario, desde la terminal, el nombre del circuito a sintetizar, eliminando así la necesidad de modificar directamente los *scripts*.

- Se llevó a cabo exitosamente la síntesis lógica de varios circuitos combinacionales y secuenciales (compuertas *not*, *and*, *xor* y *or*, nanochip El Gran Jaguar, sumadores de 4, 16 y 32 bits, restador de 16 bits, contadores de 2, 4 y 8 bits, ALUs de 4 bits, multiplicadores y divisores de 32 bits). Esto permitió identificar diversos aspectos que deben considerarse al sintetizar un circuito, especialmente en el caso de los circuitos secuenciales.

- Se implementó una nueva jerarquía de directorios para la síntesis lógica, lo cual permite organizar de manera adecuada todos los archivos de salida generados durante la síntesis lógica de cualquier circuito.

- Se mejoró el proceso de descarga de las aplicaciones de Synopsys mediante un nuevo *script* en Bash, el cual crea todas las dependencias iniciales (directorios, permisos, instalación de librerías y aplicaciones, y creación de llaves de seguridad RSA para la conexión al servidor de Synopsys). Este *script* también hace uso del *script* previamente desarrollado para la descarga de las aplicaciones.

- Se automatizó el proceso de instalación del instalador (`installer.run`) de las aplicaciones de Synopsys. El *script* utilizado para la descarga de las aplicaciones de Synopsys

(`APPS.sh`) permitía realizar dicha descarga, pero no ejecutar la instalación, proceso que anteriormente debía realizarse de forma manual. Esta tarea fue optimizada mediante un nuevo *script* que automatiza tanto la descarga como la instalación del instalador.

CAPÍTULO 17

Recomendaciones

- En el caso de que el circuito a sintetizar sea secuencial (dependa de algún *clock*) y, al momento de simular el circuito sintetizado, no se obtenga el comportamiento esperado, se recomienda verificar cómo se emplea el *clock*. No es posible utilizar un mismo *clock* tomando en cuenta ambos flancos (subida y bajada) dentro de una misma instrucción *always*, tal como se evidencia en la documentación correspondiente a la síntesis del divisor y del multiplicador de 32 bits.
- No se recomienda utilizar bloques *initial* en el archivo Verilog para inicializar variables al sintetizar un circuito. Estos bloques se emplean únicamente para definir comportamientos que ocurren una sola vez al inicio de una simulación (por ejemplo, la inicialización de registros), pero no son compatibles con la implementación en *hardware* real, lo que generará errores durante la síntesis.
- Para inicializar registros o variables, se recomienda utilizar bloques secuenciales, como los bloques *always* controlados por una señal de reloj o por una señal de *reset*, con el fin de inicializar correctamente los registros durante la operación del circuito.
- Se sugiere agregar más aplicaciones al algoritmo de descarga de las aplicaciones de Synopsys, ya que actualmente el *script* gestiona 19 aplicaciones. Sería útil explorar nuevas herramientas que puedan emplearse en distintos proyectos y ampliar así la automatización del entorno de trabajo.
- Se recomienda implementar y sintetizar diseños más grandes o con arquitecturas más sofisticadas para validar la robustez y escalabilidad del proceso. Un ejemplo de ello podría ser un microcontrolador *nibbler* de 4 bits, el cual podría escalarse posteriormente a versiones de 8, 16 y 32 bits.
- Sería útil desarrollar una interfaz gráfica que permita interactuar directamente con el sistema para seleccionar las librerías a utilizar, los archivos de entrada a sintetizar y para interactuar con las herramientas DVE o WaveView. Esto facilitaría la visualización de las señales tanto del circuito original como del sintetizado en una única interfaz, evitando la necesidad de trabajar exclusivamente mediante comandos.

- Se recomienda mejorar el *script* de descarga e instalación de las aplicaciones de Synopsys (`APPS.sh`). Actualmente, la variable `apps_real` contiene directamente los nombres iniciales de las aplicaciones, lo cual podría generar errores si Synopsys modifica dichos nombres. Se sugiere que la variable haga referencia a los nombres contenidos en el archivo `myproductsrev.txt`, asegurando que cualquier cambio se refleje automáticamente y evitando errores al intentar descargar alguna aplicación.

CAPÍTULO 18

Bibliografía

[1] J. de los Santos, "Diseño de un sumador/restador completo de 32 bits con tecnología CMOS en un proceso de 28 nanómetros usando aplicaciones de diseño de la empresa Synopsys," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2014.

[2] S. Rubio, "Definición del Flujo de Diseño para Fabricación de un Chip con Tecnologia VLSI CMOS," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2019.

[3] L. Najera, "Implementación de circuitos sintetizados a nivel netlist a partir de un diseño en lenguaje descriptivo de hardware como primer paso en el flujo de diseño de un circuito integrado," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2019.

[4] C. Cruz, "Ejecución y utilización de un flujo de diseño para el desarrollo de un chip con tecnología nanométrica: Extracción de componentes parásitos y simulaciones en HSPICE," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.

[5] M. Flores, "Corrección de anillo de entradas/salidas y pruebas de antenna y ERC para la definición del flujo de diseño del primer chip con tecnología nanométrica desarrollado en Guatemala," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.

[6] M. Sibrian, "Verificación de reglas de diseño (DRC) para el desarrollo de un flujo funcional de un circuito integrado con tecnología nanométrica," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.

[7] J. Girón, "Etapa de verificación física de Diseño en Silicio vs. Esquemático (LVS) en el flujo de diseño para un chip a nanoescala," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.

[8] J. Ruano, "Definición del flujo en la herramienta VCS para la simulación de HDLs en la Fabricación de un Chip con Tecnología Nanométrica CMOS," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2020.

[9] E. Torres, "Diseño de un circuito integrado con tecnología de 180 nm usando librerías de diseño de TSMC: Ejecución y simulación para la etapa de síntesis lógica," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2021.

[10] J. Ruiz, "Diseño de un circuito integrado con tecnología de 180 nm usando librerías de diseño de TSMC: ejecución de la fase de verificación física Layout vs Schematic (LVS)," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2021.

[11] J. Shin, "Diseño de un circuito integrado con tecnología de 180 nm usando librerías de diseño de TSMC: ejecución de la síntesis física, validación de reglas eléctricas y corrección de errores obtenidos," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2021.

[12] J. Ayala, "Diseño de un Circuito Integrado con Tecnología de 180 nm Usando Librerías de Diseño de TSMC: Ejecución de la Síntesis Física, Verificaciones de Antena y Corrección de Errores Obtenidos," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2021.

[13] A. Hernández, "Diseño de un Circuito Integrado con Tecnología de 180nm usando Librerías de Diseño de TSMC: Ejecución de la Síntesis Física, Verificación de Reglas de Diseño y Corrección de Errores Obtenidos," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2021.

[14] D. Equité, "Diseño e implementación de interfaz gráfica de la automatización de las fases de diseño de un circuito integrado y uso avanzado de IC Compiler II," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2022.

[15] A. Aguilar, "Automatización de la etapa de síntesis lógica y creación de archivos Verilog para pruebas físicas en un FPGA Genesys Xilinx Virtex-5 LX50T y automatización de la verificación extracción de parásitos," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2023.

[16] S. Schwendener, "Automatización de las verificaciones físicas de un circuito integrado con tecnología de 180 nm utilizando librerías de diseño de TSMC," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2022.

[17] P. Mendizábal, "Automatización del proceso de instalación de software de Synopsys e implementación de mejoras utilizando contenedores," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2022.

[18] C. Letona, "Diseño de un circuito integrado con tecnología de 180 nm usando librerías de diseño de TSMC: uso avanzado de StarRC para la generación de un archivo HSPICE con componentes parásitos para su correcta simulación," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2022.

[19] H. Ovalle, "Diseño de un circuito integrado con tecnología de 180 nm utilizando librerías de diseño de TSMC: Implementación de un alternativo flujo de diseño proporcionado por Synopsys con las herramientas de Formality y Design Compiler," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2022.

[20] I. Arévalo, "Diseño de un circuito integrado con tecnología de 180 nm utilizando librerías de diseño de TSMC: Simulación en PrimeSim y análisis de errores," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2022.

[21] J. Ponce, "Diseño de un circuito integrado con tecnología de 180 nm utilizando librerías de diseño de TSMC: Implementación de un alternativo flujo de diseño proporcionado por Synopsys con las herramientas PrimeTime y TetraMAX," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2022.

[22] E. Mancio, "Implementación y verificación de la funcionalidad de código obtenido durante la síntesis lógica de arquitectura del nano chip Gran Jaguar utilizando una plataforma de desarrollo con un FPGA Xilinx Virtex-5 Genesys," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2022.

[23] L. Gómez, "Procesamiento de las señales generadas por un circuito integrado con tecnología de 180nm usando librerías de diseño de TSMC montado en un FPGA Digilent Genesys Board demostrando el funcionamiento mediante una aplicación y sintetizador digital," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2022.

[24] N. Prado, "Diseño de un circuito integrado con tecnología de 180 nm, utilizando las librerías de diseño de TSMC y las librerías educativas de Synopsys: corrección de errores de densidad y polisilicio, verificación DRC y antena.," en *Trabajo de graduación en modalidad de Tesis*, Facultad de Ingeniería Universidad del Valle de Guatemala, 2024.

[25] K. Chauhan, "ASIC Design Flow in VLSI Engineering Services – A Quick Guide," einfochips, 2023.

[26] Synopsys.Inc, "VCS User Guide V-2023.12," Solvnet, 2023.

[27] Synopsys.Inc, "Design Compiler User Guide V-2023.12," Solvnet, 2023.

[28] Synopsys.Inc, "Design Vision User Guide V-2023.12," Solvnet, 2023.

[29] SSH, *SSH Protocol*, https://www.ssh.com/academy/ssh/protocol.

[30] SECTIGO, *SSH Keys*, https://www.sectigo.com/resource-library/what-is-an-ssh-key.

[31] SSH, *SSH File Transfer Protocol (SFTP)*, https://www.ssh.com/academy/ssh/sftp-ssh-file-transfer-protocol.

[32] opensource, *What is Bash?* https://opensource.com/resources/what-bash.

CAPÍTULO 19

Anexos

19.1. Circuito del nanochip "El gran Jaguar"

19.1.1. Circuito original

```
module chip_SP(q_out, reset, clk, EN, clk_s, select);
output [7:0] q_out;
input reset;
input clk;
input [1:0]select;
reg [11:0] contador;
reg [7:0] q;
input EN;
output clk_s;
wire W_1;
wire W_2;
wire W_3;
wire W_4;
wire W_5;
wire W_6;
wire W_7;
wire W_8;
wire W_9;
wire W_10;
wire W_11;
wire W_12;
wire W_13;
wire W_14;
wire W_15;
wire W_16;
wire W_17;
wire W_18;
wire W_19;
```

```
    wire clk_G;
    assign clk_s = clk_G;
    AND_2 U1(EN,EN,W_1);
    INV U2(W_1,W_2);
    INV U3(W_2,W_3);
    INV U4(W_3,W_4);
    INV U5(W_4,W_5);
    INV U6(W_5,W_6);
    INV U7(W_6,W_7);
    INV U8(W_7,W_8);
    INV U9(W_8,W_9);
    INV U10(W_9,W_10);
    INV U11(W_10,W_11);
    INV U12(W_11,W_12);
    INV U13(W_12,W_13);
    INV U14(W_13,W_14);
    INV U15(W_14,W_15);
    INV U16(W_15,W_16);
    INV U17(W_16,W_17);
    INV U18(W_17,W_18);
    INV U19(W_18,W_19);
    INV U20(W_19,clk_G);
    always @ (posedge reset or posedge clk)
    if (reset)
    contador<=12'b000000000000;
    else if(select==2'b00 || select==2'b11) begin
    if (contador <2185)
    contador <= contador + 1;
    else
    contador <= 0;
    end
    else if(select==2'b01 || select==2'b10) begin
    if (contador <2152)
    contador <= contador + 1;
    else
    contador <= 0;
    end
    always @ (posedge clk)
    if(select==2'b00 || select==2'b11)
    begin
    if(contador == 'd0)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd1)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd3)
    begin
```

```
    q<=8'b01100001;
    end
    else if(contador == 'd4)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd5)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd6)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd7)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd8)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd9)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd10)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd11)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd12)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd13)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd14)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd15)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd16)
    begin
    q<=8'b01101110;
    end
```

```
    else if(contador == 'd17)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd18)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd19)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd20)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd21)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd22)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd23)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd24)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd25)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd26)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd27)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd28)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd29)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd30)
    begin
```

```
    q<=8'b01110010;
    end
    else if(contador == 'd31)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd32)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd33)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd34)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd35)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd36)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd37)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd38)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd39)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd40)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd41)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd42)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd43)
    begin
    q<=8'b01110000;
    end
```

```
    else if(contador == 'd44)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd45)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd46)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd47)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd48)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd49)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd50)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd51)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd52)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd53)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd54)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd55)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd56)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd57)
    begin
```

```
    q<=8'b00100000;
    end
    else if(contador == 'd58)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd59)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd60)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd61)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd62)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd63)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd64)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd65)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd66)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd67)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd68)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd69)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd70)
    begin
    q<=8'b01100001;
    end
```

```
    else if(contador == 'd71)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd72)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd73)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd74)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd75)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd76)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd77)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd78)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd79)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd80)
    begin
    q<=8'b01010101;
    end
    else if(contador == 'd81)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd82)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd83)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd84)
    begin
```

```
    q<=8'b01100101;
    end
    else if(contador == 'd85)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd86)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd87)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd88)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd89)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd90)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd91)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd92)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd93)
    begin
    q<=8'b01000100;
    end
    else if(contador == 'd94)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd95)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd96)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd97)
    begin
    q<=8'b01110010;
    end
```

```
    else if(contador == 'd98)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd99)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd100)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd101)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd102)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd103)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd104)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd105)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd106)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd107)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd108)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd109)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd110)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd111)
    begin
```

```
    q<=8'b01100101;
    end
    else if(contador == 'd112)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd113)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd114)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd115)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd116)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd117)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd118)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd119)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd120)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd121)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd122)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd123)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd124)
    begin
    q<=8'b01100001;
    end
```

```
    else if(contador == 'd125)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd126)
    begin
    q<=8'b01010101;
    end
    else if(contador == 'd127)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd128)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd129)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd130)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd131)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd132)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd133)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd134)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd135)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd136)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd137)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd138)
    begin
```

```
    q<=8'b01100100;
    end
    else if(contador == 'd139)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd140)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd141)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd142)
    begin
    q<=8'b01010110;
    end
    else if(contador == 'd143)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd144)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd145)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd146)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd147)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd148)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd149)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd150)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd151)
    begin
    q<=8'b01000111;
    end
```

```
    else if(contador == 'd152)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd153)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd154)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd155)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd156)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd157)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd158)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd159)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd160)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd161)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd162)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd163)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd164)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd165)
    begin
```

```
    q<=8'b01100001;
    end
    else if(contador == 'd166)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd167)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd168)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd169)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd170)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd171)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd172)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd173)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd174)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd175)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd176)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd177)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd178)
    begin
    q<=8'b01100001;
    end
```

```
    else if(contador == 'd179)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd180)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd181)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd182)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd183)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd184)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd185)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd186)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd187)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd188)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd189)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd190)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd191)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd192)
    begin
```

```
    q<=8'b01101001;
    end
    else if(contador == 'd193)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd194)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd195)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd196)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd197)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd198)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd199)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd200)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd201)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd202)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd203)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd204)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd205)
    begin
    q<=8'b01110100;
    end
```

```
    else if(contador == 'd206)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd207)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd208)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd209)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd210)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd211)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd212)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd213)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd214)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd215)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd216)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd217)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd218)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd219)
    begin
```

```
    q<=8'b01100100;
    end
    else if(contador == 'd220)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd221)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd222)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd223)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd224)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd225)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd226)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd227)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd228)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd229)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd230)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd231)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd232)
    begin
    q<=8'b01101110;
    end
```

```
    else if(contador == 'd233)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd234)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd235)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd236)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd237)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd238)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd239)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd240)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd241)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd242)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd243)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd244)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd245)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd246)
    begin
```

```
    q<=8'b01101100;
    end
    else if(contador == 'd247)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd248)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd249)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd250)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd251)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd252)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd253)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd254)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd255)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd256)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd257)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd258)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd259)
    begin
    q<=8'b01000101;
    end
```

```
    else if(contador == 'd260)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd261)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd262)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd263)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd264)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd265)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd266)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd267)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd268)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd269)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd270)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd271)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd272)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd273)
    begin
```

```
    q<=8'b01110010;
    end
    else if(contador == 'd274)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd275)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd276)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd277)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd278)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd279)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd280)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd281)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd282)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd283)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd284)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd285)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd286)
    begin
    q<=8'b01110010;
    end
```

```
    else if(contador == 'd287)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd288)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd289)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd290)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd291)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd292)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd293)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd294)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd295)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd296)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd297)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd298)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd299)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd300)
    begin
```

```
    q<=8'b01100110;
    end
    else if(contador == 'd301)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd302)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd303)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd304)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd305)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd306)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd307)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd308)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd309)
    begin
    q<=8'b00111010;
    end
    else if(contador == 'd310)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd311)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd312)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd313)
    begin
    q<=8'b01110010;
    end
```

```
    else if(contador == 'd314)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd315)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd316)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd317)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd318)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd319)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd320)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd321)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd322)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd323)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd324)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd325)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd326)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd327)
    begin
```

```
    q<=8'b01110011;
    end
    else if(contador == 'd328)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd329)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd330)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd331)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd332)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd333)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd334)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd335)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd336)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd337)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd338)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd339)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd340)
    begin
    q<=8'b01100101;
    end
```

```
    else if(contador == 'd341)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd342)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd343)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd344)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd345)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd346)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd347)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd348)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd349)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd350)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd351)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd352)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd353)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd354)
    begin
```

```
    q<=8'b01101100;
    end
    else if(contador == 'd355)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd356)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd357)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd358)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd359)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd360)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd361)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd362)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd363)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd364)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd365)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd366)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd367)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd368)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd369)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd370)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd371)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd372)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd373)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd374)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd375)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd376)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd377)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd378)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd379)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd380)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd381)
    begin
```

```
    q<=8'b00101100;
    end
    else if(contador == 'd382)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd383)
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd384)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd385)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd386)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd387)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd388)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd389)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd390)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd391)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd392)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd393)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd394)
    begin
    q<=8'b01100101;
    end
```

```
    else if(contador == 'd395)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd396)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd397)
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd398)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd399)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd400)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd401)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd402)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd403)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd404)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd405)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd406)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd407)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd408)
    begin
```

```
    q<=8'b01110010;
    end
    else if(contador == 'd409)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd410)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd411)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd412)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd413)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd414)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd415)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd416)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd417)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd418)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd419)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd420)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd421)
    begin
    q<=8'b01100101;
    end
```

```
    else if(contador == 'd422)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd423)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd424)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd425)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd426)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd427)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd428)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd429)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd430)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd431)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd432)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd433)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd434)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd435)
    begin
```

```
    q<=8'b01110100;
    end
    else if(contador == 'd436)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd437)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd438)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd439)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd440)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd441)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd442)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd443)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd444)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd445)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd446)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd447)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd448)
    begin
    q<=8'b01110010;
    end
```

```
    else if(contador == 'd449)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd450)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd451)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd452)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd453)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd454)
    begin
    q<=8'b01001110;
    end
    else if(contador == 'd455)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd456)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd457)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd458)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd459)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd460)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd461)
    begin
    q<=8'b01010110;
    end
    else if(contador == 'd462)
    begin
```

```
    q<=8'b01100001;
    end
    else if(contador == 'd463)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd464)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd465)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd466)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd467)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd468)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd469)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd470)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd471)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd472)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd473)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd474)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd475)
    begin
    q<=8'b01101110;
    end
```

```
    else if(contador == 'd476)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd477)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd478)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd479)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd480)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd481)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd482)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd483)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd484)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd485)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd486)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd487)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd488)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd489)
    begin
```

```
    q<=8'b01010110;
    end
    else if(contador == 'd490)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd491)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd492)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd493)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd494)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd495)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd496)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd497)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd498)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd499)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd500)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd501)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd502)
    begin
    q<=8'b01101001;
    end
```

```
    else if(contador == 'd503)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd504)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd505)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd506)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd507)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd508)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd509)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd510)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd511)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd512)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd513)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd514)
    begin
    q<=8'b01000110;
    end
    else if(contador == 'd515)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd516)
    begin
```

```
    q<=8'b01101111;
    end
    else if(contador == 'd517)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd518)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd519)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd520)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd521)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd522)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd523)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd524)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd525)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd526)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd527)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd528)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd529)
    begin
    q<=8'b01001101;
    end
```

```
    else if(contador == 'd530)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd531)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd532)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd533)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd534)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd535)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd536)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd537)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd538)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd539)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd540)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd541)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd542)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd543)
    begin
```

```
    q<=8'b01100001;
    end
    else if(contador == 'd544)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd545)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd546)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd547)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd548)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd549)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd550)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd551)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd552)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd553)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd554)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd555)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd556)
    begin
    q<=8'b01001010;
    end
```

```
    else if(contador == 'd557)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd558)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd559)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd560)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd561)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd562)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd563)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd564)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd565)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd566)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd567)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd568)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd569)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd570)
    begin
```

```
    q<=8'b01010011;
    end
    else if(contador == 'd571)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd572)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd573)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd574)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd575)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd576)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd577)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd578)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd579)
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd580)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd581)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd582)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd583)
    begin
    q<=8'b01101100;
    end
```

```
    else if(contador == 'd584)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd585)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd586)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd587)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd588)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd589)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd590)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd591)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd592)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd593)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd594)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd595)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd596)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd597)
    begin
```

```
    q<=8'b01101110;
    end
    else if(contador == 'd598)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd599)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd600)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd601)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd602)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd603)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd604)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd605)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd606)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd607)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd608)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd609)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd610)
    begin
    q<=8'b01001100;
    end
```

```
    else if(contador == 'd611)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd612)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd613)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd614)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd615)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd616)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd617)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd618)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd619)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd620)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd621)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd622)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd623)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd624)
    begin
```

```
    q<=8'b01000001;
    end
    else if(contador == 'd625)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd626)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd627)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd628)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd629)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd630)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd631)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd632)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd633)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd634)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd635)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd636)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd637)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd638)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd639)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd640)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd641)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd642)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd643)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd644)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd645)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd646)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd647)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd648)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd649)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd650)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd651)
    begin
```

```
    q<=8'b01101111;
    end
    else if(contador == 'd652)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd653)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd654)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd655)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd656)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd657)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd658)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd659)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd660)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd661)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd662)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd663)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd664)
    begin
    q<=8'b01110010;
    end
```

```
    else if(contador == 'd665)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd666)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd667)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd668)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd669)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd670)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd671)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd672)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd673)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd674)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd675)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd676)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd677)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd678)
    begin
```

```
    q<=8'b01101110;
    end
    else if(contador == 'd679)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd680)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd681)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd682)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd683)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd684)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd685)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd686)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd687)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd688)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd689)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd690)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd691)
    begin
    q<=8'b01100001;
    end
```

```
    else if(contador == 'd692)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd693)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd694)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd695)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd696)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd697)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd698)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd699)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd700)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd701)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd702)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd703)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd704)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd705)
    begin
```

```
    q<=8'b01101110;
    end
    else if(contador == 'd706)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd707)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd708)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd709)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd710)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd711)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd712)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd713)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd714)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd715)
    begin
    q<=8'b01001111;
    end
    else if(contador == 'd716)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd717)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd718)
    begin
    q<=8'b01111010;
    end
```

```
    else if(contador == 'd719)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd720)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd721)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd722)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd723)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd724)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd725)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd726)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd727)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd728)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd729)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd730)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd731)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd732)
    begin
```

```
    q<=8'b01110010;
    end
    else if(contador == 'd733)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd734)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd735)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd736)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd737)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd738)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd739)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd740)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd741)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd742)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd743)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd744)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd745)
    begin
    q<=8'b01101001;
    end
```

```
    else if(contador == 'd746)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd747)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd748)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd749)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd750)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd751)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd752)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd753)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd754)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd755)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd756)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd757)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd758)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd759)
    begin
```

```
    q<=8'b01100101;
    end
    else if(contador == 'd760)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd761)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd762)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd763)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd764)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd765)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd766)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd767)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd768)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd769)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd770)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd771)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd772)
    begin
    q<=8'b01101111;
    end
```

```
    else if(contador == 'd773)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd774)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd775)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd776)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd777)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd778)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd779)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd780)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd781)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd782)
    begin
    q<=8'b01001111;
    end
    else if(contador == 'd783)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd784)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd785)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd786)
    begin
```

```
    q<=8'b01101001;
    end
    else if(contador == 'd787)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd788)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd789)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd790)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd791)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd792)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd793)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd794)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd795)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd796)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd797)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd798)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd799)
    begin
    q<=8'b01110010;
    end
```

```
    else if(contador == 'd800)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd801)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd802)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd803)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd804)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd805)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd806)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd807)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd808)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd809)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd810)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd811)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd812)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd813)
    begin
```

```
    q<=8'b01100001;
    end
    else if(contador == 'd814)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd815)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd816)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd817)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd818)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd819)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd820)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd821)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd822)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd823)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd824)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd825)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd826)
    begin
    q<=8'b01101001;
    end
```

```
    else if(contador == 'd827)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd828)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd829)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd830)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd831)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd832)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd833)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd834)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd835)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd836)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd837)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd838)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd839)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd840)
    begin
```

```
    q<=8'b01100001;
    end
    else if(contador == 'd841)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd842)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd843)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd844)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd845)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd846)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd847)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd848)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd849)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd850)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd851)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd852)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd853)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd854)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd855)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd856)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd857)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd858)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd859)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd860)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd861)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd862)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd863)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd864)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd865)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd866)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd867)
    begin
```

```
    q<=8'b00100000;
    end
    else if(contador == 'd868)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd869)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd870)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd871)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd872)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd873)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd874)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd875)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd876)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd877)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd878)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd879)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd880)
    begin
    q<=8'b01100101;
    end
```

```
    else if(contador == 'd881)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd882)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd883)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd884)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd885)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd886)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd887)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd888)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd889)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd890)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd891)
    begin
    q<=8'b01001111;
    end
    else if(contador == 'd892)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd893)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd894)
    begin
```

```
    q<=8'b01101100;
    end
    else if(contador == 'd895)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd896)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd897)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd898)
    begin
    q<=8'b01000010;
    end
    else if(contador == 'd899)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd900)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd901)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd902)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd903)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd904)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd905)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd906)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd907)
    begin
    q<=8'b01001010;
    end
```

```
    else if(contador == 'd908)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd909)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd910)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd911)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd912)
    begin
    q<=8'b01000100;
    end
    else if(contador == 'd913)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd914)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd915)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd916)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd917)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd918)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd919)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd920)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd921)
    begin
```

```
    q<=8'b01100011;
    end
    else if(contador == 'd922)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd923)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd924)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd925)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd926)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd927)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd928)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd929)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd930)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd931)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd932)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd933)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd934)
    begin
    q<=8'b01110011;
    end
```

```
    else if(contador == 'd935)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd936)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd937)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd938)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd939)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd940)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd941)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd942)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd943)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd944)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd945)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd946)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd947)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd948)
    begin
```

```
    q<=8'b01000001;
    end
    else if(contador == 'd949)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd950)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd951)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd952)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd953)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd954)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd955)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd956)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd957)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd958)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd959)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd960)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd961)
    begin
    q<=8'b01101100;
    end
```

```
    else if(contador == 'd962)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd963)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd964)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd965)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd966)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd967)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd968)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd969)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd970)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd971)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd972)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd973)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd974)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd975)
    begin
```

```
    q<=8'b01110011;
    end
    else if(contador == 'd976)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd977)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd978)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd979)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd980)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd981)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd982)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd983)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd984)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd985)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd986)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd987)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd988)
    begin
    q<=8'b01100001;
    end
```

```
    else if(contador == 'd989)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd990)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd991)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd992)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd993)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd994)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd995)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd996)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd997)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd998)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd999)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1000)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1001)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1002)
    begin
```

```
    q<=8'b01101111;
    end
    else if(contador == 'd1003)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1004)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1005)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1006)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1007)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1008)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1009)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd1010)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1011)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1012)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1013)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1014)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1015)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd1016)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1017)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1018)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1019)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1020)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1021)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1022)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1023)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1024)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1025)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1026)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1027)
    begin
    q<=8'b01000010;
    end
    else if(contador == 'd1028)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1029)
    begin
```

```
    q<=8'b01110100;
    end
    else if(contador == 'd1030)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1031)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1032)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1033)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1034)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1035)
    begin
    q<=8'b01000100;
    end
    else if(contador == 'd1036)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1037)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1038)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1039)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1040)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1041)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd1042)
    begin
    q<=8'b01101111;
    end
```

```
    else if(contador == 'd1043)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1044)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1045)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1046)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1047)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1048)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1049)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1050)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd1051)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1052)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1053)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1054)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1055)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1056)
    begin
```

```
    q<=8'b01010000;
    end
    else if(contador == 'd1057)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1058)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1059)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1060)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1061)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1062)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1063)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1064)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1065)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1066)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1067)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1068)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1069)
    begin
    q<=8'b01100100;
    end
```

```
    else if(contador == 'd1070)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1071)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1072)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1073)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1074)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1075)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1076)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1077)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1078)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1079)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1080)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1081)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1082)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1083)
    begin
```

```
    q<=8'b01100011;
    end
    else if(contador == 'd1084)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1085)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1086)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1087)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd1088)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1089)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1090)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1091)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1092)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1093)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1094)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1095)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1096)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd1097)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd1098)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1099)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1100)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1101)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1102)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1103)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1104)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1105)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1106)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1107)
    begin
    q<=8'b01111000;
    end
    else if(contador == 'd1108)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1109)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1110)
    begin
```

```
    q<=8'b01100100;
    end
    else if(contador == 'd1111)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1112)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1113)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1114)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd1115)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1116)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1117)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd1118)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1119)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1120)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1121)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1122)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1123)
    begin
    q<=8'b01100101;
    end
```

```
    else if(contador == 'd1124)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1125)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1126)
    begin
    q<=8'b01000110;
    end
    else if(contador == 'd1127)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1128)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1129)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1130)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1131)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1132)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1133)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1134)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1135)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1136)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1137)
    begin
```

```
    q<=8'b00100000;
    end
    else if(contador == 'd1138)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd1139)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1140)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1141)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1142)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1143)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd1144)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1145)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1146)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1147)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1148)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1149)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1150)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd1151)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1152)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1153)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1154)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1155)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1156)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1157)
    begin
    q<=8'b01010110;
    end
    else if(contador == 'd1158)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1159)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1160)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1161)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1162)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd1163)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1164)
    begin
```

```
    q<=8'b01100101;
    end
    else if(contador == 'd1165)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1166)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1167)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1168)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd1169)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1170)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1171)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1172)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1173)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1174)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1175)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1176)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1177)
    begin
    q<=8'b01101110;
    end
```

```
    else if(contador == 'd1178)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1179)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1180)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1181)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1182)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd1183)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1184)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1185)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1186)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1187)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1188)
    begin
    q<=8'b01001111;
    end
    else if(contador == 'd1189)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1190)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1191)
    begin
```

```
    q<=8'b01100101;
    end
    else if(contador == 'd1192)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1193)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1194)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd1195)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1196)
    begin
    q<=8'b01001110;
    end
    else if(contador == 'd1197)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1198)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1199)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1200)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1201)
    begin
    q<=8'b01000110;
    end
    else if(contador == 'd1202)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1203)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1204)
    begin
    q<=8'b01101110;
    end
```

```
    else if(contador == 'd1205)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1206)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1207)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1208)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1209)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1210)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1211)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1212)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1213)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1214)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1215)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1216)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1217)
    begin
    q<=8'b01000010;
    end
    else if(contador == 'd1218)
    begin
```

```
    q<=8'b01110101;
    end
    else if(contador == 'd1219)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1220)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1221)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1222)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1223)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1224)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1225)
    begin
    q<=8'b01000100;
    end
    else if(contador == 'd1226)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1227)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1228)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1229)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1230)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1231)
    begin
    q<=8'b01010011;
    end
```

```
    else if(contador == 'd1232)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1233)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1234)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1235)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1236)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1237)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1238)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1239)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1240)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1241)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1242)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1243)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1244)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1245)
    begin
```

```
    q<=8'b00100000;
    end
    else if(contador == 'd1246)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1247)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1248)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1249)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1250)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1251)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1252)
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd1253)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1254)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1255)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1256)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1257)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1258)
    begin
    q<=8'b01101001;
    end
```

```
    else if(contador == 'd1259)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1260)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1261)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1262)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1263)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1264)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1265)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1266)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1267)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1268)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1269)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1270)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd1271)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1272)
    begin
```

```
    q<=8'b01100011;
    end
    else if(contador == 'd1273)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1274)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1275)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1276)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1277)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1278)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1279)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1280)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1281)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd1282)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1283)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1284)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1285)
    begin
    q<=8'b01100100;
    end
```

```
    else if(contador == 'd1286)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1287)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1288)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1289)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1290)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1291)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1292)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1293)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1294)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1295)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd1296)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1297)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1298)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1299)
    begin
```

```
    q<=8'b00100000;
    end
    else if(contador == 'd1300)
    begin
    q<=8'b01010110;
    end
    else if(contador == 'd1301)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1302)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1303)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd1304)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1305)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1306)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1307)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1308)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1309)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd1310)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1311)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1312)
    begin
    q<=8'b01101110;
    end
```

```
    else if(contador == 'd1313)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1314)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1315)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1316)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1317)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1318)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1319)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1320)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1321)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1322)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1323)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1324)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1325)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1326)
    begin
```

```
    q<=8'b01110010;
    end
    else if(contador == 'd1327)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1328)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1329)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1330)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1331)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1332)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1333)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1334)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1335)
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd1336)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1337)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1338)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1339)
    begin
    q<=8'b01101110;
    end
```

```
    else if(contador == 'd1340)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1341)
    begin
    q<=8'b01000100;
    end
    else if(contador == 'd1342)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1343)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1344)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1345)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1346)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1347)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd1348)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1349)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1350)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1351)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1352)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1353)
    begin
```

```
    q<=8'b01100100;
    end
    else if(contador == 'd1354)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1355)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1356)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1357)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1358)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1359)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1360)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1361)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1362)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1363)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1364)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd1365)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1366)
    begin
    q<=8'b01001100;
    end
```

```
    else if(contador == 'd1367)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1368)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1369)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1370)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1371)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1372)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1373)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1374)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1375)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1376)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1377)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1378)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1379)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1380)
    begin
```

```
    q<=8'b01110010;
    end
    else if(contador == 'd1381)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1382)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1383)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1384)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1385)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1386)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1387)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1388)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1389)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd1390)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd1391)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd1392)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1393)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd1394)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1395)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1396)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1397)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1398)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1399)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1400)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1401)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1402)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1403)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1404)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1405)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1406)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1407)
    begin
```

```
    q<=8'b01101111;
    end
    else if(contador == 'd1408)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1409)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1410)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1411)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1412)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1413)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1414)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1415)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1416)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1417)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1418)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1419)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1420)
    begin
    q<=8'b01101100;
    end
```

```
    else if(contador == 'd1421)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1422)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1423)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1424)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1425)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1426)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1427)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1428)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1429)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1430)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1431)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1432)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1433)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1434)
    begin
```

```
    q<=8'b01100100;
    end
    else if(contador == 'd1435)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1436)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1437)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1438)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1439)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1440)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1441)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1442)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1443)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1444)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1445)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1446)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1447)
    begin
    q<=8'b01101101;
    end
```

```
    else if(contador == 'd1448)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1449)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1450)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1451)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1452)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1453)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1454)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1455)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1456)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1457)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1458)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1459)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1460)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1461)
    begin
```

```
    q<=8'b01110011;
    end
    else if(contador == 'd1462)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1463)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1464)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1465)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1466)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1467)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1468)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1469)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1470)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1471)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1472)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1473)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1474)
    begin
    q<=8'b01101110;
    end
```

```
    else if(contador == 'd1475)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1476)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1477)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1478)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1479)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1480)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1481)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1482)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1483)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1484)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1485)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1486)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1487)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1488)
    begin
```

```
    q<=8'b01000101;
    end
    else if(contador == 'd1489)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1490)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1491)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1492)
    begin
    q<=8'b00101101;
    end
    else if(contador == 'd1493)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1494)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1495)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1496)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1497)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1498)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1499)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1500)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1501)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd1502)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1503)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1504)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1505)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1506)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1507)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1508)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1509)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1510)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1511)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1512)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1513)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1514)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1515)
    begin
```

```
    q<=8'b01110010;
    end
    else if(contador == 'd1516)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1517)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1518)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1519)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1520)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1521)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1522)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1523)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1524)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1525)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1526)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1527)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1528)
    begin
    q<=8'b01100100;
    end
```

```
    else if(contador == 'd1529)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1530)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1531)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1532)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1533)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1534)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1535)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1536)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1537)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1538)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1539)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1540)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1541)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1542)
    begin
```

```
    q<=8'b01101001;
    end
    else if(contador == 'd1543)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1544)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1545)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1546)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1547)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1548)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1549)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1550)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1551)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1552)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1553)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1554)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1555)
    begin
    q<=8'b01101110;
    end
```

```
    else if(contador == 'd1556)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1557)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd1558)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1559)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1560)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1561)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1562)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1563)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd1564)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1565)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd1566)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1567)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1568)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1569)
    begin
```

```
    q<=8'b01101001;
    end
    else if(contador == 'd1570)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1571)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1572)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1573)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1574)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1575)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1576)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1577)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1578)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1579)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1580)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1581)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1582)
    begin
    q<=8'b01100100;
    end
```

```
    else if(contador == 'd1583)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1584)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1585)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd1586)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1587)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1588)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd1589)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1590)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1591)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1592)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1593)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1594)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1595)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1596)
    begin
```

```
    q<=8'b01101111;
    end
    else if(contador == 'd1597)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1598)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1599)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1600)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1601)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1602)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1603)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1604)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1605)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1606)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1607)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1608)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1609)
    begin
    q<=8'b01100010;
    end
```

```
    else if(contador == 'd1610)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1611)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1612)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1613)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1614)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1615)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1616)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1617)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1618)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1619)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1620)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1621)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1622)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1623)
    begin
```

```
    q<=8'b01110010;
    end
    else if(contador == 'd1624)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1625)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1626)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1627)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1628)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1629)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1630)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1631)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1632)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1633)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1634)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1635)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1636)
    begin
    q<=8'b01100101;
    end
```

```
    else if(contador == 'd1637)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1638)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1639)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1640)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1641)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1642)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1643)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1644)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1645)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1646)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1647)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1648)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1649)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1650)
    begin
```

```
    q<=8'b01101100;
    end
    else if(contador == 'd1651)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1652)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1653)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd1654)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1655)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1656)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd1657)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1658)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1659)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1660)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd1661)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1662)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1663)
    begin
    q<=8'b01101001;
    end
```

```
    else if(contador == 'd1664)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1665)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1666)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1667)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1668)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1669)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1670)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1671)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1672)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1673)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1674)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1675)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1676)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1677)
    begin
```

```
    q<=8'b01110101;
    end
    else if(contador == 'd1678)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1679)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1680)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1681)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1682)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1683)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1684)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1685)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1686)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1687)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1688)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd1689)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1690)
    begin
    q<=8'b01101101;
    end
```

```
    else if(contador == 'd1691)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1692)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1693)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1694)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1695)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1696)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1697)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1698)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1699)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1700)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1701)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1702)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1703)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1704)
    begin
```

```
    q<=8'b01101111;
    end
    else if(contador == 'd1705)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1706)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1707)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1708)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1709)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1710)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1711)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1712)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1713)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1714)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1715)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1716)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1717)
    begin
    q<=8'b01110100;
    end
```

```
    else if(contador == 'd1718)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1719)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1720)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1721)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1722)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1723)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1724)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1725)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1726)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1727)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1728)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1729)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1730)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1731)
    begin
```

```
    q<=8'b01101111;
    end
    else if(contador == 'd1732)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1733)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1734)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1735)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1736)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1737)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1738)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd1739)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1740)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd1741)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1742)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1743)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1744)
    begin
    q<=8'b01101001;
    end
```

```
    else if(contador == 'd1745)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1746)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1747)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1748)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1749)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1750)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1751)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1752)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1753)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1754)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1755)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1756)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd1757)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1758)
    begin
```

```
    q<=8'b00100000;
    end
    else if(contador == 'd1759)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1760)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1761)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1762)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1763)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1764)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1765)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1766)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1767)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1768)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1769)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1770)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1771)
    begin
    q<=8'b01101001;
    end
```

```
    else if(contador == 'd1772)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1773)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1774)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1775)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1776)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1777)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1778)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1779)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1780)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1781)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1782)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1783)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1784)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1785)
    begin
```

```
    q<=8'b01110010;
    end
    else if(contador == 'd1786)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1787)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1788)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1789)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1790)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1791)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1792)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1793)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1794)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1795)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1796)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1797)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1798)
    begin
    q<=8'b01101111;
    end
```

```
    else if(contador == 'd1799)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1800)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1801)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1802)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1803)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1804)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1805)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1806)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1807)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1808)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1809)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1810)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1811)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1812)
    begin
```

```
    q<=8'b01100101;
    end
    else if(contador == 'd1813)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1814)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1815)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1816)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1817)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1818)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1819)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1820)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1821)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1822)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1823)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1824)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1825)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd1826)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1827)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1828)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1829)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1830)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1831)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1832)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1833)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1834)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1835)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1836)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1837)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1838)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1839)
    begin
```

```
    q<=8'b00100000;
    end
    else if(contador == 'd1840)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1841)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1842)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1843)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1844)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1845)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1846)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1847)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1848)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1849)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1850)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1851)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1852)
    begin
    q<=8'b01110100;
    end
```

```
    else if(contador == 'd1853)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1854)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1855)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1856)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1857)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1858)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1859)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1860)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1861)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1862)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1863)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1864)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1865)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1866)
    begin
```

```
    q<=8'b01110101;
    end
    else if(contador == 'd1867)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1868)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1869)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1870)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1871)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1872)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1873)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1874)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1875)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1876)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1877)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1878)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1879)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd1880)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1881)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1882)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1883)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1884)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1885)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1886)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1887)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1888)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1889)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1890)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1891)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1892)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1893)
    begin
```

```
    q<=8'b01101110;
    end
    else if(contador == 'd1894)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1895)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1896)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1897)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1898)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1899)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1900)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd1901)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1902)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1903)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1904)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1905)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1906)
    begin
    q<=8'b01111001;
    end
```

```
    else if(contador == 'd1907)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1908)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1909)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1910)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1911)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1912)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1913)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1914)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1915)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1916)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1917)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1918)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1919)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1920)
    begin
```

```
    q<=8'b01101101;
    end
    else if(contador == 'd1921)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1922)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1923)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1924)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1925)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1926)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1927)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1928)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1929)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1930)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1931)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1932)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1933)
    begin
    q<=8'b01101100;
    end
```

```
    else if(contador == 'd1934)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1935)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1936)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1937)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1938)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1939)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1940)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1941)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1942)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1943)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1944)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1945)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1946)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1947)
    begin
```

```
    q<=8'b00101110;
    end
    else if(contador == 'd1948)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1949)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1950)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1951)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1952)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1953)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1954)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1955)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1956)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1957)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd1958)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1959)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1960)
    begin
    q<=8'b01110101;
    end
```

```
    else if(contador == 'd1961)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1962)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1963)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1964)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1965)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1966)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1967)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1968)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1969)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1970)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1971)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1972)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1973)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1974)
    begin
```

```
    q<=8'b01100101;
    end
    else if(contador == 'd1975)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1976)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1977)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1978)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1979)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1980)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1981)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1982)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1983)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1984)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1985)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1986)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1987)
    begin
    q<=8'b01101111;
    end
```

```
    else if(contador == 'd1988)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1989)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd1990)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1991)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1992)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1993)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1994)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1995)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1996)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1997)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1998)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1999)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2000)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2001)
    begin
```

```
    q<=8'b01101111;
    end
    else if(contador == 'd2002)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2003)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2004)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2005)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd2006)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2007)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2008)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2009)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2010)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2011)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd2012)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2013)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2014)
    begin
    q<=8'b01101001;
    end
```

```
    else if(contador == 'd2015)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2016)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2017)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2018)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2019)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2020)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2021)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2022)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2023)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd2024)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2025)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd2026)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2027)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd2028)
    begin
```

```
    q<=8'b01101110;
    end
    else if(contador == 'd2029)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2030)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2031)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2032)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2033)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2034)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2035)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2036)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2037)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2038)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2039)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2040)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2041)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd2042)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2043)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2044)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd2045)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2046)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2047)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2048)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2049)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd2050)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2051)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2052)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2053)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2054)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2055)
    begin
```

```
    q<=8'b00100000;
    end
    else if(contador == 'd2056)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd2057)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2058)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2059)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2060)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2061)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2062)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd2063)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd2064)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2065)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2066)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2067)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2068)
    begin
    q<=8'b00100000;
    end
```

```
    else if(contador == 'd2069)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2070)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2071)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2072)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2073)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd2074)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2075)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd2076)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2077)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2078)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2079)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2080)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2081)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2082)
    begin
```

```
    q<=8'b01101110;
    end
    else if(contador == 'd2083)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2084)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2085)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2086)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd2087)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2088)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2089)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2090)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2091)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd2092)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2093)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2094)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd2095)
    begin
    q<=8'b01101100;
    end
```

```
    else if(contador == 'd2096)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2097)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2098)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2099)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2100)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2101)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2102)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2103)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2104)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2105)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2106)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2107)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2108)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2109)
    begin
```

```
    q<=8'b01100010;
    end
    else if(contador == 'd2110)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2111)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2112)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2113)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd2114)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2115)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2116)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2117)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2118)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2119)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2120)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2121)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2122)
    begin
    q<=8'b01101100;
    end
```

```
    else if(contador == 'd2123)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2124)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2125)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2126)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2127)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2128)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2129)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2130)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2131)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2132)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2133)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2134)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd2135)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2136)
    begin
```

```
    q<=8'b01101111;
    end
    else if(contador == 'd2137)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd2138)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2139)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2140)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2141)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd2142)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2143)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd2144)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd2145)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2146)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2147)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2148)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2149)
    begin
    q<=8'b01110000;
    end
```

```
    else if(contador == 'd2150)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2151)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2152)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd2153)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2154)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2155)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2156)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2157)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2158)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2159)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2160)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2161)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2162)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2163)
    begin
```

```
    q<=8'b00100000;
    end
    else if(contador == 'd2164)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2165)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2166)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2167)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2168)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2169)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2170)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2171)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2172)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2173)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd2174)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd2175)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2176)
    begin
    q<=8'b01110100;
    end
```

```
    else if(contador == 'd2177)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2178)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd2179)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2180)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2181)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2182)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2183)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2184)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2185)
    begin
    q<=8'b00101110;
    end
    end
    else if(select==2'b01 || select==2'b10)
    begin
    if(contador == 'd0)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd1)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd2)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd3)
    begin
    q<=8'b01101100;
```

```
    end
    else if(contador == 'd4)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd5)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd6)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd7)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd8)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd9)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd10)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd11)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd12)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd13)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd14)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd15)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd16)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd17)
```

```
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd18)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd19)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd20)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd21)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd22)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd23)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd24)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd25)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd26)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd27)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd28)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd29)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd30)
    begin
    q<=8'b01101000;
```

```
    end
    else if(contador == 'd31)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd32)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd33)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd34)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd35)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd36)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd37)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd38)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd39)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd40)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd41)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd42)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd43)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd44)
```

```
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd45)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd46)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd47)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd48)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd49)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd50)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd51)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd52)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd53)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd54)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd55)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd56)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd57)
    begin
    q<=8'b00100000;
```

```
    end
    else if(contador == 'd58)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd59)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd60)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd61)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd62)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd63)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd64)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd65)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd66)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd67)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd68)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd69)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd70)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd71)
```

```
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd72)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd73)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd74)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd75)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd76)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd77)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd78)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd79)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd80)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd81)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd82)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd83)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd84)
    begin
    q<=8'b01101001;
```

```
    end
    else if(contador == 'd85)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd86)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd87)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd88)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd89)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd90)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd91)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd92)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd93)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd94)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd95)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd96)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd97)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd98)
```

```
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd99)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd100)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd101)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd102)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd103)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd104)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd105)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd106)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd107)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd108)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd109)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd110)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd111)
    begin
    q<=8'b01110000;
```

```
    end
    else if(contador == 'd112)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd113)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd114)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd115)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd116)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd117)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd118)
    begin
    q<=8'b01010101;
    end
    else if(contador == 'd119)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd120)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd121)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd122)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd123)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd124)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd125)
```

```
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd126)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd127)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd128)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd129)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd130)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd131)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd132)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd133)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd134)
    begin
    q<=8'b01010110;
    end
    else if(contador == 'd135)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd136)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd137)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd138)
    begin
    q<=8'b01100101;
```

```
    end
    else if(contador == 'd139)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd140)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd141)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd142)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd143)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd144)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd145)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd146)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd147)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd148)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd149)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd150)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd151)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd152)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd153)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd154)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd155)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd156)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd157)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd158)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd159)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd160)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd161)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd162)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd163)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd164)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd165)
    begin
    q<=8'b01110101;
```

```
    end
    else if(contador == 'd166)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd167)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd168)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd169)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd170)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd171)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd172)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd173)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd174)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd175)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd176)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd177)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd178)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd179)
```

```
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd180)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd181)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd182)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd183)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd184)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd185)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd186)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd187)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd188)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd189)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd190)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd191)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd192)
    begin
    q<=8'b01110011;
```

```
    end
    else if(contador == 'd193)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd194)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd195)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd196)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd197)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd198)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd199)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd200)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd201)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd202)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd203)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd204)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd205)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd206)
```

```
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd207)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd208)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd209)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd210)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd211)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd212)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd213)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd214)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd215)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd216)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd217)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd218)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd219)
    begin
    q<=8'b01101000;
```

```
    end
    else if(contador == 'd220)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd221)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd222)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd223)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd224)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd225)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd226)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd227)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd228)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd229)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd230)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd231)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd232)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd233)
```

```
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd234)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd235)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd236)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd237)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd238)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd239)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd240)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd241)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd242)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd243)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd244)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd245)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd246)
    begin
    q<=8'b01101111;
```

```
    end
    else if(contador == 'd247)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd248)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd249)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd250)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd251)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd252)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd253)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd254)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd255)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd256)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd257)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd258)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd259)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd260)
```

```
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd261)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd262)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd263)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd264)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd265)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd266)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd267)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd268)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd269)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd270)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd271)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd272)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd273)
    begin
    q<=8'b01101101;
```

```
    end
    else if(contador == 'd274)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd275)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd276)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd277)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd278)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd279)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd280)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd281)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd282)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd283)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd284)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd285)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd286)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd287)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd288)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd289)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd290)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd291)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd292)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd293)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd294)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd295)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd296)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd297)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd298)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd299)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd300)
    begin
    q<=8'b01110111;
```

```
    end
    else if(contador == 'd301)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd302)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd303)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd304)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd305)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd306)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd307)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd308)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd309)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd310)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd311)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd312)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd313)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd314)
```

```
    begin
    q<=8'b00111010;
    end
    else if(contador == 'd315)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd316)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd317)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd318)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd319)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd320)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd321)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd322)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd323)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd324)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd325)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd326)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd327)
    begin
    q<=8'b01110010;
```

```
    end
    else if(contador == 'd328)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd329)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd330)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd331)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd332)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd333)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd334)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd335)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd336)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd337)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd338)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd339)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd340)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd341)
```

```
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd342)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd343)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd344)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd345)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd346)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd347)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd348)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd349)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd350)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd351)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd352)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd353)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd354)
    begin
    q<=8'b01101000;
```

```
    end
    else if(contador == 'd355)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd356)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd357)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd358)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd359)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd360)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd361)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd362)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd363)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd364)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd365)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd366)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd367)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd368)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd369)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd370)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd371)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd372)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd373)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd374)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd375)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd376)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd377)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd378)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd379)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd380)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd381)
    begin
    q<=8'b01101000;
```

```
    end
    else if(contador == 'd382)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd383)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd384)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd385)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd386)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd387)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd388)
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd389)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd390)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd391)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd392)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd393)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd394)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd395)
```

```
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd396)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd397)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd398)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd399)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd400)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd401)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd402)
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd403)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd404)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd405)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd406)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd407)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd408)
    begin
    q<=8'b01110010;
```

```
    end
    else if(contador == 'd409)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd410)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd411)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd412)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd413)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd414)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd415)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd416)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd417)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd418)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd419)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd420)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd421)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd422)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd423)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd424)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd425)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd426)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd427)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd428)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd429)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd430)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd431)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd432)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd433)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd434)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd435)
    begin
    q<=8'b01110010;
```

```
    end
    else if(contador == 'd436)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd437)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd438)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd439)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd440)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd441)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd442)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd443)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd444)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd445)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd446)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd447)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd448)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd449)
```

```
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd450)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd451)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd452)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd453)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd454)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd455)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd456)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd457)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd458)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd459)
    begin
    q<=8'b01001110;
    end
    else if(contador == 'd460)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd461)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd462)
    begin
    q<=8'b01100101;
```

```
    end
    else if(contador == 'd463)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd464)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd465)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd466)
    begin
    q<=8'b01010110;
    end
    else if(contador == 'd467)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd468)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd469)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd470)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd471)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd472)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd473)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd474)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd475)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd476)
```

```
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd477)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd478)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd479)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd480)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd481)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd482)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd483)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd484)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd485)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd486)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd487)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd488)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd489)
    begin
    q<=8'b01110101;
```

```
    end
    else if(contador == 'd490)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd491)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd492)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd493)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd494)
    begin
    q<=8'b01010110;
    end
    else if(contador == 'd495)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd496)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd497)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd498)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd499)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd500)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd501)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd502)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd503)
```

```
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd504)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd505)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd506)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd507)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd508)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd509)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd510)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd511)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd512)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd513)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd514)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd515)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd516)
    begin
    q<=8'b01101110;
```

```
    end
    else if(contador == 'd517)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd518)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd519)
    begin
    q<=8'b01000110;
    end
    else if(contador == 'd520)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd521)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd522)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd523)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd524)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd525)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd526)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd527)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd528)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd529)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd530)
```

```
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd531)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd532)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd533)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd534)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd535)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd536)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd537)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd538)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd539)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd540)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd541)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd542)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd543)
    begin
    q<=8'b01010011;
```

```
    end
    else if(contador == 'd544)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd545)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd546)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd547)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd548)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd549)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd550)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd551)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd552)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd553)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd554)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd555)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd556)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd557)
```

```
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd558)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd559)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd560)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd561)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd562)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd563)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd564)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd565)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd566)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd567)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd568)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd569)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd570)
    begin
    q<=8'b01101001;
```

```
    end
    else if(contador == 'd571)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd572)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd573)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd574)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd575)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd576)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd577)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd578)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd579)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd580)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd581)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd582)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd583)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd584)
```

```
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd585)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd586)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd587)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd588)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd589)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd590)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd591)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd592)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd593)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd594)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd595)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd596)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd597)
    begin
    q<=8'b01000011;
```

```
    end
    else if(contador == 'd598)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd599)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd600)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd601)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd602)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd603)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd604)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd605)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd606)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd607)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd608)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd609)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd610)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd611)
```

```
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd612)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd613)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd614)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd615)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd616)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd617)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd618)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd619)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd620)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd621)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd622)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd623)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd624)
    begin
    q<=8'b01110010;
```

```
    end
    else if(contador == 'd625)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd626)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd627)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd628)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd629)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd630)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd631)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd632)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd633)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd634)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd635)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd636)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd637)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd638)
```

```
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd639)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd640)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd641)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd642)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd643)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd644)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd645)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd646)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd647)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd648)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd649)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd650)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd651)
    begin
    q<=8'b01100101;
```

```
    end
    else if(contador == 'd652)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd653)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd654)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd655)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd656)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd657)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd658)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd659)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd660)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd661)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd662)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd663)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd664)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd665)
```

```
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd666)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd667)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd668)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd669)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd670)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd671)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd672)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd673)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd674)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd675)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd676)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd677)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd678)
    begin
    q<=8'b01100100;
```

```
    end
    else if(contador == 'd679)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd680)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd681)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd682)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd683)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd684)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd685)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd686)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd687)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd688)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd689)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd690)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd691)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd692)
```

```
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd693)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd694)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd695)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd696)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd697)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd698)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd699)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd700)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd701)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd702)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd703)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd704)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd705)
    begin
    q<=8'b01101100;
```

```
    end
    else if(contador == 'd706)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd707)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd708)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd709)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd710)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd711)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd712)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd713)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd714)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd715)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd716)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd717)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd718)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd719)
```

```
    begin
    q<=8'b01001111;
    end
    else if(contador == 'd720)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd721)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd722)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd723)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd724)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd725)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd726)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd727)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd728)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd729)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd730)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd731)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd732)
    begin
    q<=8'b01010010;
```

```
    end
    else if(contador == 'd733)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd734)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd735)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd736)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd737)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd738)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd739)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd740)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd741)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd742)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd743)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd744)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd745)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd746)
```

```
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd747)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd748)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd749)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd750)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd751)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd752)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd753)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd754)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd755)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd756)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd757)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd758)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd759)
    begin
    q<=8'b01100101;
```

```
    end
    else if(contador == 'd760)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd761)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd762)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd763)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd764)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd765)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd766)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd767)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd768)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd769)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd770)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd771)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd772)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd773)
```

```
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd774)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd775)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd776)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd777)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd778)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd779)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd780)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd781)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd782)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd783)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd784)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd785)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd786)
    begin
    q<=8'b01001111;
```

```
    end
    else if(contador == 'd787)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd788)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd789)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd790)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd791)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd792)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd793)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd794)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd795)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd796)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd797)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd798)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd799)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd800)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd801)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd802)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd803)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd804)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd805)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd806)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd807)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd808)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd809)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd810)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd811)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd812)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd813)
    begin
    q<=8'b01100100;
```

```
    end
    else if(contador == 'd814)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd815)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd816)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd817)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd818)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd819)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd820)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd821)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd822)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd823)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd824)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd825)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd826)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd827)
```

```
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd828)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd829)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd830)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd831)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd832)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd833)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd834)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd835)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd836)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd837)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd838)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd839)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd840)
    begin
    q<=8'b01001000;
```

```
    end
    else if(contador == 'd841)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd842)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd843)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd844)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd845)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd846)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd847)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd848)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd849)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd850)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd851)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd852)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd853)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd854)
```

```
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd855)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd856)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd857)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd858)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd859)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd860)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd861)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd862)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd863)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd864)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd865)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd866)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd867)
    begin
    q<=8'b01110100;
```

```
    end
    else if(contador == 'd868)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd869)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd870)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd871)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd872)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd873)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd874)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd875)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd876)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd877)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd878)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd879)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd880)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd881)
```

```
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd882)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd883)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd884)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd885)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd886)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd887)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd888)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd889)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd890)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd891)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd892)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd893)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd894)
    begin
    q<=8'b00100000;
```

```
    end
    else if(contador == 'd895)
    begin
    q<=8'b01001111;
    end
    else if(contador == 'd896)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd897)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd898)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd899)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd900)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd901)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd902)
    begin
    q<=8'b01000010;
    end
    else if(contador == 'd903)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd904)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd905)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd906)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd907)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd908)
```

```
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd909)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd910)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd911)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd912)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd913)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd914)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd915)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd916)
    begin
    q<=8'b01000100;
    end
    else if(contador == 'd917)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd918)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd919)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd920)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd921)
    begin
    q<=8'b00100000;
```

```
    end
    else if(contador == 'd922)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd923)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd924)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd925)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd926)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd927)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd928)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd929)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd930)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd931)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd932)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd933)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd934)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd935)
```

```
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd936)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd937)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd938)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd939)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd940)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd941)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd942)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd943)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd944)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd945)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd946)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd947)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd948)
    begin
    q<=8'b01101110;
```

```
    end
    else if(contador == 'd949)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd950)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd951)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd952)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd953)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd954)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd955)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd956)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd957)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd958)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd959)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd960)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd961)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd962)
```

```
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd963)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd964)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd965)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd966)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd967)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd968)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd969)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd970)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd971)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd972)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd973)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd974)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd975)
    begin
    q<=8'b01101111;
```

```
    end
    else if(contador == 'd976)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd977)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd978)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd979)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd980)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd981)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd982)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd983)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd984)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd985)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd986)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd987)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd988)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd989)
```

```
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd990)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd991)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd992)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd993)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd994)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd995)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd996)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd997)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd998)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd999)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1000)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1001)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd1002)
    begin
    q<=8'b00101100;
```

```
    end
    else if(contador == 'd1003)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1004)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1005)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1006)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1007)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1008)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1009)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1010)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1011)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1012)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1013)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd1014)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1015)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1016)
```

```
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1017)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1018)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1019)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1020)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1021)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1022)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1023)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1024)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1025)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1026)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1027)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1028)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1029)
    begin
    q<=8'b01101100;
```

```
    end
    else if(contador == 'd1030)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1031)
    begin
    q<=8'b01000010;
    end
    else if(contador == 'd1032)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1033)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1034)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1035)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1036)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1037)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1038)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1039)
    begin
    q<=8'b01000100;
    end
    else if(contador == 'd1040)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1041)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1042)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1043)
```

```
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1044)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1045)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd1046)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1047)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1048)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1049)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1050)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1051)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1052)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1053)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1054)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd1055)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1056)
    begin
    q<=8'b01101001;
```

```
    end
    else if(contador == 'd1057)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1058)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1059)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1060)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1061)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1062)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1063)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1064)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1065)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1066)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1067)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1068)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1069)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1070)
```

```
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1071)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1072)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1073)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1074)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1075)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1076)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1077)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1078)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1079)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1080)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1081)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1082)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1083)
    begin
    q<=8'b00100000;
```

```
    end
    else if(contador == 'd1084)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1085)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1086)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1087)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1088)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1089)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1090)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1091)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd1092)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1093)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1094)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1095)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1096)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1097)
```

```
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1098)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1099)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1100)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1101)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd1102)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1103)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1104)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1105)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1106)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1107)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1108)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1109)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1110)
    begin
    q<=8'b01100101;
```

```
    end
    else if(contador == 'd1111)
    begin
    q<=8'b01111000;
    end
    else if(contador == 'd1112)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1113)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1114)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1115)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1116)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1117)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1118)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd1119)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1120)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1121)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd1122)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1123)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1124)
```

```
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1125)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1126)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1127)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1128)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1129)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1130)
    begin
    q<=8'b01000110;
    end
    else if(contador == 'd1131)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1132)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1133)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1134)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1135)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1136)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1137)
    begin
    q<=8'b01110100;
```

```
    end
    else if(contador == 'd1138)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1139)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1140)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1141)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1142)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd1143)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1144)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1145)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1146)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1147)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd1148)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1149)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1150)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1151)
```

```
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1152)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1153)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1154)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1155)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1156)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1157)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1158)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1159)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1160)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1161)
    begin
    q<=8'b01010110;
    end
    else if(contador == 'd1162)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1163)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1164)
    begin
    q<=8'b01100001;
```

```
    end
    else if(contador == 'd1165)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1166)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd1167)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1168)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1169)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1170)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1171)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1172)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd1173)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1174)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1175)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1176)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1177)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1178)
```

```
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1179)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1180)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1181)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1182)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1183)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1184)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1185)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1186)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd1187)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1188)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1189)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1190)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1191)
    begin
    q<=8'b00100000;
```

```
    end
    else if(contador == 'd1192)
    begin
    q<=8'b01001111;
    end
    else if(contador == 'd1193)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1194)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1195)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1196)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1197)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1198)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd1199)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1200)
    begin
    q<=8'b01001110;
    end
    else if(contador == 'd1201)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1202)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1203)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1204)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1205)
```

```
    begin
    q<=8'b01000110;
    end
    else if(contador == 'd1206)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1207)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1208)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1209)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1210)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1211)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1212)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1213)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1214)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1215)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1216)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1217)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1218)
    begin
    q<=8'b01100100;
```

```
    end
    else if(contador == 'd1219)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1220)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1221)
    begin
    q<=8'b01000010;
    end
    else if(contador == 'd1222)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1223)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1224)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1225)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1226)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1227)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1228)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1229)
    begin
    q<=8'b01000100;
    end
    else if(contador == 'd1230)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1231)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1232)
```

```
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1233)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1234)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1235)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd1236)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1237)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1238)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1239)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1240)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1241)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1242)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1243)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1244)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1245)
    begin
    q<=8'b01110010;
```

```
    end
    else if(contador == 'd1246)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1247)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1248)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1249)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1250)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1251)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1252)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1253)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1254)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1255)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1256)
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd1257)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1258)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1259)
```

```
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1260)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1261)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1262)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1263)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1264)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1265)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1266)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1267)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1268)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1269)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1270)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1271)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1272)
    begin
    q<=8'b01101000;
```

```
    end
    else if(contador == 'd1273)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1274)
    begin
    q<=8'b01010010;
    end
    else if(contador == 'd1275)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1276)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1277)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1278)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1279)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1280)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1281)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1282)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1283)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1284)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1285)
    begin
    q<=8'b01001100;
    end
    else if(contador == 'd1286)
```

```
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1287)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1288)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1289)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1290)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1291)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1292)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1293)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1294)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1295)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1296)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1297)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1298)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1299)
    begin
    q<=8'b01010010;
```

```
    end
    else if(contador == 'd1300)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1301)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1302)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1303)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1304)
    begin
    q<=8'b01010110;
    end
    else if(contador == 'd1305)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1306)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1307)
    begin
    q<=8'b01110001;
    end
    else if(contador == 'd1308)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1309)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1310)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1311)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1312)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1313)
```

```
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd1314)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1315)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1316)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1317)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1318)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1319)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1320)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1321)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1322)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1323)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1324)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1325)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1326)
    begin
    q<=8'b01110010;
```

```
    end
    else if(contador == 'd1327)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1328)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1329)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1330)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1331)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1332)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1333)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1334)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1335)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1336)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1337)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1338)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1339)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1340)
```

```
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1341)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1342)
    begin
    q<=8'b01001011;
    end
    else if(contador == 'd1343)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1344)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1345)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1346)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1347)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1348)
    begin
    q<=8'b01000100;
    end
    else if(contador == 'd1349)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1350)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1351)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1352)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1353)
    begin
    q<=8'b00100000;
```

```
    end
    else if(contador == 'd1354)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd1355)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1356)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1357)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1358)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1359)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1360)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1361)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1362)
    begin
    q<=8'b01111010;
    end
    else if(contador == 'd1363)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1364)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1365)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1366)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1367)
```

```
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1368)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1369)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1370)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1371)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd1372)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1373)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd1374)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1375)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1376)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1377)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1378)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1379)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1380)
    begin
    q<=8'b01100001;
```

```
    end
    else if(contador == 'd1381)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1382)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1383)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1384)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1385)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1386)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1387)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1388)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1389)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1390)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1391)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1392)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1393)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1394)
```

```
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1395)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1396)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1397)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1398)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1399)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1400)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1401)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd1402)
    begin
    q<=8'b01001000;
    end
    else if(contador == 'd1403)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd1404)
    begin
    q<=8'b01010000;
    end
    else if(contador == 'd1405)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1406)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd1407)
    begin
    q<=8'b01100001;
```

```
    end
    else if(contador == 'd1408)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1409)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1410)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1411)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1412)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1413)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1414)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1415)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1416)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1417)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1418)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1419)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1420)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1421)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1422)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1423)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1424)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1425)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1426)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1427)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1428)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1429)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1430)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1431)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1432)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1433)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1434)
    begin
    q<=8'b00100000;
```

```
    end
    else if(contador == 'd1435)
    begin
    q<=8'b00110010;
    end
    else if(contador == 'd1436)
    begin
    q<=8'b00110000;
    end
    else if(contador == 'd1437)
    begin
    q<=8'b00110001;
    end
    else if(contador == 'd1438)
    begin
    q<=8'b00111001;
    end
    else if(contador == 'd1439)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1440)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1441)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd1442)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1443)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1444)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1445)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1446)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1447)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1448)
```

```
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1449)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1450)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1451)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1452)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1453)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1454)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1455)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1456)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1457)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1458)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1459)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1460)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1461)
    begin
    q<=8'b01100101;
```

```
    end
    else if(contador == 'd1462)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1463)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1464)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1465)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1466)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1467)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1468)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1469)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1470)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1471)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1472)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1473)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1474)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1475)
```

```
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1476)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1477)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1478)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1479)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1480)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1481)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1482)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1483)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1484)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1485)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1486)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1487)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1488)
    begin
    q<=8'b01100110;
```

```
    end
    else if(contador == 'd1489)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1490)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1491)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1492)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1493)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1494)
    begin
    q<=8'b00101101;
    end
    else if(contador == 'd1495)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1496)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1497)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1498)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1499)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1500)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1501)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1502)
```

```
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1503)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1504)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1505)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1506)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1507)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1508)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1509)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1510)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1511)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1512)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1513)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1514)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1515)
    begin
    q<=8'b01100001;
```

```
    end
    else if(contador == 'd1516)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1517)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1518)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1519)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1520)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1521)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1522)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1523)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1524)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1525)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1526)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1527)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1528)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1529)
```

```
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1530)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1531)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1532)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1533)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1534)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1535)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1536)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1537)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1538)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1539)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1540)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1541)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1542)
    begin
    q<=8'b01110100;
```

```
    end
    else if(contador == 'd1543)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1544)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1545)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1546)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1547)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1548)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1549)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1550)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1551)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1552)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1553)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1554)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1555)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1556)
```

```
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1557)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1558)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd1559)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1560)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1561)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1562)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1563)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1564)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd1565)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1566)
    begin
    q<=8'b01000001;
    end
    else if(contador == 'd1567)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1568)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1569)
    begin
    q<=8'b01101111;
```

```
    end
    else if(contador == 'd1570)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1571)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1572)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1573)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1574)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1575)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1576)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1577)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1578)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1579)
    begin
    q<=8'b01001001;
    end
    else if(contador == 'd1580)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1581)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1582)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd1583)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1584)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1585)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1586)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1587)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1588)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1589)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1590)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd1591)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1592)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1593)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1594)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1595)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1596)
    begin
    q<=8'b01100011;
```

```
    end
    else if(contador == 'd1597)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1598)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1599)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1600)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1601)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1602)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1603)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1604)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1605)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1606)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1607)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1608)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1609)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1610)
```

```
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1611)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1612)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1613)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1614)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1615)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1616)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1617)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1618)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1619)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1620)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1621)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1622)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1623)
    begin
    q<=8'b01110010;
```

```
    end
    else if(contador == 'd1624)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1625)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1626)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1627)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1628)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1629)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1630)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1631)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1632)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1633)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1634)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1635)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1636)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1637)
```

```
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1638)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1639)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1640)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1641)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1642)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1643)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1644)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1645)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1646)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1647)
    begin
    q<=8'b01010100;
    end
    else if(contador == 'd1648)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1649)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1650)
    begin
    q<=8'b01110111;
```

```
    end
    else if(contador == 'd1651)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1652)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1653)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1654)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd1655)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1656)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1657)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1658)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1659)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1660)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1661)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1662)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1663)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1664)
```

```
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1665)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1666)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1667)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1668)
    begin
    q<=8'b01001101;
    end
    else if(contador == 'd1669)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1670)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1671)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1672)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1673)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1674)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1675)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1676)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1677)
    begin
    q<=8'b01110010;
```

```
    end
    else if(contador == 'd1678)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1679)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1680)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1681)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1682)
    begin
    q<=8'b01000011;
    end
    else if(contador == 'd1683)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1684)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1685)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1686)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1687)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1688)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1689)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1690)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1691)
```

```
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1692)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1693)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1694)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1695)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1696)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1697)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1698)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1699)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1700)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1701)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1702)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1703)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1704)
    begin
    q<=8'b01100110;
```

```
    end
    else if(contador == 'd1705)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1706)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1707)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1708)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1709)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1710)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1711)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1712)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1713)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1714)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1715)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1716)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1717)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1718)
```

```
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1719)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1720)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1721)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1722)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1723)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1724)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1725)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1726)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1727)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1728)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1729)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1730)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1731)
    begin
    q<=8'b01101110;
```

```
    end
    else if(contador == 'd1732)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1733)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1734)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1735)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1736)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1737)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1738)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1739)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd1740)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1741)
    begin
    q<=8'b01010111;
    end
    else if(contador == 'd1742)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1743)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1744)
    begin
    q<=8'b01101011;
    end
    else if(contador == 'd1745)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1746)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1747)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1748)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1749)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1750)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1751)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1752)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1753)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1754)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1755)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1756)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1757)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1758)
    begin
    q<=8'b01100111;
```

```
    end
    else if(contador == 'd1759)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1760)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1761)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1762)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1763)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1764)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1765)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1766)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1767)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1768)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1769)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1770)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1771)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1772)
```

```
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1773)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1774)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1775)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1776)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1777)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1778)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1779)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1780)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1781)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1782)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1783)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1784)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1785)
    begin
    q<=8'b01110010;
```

```
    end
    else if(contador == 'd1786)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1787)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1788)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1789)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1790)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1791)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1792)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1793)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1794)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1795)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1796)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1797)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1798)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1799)
```

```
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1800)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1801)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd1802)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1803)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1804)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1805)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1806)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1807)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1808)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1809)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1810)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1811)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1812)
    begin
    q<=8'b01101111;
```

```
    end
    else if(contador == 'd1813)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1814)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1815)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1816)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1817)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1818)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1819)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1820)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1821)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1822)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1823)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1824)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1825)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1826)
```

```
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1827)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1828)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1829)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1830)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1831)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1832)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1833)
    begin
    q<=8'b00101100;
    end
    else if(contador == 'd1834)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1835)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1836)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1837)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1838)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1839)
    begin
    q<=8'b00100000;
```

```
    end
    else if(contador == 'd1840)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1841)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1842)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1843)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1844)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1845)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1846)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1847)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1848)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1849)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1850)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1851)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1852)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1853)
```

```
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1854)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1855)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1856)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1857)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1858)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1859)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1860)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1861)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1862)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1863)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1864)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1865)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1866)
    begin
    q<=8'b01101001;
```

```
    end
    else if(contador == 'd1867)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1868)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1869)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1870)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1871)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1872)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1873)
    begin
    q<=8'b01010011;
    end
    else if(contador == 'd1874)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1875)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1876)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1877)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1878)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1879)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1880)
```

```
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1881)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1882)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1883)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1884)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1885)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1886)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1887)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1888)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1889)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1890)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1891)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1892)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1893)
    begin
    q<=8'b01110011;
```

```
    end
    else if(contador == 'd1894)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1895)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1896)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1897)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1898)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1899)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1900)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1901)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1902)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd1903)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1904)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1905)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1906)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1907)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1908)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1909)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1910)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1911)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1912)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1913)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1914)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1915)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1916)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1917)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1918)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1919)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1920)
    begin
    q<=8'b01100110;
```

```
    end
    else if(contador == 'd1921)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1922)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1923)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1924)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1925)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1926)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1927)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1928)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1929)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1930)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1931)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1932)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1933)
    begin
    q<=8'b00101110;
    end
    else if(contador == 'd1934)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1935)
    begin
    q<=8'b01000101;
    end
    else if(contador == 'd1936)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1937)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1938)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd1939)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1940)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1941)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1942)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1943)
    begin
    q<=8'b01001010;
    end
    else if(contador == 'd1944)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1945)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1946)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1947)
    begin
    q<=8'b01100001;
```

```
    end
    else if(contador == 'd1948)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1949)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1950)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1951)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1952)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1953)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1954)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1955)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1956)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1957)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd1958)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd1959)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1960)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1961)
```

```
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1962)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1963)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd1964)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd1965)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1966)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd1967)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd1968)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1969)
    begin
    q<=8'b01101010;
    end
    else if(contador == 'd1970)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1971)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1972)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1973)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1974)
    begin
    q<=8'b01110100;
```

```
    end
    else if(contador == 'd1975)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd1976)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1977)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1978)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1979)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd1980)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1981)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1982)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd1983)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1984)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd1985)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd1986)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd1987)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1988)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1989)
    begin
    q<=8'b01110111;
    end
    else if(contador == 'd1990)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd1991)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd1992)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1993)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd1994)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd1995)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd1996)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd1997)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd1998)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd1999)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2000)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2001)
    begin
    q<=8'b01110011;
```

```
    end
    else if(contador == 'd2002)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2003)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2004)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2005)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd2006)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd2007)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2008)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd2009)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2010)
    begin
    q<=8'b01110101;
    end
    else if(contador == 'd2011)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2012)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2013)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2014)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2015)
```

```
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2016)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2017)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2018)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2019)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2020)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2021)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2022)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd2023)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2024)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2025)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd2026)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2027)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2028)
    begin
    q<=8'b01100101;
```

```
    end
    else if(contador == 'd2029)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2030)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd2031)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2032)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd2033)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2034)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2035)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2036)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2037)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2038)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2039)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2040)
    begin
    q<=8'b01110110;
    end
    else if(contador == 'd2041)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2042)
```

```
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2043)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2044)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd2045)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd2046)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2047)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2048)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2049)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2050)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2051)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd2052)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2053)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2054)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2055)
    begin
    q<=8'b01110111;
```

```
    end
    else if(contador == 'd2056)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2057)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2058)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2059)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2060)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd2061)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2062)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2063)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2064)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2065)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd2066)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2067)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2068)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2069)
```

```
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2070)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2071)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2072)
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2073)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2074)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2075)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd2076)
    begin
    q<=8'b01110000;
    end
    else if(contador == 'd2077)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2078)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2079)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2080)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2081)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2082)
    begin
    q<=8'b01101001;
```

```
    end
    else if(contador == 'd2083)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2084)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2085)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2086)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2087)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd2088)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2089)
    begin
    q<=8'b01110010;
    end
    else if(contador == 'd2090)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2091)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2092)
    begin
    q<=8'b01101000;
    end
    else if(contador == 'd2093)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2094)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2095)
    begin
    q<=8'b01100010;
    end
    else if(contador == 'd2096)
```

```
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2097)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2098)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2099)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd2100)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2101)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2102)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2103)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2104)
    begin
    q<=8'b01100110;
    end
    else if(contador == 'd2105)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2106)
    begin
    q<=8'b01100111;
    end
    else if(contador == 'd2107)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2108)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2109)
    begin
    q<=8'b01100010;
```

```
    end
    else if(contador == 'd2110)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2111)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2112)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2113)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2114)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2115)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2116)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2117)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2118)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2119)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd2120)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2121)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2122)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2123)
```

```
    begin
    q<=8'b01100100;
    end
    else if(contador == 'd2124)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2125)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd2126)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2127)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2128)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2129)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2130)
    begin
    q<=8'b01111001;
    end
    else if(contador == 'd2131)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2132)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2133)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2134)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2135)
    begin
    q<=8'b01000111;
    end
    else if(contador == 'd2136)
    begin
    q<=8'b01110101;
```

```
    end
    else if(contador == 'd2137)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2138)
    begin
    q<=8'b01110100;
    end
    else if(contador == 'd2139)
    begin
    q<=8'b01100101;
    end
    else if(contador == 'd2140)
    begin
    q<=8'b01101101;
    end
    else if(contador == 'd2141)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2142)
    begin
    q<=8'b01101100;
    end
    else if(contador == 'd2143)
    begin
    q<=8'b01100001;
    end
    else if(contador == 'd2144)
    begin
    q<=8'b01101110;
    end
    else if(contador == 'd2145)
    begin
    q<=8'b00100000;
    end
    else if(contador == 'd2146)
    begin
    q<=8'b01110011;
    end
    else if(contador == 'd2147)
    begin
    q<=8'b01101111;
    end
    else if(contador == 'd2148)
    begin
    q<=8'b01100011;
    end
    else if(contador == 'd2149)
    begin
    q<=8'b01101001;
    end
    else if(contador == 'd2150)
```

```
begin
q<=8'b01100101;
end
else if(contador == 'd2151)
begin
q<=8'b01110100;
end
else if(contador == 'd2152)
begin
q<=8'b01111001;
end
end
assign q_out = q;
endmodule
module INV(A, B);
input A;
output B;
assign B = ~A;
endmodule
module AND_2(in1, in2, out);
input in1, in2;
output out;
assign out = in1 & in2;
endmodule
```

Cuadro 133: Archivo `nanochip.v` diseño del nanochip en Verilog

19.1.2. Circuito sintetizado junto con los pines de entradas y salidas

```
////////////////////////////////////////////////////////////
// Created by: Synopsys DC Expert(TM) in wire load mode
// Version : V-2023.12-SP1
// Date : Mon Nov 4 22:24:52 2024
////////////////////////////////////////////////////////////


module AND_2 ( in1, in2, out );
  input in1, in2;
  output out;


  AN2D0BWP7T U1 ( .A1(in2), .A2(in1), .Z(out) );
endmodule


module INV_18 ( A, B );
  input A;
  output B;


  CKND0BWP7T U1 ( .I(A), .ZN(B) );
endmodule
```

```
module INV_17 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_16 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_15 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_14 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_13 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_12 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule
```

```
module INV_11 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_10 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_9 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_8 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_7 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_6 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule
```

```
module INV_5 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_4 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_3 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_2 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_1 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule


module INV_0 ( A, B );
  input A;
  output B;


  CKNDOBWP7T U1 ( .I(A), .ZN(B) );
endmodule
```

```
module chip_SP_DW01_inc_0 ( A, SUM );
  input [11:0] A;
  output [11:0] SUM;
  wire n1, n2, n3, n4, n5, n6, n7, n8, n9, n10;

  CKXOR2D0BWP7T U1 ( .A1(A[9]), .A2(n1), .Z(SUM[9]) );
  XNR2D1BWP7T U2 ( .A1(n2), .A2(A[8]), .ZN(SUM[8]) );
  CKND2D0BWP7T U3 ( .A1(A[7]), .A2(n3), .ZN(n2) );
  CKXOR2D0BWP7T U4 ( .A1(n3), .A2(A[7]), .Z(SUM[7]) );
  CKXOR2D0BWP7T U5 ( .A1(n4), .A2(A[6]), .Z(SUM[6]) );
  XNR2D1BWP7T U6 ( .A1(n5), .A2(A[5]), .ZN(SUM[5]) );
  XNR2D1BWP7T U7 ( .A1(n6), .A2(A[4]), .ZN(SUM[4]) );
  CKND2D0BWP7T U8 ( .A1(A[3]), .A2(n7), .ZN(n6) );
  CKXOR2D0BWP7T U9 ( .A1(A[3]), .A2(n7), .Z(SUM[3]) );
  XNR2D1BWP7T U10 ( .A1(n8), .A2(A[2]), .ZN(SUM[2]) );
  XNR2D1BWP7T U11 ( .A1(A[1]), .A2(SUM[0]), .ZN(SUM[1]) );
  CKXOR2D0BWP7T U12 ( .A1(A[11]), .A2(n9), .Z(SUM[11]) );
  INR2D0BWP7T U13 ( .A1(A[10]), .B1(n10), .ZN(n9) );
  XNR2D1BWP7T U14 ( .A1(n10), .A2(A[10]), .ZN(SUM[10]) );
  CKND2D0BWP7T U15 ( .A1(n1), .A2(A[9]), .ZN(n10) );
  AN3D0BWP7T U16 ( .A1(A[7]), .A2(n3), .A3(A[8]), .Z(n1) );
  AN2D0BWP7T U17 ( .A1(A[6]), .A2(n4), .Z(n3) );
  INR2D0BWP7T U18 ( .A1(A[5]), .B1(n5), .ZN(n4) );
  ND3D0BWP7T U19 ( .A1(A[3]), .A2(n7), .A3(A[4]), .ZN(n5) );
  INR2D0BWP7T U20 ( .A1(A[2]), .B1(n8), .ZN(n7) );
  CKND2D0BWP7T U21 ( .A1(A[1]), .A2(A[0]), .ZN(n8) );
  CKND0BWP7T U22 ( .I(A[0]), .ZN(SUM[0]) );
endmodule


module chip_SP ( q_out, reset, clk, EN, clk_s, select );
  output [7:0] q_out;
  input [1:0] select;
  input reset, clk, EN;
  output clk_s;
  wire n3525, n3524, n3523, n3522, n3521, n3520, n3519, W_1, W_2, W_3, W_4,
        W_5, W_6, W_7, W_8, W_9, W_10, W_11, W_12, W_13, W_14, W_15, W_16,
        W_17, W_18, W_19, N4361, N4360, N4359, N4358, N4357, N4356, N4355,
        N4354, N4353, N4352, N4351, N4350, N4408, n6347, N4407, n6418, N4406,
        n6416, N4405, n6420, N4404, n6419, N4403, n3528, N4402, n6421, N4401,
        n6329, N4409, n3527, N4410, n6417, N4411, n6338, N4412, n310, n2912,
        n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n14, n15, n16,
        n17, n18, n19, n20, n21, n22, n23, n24, n25, n26, n27, n28, n29, n30,
        n31, n32, n33, n34, n35, n36, n37, n38, n39, n40, n41, n42, n43, n44,
        n45, n46, n47, n48, n49, n50, n51, n52, n53, n54, n55, n56, n57, n58,
        n59, n60, n61, n62, n63, n64, n65, n66, n67, n68, n69, n70, n71, n72,
        n73, n74, n75, n76, n77, n78, n79, n80, n81, n82, n83, n84, n85, n86,
        n87, n88, n89, n90, n91, n92, n93, n94, n95, n96, n97, n98, n99, n100,
        n101, n102, n103, n104, n105, n106, n107, n108, n109, n110, n111,
        n112, n113, n114, n115, n116, n117, n118, n119, n120, n121, n122,
        n123, n124, n125, n126, n127, n128, n129, n130, n131, n132, n133,
        n134, n135, n136, n137, n138, n139, n140, n141, n142, n143, n144,
```

```
        n145, n146, n147, n148, n149, n150, n151, n152, n153, n154, n155,
        n156, n157, n158, n159, n160, n161, n162, n163, n164, n165, n166,
        n167, n168, n169, n170, n171, n172, n173, n174, n175, n176, n177,
        n178, n179, n180, n181, n182, n183, n184, n185, n186, n187, n188,
        n189, n190, n191, n192, n193, n194, n195, n196, n197, n198, n199,
        n200, n201, n202, n203, n204, n205, n206, n207, n208, n209, n210,
        n211, n212, n213, n214, n215, n216, n217, n218, n219, n220, n221,
        n222, n223, n224, n225, n226, n227, n228, n229, n230, n231, n232,
        n233, n234, n235, n236, n237, n238, n239, n240, n241, n242, n243,
        n244, n245, n246, n247, n248, n249, n250, n251, n252, n253, n254,
        n255, n256, n257, n258, n259, n260, n261, n262, n263, n264, n265,
        n266, n267, n268, n269, n270, n271, n272, n273, n274, n275, n276,
        n277, n278, n279, n280, n281, n282, n283, n284, n285, n286, n287,
        n288, n289, n290, n291, n292, n293, n294, n295, n296, n297, n298,
        n299, n300, n301, n302, n303, n304, n305, n306, n307, n308, n309,
        n311, n312, n313, n314, n315, n316, n317, n318, n319, n320, n321,
        n322, n323, n324, n325, n326, n327, n328, n329, n330, n331, n332,
        n333, n334, n335, n336, n337, n338, n339, n340, n341, n342, n343,
        n344, n345, n346, n347, n348, n349, n350, n351, n352, n353, n354,
        n355, n356, n357, n358, n359, n360, n361, n362, n363, n364, n365,
        n366, n367, n368, n369, n370, n371, n372, n373, n374, n375, n376,
        n377, n378, n379, n380, n381, n382, n383, n384, n385, n386, n387,
        n388, n389, n390, n391, n392, n393, n394, n395, n396, n397, n398,
        n399, n400, n401, n402, n403, n404, n405, n406, n407, n408, n409,
        n410, n411, n412, n413, n414, n415, n416, n417, n418, n419, n420,
        n421, n422, n423, n424, n425, n426, n427, n428, n429, n430, n431,
        n432, n433, n434, n435, n436, n437, n438, n439, n440, n441, n442,
        n443, n444, n445, n446, n447, n448, n449, n450, n451, n452, n453,
        n454, n455, n456, n457, n458, n459, n460, n461, n462, n463, n464,
        n465, n466, n467, n468, n469, n470, n471, n472, n473, n474, n475,
        n476, n477, n478, n479, n480, n481, n482, n483, n484, n485, n486,
        n487, n488, n489, n490, n491, n492, n493, n494, n495, n496, n497,
        n498, n499, n500, n501, n502, n503, n504, n505, n506, n507, n508,
        n509, n510, n511, n512, n513, n514, n515, n516, n517, n518, n519,
        n520, n521, n522, n523, n524, n525, n526, n527, n528, n529, n530,
        n531, n532, n533, n534, n535, n536, n537, n538, n539, n540, n541,
        n542, n543, n544, n545, n546, n547, n548, n549, n550, n551, n552,
        n553, n554, n555, n556, n557, n558, n559, n560, n561, n562, n563,
        n564, n565, n566, n567, n568, n569, n570, n571, n572, n573, n574,
        n575, n576, n577, n578, n579, n580, n581, n582, n583, n584, n585,
        n586, n587, n588, n589, n590, n591, n592, n593, n594, n595, n596,
        n597, n598, n599, n600, n601, n602, n603, n604, n605, n606, n607,
        n608, n609, n610, n611, n612, n613, n614, n615, n616, n617, n618,
        n619, n620, n621, n622, n623, n624, n625, n626, n627, n628, n629,
        n630, n631, n632, n633, n634, n635, n636, n637, n638, n639, n640,
        n641, n642, n643, n644, n645, n646, n647, n648, n649, n650, n651,
        n652, n653, n654, n655, n656, n657, n658, n659, n660, n661, n662,
        n663, n664, n665, n666, n667, n668, n669, n670, n671, n672, n673,
        n674, n675, n676, n677, n678, n679, n680, n681, n682, n683, n684,
        n685, n686, n687, n688, n689, n690, n691, n692, n693, n694, n695,
        n696, n697, n698, n699, n700, n701, n702, n703, n704, n705, n706,
        n707, n708, n709, n710, n711, n712, n713, n714, n715, n716, n717,
        n718, n719, n720, n721, n722, n723, n724, n725, n726, n727, n728,
        n729, n730, n731, n732, n733, n734, n735, n736, n737, n738, n739,
```

```
            n740, n741, n742, n743, n744, n745, n746, n747, n748, n749, n750,
            n751, n752, n753, n754, n755, n756, n757, n758, n759, n760, n761,
            n762, n763, n764, n765, n766, n767, n768, n769, n770, n771, n772,
            n773, n774, n775, n776, n777, n778, n779, n780, n781, n782, n783,
            n784, n785, n786, n787, n788, n789, n790, n791, n792, n793, n794,
            n795, n796, n797, n798, n799, n800, n801, n802, n803, n804, n805,
            n806, n807, n808, n809, n810, n811, n812, n813, n814, n815, n816,
            n817, n818, n819, n820, n821, n822, n823, n824, n825, n826, n827,
            n828, n829, n830, n831, n832, n833, n834, n835, n836, n837, n838,
            n839, n840, n841, n842, n843, n844, n845, n846, n847, n848, n849,
            n850, n851, n852, n853, n854, n855, n856, n857, n858, n859, n860,
            n861, n862, n863, n864, n865, n866, n867, n868, n869, n870, n871,
            n872, n873, n874, n875, n876, n877, n878, n879, n880, n881, n882,
            n883, n884, n885, n886, n887, n888, n889, n890, n891, n892, n893,
            n894, n895, n896, n897, n898, n899, n900, n901, n902, n903, n904,
            n905, n906, n907, n908, n909, n910, n911, n912, n913, n914, n915,
            n916, n917, n918, n919, n920, n921, n922, n923, n924, n925, n926,
            n927, n928, n929, n930, n931, n932, n933, n934, n935, n936, n937,
            n938, n939, n940, n941, n942, n943, n944, n945, n946, n947, n948,
            n949, n950, n951, n952, n953, n954, n955, n956, n957, n958, n959,
            n960, n961, n962, n963, n964, n965, n966, n967, n968, n969, n970,
            n971, n972, n973, n974, n975, n976, n977, n978, n979, n980, n981,
            n982, n983, n984, n985, n986, n987, n988, n989, n990, n991, n992,
            n993, n994, n995, n996, n997, n998, n999, n1000, n1001, n1002, n1003,
            n1004, n1005, n1006, n1007, n1008, n1009, n1010, n1011, n1012, n1013,
            n1014, n1015, n1016, n1017, n1018, n1019, n1020, n1021, n1022, n1023,
            n1024, n1025, n1026, n1027, n1028, n1029, n1030, n1031, n1032, n1033,
            n1034, n1035, n1036, n1037, n1038, n1039, n1040, n1041, n1042, n1043,
            n1044, n1045, n1046, n1047, n1048, n1049, n1050, n1051, n1052, n1053,
            n1054, n1055, n1056, n1057, n1058, n1059, n1060, n1061, n1062, n1063,
            n1064, n1065, n1066, n1067, n1068, n1069, n1070, n1071, n1072, n1073,
            n1074, n1075, n1076, n1077, n1078, n1079, n1080, n1081, n1082, n1083,
            n1084, n1085, n1086, n1087, n1088, n1089, n1090, n1091, n1092, n1093,
            n1094, n1095, n1096, n1097, n1098, n1099, n1100, n1101, n1102, n1103,
            n1104, n1105, n1106, n1107, n1108, n1109, n1110, n1111, n1112, n1113,
            n1114, n1115, n1116, n1117, n1118, n1119, n1120, n1121, n1122, n1123,
            n1124, n1125, n1126, n1127, n1128, n1129, n1130, n1131, n1132, n1133,
            n1134, n1135, n1136, n1137, n1138, n1139, n1140, n1141, n1142, n1143,
            n1144, n1145, n1146, n1147, n1148, n1149, n1150, n1151, n1152, n1153,
            n1154, n1155, n1156, n1157, n1158, n1159, n1160, n1161, n1162, n1163,
            n1164, n1165, n1166, n1167, n1168, n1169, n1170, n1171, n1172, n1173,
            n1174, n1175, n1176, n1177, n1178, n1179, n1180, n1181, n1182, n1183,
            n1184, n1185, n1186, n1187, n1188, n1189, n1190, n1191, n1192, n1193,
            n1194, n1195, n1196, n1197, n1198, n1199, n1200, n1201, n1202, n1203,
            n1204, n1205, n1206, n1207, n1208, n1209, n1210, n1211, n1212, n1213,
            n1214, n1215, n1216, n1217, n1218, n1219, n1220, n1221, n1222, n1223,
            n1224, n1225, n1226, n1227, n1228, n1229, n1230, n1231, n1232, n1233,
            n1234, n1235, n1236, n1237, n1238, n1239, n1240, n1241, n1242, n1243,
            n1244, n1245, n1246, n1247, n1248, n1249, n1250, n1251, n1252, n1253,
            n1254, n1255, n1256, n1257, n1258, n1259, n1260, n1261, n1262, n1263,
            n1264, n1265, n1266, n1267, n1268, n1269, n1270, n1271, n1272, n1273,
            n1274, n1275, n1276, n1277, n1278, n1279, n1280, n1281, n1282, n1283,
            n1284, n1285, n1286, n1287, n1288, n1289, n1290, n1291, n1292, n1293,
            n1294, n1295, n1296, n1297, n1298, n1299, n1300, n1301, n1302, n1303,
```

```
        n1304, n1305, n1306, n1307, n1308, n1309, n1310, n1311, n1312, n1313,
        n1314, n1315, n1316, n1317, n1318, n1319, n1320, n1321, n1322, n1323,
        n1324, n1325, n1326, n1327, n1328, n1329, n1330, n1331, n1332, n1333,
        n1334, n1335, n1336, n1337, n1338, n1339, n1340, n1341, n1342, n1343,
        n1344, n1345, n1346, n1347, n1348, n1349, n1350, n1351, n1352, n1353,
        n1354, n1355, n1356, n1357, n1358, n1359, n1360, n1361, n1362, n1363,
        n1364, n1365, n1366, n1367, n1368, n1369, n1370, n1371, n1372, n1373,
        n1374, n1375, n1376, n1377, n1378, n1379, n1380, n1381, n1382, n1383,
        n1384, n1385, n1386, n1387, n1388, n1389, n1390, n1391, n1392, n1393,
        n1394, n1395, n1396, n1397, n1398, n1399, n1400, n1401, n1402, n1403,
        n1404, n1405, n1406, n1407, n1408, n1409, n1410, n1411, n1412, n1413,
        n1414, n1415, n1416, n1417, n1418, n1419, n1420, n1421, n1422, n1423,
        n1424, n1425, n1426, n1427, n1428, n1429, n1430, n1431, n1432, n1433,
        n1434, n1435, n1436, n1437, n1438, n1439, n1440, n1441, n1442, n1443,
        n1444, n1445, n1446, n1447, n1448, n1449, n1450, n1451, n1452, n1453,
        n1454, n1455, n1456, n1457, n1458, n1459, n1460, n1461, n1462, n1463,
        n1464, n1465, n1466, n1467, n1468, n1469, n1470, n1471, n1472, n1473,
        n1474, n1475, n1476, n1477, n1478, n1479, n1480, n1481, n1482, n1483,
        n1484, n1485, n1486, n1487, n1488, n1489, n1490, n1491, n1492, n1493,
        n1494, n1495, n1496, n1497, n1498, n1499, n1500, n1501, n1502, n1503,
        n1504, n1505, n1506, n1507, n1508, n1509, n1510, n1511, n1512, n1513,
        n1514, n1515, n1516, n1517, n1518, n1519, n1520, n1521, n1522, n1523,
        n1524, n1525, n1526, n1527, n1528, n1529, n1530, n1531, n1532, n1533,
        n1534, n1535, n1536, n1537, n1538, n1539, n1540, n1541, n1542, n1543,
        n1544, n1545, n1546, n1547, n1548, n1549, n1550, n1551, n1552, n1553,
        n1554, n1555, n1556, n1557, n1558, n1559, n1560, n1561, n1562, n1563,
        n1564, n1565, n1566, n1567, n1568, n1569, n1570, n1571, n1572, n1573,
        n1574, n1575, n1576, n1577, n1578, n1579, n1580, n1581, n1582, n1583,
        n1584, n1585, n1586, n1587, n1588, n1589, n1590, n1591, n1592, n1593,
        n1594, n1595, n1596, n1597, n1598, n1599, n1600, n1601, n1602, n1603,
        n1604, n1605, n1606, n1607, n1608, n1609, n1610, n1611, n1612, n1613,
        n1614, n1615, n1616, n1617, n1618, n1619, n1620, n1621, n1622, n1623,
        n1624, n1625, n1626, n1627, n1628, n1629, n1630, n1631, n1632, n1633,
        n1634, n1635, n1636, n1637, n1638, n1639, n1640, n1641, n1642, n1643,
        n1644, n1645, n1646, n1647, n1648, n1649, n1650, n1651, n1652, n1653,
        n1654, n1655, n1656, n1657, n1658, n1659, n1660, n1661, n1662, n1663,
        n1664, n1665, n1666, n1667, n1668, n1669, n1670, n1671, n1672, n1673,
        n1674, n1675, n1676, n1677, n1678, n1679, n1680, n1681, n1682, n1683,
        n1684, n1685, n1686, n1687, n1688, n1689, n1690, n1691, n1692, n1693,
        n1694, n1695, n1696, n1697, n1698, n1699, n1700, n1701, n1702, n1703,
        n1704, n1705, n1706, n1707, n1708, n1709, n1710, n1711, n1712, n1713,
        n1714, n1715, n1716, n1717, n1718, n1719, n1720, n1721, n1722, n1723,
        n1724, n1725, n1726, n1727, n1728, n1729, n1730, n1731, n1732, n1733,
        n1734, n1735, n1736, n1737, n1738, n1739, n1740, n1741, n1742, n1743,
        n1744, n1745, n1746, n1747, n1748, n1749, n1750, n1751, n1752, n1753,
        n1754, n1755, n1756, n1757, n1758, n1759, n1760, n1761, n1762, n1763,
        n1764, n1765, n1766, n1767, n1768, n1769, n1770, n1771, n1772, n1773,
        n1774, n1775, n1776, n1777, n1778, n1779, n1780, n1781, n1782, n1783,
        n1784, n1785, n1786, n1787, n1788, n1789, n1790, n1791, n1792, n1793,
        n1794, n1795, n1796, n1797, n1798, n1799, n1800, n1801, n1802, n1803,
        n1804, n1805, n1806, n1807, n1808, n1809, n1810, n1811, n1812, n1813,
        n1814, n1815, n1816, n1817, n1818, n1819, n1820, n1821, n1822, n1823,
        n1824, n1825, n1826, n1827, n1828, n1829, n1830, n1831, n1832, n1833,
        n1834, n1835, n1836, n1837, n1838, n1839, n1840, n1841, n1842, n1843,
```

```
        n1844, n1845, n1846, n1847, n1848, n1849, n1850, n1851, n1852, n1853,
        n1854, n1855, n1856, n1857, n1858, n1859, n1860, n1861, n1862, n1863,
        n1864, n1865, n1866, n1867, n1868, n1869, n1870, n1871, n1872, n1873,
        n1874, n1875, n1876, n1877, n1878, n1879, n1880, n1881, n1882, n1883,
        n1884, n1885, n1886, n1887, n1888, n1889, n1890, n1891, n1892, n1893,
        n1894, n1895, n1896, n1897, n1898, n1899, n1900, n1901, n1902, n1903,
        n1904, n1905, n1906, n1907, n1908, n1909, n1910, n1911, n1912, n1913,
        n1914, n1915, n1916, n1917, n1918, n1919, n1920, n1921, n1922, n1923,
        n1924, n1925, n1926, n1927, n1928, n1929, n1930, n1931, n1932, n1933,
        n1934, n1935, n1936, n1937, n1938, n1939, n1940, n1941, n1942, n1943,
        n1944, n1945, n1946, n1947, n1948, n1949, n1950, n1951, n1952, n1953,
        n1954, n1955, n1956, n1957, n1958, n1959, n1960, n1961, n1962, n1963,
        n1964, n1965, n1966, n1967, n1968, n1969, n1970, n1971, n1972, n1973,
        n1974, n1975, n1976, n1977, n1978, n1979, n1980, n1981, n1982, n1983,
        n1984, n1985, n1986, n1987, n1988, n1989, n1990, n1991, n1992, n1993,
        n1994, n1995, n1996, n1997, n1998, n1999, n2000, n2001, n2002, n2003,
        n2004, n2005, n2006, n2007, n2008, n2009, n2010, n2011, n2012, n2013,
        n2014, n2015, n2016, n2017, n2018, n2019, n2020, n2021, n2022, n2023,
        n2024, n2025, n2026, n2027, n2028, n2029, n2030, n2031, n2032, n2033,
        n2034, n2035, n2036, n2037, n2038, n2039, n2040, n2041, n2042, n2043,
        n2044, n2045, n2046, n2047, n2048, n2049, n2050, n2051, n2052, n2053,
        n2054, n2055, n2056, n2057, n2058, n2059, n2060, n2061, n2062, n2063,
        n2064, n2065, n2066, n2067, n2068, n2069, n2070, n2071, n2072, n2073,
        n2074, n2075, n2076, n2077, n2078, n2079, n2080, n2081, n2082, n2083,
        n2084, n2085, n2086, n2087, n2088, n2089, n2090, n2091, n2092, n2093,
        n2094, n2095, n2096, n2097, n2098, n2099, n2100, n2101, n2102, n2103,
        n2104, n2105, n2106, n2107, n2108, n2109, n2110, n2111, n2112, n2113,
        n2114, n2115, n2116, n2117, n2118, n2119, n2120, n2121, n2122, n2123,
        n2124, n2125, n2126, n2127, n2128, n2129, n2130, n2131, n2132, n2133,
        n2134, n2135, n2136, n2137, n2138, n2139, n2140, n2141, n2142, n2143,
        n2144, n2145, n2146, n2147, n2148, n2149, n2150, n2151, n2152, n2153,
        n2154, n2155, n2156, n2157, n2158, n2159, n2160, n2161, n2162, n2163,
        n2164, n2165, n2166, n2167, n2168, n2169, n2170, n2171, n2172, n2173,
        n2174, n2175, n2176, n2177, n2178, n2179, n2180, n2181, n2182, n2183,
        n2184, n2185, n2186, n2187, n2188, n2189, n2190, n2191, n2192, n2193,
        n2194, n2195, n2196, n2197, n2198, n2199, n2200, n2201, n2202, n2203,
        n2204, n2205, n2206, n2207, n2208, n2209, n2210, n2211, n2212, n2213,
        n2214, n2215, n2216, n2217, n2218, n2219, n2220, n2221, n2222, n2223,
        n2224, n2225, n2226, n2227, n2228, n2229, n2230, n2231, n2232, n2233,
        n2234, n2235, n2236, n2237, n2238, n2239, n2240, n2241, n2242, n2243,
        n2244, n2245, n2246, n2247, n2248, n2249, n2250, n2251, n2252, n2253,
        n2254, n2255, n2256, n2257, n2258, n2259, n2260, n2261, n2262, n2263,
        n2264, n2265, n2266, n2267, n2268, n2269, n2270, n2271, n2272, n2273,
        n2274, n2275, n2276, n2277, n2278, n2279, n2280, n2281, n2282, n2283,
        n2284, n2285, n2286, n2287, n2288, n2289, n2290, n2291, n2292, n2293,
        n2294, n2295, n2296, n2297, n2298, n2299, n2300, n2301, n2302, n2303,
        n2304, n2305, n2306, n2307, n2308, n2309, n2310, n2311, n2312, n2313,
        n2314, n2315, n2316, n2317, n2318, n2319, n2320, n2321, n2322, n2323,
        n2324, n2325, n2326, n2327, n2328, n2329, n2330, n2331, n2332, n2333,
        n2334, n2335, n2336, n2337, n2338, n2339, n2340, n2341, n2342, n2343,
        n2344, n2345, n2346, n2347, n2348, n2349, n2350, n2351, n2352, n2353,
        n2354, n2355, n2356, n2357, n2358, n2359, n2360, n2361, n2362, n2363,
        n2364, n2365, n2366, n2367, n2368, n2369, n2370, n2371, n2372, n2373,
        n2374, n2375, n2376, n2377, n2378, n2379, n2380, n2381, n2382, n2383,
```

```
        n2384, n2385, n2386, n2387, n2388, n2389, n2390, n2391, n2392, n2393,
        n2394, n2395, n2396, n2397, n2398, n2399, n2400, n2401, n2402, n2403,
        n2404, n2405, n2406, n2407, n2408, n2409, n2410, n2411, n2412, n2413,
        n2414, n2415, n2416, n2417, n2418, n2419, n2420, n2421, n2422, n2423,
        n2424, n2425, n2426, n2427, n2428, n2429, n2430, n2431, n2432, n2433,
        n2434, n2435, n2436, n2437, n2438, n2439, n2440, n2441, n2442, n2443,
        n2444, n2445, n2446, n2447, n2448, n2449, n2450, n2451, n2452, n2453,
        n2454, n2455, n2456, n2457, n2458, n2459, n2460, n2461, n2462, n2463,
        n2464, n2465, n2466, n2467, n2468, n2469, n2470, n2471, n2472, n2473,
        n2474, n2475, n2476, n2477, n2478, n2479, n2480, n2481, n2482, n2483,
        n2484, n2485, n2486, n2487, n2488, n2489, n2490, n2491, n2492, n2493,
        n2494, n2495, n2496, n2497, n2498, n2499, n2500, n2501, n2502, n2503,
        n2504, n2505, n2506, n2507, n2508, n2509, n2510, n2511, n2512, n2513,
        n2514, n2515, n2516, n2517, n2518, n2519, n2520, n2521, n2522, n2523,
        n2524, n2525, n2526, n2527, n2528, n2529, n2530, n2531, n2532, n2533,
        n2534, n2535, n2536, n2537, n2538, n2539, n2540, n2541, n2542, n2543,
        n2544, n2545, n2546, n2547, n2548, n2549, n2550, n2551, n2552, n2553,
        n2554, n2555, n2556, n2557, n2558, n2559, n2560, n2561, n2562, n2563,
        n2564, n2565, n2566, n2567, n2568, n2569, n2570, n2571, n2572, n2573,
        n2574, n2575, n2576, n2577, n2578, n2579, n2580, n2581, n2582, n2583,
        n2584, n2585, n2586, n2587, n2588, n2589, n2590, n2591, n2592, n2593,
        n2594, n2595, n2596, n2597, n2598, n2599, n2600, n2601, n2602, n2603,
        n2604, n2605, n2606, n2607, n2608, n2609, n2610, n2611, n2612, n2613,
        n2614, n2615, n2616, n2617, n2618, n2619, n2620, n2621, n2622, n2623,
        n2624, n2625, n2626, n2627, n2628, n2629, n2630, n2631, n2632, n2633,
        n2634, n2635, n2636, n2637, n2638, n2639, n2640, n2641, n2642, n2643,
        n2644, n2645, n2646, n2647, n2648, n2649, n2650, n2651, n2652, n2653,
        n2654, n2655, n2656, n2657, n2658, n2659, n2660, n2661, n2662, n2663,
        n2664, n2665, n2666, n2667, n2668, n2669, n2670, n2671, n2672, n2673,
        n2674, n2675, n2676, n2677, n2678, n2679, n2680, n2681, n2682, n2683,
        n2684, n2685, n2686, n2687, n2688, n2689, n2690, n2691, n2692, n2693,
        n2694, n2695, n2696, n2697, n2698, n2699, n2700, n2701, n2702, n2703,
        n2704, n2705, n2706, n2707, n2708, n2709, n2710, n2711, n2712, n2713,
        n2714, n2715, n2716, n2717, n2718, n2719, n2720, n2721, n2722, n2723,
        n2724, n2725, n2726, n2727, n2728, n2729, n2730, n2731, n2732, n2733,
        n2734, n2735, n2736, n2737, n2738, n2739, n2740, n2741, n2742, n2743,
        n2744, n2745, n2746, n2747, n2748, n2749, n2750, n2751, n2752, n2753,
        n2754, n2755, n2756, n2757, n2758, n2759, n2760, n2761, n2762, n2763,
        n2764, n2765, n2766, n2767, n2768, n2769, n2770, n2771, n2772, n2773,
        n2774, n2775, n2776, n2777, n2778, n2779, n2780, n2781, n2782, n2783,
        n2784, n2785, n2786, n2787, n2788, n2789, n2790, n2791, n2792, n2793,
        n2794, n2795, n2796, n2797, n2798, n2799, n2800, n2801, n2802, n2803,
        n2804, n2805, n2806, n2807, n2808, n2809, n2810, n2811, n2812, n2813,
        n2814, n2815, n2816, n2817, n2818, n2819, n2820, n2821, n2822, n2823,
        n2824;
    wire [11:0] contador;

    DFQD1BWP7T \q_reg[6] ( .D(n3525), .CP(clk), .Q(q_out[6]) );
    DFQD1BWP7T \q_reg[5] ( .D(n3524), .CP(clk), .Q(q_out[5]) );
    DFQD1BWP7T \q_reg[4] ( .D(n3523), .CP(clk), .Q(q_out[4]) );
    DFQD1BWP7T \q_reg[3] ( .D(n3522), .CP(clk), .Q(q_out[3]) );
    DFQD1BWP7T \q_reg[2] ( .D(n3521), .CP(clk), .Q(q_out[2]) );
    DFQD1BWP7T \q_reg[1] ( .D(n3520), .CP(clk), .Q(q_out[1]) );
    DFQD1BWP7T \q_reg[0] ( .D(n3519), .CP(clk), .Q(q_out[0]) );
```

```
    AND_2 U1 ( .in1(EN), .in2(EN), .out(W_1) );
    INV_18 U2 ( .A(W_1), .B(W_2) );
    INV_17 U3 ( .A(W_2), .B(W_3) );
    INV_16 U4 ( .A(W_3), .B(W_4) );
    INV_15 U5 ( .A(W_4), .B(W_5) );
    INV_14 U6 ( .A(W_5), .B(W_6) );
    INV_13 U7 ( .A(W_6), .B(W_7) );
    INV_12 U8 ( .A(W_7), .B(W_8) );
    INV_11 U9 ( .A(W_8), .B(W_9) );
    INV_10 U10 ( .A(W_9), .B(W_10) );
    INV_9 U11 ( .A(W_10), .B(W_11) );
    INV_8 U12 ( .A(W_11), .B(W_12) );
    INV_7 U13 ( .A(W_12), .B(W_13) );
    INV_6 U14 ( .A(W_13), .B(W_14) );
    INV_5 U15 ( .A(W_14), .B(W_15) );
    INV_4 U16 ( .A(W_15), .B(W_16) );
    INV_3 U17 ( .A(W_16), .B(W_17) );
    INV_2 U18 ( .A(W_17), .B(W_18) );
    INV_1 U19 ( .A(W_18), .B(W_19) );
    INV_0 U20 ( .A(W_19), .B(clk_s) );
    chip_SP_DW01_inc_0 r79 ( .A(contador), .SUM({N4361, N4360, N4359, N4358,
         N4357, N4356, N4355, N4354, N4353, N4352, N4351, N4350}) );
    DFCND0BWP7T \contador_reg[7] ( .D(N4408), .CP(clk), .CDN(n310), .Q(
         contador[7]), .QN(n6347) );
    DFCND0BWP7T \contador_reg[6] ( .D(N4407), .CP(clk), .CDN(n310), .Q(
         contador[6]), .QN(n6418) );
    DFCND0BWP7T \contador_reg[5] ( .D(N4406), .CP(clk), .CDN(n310), .Q(
         contador[5]), .QN(n6416) );
    DFCND0BWP7T \contador_reg[4] ( .D(N4405), .CP(clk), .CDN(n310), .Q(
         contador[4]), .QN(n6420) );
    DFCND0BWP7T \contador_reg[3] ( .D(N4404), .CP(clk), .CDN(n310), .Q(
         contador[3]), .QN(n6419) );
    DFCND0BWP7T \contador_reg[2] ( .D(N4403), .CP(clk), .CDN(n310), .Q(
         contador[2]), .QN(n3528) );
    DFCND0BWP7T \contador_reg[1] ( .D(N4402), .CP(clk), .CDN(n310), .Q(
         contador[1]), .QN(n6421) );
    DFCND0BWP7T \contador_reg[0] ( .D(N4401), .CP(clk), .CDN(n310), .Q(
         contador[0]), .QN(n6329) );
    DFCND0BWP7T \contador_reg[8] ( .D(N4409), .CP(clk), .CDN(n310), .Q(
         contador[8]), .QN(n3527) );
    DFCND0BWP7T \contador_reg[9] ( .D(N4410), .CP(clk), .CDN(n310), .Q(
         contador[9]), .QN(n6417) );
    DFCND0BWP7T \contador_reg[10] ( .D(N4411), .CP(clk), .CDN(n310), .Q(
         contador[10]), .QN(n6338) );
    DFCND0BWP7T \contador_reg[11] ( .D(N4412), .CP(clk), .CDN(n310), .Q(
         contador[11]), .QN(n2912) );
    NR3D1BWP7T U21 ( .A1(n2813), .A2(n6420), .A3(contador[3]), .ZN(n482) );
    ND2D1BWP7T U22 ( .A1(n6421), .A2(n3528), .ZN(n2813) );
    NR3D1BWP7T U23 ( .A1(n3528), .A2(n6421), .A3(n2806), .ZN(n438) );
    NR3D1BWP7T U24 ( .A1(n6419), .A2(n6420), .A3(n2813), .ZN(n293) );
    NR3D1BWP7T U25 ( .A1(contador[2]), .A2(n6421), .A3(n2806), .ZN(n460) );
    NR3D1BWP7T U26 ( .A1(contador[1]), .A2(n3528), .A3(n2806), .ZN(n266) );
    NR3D1BWP7T U27 ( .A1(n2813), .A2(n6419), .A3(contador[4]), .ZN(n296) );
    NR4D1BWP7T U28 ( .A1(contador[3]), .A2(contador[1]), .A3(n3528), .A4(n6420),
```

```
      .ZN(n294) );
    NR4D1BWP7T U29 ( .A1(contador[4]), .A2(contador[2]), .A3(n6419), .A4(n6421),
      .ZN(n268) );
    NR2D1BWP7T U30 ( .A1(n2806), .A2(n2813), .ZN(n285) );
    TIELBWP7T U31 ( .ZN(q_out[7]) );
    AO21D0BWP7T U32 ( .A1(q_out[6]), .A2(n1), .B(n2), .Z(n3525) );
    MUX2ND0BWP7T U33 ( .I0(n3), .I1(n4), .S(n5), .ZN(n2) );
    NR4D0BWP7T U34 ( .A1(n6), .A2(n7), .A3(n8), .A4(n9), .ZN(n4) );
    IIND4D0BWP7T U35 ( .A1(n10), .A2(n11), .B1(n12), .B2(n13), .ZN(n9) );
    NR3D0BWP7T U36 ( .A1(n14), .A2(n15), .A3(n16), .ZN(n13) );
    IIND4D0BWP7T U37 ( .A1(n17), .A2(n18), .B1(n19), .B2(n20), .ZN(n8) );
    NR4D0BWP7T U38 ( .A1(n21), .A2(n22), .A3(n23), .A4(n24), .ZN(n20) );
    IND4D0BWP7T U39 ( .A1(n25), .B1(n26), .B2(n27), .B3(n28), .ZN(n7) );
    AOI222D0BWP7T U40 ( .A1(n29), .A2(n30), .B1(n31), .B2(n32), .C1(n33), .C2(
      n34), .ZN(n28) );
    OAI211D0BWP7T U41 ( .A1(n35), .A2(n36), .B(n37), .C(n38), .ZN(n6) );
    NR4D0BWP7T U42 ( .A1(n39), .A2(n40), .A3(n41), .A4(n42), .ZN(n38) );
    AOI21D0BWP7T U43 ( .A1(n43), .A2(n44), .B(n45), .ZN(n37) );
    AOI31D0BWP7T U44 ( .A1(n46), .A2(n47), .A3(n48), .B(n49), .ZN(n45) );
    NR2D0BWP7T U45 ( .A1(n50), .A2(n51), .ZN(n3) );
    AO21D0BWP7T U46 ( .A1(q_out[5]), .A2(n1), .B(n52), .Z(n3524) );
    MUX2ND0BWP7T U47 ( .I0(n53), .I1(n54), .S(n5), .ZN(n52) );
    NR4D0BWP7T U48 ( .A1(n55), .A2(n56), .A3(n57), .A4(n58), .ZN(n54) );
    ND4D0BWP7T U49 ( .A1(n59), .A2(n60), .A3(n61), .A4(n62), .ZN(n58) );
    NR4D0BWP7T U50 ( .A1(n63), .A2(n64), .A3(n65), .A4(n66), .ZN(n62) );
    NR3D0BWP7T U51 ( .A1(n67), .A2(n68), .A3(n69), .ZN(n61) );
    IND4D0BWP7T U52 ( .A1(n70), .B1(n71), .B2(n72), .B3(n73), .ZN(n57) );
    INR4D0BWP7T U53 ( .A1(n74), .B1(n75), .B2(n76), .B3(n77), .ZN(n73) );
    NR3D0BWP7T U54 ( .A1(n78), .A2(n79), .A3(n80), .ZN(n72) );
    ND3D0BWP7T U55 ( .A1(n81), .A2(n82), .A3(n83), .ZN(n56) );
    NR4D0BWP7T U56 ( .A1(n84), .A2(n85), .A3(n86), .A4(n87), .ZN(n83) );
    ND3D0BWP7T U57 ( .A1(n88), .A2(n89), .A3(n90), .ZN(n85) );
    ND4D0BWP7T U58 ( .A1(n91), .A2(n92), .A3(n93), .A4(n94), .ZN(n84) );
    NR4D0BWP7T U59 ( .A1(n95), .A2(n96), .A3(n97), .A4(n98), .ZN(n82) );
    NR4D0BWP7T U60 ( .A1(n23), .A2(n99), .A3(n100), .A4(n101), .ZN(n81) );
    ND4D0BWP7T U61 ( .A1(n102), .A2(n103), .A3(n104), .A4(n105), .ZN(n23) );
    NR4D0BWP7T U62 ( .A1(n106), .A2(n107), .A3(n108), .A4(n109), .ZN(n105) );
    IND4D0BWP7T U63 ( .A1(n110), .B1(n111), .B2(n112), .B3(n113), .ZN(n109) );
    NR3D0BWP7T U64 ( .A1(n114), .A2(n115), .A3(n116), .ZN(n113) );
    IIND4D0BWP7T U65 ( .A1(n117), .A2(n118), .B1(n119), .B2(n120), .ZN(n108) );
    NR3D0BWP7T U66 ( .A1(n121), .A2(n122), .A3(n123), .ZN(n120) );
    IND4D0BWP7T U67 ( .A1(n124), .B1(n125), .B2(n126), .B3(n127), .ZN(n107) );
    NR3D0BWP7T U68 ( .A1(n128), .A2(n129), .A3(n130), .ZN(n127) );
    IIND4D0BWP7T U69 ( .A1(n131), .A2(n132), .B1(n133), .B2(n134), .ZN(n106) );
    NR4D0BWP7T U70 ( .A1(n135), .A2(n136), .A3(n137), .A4(n138), .ZN(n134) );
    NR4D0BWP7T U71 ( .A1(n139), .A2(n140), .A3(n141), .A4(n142), .ZN(n104) );
    OAI211D0BWP7T U72 ( .A1(n143), .A2(n47), .B(n144), .C(n145), .ZN(n142) );
    OAI222D0BWP7T U73 ( .A1(n146), .A2(n147), .B1(n148), .B2(n149), .C1(n150),
      .C2(n151), .ZN(n141) );
    OAI222D0BWP7T U74 ( .A1(n152), .A2(n153), .B1(n154), .B2(n155), .C1(n156),
      .C2(n157), .ZN(n140) );
    IND4D0BWP7T U75 ( .A1(n158), .B1(n159), .B2(n160), .B3(n161), .ZN(n139) );
    NR4D0BWP7T U76 ( .A1(n162), .A2(n163), .A3(n164), .A4(n165), .ZN(n103) );
    ND3D0BWP7T U77 ( .A1(n166), .A2(n167), .A3(n168), .ZN(n162) );
```

```
    INR4D0BWP7T U78 ( .A1(n169), .B1(n170), .B2(n171), .B3(n172), .ZN(n102) );
    OR3D0BWP7T U79 ( .A1(n173), .A2(n174), .A3(n175), .Z(n170) );
    ND4D0BWP7T U80 ( .A1(n176), .A2(n177), .A3(n178), .A4(n179), .ZN(n55) );
    NR4D0BWP7T U81 ( .A1(n180), .A2(n181), .A3(n182), .A4(n183), .ZN(n179) );
    IND3D0BWP7T U82 ( .A1(n184), .B1(n185), .B2(n186), .ZN(n181) );
    ND4D0BWP7T U83 ( .A1(n46), .A2(n187), .A3(n188), .A4(n189), .ZN(n180) );
    NR4D0BWP7T U84 ( .A1(n190), .A2(n191), .A3(n192), .A4(n193), .ZN(n178) );
    AOI21D0BWP7T U85 ( .A1(n194), .A2(n195), .B(n196), .ZN(n193) );
    AOI21D0BWP7T U86 ( .A1(n197), .A2(n198), .B(n199), .ZN(n192) );
    OAI211D0BWP7T U87 ( .A1(n200), .A2(n201), .B(n202), .C(n203), .ZN(n191) );
    NR2D0BWP7T U88 ( .A1(n204), .A2(n205), .ZN(n200) );
    OAI211D0BWP7T U89 ( .A1(n206), .A2(n207), .B(n208), .C(n209), .ZN(n190) );
    AOI22D0BWP7T U90 ( .A1(n210), .A2(n211), .B1(n212), .B2(n213), .ZN(n209) );
    OAI21D0BWP7T U91 ( .A1(n214), .A2(n215), .B(n216), .ZN(n208) );
    NR4D0BWP7T U92 ( .A1(n217), .A2(n218), .A3(n219), .A4(n220), .ZN(n177) );
    OAI22D0BWP7T U93 ( .A1(n221), .A2(n222), .B1(n223), .B2(n224), .ZN(n220) );
    OAI22D0BWP7T U94 ( .A1(n225), .A2(n226), .B1(n227), .B2(n228), .ZN(n219) );
    OAI22D0BWP7T U95 ( .A1(n229), .A2(n230), .B1(n231), .B2(n232), .ZN(n218) );
    OAI222D0BWP7T U96 ( .A1(n233), .A2(n234), .B1(n235), .B2(n236), .C1(n146),
         .C2(n237), .ZN(n217) );
    INR4D0BWP7T U97 ( .A1(n238), .B1(n239), .B2(n240), .B3(n241), .ZN(n176) );
    AO21D0BWP7T U98 ( .A1(n30), .A2(n242), .B(n243), .Z(n240) );
    IIND4D0BWP7T U99 ( .A1(n244), .A2(n245), .B1(n246), .B2(n247), .ZN(n239) );
    AO21D0BWP7T U100 ( .A1(q_out[4]), .A2(n1), .B(n248), .Z(n3523) );
    MUX2ND0BWP7T U101 ( .I0(n249), .I1(n250), .S(n5), .ZN(n248) );
    NR4D0BWP7T U102 ( .A1(n251), .A2(n252), .A3(n253), .A4(n254), .ZN(n250) );
    ND4D0BWP7T U103 ( .A1(n255), .A2(n256), .A3(n257), .A4(n258), .ZN(n254) );
    AOI22D0BWP7T U104 ( .A1(n259), .A2(n260), .B1(n261), .B2(n262), .ZN(n258) );
    OAI31D0BWP7T U105 ( .A1(n263), .A2(n264), .A3(n265), .B(n266), .ZN(n257) );
    OAI21D0BWP7T U106 ( .A1(n265), .A2(n267), .B(n268), .ZN(n256) );
    OAI21D0BWP7T U107 ( .A1(n210), .A2(n43), .B(n269), .ZN(n255) );
    IND4D0BWP7T U108 ( .A1(n270), .B1(n271), .B2(n272), .B3(n273), .ZN(n253) );
    NR3D0BWP7T U109 ( .A1(n274), .A2(n40), .A3(n275), .ZN(n273) );
    ND4D0BWP7T U110 ( .A1(n276), .A2(n277), .A3(n278), .A4(n279), .ZN(n252) );
    AOI222D0BWP7T U111 ( .A1(n280), .A2(n281), .B1(n282), .B2(n283), .C1(n284),
         .C2(n285), .ZN(n279) );
    AOI222D0BWP7T U112 ( .A1(n286), .A2(n287), .B1(n288), .B2(n289), .C1(n290),
         .C2(n291), .ZN(n278) );
    AOI222D0BWP7T U113 ( .A1(n292), .A2(n293), .B1(n264), .B2(n294), .C1(n295),
         .C2(n296), .ZN(n277) );
    AOI22D0BWP7T U114 ( .A1(n297), .A2(n298), .B1(n204), .B2(n299), .ZN(n276) );
    ND4D0BWP7T U115 ( .A1(n300), .A2(n301), .A3(n302), .A4(n303), .ZN(n251) );
    NR4D0BWP7T U116 ( .A1(n304), .A2(n305), .A3(n306), .A4(n307), .ZN(n303) );
    IND3D0BWP7T U117 ( .A1(n171), .B1(n308), .B2(n133), .ZN(n305) );
    OA221D0BWP7T U118 ( .A1(n309), .A2(n311), .B1(n312), .B2(n313), .C(n314),
         .Z(n133) );
    AOI22D0BWP7T U119 ( .A1(n315), .A2(n30), .B1(n316), .B2(n317), .ZN(n314) );
    CKND0BWP7T U120 ( .I(n16), .ZN(n308) );
    OAI221D0BWP7T U121 ( .A1(n318), .A2(n319), .B1(n221), .B2(n320), .C(n321),
         .ZN(n16) );
    AOI222D0BWP7T U122 ( .A1(n242), .A2(n322), .B1(n34), .B2(n323), .C1(n324),
         .C2(n30), .ZN(n321) );
    CKND2D0BWP7T U123 ( .A1(n325), .A2(n326), .ZN(n323) );
    OAI222D0BWP7T U124 ( .A1(n327), .A2(n328), .B1(n329), .B2(n330), .C1(n196),
```

```
      .C2(n236), .ZN(n171) );
    ND4D0BWP7T U125 ( .A1(n331), .A2(n332), .A3(n169), .A4(n333), .ZN(n304) );
    INR3D0BWP7T U126 ( .A1(n334), .B1(n335), .B2(n336), .ZN(n333) );
    AOI222D0BWP7T U127 ( .A1(n337), .A2(n338), .B1(n339), .B2(n205), .C1(n299),
      .C2(n340), .ZN(n169) );
    NR4D0BWP7T U128 ( .A1(n341), .A2(n342), .A3(n343), .A4(n344), .ZN(n302) );
    OAI22D0BWP7T U129 ( .A1(n345), .A2(n346), .B1(n347), .B2(n223), .ZN(n342) );
    AOI222D0BWP7T U130 ( .A1(n348), .A2(n349), .B1(n350), .B2(n351), .C1(n352),
      .C2(n353), .ZN(n301) );
    AOI222D0BWP7T U131 ( .A1(n354), .A2(n355), .B1(n356), .B2(n357), .C1(n338),
      .C2(n358), .ZN(n300) );
    NR4D0BWP7T U132 ( .A1(n359), .A2(n360), .A3(n361), .A4(n362), .ZN(n249) );
    IIND4D0BWP7T U133 ( .A1(n363), .A2(n100), .B1(n364), .B2(n365), .ZN(n362) );
    NR2D0BWP7T U134 ( .A1(n366), .A2(n367), .ZN(n365) );
    ND3D0BWP7T U135 ( .A1(n368), .A2(n369), .A3(n370), .ZN(n100) );
    AOI222D0BWP7T U136 ( .A1(n338), .A2(n371), .B1(n372), .B2(n373), .C1(n374),
      .C2(n340), .ZN(n370) );
    IND4D0BWP7T U137 ( .A1(n375), .B1(n376), .B2(n377), .B3(n378), .ZN(n361) );
    INR3D0BWP7T U138 ( .A1(n379), .B1(n380), .B2(n381), .ZN(n378) );
    IND4D0BWP7T U139 ( .A1(n382), .B1(n383), .B2(n384), .B3(n385), .ZN(n360) );
    NR4D0BWP7T U140 ( .A1(n386), .A2(n387), .A3(n388), .A4(n389), .ZN(n385) );
    ND3D0BWP7T U141 ( .A1(n390), .A2(n391), .A3(n392), .ZN(n386) );
    INR3D0BWP7T U142 ( .A1(n393), .B1(n394), .B2(n395), .ZN(n384) );
    ND4D0BWP7T U143 ( .A1(n396), .A2(n397), .A3(n398), .A4(n399), .ZN(n359) );
    NR4D0BWP7T U144 ( .A1(n400), .A2(n401), .A3(n402), .A4(n403), .ZN(n399) );
    OAI22D0BWP7T U145 ( .A1(n404), .A2(n222), .B1(n405), .B2(n346), .ZN(n403) );
    OAI222D0BWP7T U146 ( .A1(n223), .A2(n406), .B1(n407), .B2(n309), .C1(n408),
      .C2(n409), .ZN(n402) );
    OAI222D0BWP7T U147 ( .A1(n146), .A2(n410), .B1(n411), .B2(n412), .C1(n413),
      .C2(n414), .ZN(n401) );
    OAI222D0BWP7T U148 ( .A1(n415), .A2(n345), .B1(n416), .B2(n417), .C1(n418),
      .C2(n419), .ZN(n400) );
    IINR4D0BWP7T U149 ( .A1(n420), .A2(n421), .B1(n422), .B2(n423), .ZN(n398) );
    ND3D0BWP7T U150 ( .A1(n424), .A2(n425), .A3(n426), .ZN(n422) );
    AOI211D0BWP7T U151 ( .A1(n427), .A2(n428), .B(n429), .C(n430), .ZN(n397) );
    AOI21D0BWP7T U152 ( .A1(n47), .A2(n431), .B(n432), .ZN(n430) );
    AOI21D0BWP7T U153 ( .A1(n433), .A2(n434), .B(n435), .ZN(n429) );
    ND4D0BWP7T U154 ( .A1(n417), .A2(n436), .A3(n194), .A4(n437), .ZN(n428) );
    MAOI22D0BWP7T U155 ( .A1(n438), .A2(n439), .B1(n440), .B2(n441), .ZN(n396)
      );
    AO21D0BWP7T U156 ( .A1(q_out[3]), .A2(n1), .B(n442), .Z(n3522) );
    MUX2ND0BWP7T U157 ( .I0(n443), .I1(n444), .S(n5), .ZN(n442) );
    NR4D0BWP7T U158 ( .A1(n445), .A2(n446), .A3(n447), .A4(n448), .ZN(n444) );
    ND4D0BWP7T U159 ( .A1(n449), .A2(n450), .A3(n451), .A4(n452), .ZN(n448) );
    AOI222D0BWP7T U160 ( .A1(n453), .A2(n34), .B1(n265), .B2(n454), .C1(n374),
      .C2(n455), .ZN(n452) );
    MAOI22D0BWP7T U161 ( .A1(n266), .A2(n456), .B1(n227), .B2(n457), .ZN(n451)
      );
    CKND2D0BWP7T U162 ( .A1(n228), .A2(n458), .ZN(n456) );
    OAI21D0BWP7T U163 ( .A1(n459), .A2(n210), .B(n460), .ZN(n449) );
    IND4D0BWP7T U164 ( .A1(n461), .B1(n462), .B2(n463), .B3(n464), .ZN(n447) );
    NR4D0BWP7T U165 ( .A1(n465), .A2(n466), .A3(n467), .A4(n468), .ZN(n464) );
    ND4D0BWP7T U166 ( .A1(n469), .A2(n470), .A3(n471), .A4(n472), .ZN(n446) );
    AOI221D0BWP7T U167 ( .A1(n264), .A2(n293), .B1(n286), .B2(n269), .C(n473),
```

```
      .ZN(n472) );
    OAI22D0BWP7T U168 ( .A1(n435), .A2(n474), .B1(n475), .B2(n476), .ZN(n473) );
    AOI222D0BWP7T U169 ( .A1(n477), .A2(n297), .B1(n340), .B2(n478), .C1(n479),
      .C2(n480), .ZN(n471) );
    AOI222D0BWP7T U170 ( .A1(n481), .A2(n482), .B1(n337), .B2(n205), .C1(n183),
      .C2(n483), .ZN(n470) );
    AOI211D0BWP7T U171 ( .A1(n484), .A2(n485), .B(n486), .C(n487), .ZN(n469) );
    ND4D0BWP7T U172 ( .A1(n488), .A2(n489), .A3(n490), .A4(n491), .ZN(n445) );
    INR4D0BWP7T U173 ( .A1(n492), .B1(n493), .B2(n494), .B3(n495), .ZN(n491) );
    ND4D0BWP7T U174 ( .A1(n496), .A2(n497), .A3(n498), .A4(n499), .ZN(n493) );
    NR4D0BWP7T U175 ( .A1(n500), .A2(n501), .A3(n502), .A4(n503), .ZN(n490) );
    IND3D0BWP7T U176 ( .A1(n504), .B1(n505), .B2(n506), .ZN(n500) );
    NR4D0BWP7T U177 ( .A1(n507), .A2(n135), .A3(n508), .A4(n172), .ZN(n489) );
    OAI221D0BWP7T U178 ( .A1(n509), .A2(n312), .B1(n346), .B2(n510), .C(n511),
      .ZN(n172) );
    OAI221D0BWP7T U179 ( .A1(n227), .A2(n194), .B1(n512), .B2(n236), .C(n513),
      .ZN(n135) );
    AOI222D0BWP7T U180 ( .A1(n514), .A2(n515), .B1(n516), .B2(n517), .C1(n285),
      .C2(n518), .ZN(n513) );
    IND4D0BWP7T U181 ( .A1(n519), .B1(n520), .B2(n521), .B3(n522), .ZN(n507) );
    NR4D0BWP7T U182 ( .A1(n523), .A2(n524), .A3(n14), .A4(n525), .ZN(n488) );
    OAI211D0BWP7T U183 ( .A1(n526), .A2(n475), .B(n527), .C(n528), .ZN(n14) );
    AOI222D0BWP7T U184 ( .A1(n529), .A2(n351), .B1(n530), .B2(n204), .C1(n350),
      .C2(n357), .ZN(n528) );
    AOI22D0BWP7T U185 ( .A1(n531), .A2(n517), .B1(n532), .B2(n515), .ZN(n527) );
    IND3D0BWP7T U186 ( .A1(n533), .B1(n534), .B2(n535), .ZN(n523) );
    CKND0BWP7T U187 ( .I(n536), .ZN(n534) );
    NR4D0BWP7T U188 ( .A1(n537), .A2(n538), .A3(n539), .A4(n540), .ZN(n443) );
    ND4D0BWP7T U189 ( .A1(n541), .A2(n542), .A3(n543), .A4(n544), .ZN(n540) );
    NR3D0BWP7T U190 ( .A1(n545), .A2(n546), .A3(n547), .ZN(n544) );
    ND3D0BWP7T U191 ( .A1(n548), .A2(n549), .A3(n550), .ZN(n545) );
    NR3D0BWP7T U192 ( .A1(n551), .A2(n552), .A3(n553), .ZN(n543) );
    IND4D0BWP7T U193 ( .A1(n554), .B1(n555), .B2(n556), .B3(n557), .ZN(n539) );
    NR4D0BWP7T U194 ( .A1(n558), .A2(n559), .A3(n560), .A4(n561), .ZN(n557) );
    IND3D0BWP7T U195 ( .A1(n562), .B1(n563), .B2(n564), .ZN(n558) );
    CKND0BWP7T U196 ( .I(n565), .ZN(n563) );
    AN3D0BWP7T U197 ( .A1(n492), .A2(n566), .A3(n567), .Z(n556) );
    ND4D0BWP7T U198 ( .A1(n568), .A2(n569), .A3(n570), .A4(n571), .ZN(n538) );
    AOI221D0BWP7T U199 ( .A1(n572), .A2(n573), .B1(n574), .B2(n575), .C(n576),
      .ZN(n571) );
    AOI31D0BWP7T U200 ( .A1(n577), .A2(n578), .A3(n579), .B(n435), .ZN(n576) );
    CKND0BWP7T U201 ( .I(n416), .ZN(n575) );
    AOI22D0BWP7T U202 ( .A1(n580), .A2(n581), .B1(n582), .B2(n583), .ZN(n570) );
    AOI222D0BWP7T U203 ( .A1(n584), .A2(n355), .B1(n585), .B2(n285), .C1(n295),
      .C2(n259), .ZN(n569) );
    AOI22D0BWP7T U204 ( .A1(n586), .A2(n530), .B1(n587), .B2(n588), .ZN(n568) );
    ND4D0BWP7T U205 ( .A1(n589), .A2(n590), .A3(n591), .A4(n592), .ZN(n537) );
    IINR4D0BWP7T U206 ( .A1(n425), .A2(n593), .B1(n594), .B2(n595), .ZN(n592) );
    OAI211D0BWP7T U207 ( .A1(n596), .A2(n35), .B(n597), .C(n598), .ZN(n594) );
    OAI21D0BWP7T U208 ( .A1(n599), .A2(n600), .B(n601), .ZN(n598) );
    OAI21D0BWP7T U209 ( .A1(n602), .A2(n603), .B(n44), .ZN(n597) );
    NR4D0BWP7T U210 ( .A1(n604), .A2(n605), .A3(n606), .A4(n607), .ZN(n596) );
    AOI211D0BWP7T U211 ( .A1(n608), .A2(n609), .B(n610), .C(n611), .ZN(n591) );
    AOI21D0BWP7T U212 ( .A1(n36), .A2(n431), .B(n512), .ZN(n611) );
```

```
    AOI21D0BWP7T U213 ( .A1(n612), .A2(n613), .B(n614), .ZN(n610) );
    CKND2D0BWP7T U214 ( .A1(n320), .A2(n311), .ZN(n609) );
    OAI21D0BWP7T U215 ( .A1(n580), .A2(n615), .B(n269), .ZN(n590) );
    AO31D0BWP7T U216 ( .A1(n616), .A2(n617), .A3(n187), .B(n618), .Z(n589) );
    AO21D0BWP7T U217 ( .A1(q_out[2]), .A2(n1), .B(n619), .Z(n3521) );
    MUX2ND0BWP7T U218 ( .I0(n620), .I1(n621), .S(n5), .ZN(n619) );
    NR4D0BWP7T U219 ( .A1(n622), .A2(n623), .A3(n624), .A4(n625), .ZN(n621) );
    IND4D0BWP7T U220 ( .A1(n626), .B1(n627), .B2(n628), .B3(n629), .ZN(n625) );
    AOI221D0BWP7T U221 ( .A1(n630), .A2(n631), .B1(n632), .B2(n633), .C(n634),
         .ZN(n629) );
    OAI211D0BWP7T U222 ( .A1(n435), .A2(n635), .B(n636), .C(n637), .ZN(n624) );
    AOI222D0BWP7T U223 ( .A1(n280), .A2(n638), .B1(n639), .B2(n269), .C1(n43),
         .C2(n482), .ZN(n637) );
    AOI22D0BWP7T U224 ( .A1(n352), .A2(n477), .B1(n183), .B2(n640), .ZN(n636) );
    ND3D0BWP7T U225 ( .A1(n641), .A2(n642), .A3(n643), .ZN(n623) );
    NR4D0BWP7T U226 ( .A1(n644), .A2(n645), .A3(n423), .A4(n646), .ZN(n643) );
    ND3D0BWP7T U227 ( .A1(n647), .A2(n648), .A3(n202), .ZN(n644) );
    CKND0BWP7T U228 ( .I(n649), .ZN(n202) );
    AOI221D0BWP7T U229 ( .A1(n650), .A2(n282), .B1(n608), .B2(n213), .C(n651),
         .ZN(n642) );
    AOI21D0BWP7T U230 ( .A1(n437), .A2(n189), .B(n475), .ZN(n651) );
    AOI222D0BWP7T U231 ( .A1(n652), .A2(n340), .B1(n574), .B2(n259), .C1(n355),
         .C2(n479), .ZN(n641) );
    ND4D0BWP7T U232 ( .A1(n653), .A2(n654), .A3(n655), .A4(n656), .ZN(n622) );
    INR4D0BWP7T U233 ( .A1(n657), .B1(n658), .B2(n659), .B3(n660), .ZN(n656) );
    ND3D0BWP7T U234 ( .A1(n661), .A2(n662), .A3(n663), .ZN(n658) );
    NR4D0BWP7T U235 ( .A1(n664), .A2(n665), .A3(n666), .A4(n667), .ZN(n655) );
    ND3D0BWP7T U236 ( .A1(n668), .A2(n669), .A3(n670), .ZN(n664) );
    NR4D0BWP7T U237 ( .A1(n671), .A2(n672), .A3(n17), .A4(n306), .ZN(n654) );
    ND3D0BWP7T U238 ( .A1(n673), .A2(n674), .A3(n675), .ZN(n306) );
    NR4D0BWP7T U239 ( .A1(n676), .A2(n677), .A3(n678), .A4(n679), .ZN(n675) );
    OAI222D0BWP7T U240 ( .A1(n312), .A2(n680), .B1(n681), .B2(n682), .C1(n49),
         .C2(n409), .ZN(n679) );
    OAI221D0BWP7T U241 ( .A1(n683), .A2(n684), .B1(n199), .B2(n149), .C(n685),
         .ZN(n678) );
    AOI22D0BWP7T U242 ( .A1(n374), .A2(n686), .B1(n586), .B2(n687), .ZN(n685) );
    OAI211D0BWP7T U243 ( .A1(n688), .A2(n614), .B(n689), .C(n690), .ZN(n677) );
    AOI22D0BWP7T U244 ( .A1(n691), .A2(n289), .B1(n692), .B2(n693), .ZN(n690) );
    IOA21D0BWP7T U245 ( .A1(n694), .A2(n695), .B(n696), .ZN(n689) );
    IND4D0BWP7T U246 ( .A1(n41), .B1(n697), .B2(n698), .B3(n699), .ZN(n676) );
    OAI21D0BWP7T U247 ( .A1(n352), .A2(n700), .B(n701), .ZN(n697) );
    NR4D0BWP7T U248 ( .A1(n702), .A2(n703), .A3(n704), .A4(n705), .ZN(n674) );
    IIND4D0BWP7T U249 ( .A1(n706), .A2(n707), .B1(n708), .B2(n709), .ZN(n702) );
    NR4D0BWP7T U250 ( .A1(n710), .A2(n124), .A3(n138), .A4(n128), .ZN(n673) );
    ND3D0BWP7T U251 ( .A1(n711), .A2(n712), .A3(n713), .ZN(n128) );
    AOI222D0BWP7T U252 ( .A1(n714), .A2(n715), .B1(n282), .B2(n716), .C1(n615),
         .C2(n294), .ZN(n713) );
    AOI22D0BWP7T U253 ( .A1(n717), .A2(n583), .B1(n718), .B2(n482), .ZN(n711) );
    ND3D0BWP7T U254 ( .A1(n719), .A2(n720), .A3(n721), .ZN(n138) );
    AOI211D0BWP7T U255 ( .A1(n453), .A2(n640), .B(n722), .C(n723), .ZN(n721) );
    AOI21D0BWP7T U256 ( .A1(n724), .A2(n407), .B(n404), .ZN(n723) );
    AOI21D0BWP7T U257 ( .A1(n725), .A2(n726), .B(n509), .ZN(n722) );
    AOI22D0BWP7T U258 ( .A1(n216), .A2(n727), .B1(n212), .B2(n631), .ZN(n719) );
    OAI211D0BWP7T U259 ( .A1(n728), .A2(n412), .B(n729), .C(n730), .ZN(n124) );
```

```
    AOI211D0BWP7T U260 ( .A1(n349), .A2(n357), .B(n731), .C(n732), .ZN(n730) );
    AOI22D0BWP7T U261 ( .A1(n652), .A2(n733), .B1(n204), .B2(n339), .ZN(n729) );
    IIND4D0BWP7T U262 ( .A1(n734), .A2(n735), .B1(n736), .B2(n737), .ZN(n710) );
    OAI221D0BWP7T U263 ( .A1(n347), .A2(n404), .B1(n408), .B2(n738), .C(n739),
         .ZN(n17) );
    AOI222D0BWP7T U264 ( .A1(n608), .A2(n740), .B1(n350), .B2(n340), .C1(n214),
         .C2(n633), .ZN(n739) );
    ND3D0BWP7T U265 ( .A1(n126), .A2(n125), .A3(n119), .ZN(n671) );
    AN4D0BWP7T U266 ( .A1(n741), .A2(n742), .A3(n743), .A4(n744), .Z(n119) );
    NR4D0BWP7T U267 ( .A1(n745), .A2(n746), .A3(n747), .A4(n748), .ZN(n744) );
    AOI21D0BWP7T U268 ( .A1(n635), .A2(n207), .B(n408), .ZN(n748) );
    AOI21D0BWP7T U269 ( .A1(n434), .A2(n749), .B(n475), .ZN(n746) );
    OAI222D0BWP7T U270 ( .A1(n229), .A2(n750), .B1(n751), .B2(n752), .C1(n435),
         .C2(n753), .ZN(n745) );
    AOI222D0BWP7T U271 ( .A1(n266), .A2(n481), .B1(n754), .B2(n755), .C1(n43),
         .C2(n756), .ZN(n743) );
    OA21D0BWP7T U272 ( .A1(n757), .A2(n327), .B(n758), .Z(n741) );
    OA211D0BWP7T U273 ( .A1(n759), .A2(n760), .B(n761), .C(n762), .Z(n125) );
    AOI222D0BWP7T U274 ( .A1(n763), .A2(n764), .B1(n686), .B2(n372), .C1(n586),
         .C2(n358), .ZN(n762) );
    AN4D0BWP7T U275 ( .A1(n765), .A2(n766), .A3(n767), .A4(n768), .Z(n126) );
    AOI211D0BWP7T U276 ( .A1(n324), .A2(n769), .B(n770), .C(n771), .ZN(n768) );
    AOI21D0BWP7T U277 ( .A1(n406), .A2(n772), .B(n680), .ZN(n771) );
    AOI21D0BWP7T U278 ( .A1(n406), .A2(n326), .B(n309), .ZN(n770) );
    AOI222D0BWP7T U279 ( .A1(n483), .A2(n773), .B1(n774), .B2(n281), .C1(n183),
         .C2(n316), .ZN(n767) );
    AOI221D0BWP7T U280 ( .A1(n31), .A2(n775), .B1(n182), .B2(n485), .C(n776),
         .ZN(n766) );
    CKND0BWP7T U281 ( .I(n777), .ZN(n765) );
    IND3D0BWP7T U282 ( .A1(n778), .B1(n779), .B2(n780), .ZN(n777) );
    INR4D0BWP7T U283 ( .A1(n781), .B1(n782), .B2(n536), .B3(n783), .ZN(n653) );
    ND4D0BWP7T U284 ( .A1(n784), .A2(n785), .A3(n786), .A4(n787), .ZN(n536) );
    NR4D0BWP7T U285 ( .A1(n788), .A2(n789), .A3(n790), .A4(n791), .ZN(n787) );
    IND3D0BWP7T U286 ( .A1(n792), .B1(n793), .B2(n794), .ZN(n789) );
    IND3D0BWP7T U287 ( .A1(n99), .B1(n12), .B2(n795), .ZN(n788) );
    INR3D0BWP7T U288 ( .A1(n796), .B1(n129), .B2(n131), .ZN(n795) );
    ND3D0BWP7T U289 ( .A1(n797), .A2(n798), .A3(n799), .ZN(n131) );
    AOI211D0BWP7T U290 ( .A1(n297), .A2(n371), .B(n800), .C(n801), .ZN(n799) );
    AOI21D0BWP7T U291 ( .A1(n802), .A2(n803), .B(n804), .ZN(n801) );
    AOI22D0BWP7T U292 ( .A1(n353), .A2(n205), .B1(n337), .B2(n805), .ZN(n797) );
    ND4D0BWP7T U293 ( .A1(n806), .A2(n807), .A3(n808), .A4(n809), .ZN(n129) );
    NR4D0BWP7T U294 ( .A1(n810), .A2(n811), .A3(n812), .A4(n813), .ZN(n809) );
    AOI21D0BWP7T U295 ( .A1(n814), .A2(n815), .B(n327), .ZN(n813) );
    AOI21D0BWP7T U296 ( .A1(n816), .A2(n227), .B(n817), .ZN(n812) );
    OAI22D0BWP7T U297 ( .A1(n196), .A2(n635), .B1(n818), .B2(n819), .ZN(n811) );
    OAI222D0BWP7T U298 ( .A1(n228), .A2(n820), .B1(n435), .B2(n47), .C1(n512),
         .C2(n409), .ZN(n810) );
    INR3D0BWP7T U299 ( .A1(n821), .B1(n822), .B2(n823), .ZN(n808) );
    AOI22D0BWP7T U300 ( .A1(n43), .A2(n259), .B1(n282), .B2(n481), .ZN(n806) );
    OA221D0BWP7T U301 ( .A1(n824), .A2(n825), .B1(n329), .B2(n230), .C(n826),
         .Z(n12) );
    AOI222D0BWP7T U302 ( .A1(n827), .A2(n828), .B1(n350), .B2(n696), .C1(n829),
         .C2(n686), .ZN(n826) );
    ND4D0BWP7T U303 ( .A1(n830), .A2(n831), .A3(n832), .A4(n833), .ZN(n99) );
```

```
    AOI211D0BWP7T U304 ( .A1(n460), .A2(n834), .B(n835), .C(n836), .ZN(n833) );
    AOI21D0BWP7T U305 ( .A1(n837), .A2(n838), .B(n329), .ZN(n836) );
    OAI22D0BWP7T U306 ( .A1(n839), .A2(n434), .B1(n824), .B2(n750), .ZN(n835) );
    CKND0BWP7T U307 ( .I(n840), .ZN(n834) );
    AOI221D0BWP7T U308 ( .A1(n350), .A2(n297), .B1(n841), .B2(n351), .C(n842),
         .ZN(n832) );
    OAI22D0BWP7T U309 ( .A1(n196), .A2(n843), .B1(n614), .B2(n319), .ZN(n842) );
    AOI221D0BWP7T U310 ( .A1(n714), .A2(n583), .B1(n844), .B2(n296), .C(n845),
         .ZN(n831) );
    OAI22D0BWP7T U311 ( .A1(n227), .A2(n846), .B1(n512), .B2(n47), .ZN(n845) );
    AOI211D0BWP7T U312 ( .A1(n718), .A2(n266), .B(n847), .C(n848), .ZN(n830) );
    OAI22D0BWP7T U313 ( .A1(n435), .A2(n849), .B1(n35), .B2(n236), .ZN(n847) );
    AOI221D0BWP7T U314 ( .A1(n586), .A2(n841), .B1(n372), .B2(n204), .C(n850),
         .ZN(n786) );
    AO211D0BWP7T U315 ( .A1(n764), .A2(n298), .B(n851), .C(n852), .Z(n850) );
    AOI222D0BWP7T U316 ( .A1(n352), .A2(n853), .B1(n854), .B2(n696), .C1(n477),
         .C2(n686), .ZN(n785) );
    AOI22D0BWP7T U317 ( .A1(n855), .A2(n805), .B1(n717), .B2(n482), .ZN(n784) );
    IND3D0BWP7T U318 ( .A1(n856), .B1(n535), .B2(n857), .ZN(n782) );
    AN3D0BWP7T U319 ( .A1(n858), .A2(n859), .A3(n860), .Z(n535) );
    NR4D0BWP7T U320 ( .A1(n861), .A2(n862), .A3(n863), .A4(n864), .ZN(n860) );
    OAI21D0BWP7T U321 ( .A1(n865), .A2(n155), .B(n866), .ZN(n862) );
    OAI211D0BWP7T U322 ( .A1(n223), .A2(n406), .B(n867), .C(n868), .ZN(n861) );
    AOI22D0BWP7T U323 ( .A1(n31), .A2(n242), .B1(n574), .B2(n293), .ZN(n868) );
    INR4D0BWP7T U324 ( .A1(n869), .B1(n870), .B2(n871), .B3(n872), .ZN(n859) );
    NR4D0BWP7T U325 ( .A1(n873), .A2(n130), .A3(n10), .A4(n101), .ZN(n858) );
    OAI221D0BWP7T U326 ( .A1(n509), .A2(n874), .B1(n724), .B2(n875), .C(n876),
         .ZN(n101) );
    AOI22D0BWP7T U327 ( .A1(n316), .A2(n877), .B1(n878), .B2(n485), .ZN(n876) );
    OAI221D0BWP7T U328 ( .A1(n221), .A2(n879), .B1(n880), .B2(n320), .C(n881),
         .ZN(n10) );
    AOI22D0BWP7T U329 ( .A1(n632), .A2(n483), .B1(n587), .B2(n882), .ZN(n881) );
    ND3D0BWP7T U330 ( .A1(n883), .A2(n884), .A3(n885), .ZN(n130) );
    AOI211D0BWP7T U331 ( .A1(n886), .A2(n633), .B(n887), .C(n888), .ZN(n885) );
    OAI22D0BWP7T U332 ( .A1(n35), .A2(n889), .B1(n890), .B2(n325), .ZN(n887) );
    MAOI22D0BWP7T U333 ( .A1(n316), .A2(n891), .B1(n892), .B2(n893), .ZN(n884)
          );
    OA22D0BWP7T U334 ( .A1(n223), .A2(n894), .B1(n309), .B2(n895), .Z(n883) );
    NR3D0BWP7T U335 ( .A1(n896), .A2(n897), .A3(n898), .ZN(n620) );
    IND4D0BWP7T U336 ( .A1(n899), .B1(n548), .B2(n900), .B3(n901), .ZN(n898) );
    NR4D0BWP7T U337 ( .A1(n902), .A2(n903), .A3(n904), .A4(n366), .ZN(n901) );
    ND4D0BWP7T U338 ( .A1(n905), .A2(n906), .A3(n907), .A4(n908), .ZN(n366) );
    OA221D0BWP7T U339 ( .A1(n909), .A2(n910), .B1(n911), .B2(n912), .C(n913),
         .Z(n908) );
    AOI222D0BWP7T U340 ( .A1(n292), .A2(n296), .B1(n914), .B2(n438), .C1(n286),
         .C2(n266), .ZN(n907) );
    AOI211D0BWP7T U341 ( .A1(n604), .A2(n427), .B(n495), .C(n915), .ZN(n906) );
    NR3D0BWP7T U342 ( .A1(n916), .A2(n917), .A3(n918), .ZN(n905) );
    AN3D0BWP7T U343 ( .A1(n919), .A2(n920), .A3(n921), .Z(n548) );
    NR4D0BWP7T U344 ( .A1(n922), .A2(n923), .A3(n924), .A4(n925), .ZN(n921) );
    OAI222D0BWP7T U345 ( .A1(n618), .A2(n849), .B1(n49), .B2(n752), .C1(n405),
         .C2(n880), .ZN(n925) );
    OAI222D0BWP7T U346 ( .A1(n197), .A2(n926), .B1(n413), .B2(n927), .C1(n928),
         .C2(n929), .ZN(n924) );
```

```
    OAI222D0BWP7T U347 ( .A1(n435), .A2(n417), .B1(n930), .B2(n143), .C1(n146),
         .C2(n804), .ZN(n923) );
    OAI211D0BWP7T U348 ( .A1(n931), .A2(n932), .B(n933), .C(n934), .ZN(n922) );
    OAI21D0BWP7T U349 ( .A1(n605), .A2(n267), .B(n427), .ZN(n933) );
    CKND2D0BWP7T U350 ( .A1(n612), .A2(n909), .ZN(n267) );
    NR4D0BWP7T U351 ( .A1(n935), .A2(n936), .A3(n937), .A4(n938), .ZN(n920) );
    IND3D0BWP7T U352 ( .A1(n939), .B1(n940), .B2(n941), .ZN(n935) );
    CKND0BWP7T U353 ( .I(n942), .ZN(n940) );
    INR4D0BWP7T U354 ( .A1(n943), .B1(n944), .B2(n945), .B3(n946), .ZN(n919) );
    ND3D0BWP7T U355 ( .A1(n947), .A2(n948), .A3(n949), .ZN(n944) );
    IND4D0BWP7T U356 ( .A1(n950), .B1(n951), .B2(n952), .B3(n953), .ZN(n897) );
    INR4D0BWP7T U357 ( .A1(n954), .B1(n955), .B2(n956), .B3(n957), .ZN(n953) );
    ND4D0BWP7T U358 ( .A1(n958), .A2(n959), .A3(n960), .A4(n961), .ZN(n896) );
    NR4D0BWP7T U359 ( .A1(n962), .A2(n963), .A3(n964), .A4(n965), .ZN(n961) );
    AOI21D0BWP7T U360 ( .A1(n436), .A2(n911), .B(n35), .ZN(n965) );
    AOI21D0BWP7T U361 ( .A1(n189), .A2(n431), .B(n318), .ZN(n964) );
    AOI21D0BWP7T U362 ( .A1(n966), .A2(n49), .B(n612), .ZN(n963) );
    AOI221D0BWP7T U363 ( .A1(n600), .A2(n967), .B1(n438), .B2(n260), .C(n968),
         .ZN(n960) );
    AOI21D0BWP7T U364 ( .A1(n476), .A2(n579), .B(n820), .ZN(n968) );
    OR2D0BWP7T U365 ( .A1(n969), .A2(n970), .Z(n260) );
    AOI221D0BWP7T U366 ( .A1(n971), .A2(n291), .B1(n606), .B2(n972), .C(n973),
         .ZN(n959) );
    OAI22D0BWP7T U367 ( .A1(n196), .A2(n189), .B1(n974), .B2(n975), .ZN(n973) );
    CKND0BWP7T U368 ( .I(n966), .ZN(n972) );
    AOI221D0BWP7T U369 ( .A1(n976), .A2(n583), .B1(n280), .B2(n640), .C(n977),
         .ZN(n958) );
    AO21D0BWP7T U370 ( .A1(q_out[1]), .A2(n1), .B(n978), .Z(n3520) );
    MUX2ND0BWP7T U371 ( .I0(n979), .I1(n980), .S(n5), .ZN(n978) );
    NR4D0BWP7T U372 ( .A1(n981), .A2(n982), .A3(n983), .A4(n984), .ZN(n980) );
    ND4D0BWP7T U373 ( .A1(n985), .A2(n986), .A3(n987), .A4(n988), .ZN(n984) );
    INR3D0BWP7T U374 ( .A1(n145), .B1(n989), .B2(n990), .ZN(n988) );
    IND4D0BWP7T U375 ( .A1(n991), .B1(n992), .B2(n993), .B3(n994), .ZN(n983) );
    AOI222D0BWP7T U376 ( .A1(n182), .A2(n608), .B1(n995), .B2(n517), .C1(n242),
         .C2(n640), .ZN(n994) );
    IIND4D0BWP7T U377 ( .A1(n996), .A2(n524), .B1(n997), .B2(n998), .ZN(n982) );
    INR4D0BWP7T U378 ( .A1(n999), .B1(n1000), .B2(n1001), .B3(n1002), .ZN(n998)
          );
    IND3D0BWP7T U379 ( .A1(n121), .B1(n779), .B2(n111), .ZN(n1000) );
    OA221D0BWP7T U380 ( .A1(n865), .A2(n749), .B1(n824), .B2(n1003), .C(n1004),
         .Z(n111) );
    AOI22D0BWP7T U381 ( .A1(n615), .A2(n693), .B1(n583), .B2(n1005), .ZN(n1004)
          );
    AOI22D0BWP7T U382 ( .A1(n608), .A2(n774), .B1(n633), .B2(n878), .ZN(n779) );
    OAI221D0BWP7T U383 ( .A1(n874), .A2(n226), .B1(n234), .B2(n724), .C(n1006),
         .ZN(n121) );
    AOI22D0BWP7T U384 ( .A1(n1007), .A2(n259), .B1(n774), .B2(n316), .ZN(n1006)
          );
    NR3D0BWP7T U385 ( .A1(n1008), .A2(n18), .A3(n307), .ZN(n997) );
    ND4D0BWP7T U386 ( .A1(n1009), .A2(n1010), .A3(n1011), .A4(n1012), .ZN(n307)
          );
    NR4D0BWP7T U387 ( .A1(n1013), .A2(n1014), .A3(n1015), .A4(n1016), .ZN(n1012)
          );
    OAI222D0BWP7T U388 ( .A1(n312), .A2(n875), .B1(n1017), .B2(n414), .C1(n404),
```

```
      .C2(n510), .ZN(n1016) );
    OAI222D0BWP7T U389 ( .A1(n232), .A2(n147), .B1(n728), .B2(n149), .C1(n802),
      .C2(n157), .ZN(n1015) );
    IND4D0BWP7T U390 ( .A1(n1018), .B1(n954), .B2(n1019), .B3(n1020), .ZN(n1014)
      );
    INR2D0BWP7T U391 ( .A1(n1021), .B1(n1022), .ZN(n1020) );
    ND4D0BWP7T U392 ( .A1(n1023), .A2(n549), .A3(n1024), .A4(n1025), .ZN(n1013)
      );
    NR4D0BWP7T U393 ( .A1(n1026), .A2(n122), .A3(n1027), .A4(n132), .ZN(n1025)
      );
    ND4D0BWP7T U394 ( .A1(n1028), .A2(n1029), .A3(n1030), .A4(n1031), .ZN(n132)
      );
    NR4D0BWP7T U395 ( .A1(n1032), .A2(n1033), .A3(n1034), .A4(n1035), .ZN(n1031)
      );
    IND3D0BWP7T U396 ( .A1(n1036), .B1(n1037), .B2(n1038), .ZN(n1032) );
    CKND0BWP7T U397 ( .I(n1039), .ZN(n1037) );
    AOI222D0BWP7T U398 ( .A1(n1040), .A2(n204), .B1(n696), .B2(n1041), .C1(n371),
      .C2(n1042), .ZN(n1030) );
    ND3D0BWP7T U399 ( .A1(n412), .A2(n157), .A3(n760), .ZN(n1041) );
    AOI22D0BWP7T U400 ( .A1(n353), .A2(n805), .B1(n687), .B2(n351), .ZN(n1028)
      );
    ND3D0BWP7T U401 ( .A1(n1043), .A2(n1044), .A3(n1045), .ZN(n122) );
    INR4D0BWP7T U402 ( .A1(n1046), .B1(n1047), .B2(n1048), .B3(n1049), .ZN(n1045)
     );
    AOI21D0BWP7T U403 ( .A1(n820), .A2(n432), .B(n409), .ZN(n1049) );
    OAI22D0BWP7T U404 ( .A1(n974), .A2(n817), .B1(n751), .B2(n1050), .ZN(n1048)
      );
    IND3D0BWP7T U405 ( .A1(n1051), .B1(n1052), .B2(n1053), .ZN(n1047) );
    AOI22D0BWP7T U406 ( .A1(n1054), .A2(n755), .B1(n1055), .B2(n693), .ZN(n1053)
      );
    INR4D0BWP7T U407 ( .A1(n1056), .B1(n1057), .B2(n1058), .B3(n1059), .ZN(n1044)
     );
    IINR4D0BWP7T U408 ( .A1(n670), .A2(n520), .B1(n1060), .B2(n1061), .ZN(n1043)
      );
    AOI222D0BWP7T U409 ( .A1(n482), .A2(n714), .B1(n828), .B2(n1062), .C1(n517),
      .C2(n532), .ZN(n520) );
    AOI22D0BWP7T U410 ( .A1(n515), .A2(n1063), .B1(n355), .B2(n298), .ZN(n670)
      );
    IND3D0BWP7T U411 ( .A1(n1064), .B1(n1065), .B2(n1066), .ZN(n1026) );
    NR3D0BWP7T U412 ( .A1(n946), .A2(n123), .A3(n24), .ZN(n1024) );
    ND3D0BWP7T U413 ( .A1(n1067), .A2(n1068), .A3(n1069), .ZN(n24) );
    INR4D0BWP7T U414 ( .A1(n669), .B1(n665), .B2(n1070), .B3(n1071), .ZN(n1069)
      );
    OAI22D0BWP7T U415 ( .A1(n405), .A2(n875), .B1(n404), .B2(n1072), .ZN(n1070)
      );
    OAI21D0BWP7T U416 ( .A1(n1073), .A2(n1074), .B(n593), .ZN(n665) );
    AOI222D0BWP7T U417 ( .A1(n764), .A2(n353), .B1(n355), .B2(n701), .C1(n357),
      .C2(n530), .ZN(n669) );
    AOI222D0BWP7T U418 ( .A1(n1075), .A2(n355), .B1(n632), .B2(n1076), .C1(n337),
      .C2(n686), .ZN(n1068) );
    AOI222D0BWP7T U419 ( .A1(n1077), .A2(n517), .B1(n31), .B2(n1078), .C1(n1079),
      .C2(n755), .ZN(n1067) );
    ND3D0BWP7T U420 ( .A1(n1080), .A2(n1081), .A3(n1082), .ZN(n123) );
    NR4D0BWP7T U421 ( .A1(n1083), .A2(n519), .A3(n1084), .A4(n1085), .ZN(n1082)
```

```
      );
    OAI211D0BWP7T U422 ( .A1(n1086), .A2(n1087), .B(n1088), .C(n1089), .ZN(n519)
      );
    AOI22D0BWP7T U423 ( .A1(n1090), .A2(n1091), .B1(n242), .B2(n483), .ZN(n1089)
      );
    IND3D0BWP7T U424 ( .A1(n1092), .B1(n1093), .B2(n1094), .ZN(n1083) );
    AOI22D0BWP7T U425 ( .A1(n1007), .A2(n294), .B1(n1095), .B2(n29), .ZN(n1094)
      );
    AOI211D0BWP7T U426 ( .A1(n633), .A2(n1096), .B(n1097), .C(n1098), .ZN(n1081)
      );
    AOI21D0BWP7T U427 ( .A1(n895), .A2(n311), .B(n221), .ZN(n1098) );
    OAI22D0BWP7T U428 ( .A1(n875), .A2(n1099), .B1(n1100), .B2(n309), .ZN(n1097)
      );
    AOI222D0BWP7T U429 ( .A1(n1101), .A2(n1102), .B1(n33), .B2(n322), .C1(n1078),
      .C2(n34), .ZN(n1080) );
    AO21D0BWP7T U430 ( .A1(n282), .A2(n292), .B(n1103), .Z(n946) );
    AN3D0BWP7T U431 ( .A1(n1104), .A2(n1105), .A3(n1106), .Z(n549) );
    AOI22D0BWP7T U432 ( .A1(n1107), .A2(n293), .B1(n292), .B2(n693), .ZN(n1106)
      );
    IOA21D0BWP7T U433 ( .A1(n474), .A2(n433), .B(n601), .ZN(n1104) );
    NR4D0BWP7T U434 ( .A1(n1108), .A2(n1109), .A3(n1110), .A4(n1111), .ZN(n1011)
      );
    OAI22D0BWP7T U435 ( .A1(n1112), .A2(n682), .B1(n1113), .B2(n1114), .ZN(n1111)
     );
    OAI222D0BWP7T U436 ( .A1(n1115), .A2(n46), .B1(n1116), .B2(n577), .C1(n1117),
      .C2(n612), .ZN(n1110) );
    OAI222D0BWP7T U437 ( .A1(n618), .A2(n1118), .B1(n512), .B2(n155), .C1(n152),
      .C2(n189), .ZN(n1109) );
    OAI222D0BWP7T U438 ( .A1(n1119), .A2(n1120), .B1(n616), .B2(n318), .C1(n912),
      .C2(n1121), .ZN(n1108) );
    NR3D0BWP7T U439 ( .A1(n1122), .A2(n42), .A3(n599), .ZN(n616) );
    INR4D0BWP7T U440 ( .A1(n424), .B1(n1123), .B2(n1124), .B3(n1125), .ZN(n1010)
      );
    ND3D0BWP7T U441 ( .A1(n1126), .A2(n1127), .A3(n1128), .ZN(n1123) );
    OAI21D0BWP7T U442 ( .A1(n372), .A2(n855), .B(n340), .ZN(n1128) );
    OAI21D0BWP7T U443 ( .A1(n763), .A2(n479), .B(n1129), .ZN(n1126) );
    NR4D0BWP7T U444 ( .A1(n1130), .A2(n1131), .A3(n1132), .A4(n1133), .ZN(n1009)
      );
    AOI21D0BWP7T U445 ( .A1(n879), .A2(n1134), .B(n880), .ZN(n1133) );
    AOI21D0BWP7T U446 ( .A1(n819), .A2(n975), .B(n227), .ZN(n1132) );
    AOI31D0BWP7T U447 ( .A1(n578), .A2(n1135), .A3(n911), .B(n196), .ZN(n1131)
      );
    OAI22D0BWP7T U448 ( .A1(n433), .A2(n35), .B1(n1136), .B2(n1137), .ZN(n1130)
      );
    NR2D0BWP7T U449 ( .A1(n295), .A2(n582), .ZN(n433) );
    ND3D0BWP7T U450 ( .A1(n661), .A2(n1138), .A3(n1139), .ZN(n18) );
    AOI22D0BWP7T U451 ( .A1(n316), .A2(n1090), .B1(n774), .B2(n633), .ZN(n1139)
      );
    AOI22D0BWP7T U452 ( .A1(n485), .A2(n1140), .B1(n322), .B2(n215), .ZN(n661)
      );
    ND4D0BWP7T U453 ( .A1(n1141), .A2(n1142), .A3(n1143), .A4(n1144), .ZN(n524)
      );
    NR4D0BWP7T U454 ( .A1(n1145), .A2(n1146), .A3(n672), .A4(n783), .ZN(n1144)
      );
```

```
    ND4D0BWP7T U455 ( .A1(n1147), .A2(n1148), .A3(n1149), .A4(n1150), .ZN(n783)
          );
    NR4D0BWP7T U456 ( .A1(n184), .A2(n1151), .A3(n1152), .A4(n1153), .ZN(n1150)
          );
    AOI21D0BWP7T U457 ( .A1(n1120), .A2(n155), .B(n227), .ZN(n1153) );
    AOI21D0BWP7T U458 ( .A1(n895), .A2(n347), .B(n509), .ZN(n1152) );
    NR2D0BWP7T U459 ( .A1(n29), .A2(n775), .ZN(n895) );
    AOI222D0BWP7T U460 ( .A1(n886), .A2(n34), .B1(n460), .B2(n439), .C1(n1154),
         .C2(n1155), .ZN(n1149) );
    CKND0BWP7T U461 ( .I(n206), .ZN(n1155) );
    NR2D0BWP7T U462 ( .A1(n583), .A2(n268), .ZN(n206) );
    CKND2D0BWP7T U463 ( .A1(n577), .A2(n1156), .ZN(n439) );
    NR4D0BWP7T U464 ( .A1(n1157), .A2(n1158), .A3(n1159), .A4(n1160), .ZN(n1148)
          );
    NR3D0BWP7T U465 ( .A1(n118), .A2(n1161), .A3(n1162), .ZN(n1147) );
    ND4D0BWP7T U466 ( .A1(n1163), .A2(n1164), .A3(n1165), .A4(n1166), .ZN(n118)
          );
    AOI211D0BWP7T U467 ( .A1(n485), .A2(n1167), .B(n1168), .C(n1169), .ZN(n1166)
          );
    AOI21D0BWP7T U468 ( .A1(n151), .A2(n311), .B(n1073), .ZN(n1169) );
    OAI22D0BWP7T U469 ( .A1(n226), .A2(n1099), .B1(n890), .B2(n1170), .ZN(n1168)
          );
    AOI222D0BWP7T U470 ( .A1(n317), .A2(n34), .B1(n727), .B2(n322), .C1(n316),
         .C2(n1171), .ZN(n1165) );
    AOI221D0BWP7T U471 ( .A1(n1007), .A2(n583), .B1(n640), .B2(n588), .C(n1172),
         .ZN(n1164) );
    INR3D0BWP7T U472 ( .A1(n1173), .B1(n1174), .B2(n1175), .ZN(n1163) );
    ND3D0BWP7T U473 ( .A1(n1176), .A2(n1177), .A3(n1178), .ZN(n672) );
    NR4D0BWP7T U474 ( .A1(n1179), .A2(n117), .A3(n1180), .A4(n137), .ZN(n1178)
          );
    ND4D0BWP7T U475 ( .A1(n1181), .A2(n1182), .A3(n1183), .A4(n1184), .ZN(n137)
          );
    AOI22D0BWP7T U476 ( .A1(n841), .A2(n373), .B1(n204), .B2(n478), .ZN(n1184)
          );
    AOI22D0BWP7T U477 ( .A1(n855), .A2(n1185), .B1(n829), .B2(n1186), .ZN(n1183)
          );
    AOI22D0BWP7T U478 ( .A1(n374), .A2(n357), .B1(n340), .B2(n530), .ZN(n1182)
          );
    AOI21D0BWP7T U479 ( .A1(n356), .A2(n351), .B(n1187), .ZN(n1181) );
    IND4D0BWP7T U480 ( .A1(n1188), .B1(n1189), .B2(n1190), .B3(n1191), .ZN(n117)
          );
    AOI211D0BWP7T U481 ( .A1(n1192), .A2(n1193), .B(n1194), .C(n1195), .ZN(n1191)
         );
    AOI21D0BWP7T U482 ( .A1(n1196), .A2(n1197), .B(n329), .ZN(n1195) );
    OAI22D0BWP7T U483 ( .A1(n143), .A2(n635), .B1(n49), .B2(n1050), .ZN(n1194)
          );
    AOI222D0BWP7T U484 ( .A1(n583), .A2(n1198), .B1(n1199), .B2(n828), .C1(n1200)
        , .C2(n601), .ZN(n1190) );
    NR4D0BWP7T U485 ( .A1(n1201), .A2(n1202), .A3(n1203), .A4(n1204), .ZN(n1189)
          );
    NR2D0BWP7T U486 ( .A1(n910), .A2(n819), .ZN(n1201) );
    IND3D0BWP7T U487 ( .A1(n1205), .B1(n1206), .B2(n1207), .ZN(n1188) );
    IND3D0BWP7T U488 ( .A1(n1208), .B1(n1209), .B2(n1210), .ZN(n1179) );
    AOI222D0BWP7T U489 ( .A1(n210), .A2(n268), .B1(n374), .B2(n340), .C1(n349),
```

```
      .C2(n338), .ZN(n1177) );
    AOI211D0BWP7T U490 ( .A1(n604), .A2(n438), .B(n1211), .C(n1212), .ZN(n1176)
      );
    OR3D0BWP7T U491 ( .A1(n1213), .A2(n21), .A3(n15), .Z(n1146) );
    ND4D0BWP7T U492 ( .A1(n1214), .A2(n1215), .A3(n1216), .A4(n1217), .ZN(n15)
      );
    IAO21D0BWP7T U493 ( .A1(n143), .A2(n1218), .B(n646), .ZN(n1217) );
    AOI22D0BWP7T U494 ( .A1(n478), .A2(n351), .B1(n477), .B2(n204), .ZN(n1216)
      );
    AOI22D0BWP7T U495 ( .A1(n696), .A2(n479), .B1(n844), .B2(n268), .ZN(n1215)
      );
    AOI21D0BWP7T U496 ( .A1(n1200), .A2(n693), .B(n1219), .ZN(n1214) );
    ND3D0BWP7T U497 ( .A1(n1220), .A2(n1221), .A3(n1222), .ZN(n1145) );
    AOI222D0BWP7T U498 ( .A1(n295), .A2(n438), .B1(n970), .B2(n268), .C1(n600),
      .C2(n460), .ZN(n1222) );
    AOI22D0BWP7T U499 ( .A1(n603), .A2(n601), .B1(n1223), .B2(n715), .ZN(n1220)
      );
    INR4D0BWP7T U500 ( .A1(n159), .B1(n1224), .B2(n649), .B3(n158), .ZN(n1143)
      );
    ND3D0BWP7T U501 ( .A1(n1225), .A2(n161), .A3(n1226), .ZN(n1224) );
    AOI211D0BWP7T U502 ( .A1(n269), .A2(n283), .B(n1227), .C(n1228), .ZN(n1142)
      );
    AOI21D0BWP7T U503 ( .A1(n579), .A2(n1120), .B(n512), .ZN(n1228) );
    AOI21D0BWP7T U504 ( .A1(n436), .A2(n579), .B(n227), .ZN(n1227) );
    AOI222D0BWP7T U505 ( .A1(n265), .A2(n282), .B1(n580), .B2(n266), .C1(n1154),
      .C2(n693), .ZN(n1141) );
    ND4D0BWP7T U506 ( .A1(n1229), .A2(n1230), .A3(n1231), .A4(n1232), .ZN(n981)
      );
    NR4D0BWP7T U507 ( .A1(n1233), .A2(n1234), .A3(n1235), .A4(n1236), .ZN(n1232)
      );
    OAI222D0BWP7T U508 ( .A1(n865), .A2(n1237), .B1(n327), .B2(n1238), .C1(n329),
      .C2(n1239), .ZN(n1236) );
    OAI222D0BWP7T U509 ( .A1(n406), .A2(n346), .B1(n410), .B2(n682), .C1(n1170),
      .C2(n1240), .ZN(n1235) );
    OAI222D0BWP7T U510 ( .A1(n148), .A2(n149), .B1(n1241), .B2(n909), .C1(n146),
      .C2(n414), .ZN(n1234) );
    OAI222D0BWP7T U511 ( .A1(n318), .A2(n932), .B1(n35), .B2(n1242), .C1(n49),
      .C2(n1120), .ZN(n1233) );
    INR4D0BWP7T U512 ( .A1(n1243), .B1(n1244), .B2(n1245), .B3(n1246), .ZN(n1231)
     );
    AOI21D0BWP7T U513 ( .A1(n198), .A2(n1247), .B(n1248), .ZN(n1245) );
    ND3D0BWP7T U514 ( .A1(n1249), .A2(n1250), .A3(n1251), .ZN(n1244) );
    OAI21D0BWP7T U515 ( .A1(n1154), .A2(n264), .B(n427), .ZN(n1251) );
    IOA21D0BWP7T U516 ( .A1(n157), .A2(n1252), .B(n1129), .ZN(n1250) );
    OAI21D0BWP7T U517 ( .A1(n615), .A2(n43), .B(n460), .ZN(n1249) );
    AOI221D0BWP7T U518 ( .A1(n350), .A2(n455), .B1(n580), .B2(n1253), .C(n1254),
      .ZN(n1230) );
    AOI31D0BWP7T U519 ( .A1(n680), .A2(n346), .A3(n415), .B(n1255), .ZN(n1254)
      );
    CKND0BWP7T U520 ( .I(n1256), .ZN(n1253) );
    OA222D0BWP7T U521 ( .A1(n431), .A2(n1116), .B1(n474), .B2(n235), .C1(n613),
      .C2(n1257), .Z(n1229) );
    NR4D0BWP7T U522 ( .A1(n1258), .A2(n1259), .A3(n1260), .A4(n1261), .ZN(n979)
      );
```

```
    IND4D0BWP7T U523 ( .A1(n1262), .B1(n900), .B2(n857), .B3(n1263), .ZN(n1261)
          );
    NR4D0BWP7T U524 ( .A1(n1264), .A2(n367), .A3(n945), .A4(n546), .ZN(n1263) );
    ND4D0BWP7T U525 ( .A1(n1265), .A2(n1266), .A3(n1267), .A4(n1268), .ZN(n546)
          );
    NR4D0BWP7T U526 ( .A1(n1269), .A2(n1270), .A3(n1271), .A4(n1272), .ZN(n1268)
          );
    OAI211D0BWP7T U527 ( .A1(n1113), .A2(n927), .B(n1273), .C(n1274), .ZN(n1272)
          );
    CKND0BWP7T U528 ( .I(n1275), .ZN(n1274) );
    OAI211D0BWP7T U529 ( .A1(n1137), .A2(n926), .B(n1276), .C(n1277), .ZN(n1271)
          );
    MAOI22D0BWP7T U530 ( .A1(n352), .A2(n1075), .B1(n146), .B2(n1278), .ZN(n1277)
        );
    OAI21D0BWP7T U531 ( .A1(n298), .A2(n477), .B(n805), .ZN(n1276) );
    OAI222D0BWP7T U532 ( .A1(n1017), .A2(n1279), .B1(n816), .B2(n817), .C1(n1280)
       , .C2(n1248), .ZN(n1270) );
    IND4D0BWP7T U533 ( .A1(n1281), .B1(n1282), .B2(n1046), .B3(n185), .ZN(n1269)
          );
    NR4D0BWP7T U534 ( .A1(n1283), .A2(n1284), .A3(n1285), .A4(n1286), .ZN(n1267)
          );
    IND4D0BWP7T U535 ( .A1(n1287), .B1(n1288), .B2(n1289), .B3(n1290), .ZN(n1283)
        );
    NR4D0BWP7T U536 ( .A1(n1291), .A2(n1292), .A3(n1293), .A4(n1294), .ZN(n1266)
          );
    NR3D0BWP7T U537 ( .A1(n899), .A2(n1295), .A3(n533), .ZN(n1265) );
    OAI21D0BWP7T U538 ( .A1(n1296), .A2(n1248), .B(n1297), .ZN(n533) );
    IND4D0BWP7T U539 ( .A1(n851), .B1(n1298), .B2(n1299), .B3(n1300), .ZN(n899)
          );
    AOI22D0BWP7T U540 ( .A1(n298), .A2(n340), .B1(n338), .B2(n299), .ZN(n1300)
          );
    ND4D0BWP7T U541 ( .A1(n1301), .A2(n1302), .A3(n1303), .A4(n1304), .ZN(n945)
          );
    AOI211D0BWP7T U542 ( .A1(n264), .A2(n1305), .B(n1306), .C(n1307), .ZN(n1304)
          );
    AOI21D0BWP7T U543 ( .A1(n474), .A2(n436), .B(n512), .ZN(n1307) );
    OAI22D0BWP7T U544 ( .A1(n930), .A2(n1257), .B1(n1308), .B2(n579), .ZN(n1306)
          );
    AOI222D0BWP7T U545 ( .A1(n976), .A2(n269), .B1(n292), .B2(n1309), .C1(n265),
         .C2(n460), .ZN(n1303) );
    AOI222D0BWP7T U546 ( .A1(n295), .A2(n482), .B1(n970), .B2(n285), .C1(n600),
         .C2(n693), .ZN(n1302) );
    INR3D0BWP7T U547 ( .A1(n1310), .B1(n504), .B2(n1311), .ZN(n1301) );
    ND4D0BWP7T U548 ( .A1(n1312), .A2(n1313), .A3(n1314), .A4(n1315), .ZN(n367)
          );
    NR4D0BWP7T U549 ( .A1(n1316), .A2(n1317), .A3(n1318), .A4(n1319), .ZN(n1315)
          );
    OAI222D0BWP7T U550 ( .A1(n875), .A2(n1320), .B1(n345), .B2(n313), .C1(n320),
         .C2(n680), .ZN(n1319) );
    OAI222D0BWP7T U551 ( .A1(n927), .A2(n802), .B1(n1321), .B2(n201), .C1(n234),
         .C2(n1170), .ZN(n1318) );
    IND4D0BWP7T U552 ( .A1(n1322), .B1(n1323), .B2(n1324), .B3(n1325), .ZN(n1317)
        );
    NR3D0BWP7T U553 ( .A1(n1326), .A2(n1327), .A3(n1328), .ZN(n1325) );
```

```
    ND4D0BWP7T U554 ( .A1(n1023), .A2(n1329), .A3(n1330), .A4(n1331), .ZN(n1316)
          );
    INR4D0BWP7T U555 ( .A1(n1332), .B1(n1333), .B2(n1334), .B3(n1335), .ZN(n1331)
        );
    ND3D0BWP7T U556 ( .A1(n1336), .A2(n1337), .A3(n1338), .ZN(n1333) );
    NR3D0BWP7T U557 ( .A1(n1339), .A2(n1340), .A3(n1341), .ZN(n1330) );
    AOI21D0BWP7T U558 ( .A1(n296), .A2(n286), .B(n1342), .ZN(n1023) );
    NR4D0BWP7T U559 ( .A1(n1343), .A2(n1344), .A3(n1345), .A4(n1346), .ZN(n1314)
          );
    OAI222D0BWP7T U560 ( .A1(n1347), .A2(n612), .B1(n196), .B2(n579), .C1(n413),
         .C2(n231), .ZN(n1346) );
    OAI222D0BWP7T U561 ( .A1(n154), .A2(n911), .B1(n1348), .B2(n1349), .C1(n618),
         .C2(n909), .ZN(n1345) );
    OAI222D0BWP7T U562 ( .A1(n1350), .A2(n1351), .B1(n318), .B2(n153), .C1(n475),
         .C2(n1352), .ZN(n1344) );
    OAI222D0BWP7T U563 ( .A1(n416), .A2(n436), .B1(n974), .B2(n613), .C1(n408),
         .C2(n431), .ZN(n1343) );
    NR4D0BWP7T U564 ( .A1(n1353), .A2(n1354), .A3(n1355), .A4(n1356), .ZN(n1313)
          );
    AOI21D0BWP7T U565 ( .A1(n235), .A2(n1357), .B(n1135), .ZN(n1356) );
    AOI21D0BWP7T U566 ( .A1(n1358), .A2(n237), .B(n232), .ZN(n1354) );
    IND3D0BWP7T U567 ( .A1(n1359), .B1(n1225), .B2(n1360), .ZN(n1353) );
    NR4D0BWP7T U568 ( .A1(n1361), .A2(n1362), .A3(n1363), .A4(n1364), .ZN(n1312)
          );
    AOI21D0BWP7T U569 ( .A1(n617), .A2(n195), .B(n227), .ZN(n1364) );
    AOI21D0BWP7T U570 ( .A1(n194), .A2(n1156), .B(n143), .ZN(n1363) );
    AOI21D0BWP7T U571 ( .A1(n457), .A2(n189), .B(n910), .ZN(n1362) );
    NR2D0BWP7T U572 ( .A1(n603), .A2(n261), .ZN(n457) );
    OAI221D0BWP7T U573 ( .A1(n1365), .A2(n440), .B1(n1366), .B2(n1367), .C(n1368)
        , .ZN(n1361) );
    OAI21D0BWP7T U574 ( .A1(n605), .A2(n1122), .B(n269), .ZN(n1368) );
    OA211D0BWP7T U575 ( .A1(n318), .A2(n155), .B(n160), .C(n1369), .Z(n857) );
    AOI22D0BWP7T U576 ( .A1(n574), .A2(n266), .B1(n1370), .B2(n427), .ZN(n1369)
          );
    CKND0BWP7T U577 ( .I(n1371), .ZN(n900) );
    IND4D0BWP7T U578 ( .A1(n547), .B1(n1372), .B2(n1373), .B3(n1374), .ZN(n1371)
          );
    NR4D0BWP7T U579 ( .A1(n1375), .A2(n1376), .A3(n1125), .A4(n1377), .ZN(n1374)
          );
    IND3D0BWP7T U580 ( .A1(n1378), .B1(n1379), .B2(n1380), .ZN(n1375) );
    AOI222D0BWP7T U581 ( .A1(n1095), .A2(n631), .B1(n1381), .B2(n601), .C1(n606),
         .C2(n296), .ZN(n1373) );
    INR2D0BWP7T U582 ( .A1(n1382), .B1(n1383), .ZN(n1372) );
    ND4D0BWP7T U583 ( .A1(n1384), .A2(n1385), .A3(n1386), .A4(n1387), .ZN(n547)
          );
    NR4D0BWP7T U584 ( .A1(n1388), .A2(n1389), .A3(n1390), .A4(n864), .ZN(n1387)
          );
    AOI21D0BWP7T U585 ( .A1(n1072), .A2(n1391), .B(n509), .ZN(n1390) );
    OAI221D0BWP7T U586 ( .A1(n150), .A2(n407), .B1(n816), .B2(n578), .C(n1127),
         .ZN(n1388) );
    AOI211D0BWP7T U587 ( .A1(n1381), .A2(n693), .B(n1392), .C(n1393), .ZN(n1386)
          );
    OAI222D0BWP7T U588 ( .A1(n865), .A2(n1394), .B1(n143), .B2(n1118), .C1(n880),
         .C2(n320), .ZN(n1392) );
```

```
    NR3D0BWP7T U589 ( .A1(n1395), .A2(n1396), .A3(n1397), .ZN(n1385) );
    IND3D0BWP7T U590 ( .A1(n1398), .B1(n1399), .B2(n1400), .ZN(n1395) );
    INR4D0BWP7T U591 ( .A1(n1401), .B1(n1402), .B2(n1403), .B3(n1404), .ZN(n1384)
        );
    CKND2D0BWP7T U592 ( .A1(n1405), .A2(n1406), .ZN(n1402) );
    IND4D0BWP7T U593 ( .A1(n1407), .B1(n1408), .B2(n1409), .B3(n1410), .ZN(n1260)
        );
    INR4D0BWP7T U594 ( .A1(n1411), .B1(n1412), .B2(n1413), .B3(n1414), .ZN(n1410)
        );
    IND4D0BWP7T U595 ( .A1(n1415), .B1(n1416), .B2(n1417), .B3(n1418), .ZN(n1259)
        );
    NR4D0BWP7T U596 ( .A1(n1419), .A2(n1420), .A3(n1421), .A4(n1422), .ZN(n1418)
          );
    IND4D0BWP7T U597 ( .A1(n1423), .B1(n1424), .B2(n1425), .B3(n1426), .ZN(n1419)
        );
    AOI22D0BWP7T U598 ( .A1(n337), .A2(n340), .B1(n1427), .B2(n1129), .ZN(n1417)
          );
    AOI22D0BWP7T U599 ( .A1(n484), .A2(n322), .B1(n740), .B2(n30), .ZN(n1416) );
    IND3D0BWP7T U600 ( .A1(n1428), .B1(n1429), .B2(n1430), .ZN(n1415) );
    OAI22D0BWP7T U601 ( .A1(n875), .A2(n510), .B1(n1431), .B2(n889), .ZN(n1428)
          );
    ND4D0BWP7T U602 ( .A1(n1432), .A2(n1433), .A3(n1434), .A4(n1435), .ZN(n1258)
          );
    NR4D0BWP7T U603 ( .A1(n1436), .A2(n1437), .A3(n1438), .A4(n1439), .ZN(n1435)
          );
    OAI222D0BWP7T U604 ( .A1(n35), .A2(n579), .B1(n618), .B2(n46), .C1(n803),
         .C2(n237), .ZN(n1439) );
    OAI221D0BWP7T U605 ( .A1(n1440), .A2(n431), .B1(n435), .B2(n909), .C(n1441),
         .ZN(n1438) );
    AOI22D0BWP7T U606 ( .A1(n692), .A2(n44), .B1(n1442), .B2(n1443), .ZN(n1441)
          );
    AO222D0BWP7T U607 ( .A1(n1444), .A2(n268), .B1(n211), .B2(n292), .C1(n1445),
         .C2(n718), .Z(n1437) );
    OAI221D0BWP7T U608 ( .A1(n1446), .A2(n820), .B1(n1447), .B2(n1135), .C(n1448)
        , .ZN(n1436) );
    MAOI22D0BWP7T U609 ( .A1(n580), .A2(n967), .B1(n1449), .B2(n816), .ZN(n1448)
          );
    CKND2D0BWP7T U610 ( .A1(n475), .A2(n865), .ZN(n967) );
    NR4D0BWP7T U611 ( .A1(n1450), .A2(n1451), .A3(n1452), .A4(n1453), .ZN(n1434)
          );
    AOI31D0BWP7T U612 ( .A1(n1279), .A2(n1454), .A3(n1455), .B(n682), .ZN(n1453)
          );
    AOI31D0BWP7T U613 ( .A1(n412), .A2(n694), .A3(n1456), .B(n1017), .ZN(n1452)
          );
    AOI31D0BWP7T U614 ( .A1(n196), .A2(n865), .A3(n1431), .B(n617), .ZN(n1451)
          );
    ND4D0BWP7T U615 ( .A1(n1457), .A2(n1458), .A3(n1459), .A4(n1460), .ZN(n1450)
          );
    OAI21D0BWP7T U616 ( .A1(n841), .A2(n354), .B(n805), .ZN(n1460) );
    OAI21D0BWP7T U617 ( .A1(n1107), .A2(n1461), .B(n296), .ZN(n1459) );
    OAI21D0BWP7T U618 ( .A1(n1107), .A2(n600), .B(n427), .ZN(n1458) );
    OAI21D0BWP7T U619 ( .A1(n1462), .A2(n1463), .B(n583), .ZN(n1457) );
    INR4D0BWP7T U620 ( .A1(n1464), .B1(n465), .B2(n1465), .B3(n468), .ZN(n1433)
          );
```

```
    NR4D0BWP7T U621 ( .A1(n1466), .A2(n1467), .A3(n1468), .A4(n1469), .ZN(n1432)
          );
    AOI21D0BWP7T U622 ( .A1(n157), .A2(n147), .B(n1248), .ZN(n1469) );
    AOI21D0BWP7T U623 ( .A1(n910), .A2(n408), .B(n436), .ZN(n1468) );
    AOI21D0BWP7T U624 ( .A1(n1470), .A2(n865), .B(n1471), .ZN(n1467) );
    AO21D0BWP7T U625 ( .A1(q_out[0]), .A2(n1), .B(n1472), .Z(n3519) );
    MUX2ND0BWP7T U626 ( .I0(n1473), .I1(n1474), .S(n5), .ZN(n1472) );
    NR4D0BWP7T U627 ( .A1(n1475), .A2(n1476), .A3(n1477), .A4(n1478), .ZN(n1474)
          );
    IIND4D0BWP7T U628 ( .A1(n116), .A2(n115), .B1(n19), .B2(n1479), .ZN(n1478)
          );
    NR3D0BWP7T U629 ( .A1(n996), .A2(n525), .A3(n856), .ZN(n1479) );
    ND4D0BWP7T U630 ( .A1(n1480), .A2(n463), .A3(n112), .A4(n1481), .ZN(n856) );
    NR2D0BWP7T U631 ( .A1(n114), .A2(n11), .ZN(n1481) );
    ND4D0BWP7T U632 ( .A1(n987), .A2(n1482), .A3(n1483), .A4(n1484), .ZN(n11) );
    AOI211D0BWP7T U633 ( .A1(n484), .A2(n1485), .B(n1486), .C(n1487), .ZN(n1484)
          );
    AOI21D0BWP7T U634 ( .A1(n1488), .A2(n1170), .B(n223), .ZN(n1487) );
    OAI22D0BWP7T U635 ( .A1(n309), .A2(n325), .B1(n1489), .B2(n1391), .ZN(n1486)
          );
    AOI22D0BWP7T U636 ( .A1(n31), .A2(n1490), .B1(n293), .B2(n518), .ZN(n1483)
          );
    CKND2D0BWP7T U637 ( .A1(n886), .A2(n640), .ZN(n1482) );
    IND4D0BWP7T U638 ( .A1(n335), .B1(n1491), .B2(n1492), .B3(n1493), .ZN(n114)
          );
    NR4D0BWP7T U639 ( .A1(n1494), .A2(n1495), .A3(n1496), .A4(n1497), .ZN(n1493)
          );
    OAI222D0BWP7T U640 ( .A1(n196), .A2(n1156), .B1(n35), .B2(n578), .C1(n614),
         .C2(n911), .ZN(n1494) );
    NR3D0BWP7T U641 ( .A1(n1498), .A2(n1499), .A3(n1500), .ZN(n1492) );
    ND4D0BWP7T U642 ( .A1(n1501), .A2(n1502), .A3(n1503), .A4(n1504), .ZN(n335)
          );
    AOI221D0BWP7T U643 ( .A1(n650), .A2(n1505), .B1(n572), .B2(n1443), .C(n1506),
         .ZN(n1504) );
    OAI22D0BWP7T U644 ( .A1(n318), .A2(n1394), .B1(n1507), .B2(n431), .ZN(n1506)
          );
    CKND0BWP7T U645 ( .I(n1508), .ZN(n1505) );
    OA21D0BWP7T U646 ( .A1(n1121), .A2(n618), .B(n1509), .Z(n1503) );
    AN4D0BWP7T U647 ( .A1(n1510), .A2(n1511), .A3(n1512), .A4(n1513), .Z(n112)
          );
    INR4D0BWP7T U648 ( .A1(n1514), .B1(n1515), .B2(n1516), .B3(n1517), .ZN(n1513)
        );
    OAI211D0BWP7T U649 ( .A1(n1087), .A2(n223), .B(n1518), .C(n1519), .ZN(n1516)
          );
    CKND0BWP7T U650 ( .I(n1520), .ZN(n1518) );
    ND4D0BWP7T U651 ( .A1(n542), .A2(n521), .A3(n332), .A4(n999), .ZN(n1515) );
    AOI222D0BWP7T U652 ( .A1(n1078), .A2(n608), .B1(n1521), .B2(n630), .C1(n216),
         .C2(n280), .ZN(n999) );
    AN4D0BWP7T U653 ( .A1(n1522), .A2(n1523), .A3(n1524), .A4(n1525), .Z(n332)
          );
    AOI211D0BWP7T U654 ( .A1(n587), .A2(n1167), .B(n1526), .C(n1527), .ZN(n1525)
          );
    IAO21D0BWP7T U655 ( .A1(n213), .A2(n33), .B(n1073), .ZN(n1527) );
    CKND2D0BWP7T U656 ( .A1(n1528), .A2(n405), .ZN(n213) );
```

```
    MOAI22D0BWP7T U657 ( .A1(n234), .A2(n1529), .B1(n769), .B2(n215), .ZN(n1526)
          );
    CKND2D0BWP7T U658 ( .A1(n345), .A2(n874), .ZN(n1167) );
    AOI222D0BWP7T U659 ( .A1(n1101), .A2(n633), .B1(n1095), .B2(n740), .C1(n31),
         .C2(n1090), .ZN(n1524) );
    AOI222D0BWP7T U660 ( .A1(n317), .A2(n640), .B1(n774), .B2(n216), .C1(n638),
         .C2(n588), .ZN(n1523) );
    AOI221D0BWP7T U661 ( .A1(n630), .A2(n886), .B1(n316), .B2(n242), .C(n1530),
         .ZN(n1522) );
    OA221D0BWP7T U662 ( .A1(n347), .A2(n309), .B1(n1134), .B2(n680), .C(n1531),
         .Z(n521) );
    OAI21D0BWP7T U663 ( .A1(n280), .A2(n315), .B(n633), .ZN(n1531) );
    AN3D0BWP7T U664 ( .A1(n1464), .A2(n1532), .A3(n1533), .Z(n542) );
    AOI221D0BWP7T U665 ( .A1(n1534), .A2(n483), .B1(n1490), .B2(n1535), .C(n1536)
        , .ZN(n1512) );
    OAI211D0BWP7T U666 ( .A1(n326), .A2(n313), .B(n1537), .C(n1538), .ZN(n1536)
          );
    OAI21D0BWP7T U667 ( .A1(n773), .A2(n484), .B(n31), .ZN(n1538) );
    OAI31D0BWP7T U668 ( .A1(n608), .A2(n640), .A3(n638), .B(n215), .ZN(n1537) );
    AOI22D0BWP7T U669 ( .A1(n886), .A2(n1539), .B1(n216), .B2(n773), .ZN(n1511)
          );
    AOI22D0BWP7T U670 ( .A1(n293), .A2(n1007), .B1(n774), .B2(n640), .ZN(n1510)
          );
    IND4D0BWP7T U671 ( .A1(n1008), .B1(n781), .B2(n1540), .B3(n1541), .ZN(n525)
          );
    NR4D0BWP7T U672 ( .A1(n1542), .A2(n1543), .A3(n39), .A4(n1378), .ZN(n1541)
          );
    NR2D0BWP7T U673 ( .A1(n909), .A2(n614), .ZN(n1378) );
    OAI222D0BWP7T U674 ( .A1(n1257), .A2(n417), .B1(n408), .B2(n458), .C1(n618),
         .C2(n155), .ZN(n1542) );
    AOI221D0BWP7T U675 ( .A1(n1381), .A2(n460), .B1(n976), .B2(n294), .C(n110),
         .ZN(n1540) );
    ND4D0BWP7T U676 ( .A1(n1544), .A2(n1545), .A3(n1546), .A4(n1547), .ZN(n110)
          );
    AOI211D0BWP7T U677 ( .A1(n1548), .A2(n291), .B(n1549), .C(n274), .ZN(n1547)
          );
    OAI22D0BWP7T U678 ( .A1(n318), .A2(n617), .B1(n614), .B2(n153), .ZN(n1549)
          );
    AOI211D0BWP7T U679 ( .A1(n1463), .A2(n268), .B(n1550), .C(n1551), .ZN(n1546)
          );
    NR3D0BWP7T U680 ( .A1(n1552), .A2(n1553), .A3(n1554), .ZN(n1545) );
    AN3D0BWP7T U681 ( .A1(n668), .A2(n948), .A3(n1555), .Z(n1544) );
    AN3D0BWP7T U682 ( .A1(n1556), .A2(n1557), .A3(n1558), .Z(n668) );
    AOI221D0BWP7T U683 ( .A1(n1154), .A2(n1559), .B1(n585), .B2(n1560), .C(n1561)
        , .ZN(n1558) );
    AOI21D0BWP7T U684 ( .A1(n1120), .A2(n1352), .B(n143), .ZN(n1561) );
    CKND0BWP7T U685 ( .I(n1366), .ZN(n1560) );
    AOI22D0BWP7T U686 ( .A1(n582), .A2(n296), .B1(n1463), .B2(n285), .ZN(n1556)
          );
    AOI22D0BWP7T U687 ( .A1(n583), .A2(n599), .B1(n268), .B2(n42), .ZN(n781) );
    IND3D0BWP7T U688 ( .A1(n1562), .B1(n1563), .B2(n1564), .ZN(n1008) );
    AOI222D0BWP7T U689 ( .A1(n914), .A2(n1565), .B1(n603), .B2(n1566), .C1(n265),
         .C2(n294), .ZN(n1564) );
    CKND2D0BWP7T U690 ( .A1(n614), .A2(n318), .ZN(n1566) );
```

```
    AOI22D0BWP7T U691 ( .A1(n292), .A2(n1567), .B1(n261), .B2(n266), .ZN(n1563)
          );
    ND3D0BWP7T U692 ( .A1(n1568), .A2(n1569), .A3(n1570), .ZN(n996) );
    NR4D0BWP7T U693 ( .A1(n1571), .A2(n1572), .A3(n1573), .A4(n1574), .ZN(n1570)
          );
    AOI21D0BWP7T U694 ( .A1(n36), .A2(n1575), .B(n614), .ZN(n1574) );
    OAI21D0BWP7T U695 ( .A1(n1119), .A2(n909), .B(n1576), .ZN(n1572) );
    OAI21D0BWP7T U696 ( .A1(n970), .A2(n914), .B(n269), .ZN(n1576) );
    OAI221D0BWP7T U697 ( .A1(n475), .A2(n155), .B1(n512), .B2(n1135), .C(n1577),
         .ZN(n1571) );
    AOI22D0BWP7T U698 ( .A1(n265), .A2(n1578), .B1(n42), .B2(n1579), .ZN(n1577)
          );
    AOI211D0BWP7T U699 ( .A1(n212), .A2(n324), .B(n1580), .C(n1581), .ZN(n1569)
          );
    OAI22D0BWP7T U700 ( .A1(n35), .A2(n474), .B1(n910), .B2(n975), .ZN(n1580) );
    INR4D0BWP7T U701 ( .A1(n1582), .B1(n1583), .B2(n1584), .B3(n1262), .ZN(n1568)
        );
    OAI211D0BWP7T U702 ( .A1(n865), .A2(n431), .B(n1585), .C(n1586), .ZN(n1262)
          );
    AOI21D0BWP7T U703 ( .A1(n976), .A2(n583), .B(n1587), .ZN(n1586) );
    OAI21D0BWP7T U704 ( .A1(n292), .A2(n261), .B(n460), .ZN(n1585) );
    AN4D0BWP7T U705 ( .A1(n1588), .A2(n1589), .A3(n1590), .A4(n1591), .Z(n19) );
    NR4D0BWP7T U706 ( .A1(n1592), .A2(n1593), .A3(n1594), .A4(n1595), .ZN(n1591)
          );
    OAI22D0BWP7T U707 ( .A1(n614), .A2(n1596), .B1(n526), .B2(n408), .ZN(n1595)
          );
    OAI22D0BWP7T U708 ( .A1(n910), .A2(n47), .B1(n435), .B2(n1237), .ZN(n1594)
          );
    IND4D0BWP7T U709 ( .A1(n1597), .B1(n1598), .B2(n1221), .B3(n1599), .ZN(n1593)
        );
    CKND2D0BWP7T U710 ( .A1(n608), .A2(n29), .ZN(n1599) );
    IIND4D0BWP7T U711 ( .A1(n1001), .A2(n98), .B1(n1066), .B2(n1600), .ZN(n1592)
          );
    INR4D0BWP7T U712 ( .A1(n1601), .B1(n1602), .B2(n667), .B3(n1208), .ZN(n1600)
          );
    OAI222D0BWP7T U713 ( .A1(n327), .A2(n1603), .B1(n196), .B2(n753), .C1(n35),
         .C2(n409), .ZN(n1208) );
    ND3D0BWP7T U714 ( .A1(n1604), .A2(n1605), .A3(n1606), .ZN(n667) );
    NR4D0BWP7T U715 ( .A1(n1607), .A2(n1608), .A3(n1355), .A4(n1609), .ZN(n1606)
          );
    IND3D0BWP7T U716 ( .A1(n1610), .B1(n1611), .B2(n1612), .ZN(n1607) );
    AOI211D0BWP7T U717 ( .A1(n282), .A2(n1613), .B(n1281), .C(n1614), .ZN(n1605)
          );
    AOI21D0BWP7T U718 ( .A1(n409), .A2(n236), .B(n318), .ZN(n1614) );
    NR2D0BWP7T U719 ( .A1(n149), .A2(n926), .ZN(n1281) );
    CKND0BWP7T U720 ( .I(n1615), .ZN(n1613) );
    AOI22D0BWP7T U721 ( .A1(n1616), .A2(n828), .B1(n210), .B2(n583), .ZN(n1604)
          );
    OA221D0BWP7T U722 ( .A1(n1247), .A2(n1113), .B1(n413), .B2(n694), .C(n1617),
         .Z(n1066) );
    AOI22D0BWP7T U723 ( .A1(n615), .A2(n427), .B1(n1618), .B2(n296), .ZN(n1617)
          );
    ND3D0BWP7T U724 ( .A1(n1619), .A2(n1620), .A3(n1621), .ZN(n98) );
```

```
    INR4D0BWP7T U725 ( .A1(n1622), .B1(n1623), .B2(n1624), .B3(n1625), .ZN(n1621)
        );
    IND3D0BWP7T U726 ( .A1(n1626), .B1(n1627), .B2(n1628), .ZN(n1624) );
    IND4D0BWP7T U727 ( .A1(n1022), .B1(n663), .B2(n522), .B3(n1209), .ZN(n1623)
          );
    CKND0BWP7T U728 ( .I(n1629), .ZN(n1209) );
    IND4D0BWP7T U729 ( .A1(n1630), .B1(n1631), .B2(n1632), .B3(n1633), .ZN(n1629)
        );
    NR4D0BWP7T U730 ( .A1(n1634), .A2(n1635), .A3(n1359), .A4(n1636), .ZN(n1633)
          );
    AOI21D0BWP7T U731 ( .A1(n197), .A2(n1637), .B(n682), .ZN(n1636) );
    AOI31D0BWP7T U732 ( .A1(n1247), .A2(n410), .A3(n1638), .B(n1321), .ZN(n1635)
          );
    OAI222D0BWP7T U733 ( .A1(n413), .A2(n149), .B1(n1112), .B2(n1017), .C1(n1296)
        , .C2(n759), .ZN(n1634) );
    AOI211D0BWP7T U734 ( .A1(n349), .A2(n351), .B(n1639), .C(n1640), .ZN(n1632)
          );
    AN4D0BWP7T U735 ( .A1(n1641), .A2(n420), .A3(n1642), .A4(n1643), .Z(n522) );
    AOI221D0BWP7T U736 ( .A1(n298), .A2(n733), .B1(n529), .B2(n805), .C(n1644),
         .ZN(n1643) );
    CKND2D0BWP7T U737 ( .A1(n148), .A2(n1113), .ZN(n733) );
    OAI21D0BWP7T U738 ( .A1(n841), .A2(n479), .B(n352), .ZN(n1642) );
    AN2D0BWP7T U739 ( .A1(n1645), .A2(n1646), .Z(n663) );
    NR4D0BWP7T U740 ( .A1(n1647), .A2(n1648), .A3(n1649), .A4(n1650), .ZN(n1646)
          );
    OAI222D0BWP7T U741 ( .A1(n199), .A2(n1114), .B1(n1651), .B2(n1137), .C1(n231)
        , .C2(n759), .ZN(n1647) );
    NR4D0BWP7T U742 ( .A1(n1652), .A2(n363), .A3(n1653), .A4(n1654), .ZN(n1645)
          );
    IND3D0BWP7T U743 ( .A1(n1655), .B1(n1656), .B2(n1657), .ZN(n363) );
    AOI21D0BWP7T U744 ( .A1(n477), .A2(n357), .B(n1658), .ZN(n1657) );
    AOI21D0BWP7T U745 ( .A1(n694), .A2(n1659), .B(n802), .ZN(n1658) );
    ND3D0BWP7T U746 ( .A1(n1660), .A2(n1661), .A3(n1662), .ZN(n1652) );
    OAI221D0BWP7T U747 ( .A1(n1663), .A2(n1321), .B1(n1280), .B2(n1113), .C(
         n1664), .ZN(n1022) );
    AOI22D0BWP7T U748 ( .A1(n355), .A2(n339), .B1(n348), .B2(n372), .ZN(n1664)
          );
    AOI211D0BWP7T U749 ( .A1(n696), .A2(n701), .B(n1665), .C(n1666), .ZN(n1620)
          );
    AOI21D0BWP7T U750 ( .A1(n728), .A2(n148), .B(n1659), .ZN(n1666) );
    ND3D0BWP7T U751 ( .A1(n698), .A2(n1243), .A3(n1298), .ZN(n1665) );
    IINR4D0BWP7T U752 ( .A1(n1667), .A2(n1668), .B1(n1669), .B2(n1670), .ZN(
         n1619) );
    OAI22D0BWP7T U753 ( .A1(n759), .A2(n927), .B1(n1017), .B2(n1671), .ZN(n1669)
          );
    ND4D0BWP7T U754 ( .A1(n1672), .A2(n1673), .A3(n1674), .A4(n1675), .ZN(n1001)
          );
    AOI211D0BWP7T U755 ( .A1(n584), .A2(n1676), .B(n645), .C(n1677), .ZN(n1675)
          );
    AOI222D0BWP7T U756 ( .A1(n829), .A2(n205), .B1(n1678), .B2(n1445), .C1(n1679)
        , .C2(n828), .ZN(n1674) );
    AOI222D0BWP7T U757 ( .A1(n1680), .A2(n482), .B1(n1461), .B2(n601), .C1(n1681)
        , .C2(n693), .ZN(n1673) );
    AOI221D0BWP7T U758 ( .A1(n459), .A2(n259), .B1(n1682), .B2(n1683), .C(n666),
```

```
      .ZN(n1672) );
    OAI221D0BWP7T U759 ( .A1(n618), .A2(n1050), .B1(n196), .B2(n846), .C(n1684),
      .ZN(n666) );
    NR4D0BWP7T U760 ( .A1(n1685), .A2(n1686), .A3(n1687), .A4(n1688), .ZN(n1590)
      );
    AOI21D0BWP7T U761 ( .A1(n224), .A2(n724), .B(n509), .ZN(n1688) );
    AOI21D0BWP7T U762 ( .A1(n512), .A2(n432), .B(n635), .ZN(n1687) );
    OAI22D0BWP7T U763 ( .A1(n327), .A2(n1689), .B1(n1690), .B2(n231), .ZN(n1686)
      );
    OAI221D0BWP7T U764 ( .A1(n319), .A2(n35), .B1(n1241), .B2(n752), .C(n1691),
      .ZN(n1685) );
    AOI22D0BWP7T U765 ( .A1(n349), .A2(n297), .B1(n348), .B2(n374), .ZN(n1691)
      );
    INR4D0BWP7T U766 ( .A1(n699), .B1(n1124), .B2(n467), .B3(n1692), .ZN(n1589)
      );
    NR2D0BWP7T U767 ( .A1(n187), .A2(n49), .ZN(n467) );
    IINR4D0BWP7T U768 ( .A1(n1127), .A2(n462), .B1(n1693), .B2(n852), .ZN(n1588)
      );
    ND4D0BWP7T U769 ( .A1(n1694), .A2(n1695), .A3(n1696), .A4(n1697), .ZN(n115)
      );
    NR4D0BWP7T U770 ( .A1(n1698), .A2(n1699), .A3(n1700), .A4(n1701), .ZN(n1697)
      );
    OAI22D0BWP7T U771 ( .A1(n1702), .A2(n319), .B1(n1703), .B2(n347), .ZN(n1701)
      );
    MOAI22D0BWP7T U772 ( .A1(n1073), .A2(n879), .B1(n1704), .B2(n216), .ZN(n1700)
     );
    OAI222D0BWP7T U773 ( .A1(n226), .A2(n772), .B1(n318), .B2(n889), .C1(n1086),
      .C2(n1255), .ZN(n1699) );
    IND4D0BWP7T U774 ( .A1(n1465), .B1(n1705), .B2(n1706), .B3(n1707), .ZN(n1698)
     );
    AOI211D0BWP7T U775 ( .A1(n587), .A2(n1708), .B(n1709), .C(n1710), .ZN(n1707)
      );
    AOI21D0BWP7T U776 ( .A1(n880), .A2(n313), .B(n222), .ZN(n1710) );
    AOI21D0BWP7T U777 ( .A1(n894), .A2(n1391), .B(n150), .ZN(n1709) );
    NR3D0BWP7T U778 ( .A1(n183), .A2(n740), .A3(n632), .ZN(n894) );
    AOI21D0BWP7T U779 ( .A1(n727), .A2(n633), .B(n1711), .ZN(n1706) );
    OAI21D0BWP7T U780 ( .A1(n30), .A2(n1102), .B(n886), .ZN(n1705) );
    INR4D0BWP7T U781 ( .A1(n1712), .B1(n1713), .B2(n1714), .B3(n1715), .ZN(n1696)
     );
    OAI211D0BWP7T U782 ( .A1(n221), .A2(n1074), .B(n1716), .C(n1717), .ZN(n1713)
      );
    AOI22D0BWP7T U783 ( .A1(n32), .A2(n34), .B1(n31), .B2(n29), .ZN(n1717) );
    OAI21D0BWP7T U784 ( .A1(n1101), .A2(n1140), .B(n640), .ZN(n1716) );
    NR3D0BWP7T U785 ( .A1(n1718), .A2(n1719), .A3(n1720), .ZN(n1695) );
    NR3D0BWP7T U786 ( .A1(n1162), .A2(n1064), .A3(n1002), .ZN(n1694) );
    ND4D0BWP7T U787 ( .A1(n1721), .A2(n1722), .A3(n1723), .A4(n1724), .ZN(n1002)
      );
    AOI211D0BWP7T U788 ( .A1(n1095), .A2(n280), .B(n1725), .C(n1726), .ZN(n1724)
      );
    OAI22D0BWP7T U789 ( .A1(n1072), .A2(n313), .B1(n1086), .B2(n874), .ZN(n1725)
      );
    AOI22D0BWP7T U790 ( .A1(n30), .A2(n1727), .B1(n588), .B2(n322), .ZN(n1723)
      );
    CKND2D0BWP7T U791 ( .A1(n1528), .A2(n510), .ZN(n1727) );
```

```
    OAI21D0BWP7T U792 ( .A1(n1171), .A2(n1140), .B(n34), .ZN(n1722) );
    OAI21D0BWP7T U793 ( .A1(n638), .A2(n1076), .B(n631), .ZN(n1721) );
    OAI211D0BWP7T U794 ( .A1(n1728), .A2(n1488), .B(n1729), .C(n1730), .ZN(n1064)
        );
    AOI222D0BWP7T U795 ( .A1(n1095), .A2(n588), .B1(n216), .B2(n877), .C1(n453),
         .C2(n1731), .ZN(n1730) );
    CKND0BWP7T U796 ( .I(n1732), .ZN(n1731) );
    MAOI22D0BWP7T U797 ( .A1(n630), .A2(n1733), .B1(n309), .B2(n1734), .ZN(n1729)
        );
    ND3D0BWP7T U798 ( .A1(n347), .A2(n345), .A3(n326), .ZN(n1733) );
    ND4D0BWP7T U799 ( .A1(n1735), .A2(n1736), .A3(n1737), .A4(n1738), .ZN(n1162)
          );
    INR4D0BWP7T U800 ( .A1(n1739), .B1(n1740), .B2(n1741), .B3(n1742), .ZN(n1738)
        );
    ND3D0BWP7T U801 ( .A1(n1743), .A2(n1744), .A3(n1745), .ZN(n1740) );
    AOI22D0BWP7T U802 ( .A1(n518), .A2(n268), .B1(n630), .B2(n484), .ZN(n1745)
          );
    AOI211D0BWP7T U803 ( .A1(n587), .A2(n1746), .B(n1747), .C(n1748), .ZN(n1737)
          );
    AOI31D0BWP7T U804 ( .A1(n311), .A2(n879), .A3(n1749), .B(n313), .ZN(n1748)
          );
    CKND2D0BWP7T U805 ( .A1(n1750), .A2(n1751), .ZN(n1747) );
    OAI21D0BWP7T U806 ( .A1(n882), .A2(n29), .B(n322), .ZN(n1751) );
    OAI21D0BWP7T U807 ( .A1(n183), .A2(n33), .B(n485), .ZN(n1750) );
    AOI22D0BWP7T U808 ( .A1(n1078), .A2(n1485), .B1(n453), .B2(n608), .ZN(n1736)
          );
    CKND0BWP7T U809 ( .I(n1752), .ZN(n1485) );
    AOI22D0BWP7T U810 ( .A1(n183), .A2(n216), .B1(n1534), .B2(n316), .ZN(n1735)
          );
    ND4D0BWP7T U811 ( .A1(n1753), .A2(n1754), .A3(n1755), .A4(n1756), .ZN(n116)
          );
    INR4D0BWP7T U812 ( .A1(n1757), .B1(n1758), .B2(n1759), .B3(n1760), .ZN(n1756)
        );
    IND3D0BWP7T U813 ( .A1(n1761), .B1(n1762), .B2(n1763), .ZN(n1759) );
    IND3D0BWP7T U814 ( .A1(n1161), .B1(n1065), .B2(n1764), .ZN(n1758) );
    NR3D0BWP7T U815 ( .A1(n1765), .A2(n1766), .A3(n1767), .ZN(n1764) );
    AN2D0BWP7T U816 ( .A1(n1768), .A2(n1769), .Z(n1065) );
    AOI211D0BWP7T U817 ( .A1(n599), .A2(n1309), .B(n1770), .C(n1771), .ZN(n1769)
          );
    AOI21D0BWP7T U818 ( .A1(n203), .A2(n1772), .B(n1350), .ZN(n1771) );
    AOI21D0BWP7T U819 ( .A1(n614), .A2(n475), .B(n1118), .ZN(n1770) );
    AOI211D0BWP7T U820 ( .A1(n1773), .A2(n460), .B(n1774), .C(n1775), .ZN(n1768)
          );
    ND4D0BWP7T U821 ( .A1(n1776), .A2(n1777), .A3(n1778), .A4(n1779), .ZN(n1161)
          );
    INR4D0BWP7T U822 ( .A1(n1780), .B1(n1781), .B2(n1782), .B3(n1783), .ZN(n1779)
        );
    OAI22D0BWP7T U823 ( .A1(n227), .A2(n1449), .B1(n35), .B2(n1784), .ZN(n1782)
          );
    IND3D0BWP7T U824 ( .A1(n1785), .B1(n1786), .B2(n1787), .ZN(n1781) );
    AOI222D0BWP7T U825 ( .A1(n1788), .A2(n294), .B1(n288), .B2(n1443), .C1(n580),
         .C2(n293), .ZN(n1778) );
    AOI22D0BWP7T U826 ( .A1(n438), .A2(n1789), .B1(n1790), .B2(n454), .ZN(n1777)
          );
```

```
    CKND2D0BWP7T U827 ( .A1(n614), .A2(n865), .ZN(n454) );
    AOI22D0BWP7T U828 ( .A1(n574), .A2(n1791), .B1(n1792), .B2(n1579), .ZN(n1776)
        );
    NR4D0BWP7T U829 ( .A1(n1793), .A2(n1794), .A3(n1795), .A4(n1796), .ZN(n1755)
          );
    CKND2D0BWP7T U830 ( .A1(n1797), .A2(n1798), .ZN(n1794) );
    AOI211D0BWP7T U831 ( .A1(n1799), .A2(n1800), .B(n1801), .C(n1802), .ZN(n1754)
        );
    CKND0BWP7T U832 ( .I(n1803), .ZN(n1800) );
    AOI22D0BWP7T U833 ( .A1(n1381), .A2(n715), .B1(n585), .B2(n1804), .ZN(n1753)
          );
    IND4D0BWP7T U834 ( .A1(n136), .B1(n511), .B2(n742), .B3(n1805), .ZN(n1477)
          );
    INR4D0BWP7T U835 ( .A1(n1138), .B1(n1806), .B2(n1807), .B3(n1071), .ZN(n1805)
        );
    OAI222D0BWP7T U836 ( .A1(n404), .A2(n311), .B1(n346), .B2(n1099), .C1(n1086),
         .C2(n224), .ZN(n1071) );
    AOI21D0BWP7T U837 ( .A1(n1490), .A2(n608), .B(n501), .ZN(n1138) );
    OAI21D0BWP7T U838 ( .A1(n509), .A2(n892), .B(n1808), .ZN(n501) );
    AN3D0BWP7T U839 ( .A1(n1809), .A2(n1810), .A3(n1811), .Z(n742) );
    NR4D0BWP7T U840 ( .A1(n1812), .A2(n1813), .A3(n1246), .A4(n1814), .ZN(n1811)
          );
    AOI21D0BWP7T U841 ( .A1(n1815), .A2(n843), .B(n475), .ZN(n1814) );
    OAI22D0BWP7T U842 ( .A1(n199), .A2(n412), .B1(n1816), .B2(n846), .ZN(n1813)
          );
    OAI221D0BWP7T U843 ( .A1(n1817), .A2(n910), .B1(n618), .B2(n752), .C(n1818),
         .ZN(n1812) );
    AOI22D0BWP7T U844 ( .A1(n531), .A2(n1683), .B1(n1678), .B2(n269), .ZN(n1818)
          );
    CKND0BWP7T U845 ( .I(n1819), .ZN(n531) );
    AOI211D0BWP7T U846 ( .A1(n718), .A2(n460), .B(n1820), .C(n1821), .ZN(n1810)
          );
    OAI22D0BWP7T U847 ( .A1(n327), .A2(n837), .B1(n435), .B2(n1822), .ZN(n1820)
          );
    NR4D0BWP7T U848 ( .A1(n1823), .A2(n1824), .A3(n1825), .A4(n1826), .ZN(n1809)
          );
    AN4D0BWP7T U849 ( .A1(n1827), .A2(n1828), .A3(n1829), .A4(n1830), .Z(n511)
          );
    IINR4D0BWP7T U850 ( .A1(n331), .A2(n541), .B1(n1831), .B2(n1832), .ZN(n1830)
          );
    AO21D0BWP7T U851 ( .A1(n588), .A2(n316), .B(n1833), .Z(n1832) );
    OAI221D0BWP7T U852 ( .A1(n875), .A2(n320), .B1(n1834), .B2(n1488), .C(n1835),
         .ZN(n1831) );
    AOI22D0BWP7T U853 ( .A1(n630), .A2(n882), .B1(n1095), .B2(n242), .ZN(n1835)
          );
    AOI222D0BWP7T U854 ( .A1(n485), .A2(n740), .B1(n34), .B2(n324), .C1(n693),
         .C2(n518), .ZN(n541) );
    AOI22D0BWP7T U855 ( .A1(n588), .A2(n608), .B1(n638), .B2(n727), .ZN(n331) );
    NR4D0BWP7T U856 ( .A1(n1836), .A2(n1837), .A3(n1838), .A4(n1839), .ZN(n1829)
          );
    AOI21D0BWP7T U857 ( .A1(n347), .A2(n406), .B(n313), .ZN(n1839) );
    AOI21D0BWP7T U858 ( .A1(n226), .A2(n221), .B(n312), .ZN(n1838) );
    AOI31D0BWP7T U859 ( .A1(n510), .A2(n1170), .A3(n345), .B(n234), .ZN(n1837)
          );
```

```
    OAI22D0BWP7T U860 ( .A1(n233), .A2(n346), .B1(n1840), .B2(n1529), .ZN(n1836)
          );
    AOI22D0BWP7T U861 ( .A1(n631), .A2(n281), .B1(n183), .B2(n638), .ZN(n1828)
          );
    AOI22D0BWP7T U862 ( .A1(n31), .A2(n215), .B1(n774), .B2(n483), .ZN(n1827) );
    ND4D0BWP7T U863 ( .A1(n1841), .A2(n1842), .A3(n1843), .A4(n1844), .ZN(n136)
          );
    NR4D0BWP7T U864 ( .A1(n1845), .A2(n1846), .A3(n1847), .A4(n1848), .ZN(n1844)
          );
    OAI222D0BWP7T U865 ( .A1(n432), .A2(n752), .B1(n618), .B2(n1849), .C1(n229),
         .C2(n1850), .ZN(n1848) );
    AO222D0BWP7T U866 ( .A1(n482), .A2(n1618), .B1(n1851), .B2(n718), .C1(n1683),
         .C2(n1852), .Z(n1847) );
    OAI222D0BWP7T U867 ( .A1(n928), .A2(n186), .B1(n1257), .B2(n1822), .C1(n1853)
        , .C2(n1239), .ZN(n1846) );
    CKND0BWP7T U868 ( .I(n514), .ZN(n1239) );
    IND4D0BWP7T U869 ( .A1(n595), .B1(n1854), .B2(n1855), .B3(n1856), .ZN(n1845)
          );
    OAI21D0BWP7T U870 ( .A1(n296), .A2(n269), .B(n210), .ZN(n1856) );
    OAI21D0BWP7T U871 ( .A1(n1857), .A2(n1063), .B(n828), .ZN(n1855) );
    OAI31D0BWP7T U872 ( .A1(n1858), .A2(n615), .A3(n1859), .B(n583), .ZN(n1854)
          );
    NR4D0BWP7T U873 ( .A1(n1860), .A2(n1861), .A3(n1862), .A4(n991), .ZN(n1843)
          );
    AO221D0BWP7T U874 ( .A1(n259), .A2(n604), .B1(n481), .B2(n293), .C(n1863),
         .Z(n1860) );
    NR3D0BWP7T U875 ( .A1(n1027), .A2(n1864), .A3(n1865), .ZN(n1842) );
    ND4D0BWP7T U876 ( .A1(n1866), .A2(n1867), .A3(n1868), .A4(n1869), .ZN(n1027)
          );
    NR4D0BWP7T U877 ( .A1(n1870), .A2(n1871), .A3(n1872), .A4(n1873), .ZN(n1869)
          );
    AOI21D0BWP7T U878 ( .A1(n1596), .A2(n738), .B(n196), .ZN(n1873) );
    AOI21D0BWP7T U879 ( .A1(n1218), .A2(n1817), .B(n318), .ZN(n1872) );
    NR2D0BWP7T U880 ( .A1(n1859), .A2(n1198), .ZN(n1218) );
    AOI21D0BWP7T U881 ( .A1(n1874), .A2(n635), .B(n614), .ZN(n1871) );
    OAI222D0BWP7T U882 ( .A1(n329), .A2(n1603), .B1(n618), .B2(n749), .C1(n738),
         .C2(n820), .ZN(n1870) );
    AOI222D0BWP7T U883 ( .A1(n1199), .A2(n517), .B1(n1223), .B2(n1565), .C1(n844)
        , .C2(n601), .ZN(n1868) );
    AOI22D0BWP7T U884 ( .A1(n1200), .A2(n259), .B1(n459), .B2(n282), .ZN(n1866)
          );
    NR3D0BWP7T U885 ( .A1(n508), .A2(n734), .A3(n1180), .ZN(n1841) );
    ND4D0BWP7T U886 ( .A1(n1875), .A2(n1876), .A3(n1877), .A4(n1878), .ZN(n1180)
          );
    INR4D0BWP7T U887 ( .A1(n1879), .B1(n1880), .B2(n1881), .B3(n1882), .ZN(n1878)
         );
    IND3D0BWP7T U888 ( .A1(n1883), .B1(n1884), .B2(n1885), .ZN(n1881) );
    OAI221D0BWP7T U889 ( .A1(n432), .A2(n437), .B1(n49), .B2(n1815), .C(n1886),
         .ZN(n1880) );
    AOI222D0BWP7T U890 ( .A1(n459), .A2(n294), .B1(n1887), .B2(n1683), .C1(n714),
         .C2(n427), .ZN(n1886) );
    NR4D0BWP7T U891 ( .A1(n1888), .A2(n1889), .A3(n1890), .A4(n1891), .ZN(n1877)
          );
    AOI21D0BWP7T U892 ( .A1(n1819), .A2(n1892), .B(n327), .ZN(n1891) );
```

```
    OA21D0BWP7T U893 ( .A1(n1893), .A2(n1894), .B(n517), .Z(n1890) );
    OA31D0BWP7T U894 ( .A1(n481), .A2(n1681), .A3(n459), .B(n268), .Z(n1889) );
    OAI22D0BWP7T U895 ( .A1(n1308), .A2(n635), .B1(n1895), .B2(n435), .ZN(n1888)
          );
    NR2D0BWP7T U896 ( .A1(n583), .A2(n427), .ZN(n1308) );
    AOI222D0BWP7T U897 ( .A1(n1680), .A2(n1896), .B1(n844), .B2(n1897), .C1(n615)
        , .C2(n1559), .ZN(n1876) );
    CKND0BWP7T U898 ( .I(n818), .ZN(n1559) );
    CKND0BWP7T U899 ( .I(n839), .ZN(n1896) );
    AOI22D0BWP7T U900 ( .A1(n266), .A2(n518), .B1(n1898), .B2(n515), .ZN(n1875)
          );
    OAI211D0BWP7T U901 ( .A1(n820), .A2(n817), .B(n1899), .C(n1900), .ZN(n734)
          );
    AOI222D0BWP7T U902 ( .A1(n1079), .A2(n1193), .B1(n1198), .B2(n1309), .C1(
         n1894), .C2(n515), .ZN(n1900) );
    ND4D0BWP7T U903 ( .A1(n1901), .A2(n1902), .A3(n1903), .A4(n1904), .ZN(n508)
          );
    INR4D0BWP7T U904 ( .A1(n271), .B1(n1905), .B2(n1906), .B3(n1907), .ZN(n1904)
          );
    AOI21D0BWP7T U905 ( .A1(n475), .A2(n408), .B(n409), .ZN(n1905) );
    AOI221D0BWP7T U906 ( .A1(n43), .A2(n438), .B1(n210), .B2(n693), .C(n1908),
         .ZN(n1903) );
    AOI222D0BWP7T U907 ( .A1(n515), .A2(n1909), .B1(n639), .B2(n1791), .C1(n615),
         .C2(n1309), .ZN(n1902) );
    CKND2D0BWP7T U908 ( .A1(n49), .A2(n432), .ZN(n1309) );
    CKND0BWP7T U909 ( .I(n1119), .ZN(n1791) );
    NR2D0BWP7T U910 ( .A1(n285), .A2(n482), .ZN(n1119) );
    OAI21D0BWP7T U911 ( .A1(n1910), .A2(n475), .B(n1197), .ZN(n1909) );
    NR3D0BWP7T U912 ( .A1(n1911), .A2(n1912), .A3(n1913), .ZN(n1901) );
    ND4D0BWP7T U913 ( .A1(n1914), .A2(n1915), .A3(n1916), .A4(n1917), .ZN(n1476)
          );
    AOI222D0BWP7T U914 ( .A1(n295), .A2(n460), .B1(n263), .B2(n285), .C1(n284),
         .C2(n438), .ZN(n1917) );
    CKND0BWP7T U915 ( .I(n458), .ZN(n284) );
    ND4D0BWP7T U916 ( .A1(n1918), .A2(n1919), .A3(n1920), .A4(n1921), .ZN(n1475)
          );
    INR4D0BWP7T U917 ( .A1(n934), .B1(n1922), .B2(n1923), .B3(n1924), .ZN(n1921)
          );
    AOI21D0BWP7T U918 ( .A1(n1440), .A2(n227), .B(n36), .ZN(n1924) );
    AOI21D0BWP7T U919 ( .A1(n1116), .A2(n966), .B(n911), .ZN(n1923) );
    OAI21D0BWP7T U920 ( .A1(n261), .A2(n969), .B(n693), .ZN(n1920) );
    OAI21D0BWP7T U921 ( .A1(n1925), .A2(n1579), .B(n603), .ZN(n1919) );
    CKND0BWP7T U922 ( .I(n235), .ZN(n1579) );
    NR2D0BWP7T U923 ( .A1(n293), .A2(n285), .ZN(n235) );
    OAI21D0BWP7T U924 ( .A1(n265), .A2(n1926), .B(n296), .ZN(n1918) );
    NR3D0BWP7T U925 ( .A1(n1927), .A2(n1928), .A3(n1929), .ZN(n1473) );
    ND3D0BWP7T U926 ( .A1(n943), .A2(n1930), .A3(n550), .ZN(n1929) );
    CKND0BWP7T U927 ( .I(n1931), .ZN(n550) );
    IND4D0BWP7T U928 ( .A1(n1932), .B1(n1933), .B2(n1934), .B3(n1935), .ZN(n1931)
        );
    AOI221D0BWP7T U929 ( .A1(n337), .A2(n696), .B1(n853), .B2(n764), .C(n1936),
         .ZN(n1935) );
    AO221D0BWP7T U930 ( .A1(n1937), .A2(n339), .B1(n338), .B2(n1938), .C(n275),
         .Z(n1936) );
```

```
    AOI222D0BWP7T U931 ( .A1(n1198), .A2(n438), .B1(n763), .B2(n355), .C1(n639),
         .C2(n296), .ZN(n1934) );
    AOI22D0BWP7T U932 ( .A1(n638), .A2(n632), .B1(n356), .B2(n1129), .ZN(n943)
         );
    IND4D0BWP7T U933 ( .A1(n1939), .B1(n1940), .B2(n1941), .B3(n1942), .ZN(n1928)
        );
    IND4D0BWP7T U934 ( .A1(n1943), .B1(n1944), .B2(n26), .B3(n1945), .ZN(n1927)
         );
    AOI211D0BWP7T U935 ( .A1(n631), .A2(n608), .B(n1946), .C(n1947), .ZN(n1945)
         );
    CKND2D0BWP7T U936 ( .A1(n426), .A2(n425), .ZN(n1946) );
    CKND0BWP7T U937 ( .I(n1948), .ZN(n426) );
    OA211D0BWP7T U938 ( .A1(n1949), .A2(n1950), .B(n53), .C(n1951), .Z(n1) );
    CKND0BWP7T U939 ( .I(n51), .ZN(n1951) );
    ND4D0BWP7T U940 ( .A1(n1952), .A2(n1953), .A3(n1954), .A4(n1955), .ZN(n51)
         );
    NR4D0BWP7T U941 ( .A1(n1956), .A2(n1957), .A3(n915), .A4(n991), .ZN(n1955)
         );
    OAI22D0BWP7T U942 ( .A1(n614), .A2(n194), .B1(n435), .B2(n434), .ZN(n991) );
    OAI222D0BWP7T U943 ( .A1(n1637), .A2(n1017), .B1(n35), .B2(n1784), .C1(n880),
         .C2(n510), .ZN(n915) );
    ND3D0BWP7T U944 ( .A1(n1297), .A2(n1382), .A3(n247), .ZN(n1957) );
    AOI22D0BWP7T U945 ( .A1(n30), .A2(n215), .B1(n515), .B2(n1616), .ZN(n247) );
    AOI222D0BWP7T U946 ( .A1(n322), .A2(n588), .B1(n485), .B2(n727), .C1(n438),
         .C2(n717), .ZN(n1382) );
    AOI22D0BWP7T U947 ( .A1(n696), .A2(n829), .B1(n204), .B2(n350), .ZN(n1297)
         );
    OAI211D0BWP7T U948 ( .A1(n234), .A2(n1099), .B(n1532), .C(n1958), .ZN(n1956)
         );
    INR3D0BWP7T U949 ( .A1(n1298), .B1(n595), .B2(n275), .ZN(n1958) );
    CKND0BWP7T U950 ( .I(n1641), .ZN(n275) );
    CKND2D0BWP7T U951 ( .A1(n351), .A2(n299), .ZN(n1641) );
    NR2D0BWP7T U952 ( .A1(n1959), .A2(n229), .ZN(n595) );
    CKND2D0BWP7T U953 ( .A1(n339), .A2(n357), .ZN(n1298) );
    CKND2D0BWP7T U954 ( .A1(n727), .A2(n483), .ZN(n1532) );
    NR4D0BWP7T U955 ( .A1(n560), .A2(n1275), .A3(n388), .A4(n1823), .ZN(n1954)
         );
    OAI211D0BWP7T U956 ( .A1(n48), .A2(n227), .B(n913), .C(n1960), .ZN(n1823) );
    AOI21D0BWP7T U957 ( .A1(n639), .A2(n296), .B(n565), .ZN(n1960) );
    OAI22D0BWP7T U958 ( .A1(n526), .A2(n49), .B1(n824), .B2(n1850), .ZN(n565) );
    CKND2D0BWP7T U959 ( .A1(n515), .A2(n1961), .ZN(n913) );
    OAI221D0BWP7T U960 ( .A1(n319), .A2(n432), .B1(n724), .B2(n221), .C(n1962),
         .ZN(n388) );
    AOI222D0BWP7T U961 ( .A1(n30), .A2(n773), .B1(n34), .B2(n1963), .C1(n882),
         .C2(n322), .ZN(n1962) );
    CKND2D0BWP7T U962 ( .A1(n510), .A2(n1134), .ZN(n1963) );
    OAI211D0BWP7T U963 ( .A1(n1446), .A2(n49), .B(n698), .C(n1964), .ZN(n1275)
         );
    AOI21D0BWP7T U964 ( .A1(n1680), .A2(n266), .B(n243), .ZN(n1964) );
    OAI211D0BWP7T U965 ( .A1(n1822), .A2(n35), .B(n1965), .C(n1966), .ZN(n243)
         );
    AOI222D0BWP7T U966 ( .A1(n1198), .A2(n482), .B1(n1223), .B2(n427), .C1(n460),
         .C2(n1200), .ZN(n1966) );
    AOI22D0BWP7T U967 ( .A1(n1967), .A2(n269), .B1(n285), .B2(n1005), .ZN(n1965)
```

```
      );
    CKND2D0BWP7T U968 ( .A1(n479), .A2(n351), .ZN(n698) );
    NR2D0BWP7T U969 ( .A1(n1381), .A2(n650), .ZN(n1446) );
    OAI221D0BWP7T U970 ( .A1(n1137), .A2(n146), .B1(n197), .B2(n199), .C(n1968),
         .ZN(n560) );
    OAI21D0BWP7T U971 ( .A1(n349), .A2(n529), .B(n351), .ZN(n1968) );
    NR3D0BWP7T U972 ( .A1(n1322), .A2(n1421), .A3(n942), .ZN(n1953) );
    OAI221D0BWP7T U973 ( .A1(n724), .A2(n880), .B1(n221), .B2(n222), .C(n1969),
         .ZN(n942) );
    AOI211D0BWP7T U974 ( .A1(n1090), .A2(n30), .B(n1692), .C(n1970), .ZN(n1969)
         );
    AOI21D0BWP7T U975 ( .A1(n1971), .A2(n1972), .B(n327), .ZN(n1970) );
    AOI21D0BWP7T U976 ( .A1(n197), .A2(n1454), .B(n148), .ZN(n1692) );
    OAI221D0BWP7T U977 ( .A1(n225), .A2(n226), .B1(n1074), .B2(n223), .C(n1973),
         .ZN(n1421) );
    AOI222D0BWP7T U978 ( .A1(n886), .A2(n316), .B1(n1534), .B2(n633), .C1(n43),
         .C2(n438), .ZN(n1973) );
    ND4D0BWP7T U979 ( .A1(n1974), .A2(n1975), .A3(n1976), .A4(n1977), .ZN(n1322)
         );
    AOI32D0BWP7T U980 ( .A1(n268), .A2(n755), .A3(n1978), .B1(n324), .B2(n1076),
         .ZN(n1977) );
    CKND2D0BWP7T U981 ( .A1(n509), .A2(n875), .ZN(n1076) );
    MAOI22D0BWP7T U982 ( .A1(n686), .A2(n1040), .B1(n404), .B2(n1979), .ZN(n1976)
        );
    AOI22D0BWP7T U983 ( .A1(n355), .A2(n299), .B1(n31), .B2(n183), .ZN(n1975) );
    AOI21D0BWP7T U984 ( .A1(n1055), .A2(n693), .B(n977), .ZN(n1974) );
    OAI221D0BWP7T U985 ( .A1(n728), .A2(n1980), .B1(n1073), .B2(n1528), .C(n1981)
       , .ZN(n977) );
    AOI222D0BWP7T U986 ( .A1(n353), .A2(n355), .B1(n453), .B2(n633), .C1(n478),
         .C2(n357), .ZN(n1981) );
    NR2D0BWP7T U987 ( .A1(n1943), .A2(n955), .ZN(n1952) );
    OAI211D0BWP7T U988 ( .A1(n512), .A2(n437), .B(n1982), .C(n1983), .ZN(n955)
         );
    AOI222D0BWP7T U989 ( .A1(n1490), .A2(n633), .B1(n316), .B2(n1078), .C1(n1534)
       , .C2(n483), .ZN(n1983) );
    CKND0BWP7T U990 ( .I(n1947), .ZN(n1982) );
    ND4D0BWP7T U991 ( .A1(n1984), .A2(n1985), .A3(n1986), .A4(n1987), .ZN(n1947)
         );
    NR4D0BWP7T U992 ( .A1(n1988), .A2(n1989), .A3(n1990), .A4(n1991), .ZN(n1987)
         );
    OAI222D0BWP7T U993 ( .A1(n347), .A2(n875), .B1(n197), .B2(n232), .C1(n435),
         .C2(n319), .ZN(n1991) );
    OAI222D0BWP7T U994 ( .A1(n1112), .A2(n146), .B1(n618), .B2(n431), .C1(n1296),
         .C2(n926), .ZN(n1990) );
    OAI221D0BWP7T U995 ( .A1(n1615), .A2(n475), .B1(n1752), .B2(n407), .C(n1992),
         .ZN(n1989) );
    OAI21D0BWP7T U996 ( .A1(n599), .A2(n604), .B(n601), .ZN(n1992) );
    NR2D0BWP7T U997 ( .A1(n718), .A2(n1005), .ZN(n1615) );
    IIND4D0BWP7T U998 ( .A1(n645), .A2(n863), .B1(n1993), .B2(n1360), .ZN(n1988)
         );
    CKND0BWP7T U999 ( .I(n1802), .ZN(n1360) );
    NR2D0BWP7T U1000 ( .A1(n1242), .A2(n432), .ZN(n1802) );
    OAI21D0BWP7T U1001 ( .A1(n686), .A2(n1185), .B(n372), .ZN(n1993) );
    CKND0BWP7T U1002 ( .I(n156), .ZN(n1185) );
```

```
    NR2D0BWP7T U1003 ( .A1(n153), .A2(n35), .ZN(n863) );
    NR2D0BWP7T U1004 ( .A1(n410), .A2(n803), .ZN(n645) );
    INR4D0BWP7T U1005 ( .A1(n1273), .B1(n1994), .B2(n1202), .B3(n1863), .ZN(
         n1986) );
    OAI221D0BWP7T U1006 ( .A1(n327), .A2(n330), .B1(n910), .B2(n1596), .C(n1430),
         .ZN(n1863) );
    AOI22D0BWP7T U1007 ( .A1(n282), .A2(n1200), .B1(n285), .B2(n714), .ZN(n1430)
          );
    OAI21D0BWP7T U1008 ( .A1(n199), .A2(n237), .B(n1429), .ZN(n1202) );
    AOI222D0BWP7T U1009 ( .A1(n515), .A2(n1893), .B1(n355), .B2(n687), .C1(n266),
         .C2(n1995), .ZN(n1429) );
    ND3D0BWP7T U1010 ( .A1(n1425), .A2(n1426), .A3(n144), .ZN(n1994) );
    AOI22D0BWP7T U1011 ( .A1(n482), .A2(n692), .B1(n282), .B2(n650), .ZN(n144)
          );
    AOI22D0BWP7T U1012 ( .A1(n351), .A2(n356), .B1(n1040), .B2(n764), .ZN(n1426)
          );
    AOI221D0BWP7T U1013 ( .A1(n1090), .A2(n633), .B1(n485), .B2(n588), .C(n184),
         .ZN(n1425) );
    NR2D0BWP7T U1014 ( .A1(n1073), .A2(n312), .ZN(n184) );
    AOI222D0BWP7T U1015 ( .A1(n1683), .A2(n514), .B1(n1996), .B2(n828), .C1(n583)
        , .C2(n604), .ZN(n1273) );
    AOI221D0BWP7T U1016 ( .A1(n1198), .A2(n269), .B1(n317), .B2(n638), .C(n1997),
         .ZN(n1985) );
    OAI22D0BWP7T U1017 ( .A1(n318), .A2(n1050), .B1(n1116), .B2(n738), .ZN(n1997)
         );
    AOI211D0BWP7T U1018 ( .A1(n587), .A2(n242), .B(n562), .C(n241), .ZN(n1984)
          );
    OAI22D0BWP7T U1019 ( .A1(n475), .A2(n1449), .B1(n432), .B2(n1156), .ZN(n241)
          );
    OAI22D0BWP7T U1020 ( .A1(n1489), .A2(n1087), .B1(n509), .B2(n874), .ZN(n562)
          );
    ND4D0BWP7T U1021 ( .A1(n1998), .A2(n1999), .A3(n2000), .A4(n2001), .ZN(n1943)
         );
    NR4D0BWP7T U1022 ( .A1(n2002), .A2(n2003), .A3(n2004), .A4(n2005), .ZN(n2001)
         );
    OAI22D0BWP7T U1023 ( .A1(n1137), .A2(n926), .B1(n824), .B2(n1197), .ZN(n2005)
         );
    OAI22D0BWP7T U1024 ( .A1(n329), .A2(n1972), .B1(n725), .B2(n509), .ZN(n2004)
          );
    NR2D0BWP7T U1025 ( .A1(n453), .A2(n317), .ZN(n725) );
    OAI222D0BWP7T U1026 ( .A1(n312), .A2(n346), .B1(n1651), .B2(n149), .C1(n327),
         .C2(n1196), .ZN(n2003) );
    ND4D0BWP7T U1027 ( .A1(n2006), .A2(n1243), .A3(n2007), .A4(n2008), .ZN(n2002)
         );
    AN4D0BWP7T U1028 ( .A1(n185), .A2(n160), .A3(n1379), .A4(n1464), .Z(n2008)
          );
    CKND2D0BWP7T U1029 ( .A1(n214), .A2(n30), .ZN(n1464) );
    CKND0BWP7T U1030 ( .I(n466), .ZN(n1379) );
    CKND2D0BWP7T U1031 ( .A1(n599), .A2(n282), .ZN(n160) );
    CKND2D0BWP7T U1032 ( .A1(n1680), .A2(n296), .ZN(n185) );
    OAI21D0BWP7T U1033 ( .A1(n42), .A2(n518), .B(n583), .ZN(n2007) );
    CKND2D0BWP7T U1034 ( .A1(n340), .A2(n854), .ZN(n1243) );
    OAI21D0BWP7T U1035 ( .A1(n1381), .A2(n1198), .B(n259), .ZN(n2006) );
```

```
    NR4D0BWP7T U1036 ( .A1(n2009), .A2(n1203), .A3(n1328), .A4(n1821), .ZN(n2000)
        );
    OAI22D0BWP7T U1037 ( .A1(n526), .A2(n614), .B1(n229), .B2(n1603), .ZN(n1821)
         );
    AO21D0BWP7T U1038 ( .A1(n291), .A2(n1548), .B(n1801), .Z(n1328) );
    NR2D0BWP7T U1039 ( .A1(n1352), .A2(n49), .ZN(n1801) );
    OAI22D0BWP7T U1040 ( .A1(n865), .A2(n635), .B1(n752), .B2(n820), .ZN(n1203)
         );
    OAI221D0BWP7T U1041 ( .A1(n226), .A2(n1072), .B1(n1086), .B2(n2010), .C(
        n2011), .ZN(n2009) );
    AOI22D0BWP7T U1042 ( .A1(n459), .A2(n427), .B1(n718), .B2(n268), .ZN(n2011)
         );
    INR3D0BWP7T U1043 ( .A1(n564), .B1(n1862), .B2(n1326), .ZN(n1999) );
    OAI211D0BWP7T U1044 ( .A1(n1170), .A2(n346), .B(n1712), .C(n2012), .ZN(n1326)
        );
    AOI22D0BWP7T U1045 ( .A1(n297), .A2(n479), .B1(n374), .B2(n357), .ZN(n2012)
         );
    AOI22D0BWP7T U1046 ( .A1(n316), .A2(n1490), .B1(n633), .B2(n317), .ZN(n1712)
         );
    OAI211D0BWP7T U1047 ( .A1(n432), .A2(n1237), .B(n2013), .C(n2014), .ZN(n1862)
        );
    AOI221D0BWP7T U1048 ( .A1(n293), .A2(n1678), .B1(n1223), .B2(n266), .C(n1327)
       , .ZN(n2014) );
    OAI22D0BWP7T U1049 ( .A1(n143), .A2(n236), .B1(n229), .B2(n2015), .ZN(n1327)
         );
    AOI22D0BWP7T U1050 ( .A1(n1055), .A2(n285), .B1(n1077), .B2(n828), .ZN(n2013)
        );
    AOI221D0BWP7T U1051 ( .A1(n268), .A2(n1370), .B1(n608), .B2(n324), .C(n1465),
        .ZN(n564) );
    NR2D0BWP7T U1052 ( .A1(n405), .A2(n234), .ZN(n1465) );
    INR3D0BWP7T U1053 ( .A1(n1424), .B1(n1423), .B2(n732), .ZN(n1998) );
    OAI221D0BWP7T U1054 ( .A1(n1980), .A2(n1113), .B1(n146), .B2(n1638), .C(
        n2016), .ZN(n732) );
    AOI22D0BWP7T U1055 ( .A1(n205), .A2(n350), .B1(n686), .B2(n358), .ZN(n2016)
         );
    ND3D0BWP7T U1056 ( .A1(n1885), .A2(n238), .A3(n2017), .ZN(n1423) );
    AOI22D0BWP7T U1057 ( .A1(n1857), .A2(n1683), .B1(n2018), .B2(n828), .ZN(
        n2017) );
    AOI22D0BWP7T U1058 ( .A1(n285), .A2(n604), .B1(n266), .B2(n1678), .ZN(n238)
         );
    AOI222D0BWP7T U1059 ( .A1(n482), .A2(n1461), .B1(n1683), .B2(n2019), .C1(
        n460), .C2(n1007), .ZN(n1885) );
    AOI22D0BWP7T U1060 ( .A1(n687), .A2(n340), .B1(n1676), .B2(n298), .ZN(n1424)
         );
    CKND2D0BWP7T U1061 ( .A1(n1321), .A2(n1113), .ZN(n1676) );
    AN4D0BWP7T U1062 ( .A1(n2020), .A2(n2021), .A3(n2022), .A4(n2023), .Z(n53)
         );
    INR4D0BWP7T U1063 ( .A1(n2024), .B1(n2025), .B2(n2026), .B3(n2027), .ZN(
        n2023) );
    IIND4D0BWP7T U1064 ( .A1(n1654), .A2(n1625), .B1(n1491), .B2(n2028), .ZN(
        n2027) );
    NR3D0BWP7T U1065 ( .A1(n173), .A2(n1530), .A3(n1517), .ZN(n2028) );
    IND4D0BWP7T U1066 ( .A1(n1393), .B1(n941), .B2(n2029), .B3(n2030), .ZN(n1517)
        );
```

```
    AOI221D0BWP7T U1067 ( .A1(n216), .A2(n1078), .B1(n212), .B2(n280), .C(n2031),
         .ZN(n2030) );
    OAI211D0BWP7T U1068 ( .A1(n326), .A2(n223), .B(n2032), .C(n425), .ZN(n2031)
         );
    CKND2D0BWP7T U1069 ( .A1(n638), .A2(n882), .ZN(n425) );
    OAI31D0BWP7T U1070 ( .A1(n878), .A2(n886), .A3(n315), .B(n322), .ZN(n2032)
         );
    AOI22D0BWP7T U1071 ( .A1(n33), .A2(n638), .B1(n182), .B2(n316), .ZN(n2029)
         );
    MAOI22D0BWP7T U1072 ( .A1(n34), .A2(n484), .B1(n2033), .B2(n404), .ZN(n941)
         );
    OAI22D0BWP7T U1073 ( .A1(n1099), .A2(n880), .B1(n509), .B2(n406), .ZN(n1393)
         );
    OAI221D0BWP7T U1074 ( .A1(n313), .A2(n325), .B1(n221), .B2(n347), .C(n2034),
         .ZN(n1530) );
    OAI21D0BWP7T U1075 ( .A1(n34), .A2(n630), .B(n215), .ZN(n2034) );
    CKND2D0BWP7T U1076 ( .A1(n2035), .A2(n2036), .ZN(n173) );
    AOI211D0BWP7T U1077 ( .A1(n574), .A2(n460), .B(n2037), .C(n864), .ZN(n2036)
         );
    NR2D0BWP7T U1078 ( .A1(n155), .A2(n432), .ZN(n864) );
    AO21D0BWP7T U1079 ( .A1(n268), .A2(n264), .B(n634), .Z(n2037) );
    OAI211D0BWP7T U1080 ( .A1(n1170), .A2(n1240), .B(n2038), .C(n2039), .ZN(n634)
        );
    AOI211D0BWP7T U1081 ( .A1(n605), .A2(n266), .B(n2040), .C(n2041), .ZN(n2039)
         );
    AOI21D0BWP7T U1082 ( .A1(n2042), .A2(n1352), .B(n196), .ZN(n2041) );
    OAI22D0BWP7T U1083 ( .A1(n408), .A2(n2043), .B1(n475), .B2(n932), .ZN(n2040)
         );
    AOI22D0BWP7T U1084 ( .A1(n2044), .A2(n573), .B1(n1799), .B2(n482), .ZN(n2038)
        );
    NR4D0BWP7T U1085 ( .A1(n707), .A2(n341), .A3(n1157), .A4(n1581), .ZN(n2035)
         );
    OAI211D0BWP7T U1086 ( .A1(n49), .A2(n1449), .B(n2045), .C(n2046), .ZN(n1581)
         );
    AOI211D0BWP7T U1087 ( .A1(n2047), .A2(n2048), .B(n2049), .C(n2050), .ZN(
        n2046) );
    AOI21D0BWP7T U1088 ( .A1(n2051), .A2(n2052), .B(n1348), .ZN(n2050) );
    AOI21D0BWP7T U1089 ( .A1(n577), .A2(n1121), .B(n227), .ZN(n2049) );
    AOI22D0BWP7T U1090 ( .A1(n580), .A2(n259), .B1(n1773), .B2(n285), .ZN(n2045)
         );
    OAI221D0BWP7T U1091 ( .A1(n408), .A2(n195), .B1(n227), .B2(n189), .C(n2053),
         .ZN(n1157) );
    AOI22D0BWP7T U1092 ( .A1(n1154), .A2(n285), .B1(n585), .B2(n438), .ZN(n2053)
         );
    OAI221D0BWP7T U1093 ( .A1(n35), .A2(n1352), .B1(n143), .B2(n1242), .C(n2054),
         .ZN(n341) );
    AOI22D0BWP7T U1094 ( .A1(n606), .A2(n438), .B1(n630), .B2(n317), .ZN(n2054)
         );
    OAI221D0BWP7T U1095 ( .A1(n512), .A2(n578), .B1(n1156), .B2(n820), .C(n2055),
         .ZN(n707) );
    OAI21D0BWP7T U1096 ( .A1(n1792), .A2(n582), .B(n482), .ZN(n2055) );
    OA221D0BWP7T U1097 ( .A1(n408), .A2(n1135), .B1(n35), .B2(n1156), .C(n2056),
         .Z(n1491) );
```

```
    AOI222D0BWP7T U1098 ( .A1(n602), .A2(n296), .B1(n2057), .B2(n289), .C1(n1122)
        , .C2(n482), .ZN(n2056) );
    ND3D0BWP7T U1099 ( .A1(n2058), .A2(n2059), .A3(n2060), .ZN(n1625) );
    NR4D0BWP7T U1100 ( .A1(n487), .A2(n706), .A3(n792), .A4(n1103), .ZN(n2060)
          );
    OAI21D0BWP7T U1101 ( .A1(n803), .A2(n1980), .B(n2061), .ZN(n1103) );
    OAI21D0BWP7T U1102 ( .A1(n350), .A2(n529), .B(n686), .ZN(n2061) );
    OAI221D0BWP7T U1103 ( .A1(n1659), .A2(n1321), .B1(n1248), .B2(n414), .C(
         n2062), .ZN(n792) );
    AOI22D0BWP7T U1104 ( .A1(n586), .A2(n478), .B1(n349), .B2(n355), .ZN(n2062)
          );
    OAI221D0BWP7T U1105 ( .A1(n728), .A2(n760), .B1(n759), .B2(n201), .C(n2063),
         .ZN(n706) );
    AOI222D0BWP7T U1106 ( .A1(n338), .A2(n299), .B1(n1938), .B2(n357), .C1(n530),
         .C2(n355), .ZN(n2063) );
    OAI221D0BWP7T U1107 ( .A1(n728), .A2(n2064), .B1(n197), .B2(n1321), .C(n2065)
        , .ZN(n487) );
    AOI222D0BWP7T U1108 ( .A1(n338), .A2(n701), .B1(n841), .B2(n340), .C1(n337),
         .C2(n204), .ZN(n2065) );
    AOI221D0BWP7T U1109 ( .A1(n374), .A2(n1186), .B1(n348), .B2(n478), .C(n2066),
         .ZN(n2059) );
    AOI21D0BWP7T U1110 ( .A1(n803), .A2(n1248), .B(n1659), .ZN(n2066) );
    AOI22D0BWP7T U1111 ( .A1(n1938), .A2(n204), .B1(n686), .B2(n479), .ZN(n2058)
          );
    ND4D0BWP7T U1112 ( .A1(n2067), .A2(n2068), .A3(n2069), .A4(n2070), .ZN(n1654)
        );
    OA221D0BWP7T U1113 ( .A1(n412), .A2(n413), .B1(n927), .B2(n926), .C(n992),
         .Z(n2070) );
    AOI22D0BWP7T U1114 ( .A1(n351), .A2(n853), .B1(n478), .B2(n338), .ZN(n992)
          );
    OAI21D0BWP7T U1115 ( .A1(n530), .A2(n855), .B(n338), .ZN(n2069) );
    IOA21D0BWP7T U1116 ( .A1(n1280), .A2(n1455), .B(n805), .ZN(n2068) );
    NR2D0BWP7T U1117 ( .A1(n1427), .A2(n349), .ZN(n1455) );
    OAI21D0BWP7T U1118 ( .A1(n339), .A2(n374), .B(n764), .ZN(n2067) );
    IIND4D0BWP7T U1119 ( .A1(n1774), .A2(n1765), .B1(n1555), .B2(n2071), .ZN(
         n2026) );
    NR4D0BWP7T U1120 ( .A1(n822), .A2(n1826), .A3(n1882), .A4(n1833), .ZN(n2071)
          );
    OAI221D0BWP7T U1121 ( .A1(n1073), .A2(n1087), .B1(n879), .B2(n223), .C(n2072)
        , .ZN(n1833) );
    AOI222D0BWP7T U1122 ( .A1(n608), .A2(n1534), .B1(n183), .B2(n633), .C1(n640),
         .C2(n1090), .ZN(n2072) );
    ND4D0BWP7T U1123 ( .A1(n2073), .A2(n2074), .A3(n2075), .A4(n2076), .ZN(n1882)
        );
    AOI222D0BWP7T U1124 ( .A1(n604), .A2(n266), .B1(n293), .B2(n1461), .C1(n282),
         .C2(n639), .ZN(n2076) );
    AOI22D0BWP7T U1125 ( .A1(n482), .A2(n2077), .B1(n2078), .B2(n460), .ZN(n2075)
        );
    CKND2D0BWP7T U1126 ( .A1(n819), .A2(n47), .ZN(n2077) );
    OAI21D0BWP7T U1127 ( .A1(n1192), .A2(n2079), .B(n515), .ZN(n2074) );
    CKND0BWP7T U1128 ( .I(n1972), .ZN(n1192) );
    OAI21D0BWP7T U1129 ( .A1(n1857), .A2(n1616), .B(n517), .ZN(n2073) );
    OAI211D0BWP7T U1130 ( .A1(n526), .A2(n143), .B(n2080), .C(n2081), .ZN(n1826)
          );
```

```
    AOI222D0BWP7T U1131 ( .A1(n1005), .A2(n427), .B1(n1062), .B2(n515), .C1(n615)
        , .C2(n438), .ZN(n2081) );
    AOI22D0BWP7T U1132 ( .A1(n43), .A2(n282), .B1(n1995), .B2(n268), .ZN(n2080)
          );
    OAI221D0BWP7T U1133 ( .A1(n196), .A2(n47), .B1(n526), .B2(n910), .C(n2082),
         .ZN(n822) );
    AOI222D0BWP7T U1134 ( .A1(n1616), .A2(n1683), .B1(n2019), .B2(n517), .C1(
         n1996), .C2(n515), .ZN(n2082) );
    CKND0BWP7T U1135 ( .I(n929), .ZN(n1616) );
    AOI21D0BWP7T U1136 ( .A1(n291), .A2(n1442), .B(n343), .ZN(n1555) );
    OAI222D0BWP7T U1137 ( .A1(n143), .A2(n431), .B1(n1350), .B2(n2083), .C1(n512)
        , .C2(n195), .ZN(n343) );
    ND4D0BWP7T U1138 ( .A1(n2084), .A2(n2085), .A3(n2086), .A4(n2087), .ZN(n1765)
         );
    AOI22D0BWP7T U1139 ( .A1(n42), .A2(n693), .B1(n605), .B2(n268), .ZN(n2087)
          );
    AOI22D0BWP7T U1140 ( .A1(n691), .A2(n291), .B1(n574), .B2(n269), .ZN(n2086)
          );
    AOI22D0BWP7T U1141 ( .A1(n293), .A2(n1122), .B1(n582), .B2(n266), .ZN(n2085)
          );
    AOI22D0BWP7T U1142 ( .A1(n630), .A2(n727), .B1(n212), .B2(n773), .ZN(n2084)
          );
    OAI221D0BWP7T U1143 ( .A1(n1350), .A2(n2088), .B1(n432), .B2(n195), .C(n2089)
        , .ZN(n1774) );
    AOI22D0BWP7T U1144 ( .A1(n1154), .A2(n294), .B1(n2090), .B2(n289), .ZN(n2089)
         );
    IIND4D0BWP7T U1145 ( .A1(n888), .A2(n97), .B1(n949), .B2(n2091), .ZN(n2025)
          );
    NR4D0BWP7T U1146 ( .A1(n1061), .A2(n1865), .A3(n2092), .A4(n1718), .ZN(n2091)
         );
    ND4D0BWP7T U1147 ( .A1(n2093), .A2(n2094), .A3(n2095), .A4(n2096), .ZN(n1718)
         );
    AOI211D0BWP7T U1148 ( .A1(n640), .A2(n1704), .B(n2097), .C(n2098), .ZN(n2096)
         );
    IAO21D0BWP7T U1149 ( .A1(n1096), .A2(n453), .B(n223), .ZN(n2098) );
    CKND2D0BWP7T U1150 ( .A1(n2010), .A2(n892), .ZN(n1096) );
    AOI21D0BWP7T U1151 ( .A1(n510), .A2(n347), .B(n226), .ZN(n2097) );
    CKND2D0BWP7T U1152 ( .A1(n345), .A2(n312), .ZN(n1704) );
    AOI22D0BWP7T U1153 ( .A1(n31), .A2(n2099), .B1(n324), .B2(n485), .ZN(n2095)
          );
    AOI22D0BWP7T U1154 ( .A1(n774), .A2(n322), .B1(n216), .B2(n32), .ZN(n2094)
          );
    AOI22D0BWP7T U1155 ( .A1(n631), .A2(n34), .B1(n182), .B2(n483), .ZN(n2093)
          );
    ND3D0BWP7T U1156 ( .A1(n2100), .A2(n708), .A3(n2101), .ZN(n1865) );
    AOI211D0BWP7T U1157 ( .A1(n350), .A2(n355), .B(n2102), .C(n851), .ZN(n2101)
          );
    NR2D0BWP7T U1158 ( .A1(n409), .A2(n196), .ZN(n851) );
    OAI22D0BWP7T U1159 ( .A1(n318), .A2(n635), .B1(n614), .B2(n752), .ZN(n2102)
          );
    OA222D0BWP7T U1160 ( .A1(n824), .A2(n330), .B1(n229), .B2(n1972), .C1(n865),
         .C2(n817), .Z(n708) );
    CKND2D0BWP7T U1161 ( .A1(n285), .A2(n2103), .ZN(n1972) );
    AOI21D0BWP7T U1162 ( .A1(n995), .A2(n1683), .B(n990), .ZN(n2100) );
```

```
    OAI22D0BWP7T U1163 ( .A1(n1822), .A2(n820), .B1(n319), .B2(n408), .ZN(n990)
          );
    ND4D0BWP7T U1164 ( .A1(n2104), .A2(n2105), .A3(n2106), .A4(n2107), .ZN(n1061)
        );
    AOI22D0BWP7T U1165 ( .A1(n1200), .A2(n1897), .B1(n754), .B2(n1683), .ZN(
         n2107) );
    CKND0BWP7T U1166 ( .I(n1892), .ZN(n754) );
    AOI22D0BWP7T U1167 ( .A1(n1054), .A2(n828), .B1(n2018), .B2(n515), .ZN(n2106)
        );
    AOI22D0BWP7T U1168 ( .A1(n844), .A2(n266), .B1(n639), .B2(n269), .ZN(n2105)
          );
    AOI21D0BWP7T U1169 ( .A1(n293), .A2(n1995), .B(n486), .ZN(n2104) );
    OAI211D0BWP7T U1170 ( .A1(n526), .A2(n432), .B(n2108), .C(n2109), .ZN(n486)
          );
    AOI222D0BWP7T U1171 ( .A1(n1461), .A2(n285), .B1(n1894), .B2(n828), .C1(
         n2110), .C2(n1683), .ZN(n2109) );
    AOI22D0BWP7T U1172 ( .A1(n514), .A2(n517), .B1(n259), .B2(n639), .ZN(n2108)
          );
    AN4D0BWP7T U1173 ( .A1(n2111), .A2(n2112), .A3(n2113), .A4(n2114), .Z(n949)
          );
    INR4D0BWP7T U1174 ( .A1(n807), .B1(n2115), .B2(n2116), .B3(n1051), .ZN(n2114)
        );
    OAI222D0BWP7T U1175 ( .A1(n1280), .A2(n199), .B1(n840), .B2(n614), .C1(n1850)
        , .C2(n329), .ZN(n1051) );
    NR2D0BWP7T U1176 ( .A1(n639), .A2(n1461), .ZN(n840) );
    OAI22D0BWP7T U1177 ( .A1(n509), .A2(n151), .B1(n405), .B2(n313), .ZN(n2116)
          );
    OAI221D0BWP7T U1178 ( .A1(n432), .A2(n1822), .B1(n47), .B2(n820), .C(n2117),
         .ZN(n2115) );
    AOI22D0BWP7T U1179 ( .A1(n1852), .A2(n1683), .B1(n514), .B2(n828), .ZN(n2117)
        );
    AOI22D0BWP7T U1180 ( .A1(n515), .A2(n1682), .B1(n296), .B2(n718), .ZN(n807)
          );
    AOI221D0BWP7T U1181 ( .A1(n727), .A2(n633), .B1(n183), .B2(n587), .C(n2118),
         .ZN(n2113) );
    OAI222D0BWP7T U1182 ( .A1(n839), .A2(n846), .B1(n875), .B2(n1391), .C1(n803),
         .C2(n804), .ZN(n2118) );
    AOI22D0BWP7T U1183 ( .A1(n1171), .A2(n485), .B1(n31), .B2(n1090), .ZN(n2112)
          );
    AOI22D0BWP7T U1184 ( .A1(n518), .A2(n268), .B1(n615), .B2(n294), .ZN(n2111)
          );
    ND4D0BWP7T U1185 ( .A1(n2119), .A2(n2120), .A3(n2121), .A4(n2122), .ZN(n97)
          );
    NR4D0BWP7T U1186 ( .A1(n2123), .A2(n2124), .A3(n2125), .A4(n2126), .ZN(n2122)
        );
    OAI221D0BWP7T U1187 ( .A1(n309), .A2(n151), .B1(n222), .B2(n680), .C(n930),
         .ZN(n2126) );
    OAI222D0BWP7T U1188 ( .A1(n234), .A2(n1320), .B1(n150), .B2(n2010), .C1(n143)
        , .C2(n1822), .ZN(n2125) );
    OAI222D0BWP7T U1189 ( .A1(n926), .A2(n1638), .B1(n227), .B2(n1394), .C1(
         n1248), .C2(n927), .ZN(n2124) );
    OAI222D0BWP7T U1190 ( .A1(n2127), .A2(n1663), .B1(n1252), .B2(n803), .C1(
         n435), .C2(n153), .ZN(n2123) );
    NR2D0BWP7T U1191 ( .A1(n354), .A2(n701), .ZN(n1252) );
```

```
    NR4D0BWP7T U1192 ( .A1(n2128), .A2(n2129), .A3(n2130), .A4(n2131), .ZN(n2121)
        );
    AOI21D0BWP7T U1193 ( .A1(n1734), .A2(n1087), .B(n313), .ZN(n2131) );
    NR2D0BWP7T U1194 ( .A1(n315), .A2(n453), .ZN(n1734) );
    AOI21D0BWP7T U1195 ( .A1(n2132), .A2(n1659), .B(n440), .ZN(n2130) );
    AOI21D0BWP7T U1196 ( .A1(n1449), .A2(n932), .B(n35), .ZN(n2129) );
    OAI211D0BWP7T U1197 ( .A1(n2133), .A2(n759), .B(n2134), .C(n2135), .ZN(n2128)
        );
    OAI21D0BWP7T U1198 ( .A1(n1370), .A2(n582), .B(n294), .ZN(n2135) );
    OAI31D0BWP7T U1199 ( .A1(n32), .A2(n324), .A3(n1490), .B(n630), .ZN(n2134)
        );
    NR4D0BWP7T U1200 ( .A1(n2047), .A2(n2136), .A3(n600), .A4(n1462), .ZN(n2120)
        );
    AOI211D0BWP7T U1201 ( .A1(n686), .A2(n2137), .B(n2138), .C(n1389), .ZN(n2119)
        );
    NR2D0BWP7T U1202 ( .A1(n1242), .A2(n196), .ZN(n1389) );
    CKND2D0BWP7T U1203 ( .A1(n231), .A2(n157), .ZN(n2137) );
    OAI211D0BWP7T U1204 ( .A1(n150), .A2(n406), .B(n2139), .C(n2140), .ZN(n888)
        );
    AOI22D0BWP7T U1205 ( .A1(n1007), .A2(n268), .B1(n315), .B2(n630), .ZN(n2140)
        );
    OAI21D0BWP7T U1206 ( .A1(n1095), .A2(n608), .B(n775), .ZN(n2139) );
    INR4D0BWP7T U1207 ( .A1(n2141), .B1(n1187), .B2(n22), .B3(n1205), .ZN(n2024)
        );
    OAI211D0BWP7T U1208 ( .A1(n824), .A2(n1238), .B(n2142), .C(n2143), .ZN(n1205)
        );
    AOI222D0BWP7T U1209 ( .A1(n1995), .A2(n482), .B1(n1680), .B2(n693), .C1(n717)
       , .C2(n601), .ZN(n2143) );
    AOI22D0BWP7T U1210 ( .A1(n1054), .A2(n1683), .B1(n1055), .B2(n269), .ZN(
        n2142) );
    OAI211D0BWP7T U1211 ( .A1(n1241), .A2(n1120), .B(n709), .C(n1480), .ZN(n22)
        );
    OA221D0BWP7T U1212 ( .A1(n435), .A2(n155), .B1(n432), .B2(n1120), .C(n993),
        .Z(n1480) );
    AOI22D0BWP7T U1213 ( .A1(n285), .A2(n1788), .B1(n482), .B2(n1381), .ZN(n993)
        );
    AOI22D0BWP7T U1214 ( .A1(n294), .A2(n585), .B1(n601), .B2(n1154), .ZN(n709)
        );
    OAI221D0BWP7T U1215 ( .A1(n1112), .A2(n1321), .B1(n1296), .B2(n682), .C(
        n2144), .ZN(n1187) );
    AOI211D0BWP7T U1216 ( .A1(n1938), .A2(n340), .B(n2145), .C(n2146), .ZN(n2144)
        );
    AOI21D0BWP7T U1217 ( .A1(n198), .A2(n1358), .B(n926), .ZN(n2146) );
    AOI21D0BWP7T U1218 ( .A1(n1980), .A2(n1663), .B(n1248), .ZN(n2145) );
    NR4D0BWP7T U1219 ( .A1(n1174), .A2(n1085), .A3(n731), .A4(n1039), .ZN(n2141)
        );
    ND3D0BWP7T U1220 ( .A1(n2147), .A2(n627), .A3(n2148), .ZN(n1039) );
    AOI211D0BWP7T U1221 ( .A1(n297), .A2(n854), .B(n2149), .C(n2150), .ZN(n2148)
        );
    AOI21D0BWP7T U1222 ( .A1(n1638), .A2(n1637), .B(n413), .ZN(n2150) );
    AOI21D0BWP7T U1223 ( .A1(n1980), .A2(n1114), .B(n148), .ZN(n2149) );
    AOI222D0BWP7T U1224 ( .A1(n764), .A2(n349), .B1(n339), .B2(n352), .C1(n357),
        .C2(n1427), .ZN(n627) );
    AOI22D0BWP7T U1225 ( .A1(n371), .A2(n204), .B1(n853), .B2(n1129), .ZN(n2147)
```

```
      );
    OAI211D0BWP7T U1226 ( .A1(n410), .A2(n232), .B(n2151), .C(n2152), .ZN(n731)
      );
    AOI22D0BWP7T U1227 ( .A1(n696), .A2(n371), .B1(n586), .B2(n479), .ZN(n2152)
      );
    OAI21D0BWP7T U1228 ( .A1(n204), .A2(n352), .B(n298), .ZN(n2151) );
    ND4D0BWP7T U1229 ( .A1(n2153), .A2(n2154), .A3(n2155), .A4(n2156), .ZN(n1085)
     );
    AOI222D0BWP7T U1230 ( .A1(n484), .A2(n608), .B1(n214), .B2(n34), .C1(n632),
     .C2(n638), .ZN(n2156) );
    OAI21D0BWP7T U1231 ( .A1(n886), .A2(n588), .B(n216), .ZN(n2155) );
    OAI21D0BWP7T U1232 ( .A1(n183), .A2(n775), .B(n483), .ZN(n2154) );
    OAI21D0BWP7T U1233 ( .A1(n183), .A2(n214), .B(n640), .ZN(n2153) );
    OAI221D0BWP7T U1234 ( .A1(n234), .A2(n320), .B1(n509), .B2(n1074), .C(n2157),
     .ZN(n1174) );
    AOI221D0BWP7T U1235 ( .A1(n1490), .A2(n587), .B1(n638), .B2(n2158), .C(n2159)
    , .ZN(n2157) );
    AOI21D0BWP7T U1236 ( .A1(n680), .A2(n1086), .B(n1528), .ZN(n2159) );
    CKND0BWP7T U1237 ( .I(n2160), .ZN(n2158) );
    NR4D0BWP7T U1238 ( .A1(n2161), .A2(n2162), .A3(n1911), .A4(n2163), .ZN(n2022)
     );
    OAI22D0BWP7T U1239 ( .A1(n345), .A2(n313), .B1(n512), .B2(n236), .ZN(n2163)
      );
    ND4D0BWP7T U1240 ( .A1(n2164), .A2(n1299), .A3(n2165), .A4(n2166), .ZN(n1911)
     );
    AOI211D0BWP7T U1241 ( .A1(n516), .A2(n1683), .B(n2167), .C(n2168), .ZN(n2166)
     );
    AOI21D0BWP7T U1242 ( .A1(n849), .A2(n1237), .B(n618), .ZN(n2168) );
    OAI22D0BWP7T U1243 ( .A1(n327), .A2(n1003), .B1(n143), .B2(n738), .ZN(n2167)
      );
    AOI22D0BWP7T U1244 ( .A1(n583), .A2(n844), .B1(n372), .B2(n355), .ZN(n2165)
      );
    CKND2D0BWP7T U1245 ( .A1(n1198), .A2(n282), .ZN(n1299) );
    AOI22D0BWP7T U1246 ( .A1(n459), .A2(n296), .B1(n718), .B2(n427), .ZN(n2164)
      );
    OAI221D0BWP7T U1247 ( .A1(n1116), .A2(n752), .B1(n143), .B2(n47), .C(n2169),
     .ZN(n2162) );
    AOI22D0BWP7T U1248 ( .A1(n353), .A2(n764), .B1(n2170), .B2(n1567), .ZN(n2169)
     );
    IND4D0BWP7T U1249 ( .A1(n50), .B1(n1739), .B2(n2171), .B3(n2172), .ZN(n2161)
      );
    NR3D0BWP7T U1250 ( .A1(n778), .A2(n503), .A3(n1630), .ZN(n2172) );
    OAI221D0BWP7T U1251 ( .A1(n804), .A2(n1321), .B1(n198), .B2(n802), .C(n2173),
     .ZN(n1630) );
    AOI211D0BWP7T U1252 ( .A1(n479), .A2(n455), .B(n2174), .C(n2175), .ZN(n2173)
      );
    AOI21D0BWP7T U1253 ( .A1(n199), .A2(n803), .B(n681), .ZN(n2175) );
    AOI21D0BWP7T U1254 ( .A1(n1279), .A2(n410), .B(n1113), .ZN(n2174) );
    OAI222D0BWP7T U1255 ( .A1(n874), .A2(n880), .B1(n150), .B2(n151), .C1(n879),
     .C2(n404), .ZN(n503) );
    OAI211D0BWP7T U1256 ( .A1(n405), .A2(n509), .B(n2176), .C(n2177), .ZN(n778)
      );
    AOI211D0BWP7T U1257 ( .A1(n31), .A2(n214), .B(n1948), .C(n2178), .ZN(n2177)
      );
```

```
    AOI21D0BWP7T U1258 ( .A1(n345), .A2(n1134), .B(n309), .ZN(n2178) );
    NR2D0BWP7T U1259 ( .A1(n313), .A2(n407), .ZN(n1948) );
    AOI22D0BWP7T U1260 ( .A1(n216), .A2(n1490), .B1(n1007), .B2(n427), .ZN(n2176)
        );
    AOI22D0BWP7T U1261 ( .A1(n34), .A2(n632), .B1(n322), .B2(n182), .ZN(n1739)
         );
    ND4D0BWP7T U1262 ( .A1(n2179), .A2(n2180), .A3(n2181), .A4(n2182), .ZN(n50)
         );
    NR4D0BWP7T U1263 ( .A1(n2183), .A2(n2184), .A3(n551), .A4(n380), .ZN(n2182)
         );
    IND4D0BWP7T U1264 ( .A1(n1294), .B1(n1757), .B2(n2185), .B3(n2186), .ZN(n380)
        );
    AOI211D0BWP7T U1265 ( .A1(n727), .A2(n638), .B(n2187), .C(n2188), .ZN(n2186)
         );
    AOI21D0BWP7T U1266 ( .A1(n222), .A2(n311), .B(n313), .ZN(n2188) );
    OAI22D0BWP7T U1267 ( .A1(n614), .A2(n817), .B1(n865), .B2(n1849), .ZN(n2187)
         );
    OA21D0BWP7T U1268 ( .A1(n724), .A2(n509), .B(n91), .Z(n2185) );
    AOI22D0BWP7T U1269 ( .A1(n482), .A2(n615), .B1(n693), .B2(n2170), .ZN(n91)
         );
    AOI22D0BWP7T U1270 ( .A1(n317), .A2(n1095), .B1(n1443), .B2(n290), .ZN(n1757)
        );
    ND4D0BWP7T U1271 ( .A1(n2189), .A2(n1879), .A3(n2190), .A4(n2191), .ZN(n1294)
        );
    AOI22D0BWP7T U1272 ( .A1(n1055), .A2(n583), .B1(n1461), .B2(n266), .ZN(n2191)
        );
    MAOI22D0BWP7T U1273 ( .A1(n714), .A2(n294), .B1(n1850), .B2(n229), .ZN(n2190)
        );
    AOI22D0BWP7T U1274 ( .A1(n259), .A2(n2170), .B1(n1683), .B2(n1679), .ZN(
        n1879) );
    AOI21D0BWP7T U1275 ( .A1(n459), .A2(n460), .B(n1060), .ZN(n2189) );
    OAI222D0BWP7T U1276 ( .A1(n199), .A2(n1279), .B1(n1816), .B2(n526), .C1(n408)
       , .C2(n1815), .ZN(n1060) );
    ND4D0BWP7T U1277 ( .A1(n2192), .A2(n2193), .A3(n2194), .A4(n2195), .ZN(n551)
         );
    NR4D0BWP7T U1278 ( .A1(n2196), .A2(n2197), .A3(n2198), .A4(n2199), .ZN(n2195)
        );
    OAI22D0BWP7T U1279 ( .A1(n1431), .A2(n889), .B1(n1170), .B2(n1073), .ZN(
        n2199) );
    OAI222D0BWP7T U1280 ( .A1(n326), .A2(n1240), .B1(n150), .B2(n1529), .C1(n347)
       , .C2(n313), .ZN(n2198) );
    OAI222D0BWP7T U1281 ( .A1(n1248), .A2(n201), .B1(n803), .B2(n231), .C1(n1280)
       , .C2(n440), .ZN(n2197) );
    OAI211D0BWP7T U1282 ( .A1(n1112), .A2(n1017), .B(n2200), .C(n2201), .ZN(
        n2196) );
    CKND0BWP7T U1283 ( .I(n39), .ZN(n2201) );
    NR2D0BWP7T U1284 ( .A1(n1156), .A2(n865), .ZN(n39) );
    OAI21D0BWP7T U1285 ( .A1(n350), .A2(n371), .B(n297), .ZN(n2200) );
    INR3D0BWP7T U1286 ( .A1(n761), .B1(n2202), .B2(n1760), .ZN(n2194) );
    OAI221D0BWP7T U1287 ( .A1(n35), .A2(n613), .B1(n510), .B2(n1240), .C(n2203),
        .ZN(n1760) );
    AOI22D0BWP7T U1288 ( .A1(n650), .A2(n460), .B1(n1370), .B2(n296), .ZN(n2203)
         );
```

```
    OAI211D0BWP7T U1289 ( .A1(n910), .A2(n1849), .B(n1514), .C(n1941), .ZN(n2202)
        );
    AN4D0BWP7T U1290 ( .A1(n2204), .A2(n2205), .A3(n2206), .A4(n2207), .Z(n1941)
         );
    NR4D0BWP7T U1291 ( .A1(n2208), .A2(n2209), .A3(n2210), .A4(n2211), .ZN(n2207)
        );
    AOI21D0BWP7T U1292 ( .A1(n225), .A2(n510), .B(n221), .ZN(n2211) );
    AOI21D0BWP7T U1293 ( .A1(n724), .A2(n1134), .B(n875), .ZN(n2210) );
    AOI21D0BWP7T U1294 ( .A1(n2212), .A2(n880), .B(n1320), .ZN(n2209) );
    AOI21D0BWP7T U1295 ( .A1(n893), .A2(n1073), .B(n1072), .ZN(n2208) );
    NR2D0BWP7T U1296 ( .A1(n216), .A2(n212), .ZN(n893) );
    OA222D0BWP7T U1297 ( .A1(n1488), .A2(n1086), .B1(n224), .B2(n1840), .C1(n309)
       , .C2(n1074), .Z(n2206) );
    NR2D0BWP7T U1298 ( .A1(n1535), .A2(n587), .ZN(n1840) );
    AOI222D0BWP7T U1299 ( .A1(n242), .A2(n1535), .B1(n630), .B2(n882), .C1(n1095)
       , .C2(n29), .ZN(n2205) );
    CKND0BWP7T U1300 ( .I(n1834), .ZN(n1535) );
    NR2D0BWP7T U1301 ( .A1(n30), .A2(n485), .ZN(n1834) );
    AOI221D0BWP7T U1302 ( .A1(n484), .A2(n316), .B1(n632), .B2(n587), .C(n1720),
         .ZN(n2204) );
    AO222D0BWP7T U1303 ( .A1(n317), .A2(n608), .B1(n1746), .B2(n31), .C1(n216),
         .C2(n315), .Z(n1720) );
    CKND2D0BWP7T U1304 ( .A1(n151), .A2(n1074), .ZN(n1746) );
    AOI22D0BWP7T U1305 ( .A1(n316), .A2(n632), .B1(n638), .B2(n1534), .ZN(n1514)
         );
    AOI22D0BWP7T U1306 ( .A1(n805), .A2(n374), .B1(n584), .B2(n338), .ZN(n761)
         );
    AOI222D0BWP7T U1307 ( .A1(n266), .A2(n518), .B1(n774), .B2(n485), .C1(n631),
         .C2(n30), .ZN(n2193) );
    AOI222D0BWP7T U1308 ( .A1(n1618), .A2(n482), .B1(n615), .B2(n427), .C1(n1681)
       , .C2(n583), .ZN(n2192) );
    OAI222D0BWP7T U1309 ( .A1(n1659), .A2(n802), .B1(n432), .B2(n36), .C1(n1247),
         .C2(n1321), .ZN(n2184) );
    IND4D0BWP7T U1310 ( .A1(n1342), .B1(n1046), .B2(n1127), .B3(n2213), .ZN(
        n2183) );
    AOI21D0BWP7T U1311 ( .A1(n372), .A2(n373), .B(n2214), .ZN(n2213) );
    AOI21D0BWP7T U1312 ( .A1(n614), .A2(n143), .B(n417), .ZN(n2214) );
    CKND2D0BWP7T U1313 ( .A1(n1321), .A2(n1248), .ZN(n373) );
    CKND2D0BWP7T U1314 ( .A1(n1107), .A2(n693), .ZN(n1127) );
    CKND2D0BWP7T U1315 ( .A1(n460), .A2(n1967), .ZN(n1046) );
    NR4D0BWP7T U1316 ( .A1(n2215), .A2(n2216), .A3(n936), .A4(n916), .ZN(n2181)
         );
    AO211D0BWP7T U1317 ( .A1(n297), .A2(n349), .B(n646), .C(n87), .Z(n916) );
    OAI221D0BWP7T U1318 ( .A1(n1017), .A2(n157), .B1(n148), .B2(n760), .C(n2217),
         .ZN(n87) );
    AOI222D0BWP7T U1319 ( .A1(n297), .A2(n339), .B1(n829), .B2(n2218), .C1(n298),
         .C2(n357), .ZN(n2217) );
    CKND2D0BWP7T U1320 ( .A1(n682), .A2(n232), .ZN(n2218) );
    NR2D0BWP7T U1321 ( .A1(n1980), .A2(n413), .ZN(n646) );
    ND4D0BWP7T U1322 ( .A1(n2219), .A2(n2220), .A3(n2221), .A4(n2222), .ZN(n936)
         );
    NR3D0BWP7T U1323 ( .A1(n2223), .A2(n2224), .A3(n2225), .ZN(n2222) );
    OAI222D0BWP7T U1324 ( .A1(n1507), .A2(n526), .B1(n1440), .B2(n194), .C1(n318)
       , .C2(n819), .ZN(n2225) );
```

```
    OAI222D0BWP7T U1325 ( .A1(n152), .A2(n635), .B1(n327), .B2(n1603), .C1(n475),
         .C2(n889), .ZN(n2224) );
    ND4D0BWP7T U1326 ( .A1(n1601), .A2(n1207), .A3(n94), .A4(n2226), .ZN(n2223)
         );
    AOI222D0BWP7T U1327 ( .A1(n315), .A2(n30), .B1(n693), .B2(n639), .C1(n293),
         .C2(n604), .ZN(n2226) );
    OA221D0BWP7T U1328 ( .A1(n618), .A2(n817), .B1(n432), .B2(n1815), .C(n2227),
         .Z(n94) );
    AOI22D0BWP7T U1329 ( .A1(n2019), .A2(n515), .B1(n615), .B2(n296), .ZN(n2227)
         );
    AOI22D0BWP7T U1330 ( .A1(n601), .A2(n718), .B1(n1967), .B2(n285), .ZN(n1207)
         );
    AOI22D0BWP7T U1331 ( .A1(n517), .A2(n2079), .B1(n828), .B2(n1852), .ZN(n1601)
        );
    NR4D0BWP7T U1332 ( .A1(n2228), .A2(n2229), .A3(n2230), .A4(n2231), .ZN(n2221)
        );
    AOI21D0BWP7T U1333 ( .A1(n47), .A2(n889), .B(n35), .ZN(n2231) );
    AOI21D0BWP7T U1334 ( .A1(n186), .A2(n2232), .B(n229), .ZN(n2230) );
    AOI21D0BWP7T U1335 ( .A1(n434), .A2(n1120), .B(n143), .ZN(n2229) );
    OAI211D0BWP7T U1336 ( .A1(n2233), .A2(n408), .B(n2234), .C(n2235), .ZN(n2228)
        );
    OAI21D0BWP7T U1337 ( .A1(n775), .A2(n1140), .B(n316), .ZN(n2235) );
    OAI21D0BWP7T U1338 ( .A1(n324), .A2(n280), .B(n587), .ZN(n2234) );
    AOI222D0BWP7T U1339 ( .A1(n453), .A2(n608), .B1(n459), .B2(n211), .C1(n1171),
         .C2(n2236), .ZN(n2220) );
    CKND2D0BWP7T U1340 ( .A1(n143), .A2(n196), .ZN(n211) );
    AOI222D0BWP7T U1341 ( .A1(n1101), .A2(n633), .B1(n183), .B2(n322), .C1(n216),
         .C2(n214), .ZN(n2219) );
    IND3D0BWP7T U1342 ( .A1(n918), .B1(n2237), .B2(n364), .ZN(n2216) );
    AOI22D0BWP7T U1343 ( .A1(n586), .A2(n349), .B1(n205), .B2(n353), .ZN(n364)
         );
    ND4D0BWP7T U1344 ( .A1(n2238), .A2(n2239), .A3(n2240), .A4(n2241), .ZN(n918)
         );
    NR4D0BWP7T U1345 ( .A1(n2242), .A2(n2243), .A3(n1761), .A4(n1500), .ZN(n2241)
        );
    OAI221D0BWP7T U1346 ( .A1(n49), .A2(n1394), .B1(n222), .B2(n1240), .C(n2244),
         .ZN(n1500) );
    AOI222D0BWP7T U1347 ( .A1(n1107), .A2(n266), .B1(n1799), .B2(n2245), .C1(
         n572), .C2(n289), .ZN(n2244) );
    OAI221D0BWP7T U1348 ( .A1(n432), .A2(n1449), .B1(n1350), .B2(n2051), .C(
         n2246), .ZN(n1761) );
    AOI221D0BWP7T U1349 ( .A1(n606), .A2(n259), .B1(n602), .B2(n269), .C(n2247),
         .ZN(n2246) );
    AOI21D0BWP7T U1350 ( .A1(n1784), .A2(n2043), .B(n196), .ZN(n2247) );
    ND3D0BWP7T U1351 ( .A1(n92), .A2(n93), .A3(n168), .ZN(n2243) );
    AN4D0BWP7T U1352 ( .A1(n1021), .A2(n657), .A3(n2248), .A4(n2249), .Z(n168)
         );
    AOI211D0BWP7T U1353 ( .A1(n42), .A2(n285), .B(n2250), .C(n2251), .ZN(n2249)
         );
    AOI21D0BWP7T U1354 ( .A1(n35), .A2(n196), .B(n684), .ZN(n2251) );
    OAI22D0BWP7T U1355 ( .A1(n1099), .A2(n313), .B1(n408), .B2(n1156), .ZN(n2250)
        );
    INR2D0BWP7T U1356 ( .A1(n736), .B1(n873), .ZN(n2248) );
```

```
    OAI221D0BWP7T U1357 ( .A1(n408), .A2(n1449), .B1(n1099), .B2(n680), .C(n2252)
        , .ZN(n873) );
    AOI22D0BWP7T U1358 ( .A1(n574), .A2(n601), .B1(n1462), .B2(n269), .ZN(n2252)
          );
    AOI22D0BWP7T U1359 ( .A1(n291), .A2(n2044), .B1(n693), .B2(n605), .ZN(n736)
          );
    CKND0BWP7T U1360 ( .I(n2253), .ZN(n2044) );
    AOI22D0BWP7T U1361 ( .A1(n296), .A2(n264), .B1(n482), .B2(n605), .ZN(n657)
          );
    AOI22D0BWP7T U1362 ( .A1(n630), .A2(n774), .B1(n438), .B2(n1122), .ZN(n1021)
          );
    AOI222D0BWP7T U1363 ( .A1(n32), .A2(n212), .B1(n640), .B2(n878), .C1(n34),
         .C2(n280), .ZN(n93) );
    OA221D0BWP7T U1364 ( .A1(n2254), .A2(n2255), .B1(n865), .B2(n1135), .C(n2256)
        , .Z(n92) );
    AOI22D0BWP7T U1365 ( .A1(n1790), .A2(n269), .B1(n1462), .B2(n293), .ZN(n2256)
         );
    ND3D0BWP7T U1366 ( .A1(n1173), .A2(n780), .A3(n2257), .ZN(n2242) );
    AOI222D0BWP7T U1367 ( .A1(n1095), .A2(n280), .B1(n638), .B2(n1101), .C1(n315)
        , .C2(n640), .ZN(n2257) );
    AOI22D0BWP7T U1368 ( .A1(n29), .A2(n640), .B1(n1490), .B2(n212), .ZN(n780)
          );
    AOI22D0BWP7T U1369 ( .A1(n216), .A2(n33), .B1(n183), .B2(n630), .ZN(n1173)
          );
    AOI221D0BWP7T U1370 ( .A1(n632), .A2(n322), .B1(n878), .B2(n34), .C(n2258),
         .ZN(n2240) );
    OAI222D0BWP7T U1371 ( .A1(n1431), .A2(n1394), .B1(n614), .B2(n153), .C1(n154)
        , .C2(n1121), .ZN(n2258) );
    AOI221D0BWP7T U1372 ( .A1(n2057), .A2(n573), .B1(n453), .B2(n2259), .C(n2260)
        , .ZN(n2239) );
    AOI31D0BWP7T U1373 ( .A1(n222), .A2(n892), .A3(n2033), .B(n509), .ZN(n2260)
          );
    AOI22D0BWP7T U1374 ( .A1(n630), .A2(n1521), .B1(n580), .B2(n293), .ZN(n2238)
          );
    IND4D0BWP7T U1375 ( .A1(n1339), .B1(n1409), .B2(n948), .B3(n2261), .ZN(n2215)
         );
    AOI222D0BWP7T U1376 ( .A1(n259), .A2(n1788), .B1(n289), .B2(n288), .C1(n293),
         .C2(n602), .ZN(n948) );
    AN4D0BWP7T U1377 ( .A1(n2262), .A2(n88), .A3(n2263), .A4(n2264), .Z(n1409)
          );
    AOI22D0BWP7T U1378 ( .A1(n1090), .A2(n2259), .B1(n1857), .B2(n828), .ZN(
         n2264) );
    CKND0BWP7T U1379 ( .I(n1197), .ZN(n1857) );
    CKND2D0BWP7T U1380 ( .A1(n2265), .A2(n294), .ZN(n1197) );
    CKND2D0BWP7T U1381 ( .A1(n404), .A2(n309), .ZN(n2259) );
    AOI22D0BWP7T U1382 ( .A1(n453), .A2(n34), .B1(n1078), .B2(n638), .ZN(n2263)
          );
    AOI222D0BWP7T U1383 ( .A1(n269), .A2(n1005), .B1(n515), .B2(n1679), .C1(n285)
        , .C2(n1198), .ZN(n88) );
    AOI22D0BWP7T U1384 ( .A1(n1534), .A2(n316), .B1(n1095), .B2(n242), .ZN(n2262)
         );
    ND4D0BWP7T U1385 ( .A1(n2266), .A2(n2267), .A3(n2268), .A4(n2269), .ZN(n1339)
         );
```

```
    NR4D0BWP7T U1386 ( .A1(n2270), .A2(n2271), .A3(n2272), .A4(n2273), .ZN(n2269)
        );
    OAI22D0BWP7T U1387 ( .A1(n1134), .A2(n680), .B1(n327), .B2(n2274), .ZN(n2273)
        );
    OAI22D0BWP7T U1388 ( .A1(n196), .A2(n738), .B1(n234), .B2(n879), .ZN(n2272)
         );
    OAI221D0BWP7T U1389 ( .A1(n1732), .A2(n311), .B1(n1489), .B2(n406), .C(n2275)
       , .ZN(n2271) );
    MAOI22D0BWP7T U1390 ( .A1(n2276), .A2(n755), .B1(n1050), .B2(n1702), .ZN(
        n2275) );
    ND4D0BWP7T U1391 ( .A1(n2277), .A2(n2278), .A3(n2279), .A4(n2280), .ZN(n2270)
        );
    NR4D0BWP7T U1392 ( .A1(n1608), .A2(n2281), .A3(n2282), .A4(n2283), .ZN(n2280)
        );
    AOI21D0BWP7T U1393 ( .A1(n2010), .A2(n724), .B(n309), .ZN(n2283) );
    AOI21D0BWP7T U1394 ( .A1(n2284), .A2(n1892), .B(n824), .ZN(n2282) );
    AOI21D0BWP7T U1395 ( .A1(n2285), .A2(n435), .B(n846), .ZN(n2281) );
    NR2D0BWP7T U1396 ( .A1(n1156), .A2(n475), .ZN(n1608) );
    OA22D0BWP7T U1397 ( .A1(n318), .A2(n2233), .B1(n329), .B2(n2286), .Z(n2279)
         );
    NR2D0BWP7T U1398 ( .A1(n43), .A2(n844), .ZN(n2233) );
    OAI21D0BWP7T U1399 ( .A1(n453), .A2(n1490), .B(n485), .ZN(n2278) );
    IOA21D0BWP7T U1400 ( .A1(n889), .A2(n1895), .B(n293), .ZN(n2277) );
    NR2D0BWP7T U1401 ( .A1(n481), .A2(n639), .ZN(n1895) );
    IINR4D0BWP7T U1402 ( .A1(n1501), .A2(n987), .B1(n2287), .B2(n1552), .ZN(
        n2268) );
    OAI21D0BWP7T U1403 ( .A1(n475), .A2(n1367), .B(n2288), .ZN(n1552) );
    OAI21D0BWP7T U1404 ( .A1(n605), .A2(n1799), .B(n601), .ZN(n2288) );
    IIND4D0BWP7T U1405 ( .A1(n553), .A2(n86), .B1(n1762), .B2(n90), .ZN(n2287)
         );
    AN4D0BWP7T U1406 ( .A1(n2289), .A2(n463), .A3(n2290), .A4(n2291), .Z(n90) );
    AOI221D0BWP7T U1407 ( .A1(n774), .A2(n587), .B1(n280), .B2(n640), .C(n2292),
        .ZN(n2291) );
    OAI22D0BWP7T U1408 ( .A1(n1086), .A2(n510), .B1(n221), .B2(n1134), .ZN(n2292)
        );
    AOI22D0BWP7T U1409 ( .A1(n216), .A2(n2293), .B1(n453), .B2(n483), .ZN(n2290)
         );
    CKND2D0BWP7T U1410 ( .A1(n1528), .A2(n347), .ZN(n2293) );
    CKND2D0BWP7T U1411 ( .A1(n638), .A2(n1171), .ZN(n463) );
    IOA21D0BWP7T U1412 ( .A1(n309), .A2(n1728), .B(n1490), .ZN(n2289) );
    OA211D0BWP7T U1413 ( .A1(n224), .A2(n680), .B(n2294), .C(n2295), .Z(n1762)
         );
    AOI222D0BWP7T U1414 ( .A1(n1370), .A2(n266), .B1(n1381), .B2(n583), .C1(
        n1122), .C2(n296), .ZN(n2295) );
    AOI22D0BWP7T U1415 ( .A1(n1790), .A2(n693), .B1(n1799), .B2(n294), .ZN(n2294)
        );
    ND4D0BWP7T U1416 ( .A1(n2296), .A2(n2297), .A3(n2298), .A4(n2299), .ZN(n86)
         );
    AOI221D0BWP7T U1417 ( .A1(n714), .A2(n460), .B1(n293), .B2(n1198), .C(n2300),
        .ZN(n2299) );
    OAI22D0BWP7T U1418 ( .A1(n435), .A2(n409), .B1(n196), .B2(n228), .ZN(n2300)
         );
    AOI22D0BWP7T U1419 ( .A1(n2276), .A2(n828), .B1(n1852), .B2(n515), .ZN(n2298)
        );
```

```
    CKND0BWP7T U1420 ( .I(n750), .ZN(n2276) );
    OAI21D0BWP7T U1421 ( .A1(n827), .A2(n1199), .B(n1683), .ZN(n2297) );
    OAI21D0BWP7T U1422 ( .A1(n1223), .A2(n1995), .B(n438), .ZN(n2296) );
    OAI221D0BWP7T U1423 ( .A1(n1086), .A2(n1072), .B1(n404), .B2(n320), .C(n2301)
        , .ZN(n553) );
    AOI222D0BWP7T U1424 ( .A1(n453), .A2(n322), .B1(n886), .B2(n1091), .C1(n483),
         .C2(n2099), .ZN(n2301) );
    CKND2D0BWP7T U1425 ( .A1(n1100), .A2(n405), .ZN(n2099) );
    CKND2D0BWP7T U1426 ( .A1(n509), .A2(n1086), .ZN(n1091) );
    AOI22D0BWP7T U1427 ( .A1(n485), .A2(n215), .B1(n773), .B2(n640), .ZN(n987)
          );
    AOI22D0BWP7T U1428 ( .A1(n601), .A2(n1792), .B1(n282), .B2(n580), .ZN(n1501)
          );
    AOI211D0BWP7T U1429 ( .A1(n632), .A2(n31), .B(n2302), .C(n902), .ZN(n2267)
          );
    OAI22D0BWP7T U1430 ( .A1(n1086), .A2(n1099), .B1(n726), .B2(n1073), .ZN(n902)
         );
    OAI22D0BWP7T U1431 ( .A1(n509), .A2(n1488), .B1(n346), .B2(n510), .ZN(n2302)
          );
    IINR4D0BWP7T U1432 ( .A1(n720), .A2(n1684), .B1(n1602), .B2(n1883), .ZN(
         n2266) );
    OAI221D0BWP7T U1433 ( .A1(n910), .A2(n889), .B1(n865), .B2(n409), .C(n2303),
         .ZN(n1883) );
    AOI222D0BWP7T U1434 ( .A1(n293), .A2(n1005), .B1(n517), .B2(n2304), .C1(
         n1461), .C2(n268), .ZN(n2303) );
    CKND2D0BWP7T U1435 ( .A1(n330), .A2(n838), .ZN(n2304) );
    CKND0BWP7T U1436 ( .I(n2018), .ZN(n838) );
    OAI222D0BWP7T U1437 ( .A1(n408), .A2(n819), .B1(n49), .B2(n817), .C1(n865),
         .C2(n849), .ZN(n1602) );
    AOI222D0BWP7T U1438 ( .A1(n355), .A2(n1427), .B1(n1445), .B2(n1995), .C1(
         n285), .C2(n1681), .ZN(n1684) );
    AOI222D0BWP7T U1439 ( .A1(n34), .A2(n183), .B1(n322), .B2(n317), .C1(n640),
         .C2(n1490), .ZN(n720) );
    INR4D0BWP7T U1440 ( .A1(n2305), .B1(n2306), .B2(n95), .B3(n174), .ZN(n2180)
          );
    ND4D0BWP7T U1441 ( .A1(n1582), .A2(n1336), .A3(n2307), .A4(n2308), .ZN(n174)
          );
    NR3D0BWP7T U1442 ( .A1(n2309), .A2(n703), .A3(n938), .ZN(n2308) );
    OAI222D0BWP7T U1443 ( .A1(n2254), .A2(n1349), .B1(n435), .B2(n195), .C1(n418)
        , .C2(n2310), .ZN(n938) );
    OAI22D0BWP7T U1444 ( .A1(n318), .A2(n1352), .B1(n435), .B2(n578), .ZN(n703)
          );
    OAI222D0BWP7T U1445 ( .A1(n318), .A2(n431), .B1(n416), .B2(n189), .C1(n227),
         .C2(n1471), .ZN(n2309) );
    NR2D0BWP7T U1446 ( .A1(n293), .A2(n583), .ZN(n416) );
    NR3D0BWP7T U1447 ( .A1(n1396), .A2(n660), .A3(n1158), .ZN(n2307) );
    ND4D0BWP7T U1448 ( .A1(n1400), .A2(n393), .A3(n2311), .A4(n2312), .ZN(n1158)
          );
    AOI211D0BWP7T U1449 ( .A1(n460), .A2(n1444), .B(n2313), .C(n1376), .ZN(n2312)
         );
    NR2D0BWP7T U1450 ( .A1(n2314), .A2(n2254), .ZN(n1376) );
    OAI22D0BWP7T U1451 ( .A1(n227), .A2(n1118), .B1(n318), .B2(n153), .ZN(n2313)
          );
    AOI22D0BWP7T U1452 ( .A1(n971), .A2(n291), .B1(n606), .B2(n294), .ZN(n2311)
```

```
      );
    AOI22D0BWP7T U1453 ( .A1(n294), .A2(n1792), .B1(n259), .B2(n1773), .ZN(n393)
      );
    OA21D0BWP7T U1454 ( .A1(n418), .A2(n2088), .B(n2315), .Z(n1400) );
    OAI21D0BWP7T U1455 ( .A1(n1534), .A2(n1140), .B(n1095), .ZN(n2315) );
    OAI211D0BWP7T U1456 ( .A1(n475), .A2(n684), .B(n986), .C(n2316), .ZN(n660)
      );
    AOI22D0BWP7T U1457 ( .A1(n2317), .A2(n291), .B1(n606), .B2(n282), .ZN(n2316)
      );
    AOI21D0BWP7T U1458 ( .A1(n294), .A2(n1773), .B(n1377), .ZN(n986) );
    NR2D0BWP7T U1459 ( .A1(n680), .A2(n874), .ZN(n1377) );
    ND3D0BWP7T U1460 ( .A1(n867), .A2(n334), .A3(n2318), .ZN(n1396) );
    AOI222D0BWP7T U1461 ( .A1(n692), .A2(n285), .B1(n438), .B2(n1444), .C1(n650),
         .C2(n294), .ZN(n2318) );
    CKND0BWP7T U1462 ( .I(n688), .ZN(n1444) );
    NR2D0BWP7T U1463 ( .A1(n602), .A2(n42), .ZN(n688) );
    AOI22D0BWP7T U1464 ( .A1(n427), .A2(n1792), .B1(n289), .B2(n1548), .ZN(n334)
      );
    CKND0BWP7T U1465 ( .I(n2052), .ZN(n1548) );
    AOI22D0BWP7T U1466 ( .A1(n269), .A2(n582), .B1(n460), .B2(n585), .ZN(n867)
      );
    OA221D0BWP7T U1467 ( .A1(n49), .A2(n577), .B1(n312), .B2(n680), .C(n2319),
         .Z(n1336) );
    AOI221D0BWP7T U1468 ( .A1(n1790), .A2(n293), .B1(n650), .B2(n583), .C(n2320),
         .ZN(n2319) );
    AOI21D0BWP7T U1469 ( .A1(n1242), .A2(n189), .B(n35), .ZN(n2320) );
    AOI21D0BWP7T U1470 ( .A1(n269), .A2(n580), .B(n957), .ZN(n1582) );
    OAI222D0BWP7T U1471 ( .A1(n196), .A2(n932), .B1(n865), .B2(n578), .C1(n408),
         .C2(n1120), .ZN(n957) );
    ND4D0BWP7T U1472 ( .A1(n1942), .A2(n26), .A3(n2321), .A4(n2322), .ZN(n95) );
    AOI221D0BWP7T U1473 ( .A1(n2136), .A2(n293), .B1(n261), .B2(n294), .C(n2323),
         .ZN(n2322) );
    OAI222D0BWP7T U1474 ( .A1(n432), .A2(n911), .B1(n2324), .B2(n975), .C1(n1507)
        , .C2(n579), .ZN(n2323) );
    AOI21D0BWP7T U1475 ( .A1(n603), .A2(n269), .B(n25), .ZN(n2321) );
    ND3D0BWP7T U1476 ( .A1(n2325), .A2(n2326), .A3(n2327), .ZN(n25) );
    AOI211D0BWP7T U1477 ( .A1(n603), .A2(n293), .B(n2328), .C(n2329), .ZN(n2327)
      );
    CKND0BWP7T U1478 ( .I(n1916), .ZN(n2329) );
    AOI211D0BWP7T U1479 ( .A1(n427), .A2(n603), .B(n1125), .C(n2330), .ZN(n1916)
      );
    IND2D0BWP7T U1480 ( .A1(n495), .B1(n952), .ZN(n2330) );
    AOI22D0BWP7T U1481 ( .A1(n268), .A2(n261), .B1(n427), .B2(n914), .ZN(n952)
      );
    OAI22D0BWP7T U1482 ( .A1(n910), .A2(n476), .B1(n196), .B2(n909), .ZN(n495)
      );
    NR2D0BWP7T U1483 ( .A1(n36), .A2(n820), .ZN(n1125) );
    OAI22D0BWP7T U1484 ( .A1(n35), .A2(n436), .B1(n318), .B2(n579), .ZN(n2328)
      );
    AOI211D0BWP7T U1485 ( .A1(n482), .A2(n1926), .B(n2331), .C(n2332), .ZN(n2326)
     );
    AOI21D0BWP7T U1486 ( .A1(n152), .A2(n435), .B(n975), .ZN(n2332) );
    AOI21D0BWP7T U1487 ( .A1(n1117), .A2(n816), .B(n909), .ZN(n2331) );
    CKND0BWP7T U1488 ( .I(n930), .ZN(n1926) );
```

```
    NR2D0BWP7T U1489 ( .A1(n976), .A2(n914), .ZN(n930) );
    AOI22D0BWP7T U1490 ( .A1(n264), .A2(n1565), .B1(n970), .B2(n285), .ZN(n2325)
          );
    AN4D0BWP7T U1491 ( .A1(n1310), .A2(n1914), .A3(n2333), .A4(n2334), .Z(n26)
          );
    AOI211D0BWP7T U1492 ( .A1(n603), .A2(n2335), .B(n2336), .C(n2337), .ZN(n2334)
         );
    AOI31D0BWP7T U1493 ( .A1(n974), .A2(n143), .A3(n1347), .B(n474), .ZN(n2337)
          );
    OAI22D0BWP7T U1494 ( .A1(n408), .A2(n417), .B1(n1803), .B2(n187), .ZN(n2336)
          );
    AOI22D0BWP7T U1495 ( .A1(n600), .A2(n583), .B1(n286), .B2(n438), .ZN(n2333)
          );
    AN4D0BWP7T U1496 ( .A1(n496), .A2(n954), .A3(n492), .A4(n2338), .Z(n1914) );
    AOI221D0BWP7T U1497 ( .A1(n286), .A2(n282), .B1(n970), .B2(n438), .C(n962),
         .ZN(n2338) );
    AOI21D0BWP7T U1498 ( .A1(n474), .A2(n909), .B(n432), .ZN(n962) );
    AOI22D0BWP7T U1499 ( .A1(n266), .A2(n914), .B1(n262), .B2(n2136), .ZN(n492)
          );
    CKND2D0BWP7T U1500 ( .A1(n143), .A2(n435), .ZN(n262) );
    AOI22D0BWP7T U1501 ( .A1(n259), .A2(n603), .B1(n482), .B2(n2136), .ZN(n954)
          );
    OA222D0BWP7T U1502 ( .A1(n1816), .A2(n36), .B1(n618), .B2(n46), .C1(n432),
         .C2(n579), .Z(n496) );
    AOI222D0BWP7T U1503 ( .A1(n296), .A2(n600), .B1(n269), .B2(n264), .C1(n2339),
         .C2(n295), .ZN(n1310) );
    AN3D0BWP7T U1504 ( .A1(n2340), .A2(n1338), .A3(n2341), .Z(n1942) );
    AOI211D0BWP7T U1505 ( .A1(n286), .A2(n293), .B(n2342), .C(n2343), .ZN(n2341)
          );
    AOI21D0BWP7T U1506 ( .A1(n36), .A2(n417), .B(n435), .ZN(n2343) );
    OAI22D0BWP7T U1507 ( .A1(n143), .A2(n476), .B1(n2344), .B2(n436), .ZN(n2342)
          );
    AOI22D0BWP7T U1508 ( .A1(n438), .A2(n292), .B1(n294), .B2(n295), .ZN(n1338)
          );
    AOI22D0BWP7T U1509 ( .A1(n261), .A2(n296), .B1(n603), .B2(n460), .ZN(n2340)
          );
    IND3D0BWP7T U1510 ( .A1(n903), .B1(n1933), .B2(n1940), .ZN(n2306) );
    AN4D0BWP7T U1511 ( .A1(n2345), .A2(n2346), .A3(n2347), .A4(n2348), .Z(n1940)
          );
    NR4D0BWP7T U1512 ( .A1(n2349), .A2(n2350), .A3(n2351), .A4(n2352), .ZN(n2348)
         );
    ND4D0BWP7T U1513 ( .A1(n2353), .A2(n2354), .A3(n2355), .A4(n2356), .ZN(n2352)
         );
    OAI21D0BWP7T U1514 ( .A1(n878), .A2(n774), .B(n633), .ZN(n2356) );
    OAI21D0BWP7T U1515 ( .A1(n585), .A2(n2078), .B(n583), .ZN(n2355) );
    IOA21D0BWP7T U1516 ( .A1(n1072), .A2(n225), .B(n31), .ZN(n2354) );
    OAI21D0BWP7T U1517 ( .A1(n574), .A2(n518), .B(n285), .ZN(n2353) );
    IND4D0BWP7T U1518 ( .A1(n1906), .B1(n2357), .B2(n271), .B3(n1808), .ZN(n2351)
         );
    CKND2D0BWP7T U1519 ( .A1(n280), .A2(n608), .ZN(n1808) );
    CKND2D0BWP7T U1520 ( .A1(n1223), .A2(n296), .ZN(n271) );
    OAI21D0BWP7T U1521 ( .A1(n280), .A2(n1521), .B(n322), .ZN(n2357) );
    NR2D0BWP7T U1522 ( .A1(n1238), .A2(n329), .ZN(n1906) );
```

```
    ND4D0BWP7T U1523 ( .A1(n2358), .A2(n2359), .A3(n2360), .A4(n2361), .ZN(n2350)
        );
    AOI222D0BWP7T U1524 ( .A1(n615), .A2(n1578), .B1(n33), .B2(n769), .C1(n216),
         .C2(n877), .ZN(n2361) );
    CKND2D0BWP7T U1525 ( .A1(n151), .A2(n1391), .ZN(n877) );
    CKND2D0BWP7T U1526 ( .A1(n1086), .A2(n1073), .ZN(n769) );
    CKND0BWP7T U1527 ( .I(n974), .ZN(n1578) );
    NR2D0BWP7T U1528 ( .A1(n693), .A2(n460), .ZN(n974) );
    AOI22D0BWP7T U1529 ( .A1(n269), .A2(n1858), .B1(n1198), .B2(n1305), .ZN(
        n2360) );
    CKND2D0BWP7T U1530 ( .A1(n227), .A2(n432), .ZN(n1305) );
    CKND2D0BWP7T U1531 ( .A1(n47), .A2(n1050), .ZN(n1858) );
    OAI31D0BWP7T U1532 ( .A1(n1799), .A2(n718), .A3(n210), .B(n266), .ZN(n2359)
         );
    OAI31D0BWP7T U1533 ( .A1(n1199), .A2(n1893), .A3(n532), .B(n517), .ZN(n2358)
         );
    ND4D0BWP7T U1534 ( .A1(n2362), .A2(n2363), .A3(n2364), .A4(n2365), .ZN(n2349)
        );
    AOI221D0BWP7T U1535 ( .A1(n587), .A2(n1521), .B1(n484), .B2(n281), .C(n2366),
         .ZN(n2365) );
    OAI22D0BWP7T U1536 ( .A1(n196), .A2(n431), .B1(n1749), .B2(n234), .ZN(n2366)
         );
    CKND0BWP7T U1537 ( .I(n415), .ZN(n281) );
    NR2D0BWP7T U1538 ( .A1(n31), .A2(n485), .ZN(n415) );
    CKND0BWP7T U1539 ( .I(n233), .ZN(n1521) );
    NR2D0BWP7T U1540 ( .A1(n29), .A2(n740), .ZN(n233) );
    AOI221D0BWP7T U1541 ( .A1(n582), .A2(n296), .B1(n1095), .B2(n740), .C(n2367),
         .ZN(n2364) );
    OAI22D0BWP7T U1542 ( .A1(n404), .A2(n325), .B1(n406), .B2(n226), .ZN(n2367)
         );
    AOI221D0BWP7T U1543 ( .A1(n1054), .A2(n755), .B1(n1678), .B2(n2368), .C(
        n2369), .ZN(n2363) );
    OAI22D0BWP7T U1544 ( .A1(n408), .A2(n1050), .B1(n154), .B2(n752), .ZN(n2369)
         );
    AN2D0BWP7T U1545 ( .A1(n2103), .A2(n438), .Z(n1054) );
    AOI222D0BWP7T U1546 ( .A1(n1887), .A2(n828), .B1(n1618), .B2(n1804), .C1(
        n282), .C2(n1005), .ZN(n2362) );
    NR4D0BWP7T U1547 ( .A1(n2370), .A2(n2371), .A3(n2372), .A4(n2373), .ZN(n2347)
        );
    OAI22D0BWP7T U1548 ( .A1(n2010), .A2(n313), .B1(n1073), .B2(n222), .ZN(n2373)
        );
    OAI22D0BWP7T U1549 ( .A1(n526), .A2(n435), .B1(n614), .B2(n1822), .ZN(n2372)
         );
    ND4D0BWP7T U1550 ( .A1(n1502), .A2(n506), .A3(n1519), .A4(n951), .ZN(n2371)
         );
    AN4D0BWP7T U1551 ( .A1(n2374), .A2(n2375), .A3(n2376), .A4(n2377), .Z(n951)
         );
    NR4D0BWP7T U1552 ( .A1(n2378), .A2(n2379), .A3(n2380), .A4(n2381), .ZN(n2377)
        );
    IND4D0BWP7T U1553 ( .A1(n1292), .B1(n376), .B2(n377), .B3(n2382), .ZN(n2381)
         );
    NR3D0BWP7T U1554 ( .A1(n21), .A2(n1058), .A3(n1204), .ZN(n2382) );
    OAI221D0BWP7T U1555 ( .A1(n435), .A2(n194), .B1(n228), .B2(n432), .C(n2383),
         .ZN(n1204) );
```

```
    AOI222D0BWP7T U1556 ( .A1(n855), .A2(n355), .B1(n1683), .B2(n1961), .C1(
        n1200), .C2(n266), .ZN(n2383) );
    OAI221D0BWP7T U1557 ( .A1(n1850), .A2(n327), .B1(n846), .B2(n820), .C(n2384),
        .ZN(n1058) );
    AOI222D0BWP7T U1558 ( .A1(n460), .A2(n1223), .B1(n1199), .B2(n515), .C1(n43),
        .C2(n1567), .ZN(n2384) );
    CKND0BWP7T U1559 ( .I(n2274), .ZN(n1199) );
    CKND2D0BWP7T U1560 ( .A1(n294), .A2(n2103), .ZN(n2274) );
    CKND2D0BWP7T U1561 ( .A1(n482), .A2(n2103), .ZN(n1850) );
    OAI22D0BWP7T U1562 ( .A1(n223), .A2(n1072), .B1(n407), .B2(n880), .ZN(n21)
        );
    AN4D0BWP7T U1563 ( .A1(n2385), .A2(n2386), .A3(n2387), .A4(n2388), .Z(n377)
        );
    AOI222D0BWP7T U1564 ( .A1(n214), .A2(n638), .B1(n587), .B2(n2389), .C1(n1095)
      , .C2(n183), .ZN(n2388) );
    CKND2D0BWP7T U1565 ( .A1(n151), .A2(n405), .ZN(n2389) );
    AOI222D0BWP7T U1566 ( .A1(n216), .A2(n1534), .B1(n1090), .B2(n34), .C1(n1078)
      , .C2(n485), .ZN(n2387) );
    AOI222D0BWP7T U1567 ( .A1(n212), .A2(n317), .B1(n630), .B2(n1101), .C1(n588),
        .C2(n483), .ZN(n2386) );
    AOI221D0BWP7T U1568 ( .A1(n632), .A2(n640), .B1(n886), .B2(n31), .C(n69),
        .ZN(n2385) );
    OAI211D0BWP7T U1569 ( .A1(n405), .A2(n404), .B(n2390), .C(n2391), .ZN(n69)
        );
    AOI211D0BWP7T U1570 ( .A1(n740), .A2(n483), .B(n2392), .C(n2393), .ZN(n2391)
        );
    AOI21D0BWP7T U1571 ( .A1(n1086), .A2(n234), .B(n1074), .ZN(n2393) );
    OAI22D0BWP7T U1572 ( .A1(n1086), .A2(n406), .B1(n312), .B2(n880), .ZN(n2392)
        );
    AOI22D0BWP7T U1573 ( .A1(n878), .A2(n608), .B1(n1490), .B2(n34), .ZN(n2390)
        );
    CKND0BWP7T U1574 ( .I(n222), .ZN(n632) );
    OA211D0BWP7T U1575 ( .A1(n327), .A2(n929), .B(n2394), .C(n2395), .Z(n376) );
    AOI222D0BWP7T U1576 ( .A1(n1898), .A2(n515), .B1(n210), .B2(n1851), .C1(n459)
      , .C2(n756), .ZN(n2395) );
    CKND0BWP7T U1577 ( .I(n1241), .ZN(n756) );
    AOI31D0BWP7T U1578 ( .A1(n268), .A2(n1193), .A3(n1978), .B(n2396), .ZN(n2394)
      );
    AOI21D0BWP7T U1579 ( .A1(n1819), .A2(n1238), .B(n229), .ZN(n2396) );
    CKND0BWP7T U1580 ( .I(n1077), .ZN(n1238) );
    NR2D0BWP7T U1581 ( .A1(n820), .A2(n1910), .ZN(n1077) );
    CKND2D0BWP7T U1582 ( .A1(n2265), .A2(n438), .ZN(n929) );
    ND4D0BWP7T U1583 ( .A1(n2397), .A2(n2398), .A3(n2399), .A4(n2400), .ZN(n1292)
      );
    NR4D0BWP7T U1584 ( .A1(n2401), .A2(n2402), .A3(n68), .A4(n1912), .ZN(n2400)
        );
    OAI222D0BWP7T U1585 ( .A1(n512), .A2(n194), .B1(n910), .B2(n738), .C1(n229),
        .C2(n1003), .ZN(n1912) );
    CKND0BWP7T U1586 ( .I(n2110), .ZN(n1003) );
    OAI221D0BWP7T U1587 ( .A1(n227), .A2(n1822), .B1(n228), .B2(n475), .C(n2403),
        .ZN(n68) );
    OA222D0BWP7T U1588 ( .A1(n2015), .A2(n329), .B1(n1603), .B2(n824), .C1(n1050)
      , .C2(n1241), .Z(n2403) );
    ND3D0BWP7T U1589 ( .A1(n1056), .A2(n1899), .A3(n1867), .ZN(n2402) );
```

```
    AOI22D0BWP7T U1590 ( .A1(n427), .A2(n1995), .B1(n1961), .B2(n828), .ZN(n1867)
        );
    CKND0BWP7T U1591 ( .I(n1959), .ZN(n1961) );
    CKND2D0BWP7T U1592 ( .A1(n2103), .A2(n296), .ZN(n1959) );
    CKND0BWP7T U1593 ( .I(n1815), .ZN(n1995) );
    MAOI22D0BWP7T U1594 ( .A1(n517), .A2(n1852), .B1(n327), .B2(n825), .ZN(n1899)
        );
    NR2D0BWP7T U1595 ( .A1(n512), .A2(n1910), .ZN(n1852) );
    AOI22D0BWP7T U1596 ( .A1(n266), .A2(n1618), .B1(n460), .B2(n1681), .ZN(n1056)
        );
    CKND0BWP7T U1597 ( .I(n434), .ZN(n1618) );
    OAI211D0BWP7T U1598 ( .A1(n865), .A2(n437), .B(n2404), .C(n2405), .ZN(n2401)
         );
    AOI222D0BWP7T U1599 ( .A1(n717), .A2(n1804), .B1(n514), .B2(n515), .C1(n1678)
       , .C2(n715), .ZN(n2405) );
    CKND0BWP7T U1600 ( .I(n1357), .ZN(n715) );
    NR2D0BWP7T U1601 ( .A1(n35), .A2(n1910), .ZN(n514) );
    CKND0BWP7T U1602 ( .I(n1440), .ZN(n1804) );
    NR2D0BWP7T U1603 ( .A1(n583), .A2(n296), .ZN(n1440) );
    MAOI22D0BWP7T U1604 ( .A1(n844), .A2(n268), .B1(n409), .B2(n1431), .ZN(n2404)
        );
    NR2D0BWP7T U1605 ( .A1(n294), .A2(n296), .ZN(n1431) );
    INR4D0BWP7T U1606 ( .A1(n161), .B1(n2406), .B2(n2407), .B3(n41), .ZN(n2399)
         );
    NR2D0BWP7T U1607 ( .A1(n1819), .A2(n824), .ZN(n41) );
    AOI21D0BWP7T U1608 ( .A1(n752), .A2(n1849), .B(n143), .ZN(n2407) );
    OAI221D0BWP7T U1609 ( .A1(n2286), .A2(n327), .B1(n1115), .B2(n47), .C(n2408),
        .ZN(n2406) );
    AO31D0BWP7T U1610 ( .A1(n814), .A2(n330), .A3(n328), .B(n329), .Z(n2408) );
    CKND2D0BWP7T U1611 ( .A1(n2265), .A2(n268), .ZN(n330) );
    NR2D0BWP7T U1612 ( .A1(n2079), .A2(n2018), .ZN(n814) );
    NR2D0BWP7T U1613 ( .A1(n1910), .A2(n865), .ZN(n2018) );
    CKND0BWP7T U1614 ( .I(n1196), .ZN(n2079) );
    CKND2D0BWP7T U1615 ( .A1(n269), .A2(n2103), .ZN(n1196) );
    NR2D0BWP7T U1616 ( .A1(n1079), .A2(n1063), .ZN(n2286) );
    CKND2D0BWP7T U1617 ( .A1(n1007), .A2(n266), .ZN(n161) );
    AOI222D0BWP7T U1618 ( .A1(n1005), .A2(n693), .B1(n1859), .B2(n1851), .C1(
        n1461), .C2(n282), .ZN(n2398) );
    AOI22D0BWP7T U1619 ( .A1(n459), .A2(n1565), .B1(n481), .B2(n2339), .ZN(n2397)
        );
    CKND0BWP7T U1620 ( .I(n1347), .ZN(n1565) );
    IND4D0BWP7T U1621 ( .A1(n1414), .B1(n60), .B2(n1093), .B3(n2409), .ZN(n2380)
         );
    NR4D0BWP7T U1622 ( .A1(n67), .A2(n1795), .A3(n1495), .A4(n1726), .ZN(n2409)
         );
    OAI22D0BWP7T U1623 ( .A1(n880), .A2(n325), .B1(n404), .B2(n1099), .ZN(n1726)
         );
    OAI221D0BWP7T U1624 ( .A1(n618), .A2(n1242), .B1(n512), .B2(n1121), .C(n2410)
       , .ZN(n1495) );
    AOI22D0BWP7T U1625 ( .A1(n2047), .A2(n583), .B1(n1107), .B2(n293), .ZN(n2410)
        );
    ND3D0BWP7T U1626 ( .A1(n2411), .A2(n555), .A3(n2412), .ZN(n1795) );
    AOI221D0BWP7T U1627 ( .A1(n1462), .A2(n296), .B1(n574), .B2(n693), .C(n2413),
        .ZN(n2412) );
```

```
    AOI21D0BWP7T U1628 ( .A1(n1242), .A2(n195), .B(n227), .ZN(n2413) );
    AOI22D0BWP7T U1629 ( .A1(n773), .A2(n1095), .B1(n289), .B2(n2414), .ZN(n555)
          );
    AOI22D0BWP7T U1630 ( .A1(n264), .A2(n266), .B1(n2136), .B2(n601), .ZN(n2411)
          );
    ND4D0BWP7T U1631 ( .A1(n2415), .A2(n2416), .A3(n2417), .A4(n2418), .ZN(n67)
          );
    AOI22D0BWP7T U1632 ( .A1(n1095), .A2(n2419), .B1(n1154), .B2(n259), .ZN(
         n2418) );
    CKND2D0BWP7T U1633 ( .A1(n510), .A2(n325), .ZN(n2419) );
    AOI22D0BWP7T U1634 ( .A1(n1773), .A2(n581), .B1(n1370), .B2(n693), .ZN(n2417)
         );
    AOI22D0BWP7T U1635 ( .A1(n1790), .A2(n2339), .B1(n692), .B2(n296), .ZN(n2416)
         );
    CKND0BWP7T U1636 ( .I(n152), .ZN(n2339) );
    AOI22D0BWP7T U1637 ( .A1(n1462), .A2(n583), .B1(n585), .B2(n601), .ZN(n2415)
          );
    OA221D0BWP7T U1638 ( .A1(n509), .A2(n772), .B1(n221), .B2(n326), .C(n2420),
         .Z(n1093) );
    AOI22D0BWP7T U1639 ( .A1(n774), .A2(n638), .B1(n608), .B2(n882), .ZN(n2420)
          );
    OA221D0BWP7T U1640 ( .A1(n875), .A2(n892), .B1(n404), .B2(n224), .C(n2421),
         .Z(n60) );
    AOI222D0BWP7T U1641 ( .A1(n1095), .A2(n1090), .B1(n322), .B2(n2422), .C1(
         n608), .C2(n214), .ZN(n2421) );
    CKND2D0BWP7T U1642 ( .A1(n347), .A2(n312), .ZN(n2422) );
    OAI211D0BWP7T U1643 ( .A1(n345), .A2(n346), .B(n2423), .C(n2424), .ZN(n1414)
          );
    AOI211D0BWP7T U1644 ( .A1(n183), .A2(n2425), .B(n2426), .C(n2427), .ZN(n2424)
         );
    AOI21D0BWP7T U1645 ( .A1(n1488), .A2(n1072), .B(n309), .ZN(n2427) );
    OAI22D0BWP7T U1646 ( .A1(n406), .A2(n680), .B1(n312), .B2(n226), .ZN(n2426)
          );
    CKND0BWP7T U1647 ( .I(n2212), .ZN(n2425) );
    AOI22D0BWP7T U1648 ( .A1(n878), .A2(n485), .B1(n630), .B2(n886), .ZN(n2423)
          );
    CKND0BWP7T U1649 ( .I(n1134), .ZN(n886) );
    ND4D0BWP7T U1650 ( .A1(n383), .A2(n1787), .A3(n167), .A4(n2428), .ZN(n2379)
          );
    INR4D0BWP7T U1651 ( .A1(n1088), .B1(n776), .B2(n1824), .B3(n1553), .ZN(n2428)
         );
    OAI211D0BWP7T U1652 ( .A1(n311), .A2(n680), .B(n2429), .C(n2430), .ZN(n1553)
          );
    AOI221D0BWP7T U1653 ( .A1(n605), .A2(n296), .B1(n42), .B2(n294), .C(n2431),
         .ZN(n2430) );
    AOI21D0BWP7T U1654 ( .A1(n436), .A2(n1352), .B(n910), .ZN(n2431) );
    AOI22D0BWP7T U1655 ( .A1(n1462), .A2(n427), .B1(n585), .B2(n269), .ZN(n2429)
          );
    OAI222D0BWP7T U1656 ( .A1(n49), .A2(n843), .B1(n229), .B2(n815), .C1(n143),
         .C2(n1237), .ZN(n1824) );
    OAI22D0BWP7T U1657 ( .A1(n346), .A2(n222), .B1(n319), .B2(n512), .ZN(n776)
          );
    AOI222D0BWP7T U1658 ( .A1(n322), .A2(n775), .B1(n727), .B2(n31), .C1(n693),
         .C2(n1007), .ZN(n1088) );
```

```
    CKND0BWP7T U1659 ( .I(n2432), .ZN(n167) );
    IND4D0BWP7T U1660 ( .A1(n872), .B1(n737), .B2(n2433), .B3(n2434), .ZN(n2432)
        );
    AOI211D0BWP7T U1661 ( .A1(n599), .A2(n296), .B(n2435), .C(n494), .ZN(n2434)
        );
    OAI22D0BWP7T U1662 ( .A1(n432), .A2(n684), .B1(n227), .B2(n1367), .ZN(n494)
        );
    OAI221D0BWP7T U1663 ( .A1(n865), .A2(n431), .B1(n143), .B2(n189), .C(n424),
        .ZN(n2435) );
    CKND2D0BWP7T U1664 ( .A1(n1773), .A2(n601), .ZN(n424) );
    NR3D0BWP7T U1665 ( .A1(n336), .A2(n1584), .A3(n1160), .ZN(n2433) );
    OAI221D0BWP7T U1666 ( .A1(n432), .A2(n431), .B1(n35), .B2(n1118), .C(n1411),
        .ZN(n1160) );
    AOI22D0BWP7T U1667 ( .A1(n630), .A2(n878), .B1(n693), .B2(n606), .ZN(n1411)
        );
    OAI22D0BWP7T U1668 ( .A1(n618), .A2(n1449), .B1(n820), .B2(n577), .ZN(n1584)
        );
    OAI222D0BWP7T U1669 ( .A1(n475), .A2(n1394), .B1(n408), .B2(n1352), .C1(
        n1170), .C2(n309), .ZN(n336) );
    OA21D0BWP7T U1670 ( .A1(n155), .A2(n820), .B(n2436), .Z(n737) );
    OAI21D0BWP7T U1671 ( .A1(n1154), .A2(n1370), .B(n438), .ZN(n2436) );
    OAI221D0BWP7T U1672 ( .A1(n475), .A2(n1352), .B1(n614), .B2(n577), .C(n2437),
        .ZN(n872) );
    AOI22D0BWP7T U1673 ( .A1(n1462), .A2(n2245), .B1(n1442), .B2(n1443), .ZN(
        n2437) );
    CKND0BWP7T U1674 ( .I(n2083), .ZN(n1442) );
    CKND2D0BWP7T U1675 ( .A1(n910), .A2(n196), .ZN(n2245) );
    CKND0BWP7T U1676 ( .I(n613), .ZN(n1462) );
    OA211D0BWP7T U1677 ( .A1(n408), .A2(n911), .B(n2438), .C(n2439), .Z(n1787)
        );
    AOI222D0BWP7T U1678 ( .A1(n606), .A2(n601), .B1(n1154), .B2(n296), .C1(n1790)
       , .C2(n583), .ZN(n2439) );
    AOI22D0BWP7T U1679 ( .A1(n582), .A2(n427), .B1(n1799), .B2(n693), .ZN(n2438)
        );
    AN3D0BWP7T U1680 ( .A1(n2440), .A2(n2441), .A3(n2442), .Z(n383) );
    AOI211D0BWP7T U1681 ( .A1(n1463), .A2(n2443), .B(n2444), .C(n1793), .ZN(
        n2442) );
    OAI22D0BWP7T U1682 ( .A1(n1120), .A2(n820), .B1(n318), .B2(n1121), .ZN(n1793)
        );
    OAI222D0BWP7T U1683 ( .A1(n143), .A2(n1352), .B1(n35), .B2(n578), .C1(n1816),
        .C2(n189), .ZN(n2444) );
    CKND0BWP7T U1684 ( .I(n2285), .ZN(n2443) );
    AOI221D0BWP7T U1685 ( .A1(n2414), .A2(n573), .B1(n438), .B2(n1789), .C(n274),
        .ZN(n2441) );
    NR2D0BWP7T U1686 ( .A1(n1351), .A2(n418), .ZN(n274) );
    CKND2D0BWP7T U1687 ( .A1(n1120), .A2(n2043), .ZN(n1789) );
    AOI22D0BWP7T U1688 ( .A1(n580), .A2(n268), .B1(n1381), .B2(n460), .ZN(n2440)
        );
    ND4D0BWP7T U1689 ( .A1(n1509), .A2(n27), .A3(n1743), .A4(n2445), .ZN(n2378)
        );
    AOI221D0BWP7T U1690 ( .A1(n995), .A2(n828), .B1(n215), .B2(n316), .C(n2446),
        .ZN(n2445) );
    OAI22D0BWP7T U1691 ( .A1(n223), .A2(n1134), .B1(n1087), .B2(n309), .ZN(n2446)
        );
```

```
    AOI22D0BWP7T U1692 ( .A1(n315), .A2(n1095), .B1(n882), .B2(n633), .ZN(n1743)
          );
    AOI22D0BWP7T U1693 ( .A1(n1925), .A2(n599), .B1(n266), .B2(n580), .ZN(n27)
          );
    AOI22D0BWP7T U1694 ( .A1(n460), .A2(n2047), .B1(n1381), .B2(n293), .ZN(n1509)
         );
    NR4D0BWP7T U1695 ( .A1(n2447), .A2(n2448), .A3(n2449), .A4(n2450), .ZN(n2376)
         );
    OAI222D0BWP7T U1696 ( .A1(n329), .A2(n757), .B1(n2285), .B2(n526), .C1(n1116)
        , .C2(n1815), .ZN(n2450) );
    NR2D0BWP7T U1697 ( .A1(n282), .A2(n269), .ZN(n1116) );
    OAI222D0BWP7T U1698 ( .A1(n227), .A2(n635), .B1(n318), .B2(n752), .C1(n512),
         .C2(n846), .ZN(n2449) );
    OAI222D0BWP7T U1699 ( .A1(n408), .A2(n207), .B1(n1489), .B2(n1170), .C1(n152)
        , .C2(n409), .ZN(n2448) );
    NR2D0BWP7T U1700 ( .A1(n282), .A2(n460), .ZN(n152) );
    CKND0BWP7T U1701 ( .I(n1055), .ZN(n207) );
    NR2D0BWP7T U1702 ( .A1(n1910), .A2(n229), .ZN(n1055) );
    OAI221D0BWP7T U1703 ( .A1(n475), .A2(n1118), .B1(n49), .B2(n1135), .C(n2451),
         .ZN(n2447) );
    AOI22D0BWP7T U1704 ( .A1(n1101), .A2(n1539), .B1(n32), .B2(n485), .ZN(n2451)
          );
    CKND2D0BWP7T U1705 ( .A1(n1703), .A2(n404), .ZN(n1539) );
    NR4D0BWP7T U1706 ( .A1(n2452), .A2(n2453), .A3(n2454), .A4(n2455), .ZN(n2375)
         );
    AOI21D0BWP7T U1707 ( .A1(n48), .A2(n849), .B(n143), .ZN(n2455) );
    NR2D0BWP7T U1708 ( .A1(n43), .A2(n604), .ZN(n48) );
    AOI21D0BWP7T U1709 ( .A1(n892), .A2(n406), .B(n313), .ZN(n2454) );
    AOI21D0BWP7T U1710 ( .A1(n1255), .A2(n325), .B(n509), .ZN(n2453) );
    IND4D0BWP7T U1711 ( .A1(n1124), .B1(n2456), .B2(n1533), .B3(n593), .ZN(n2452)
         );
    CKND2D0BWP7T U1712 ( .A1(n633), .A2(n775), .ZN(n593) );
    CKND2D0BWP7T U1713 ( .A1(n242), .A2(n633), .ZN(n1533) );
    OAI21D0BWP7T U1714 ( .A1(n532), .A2(n2457), .B(n515), .ZN(n2456) );
    NR2D0BWP7T U1715 ( .A1(n2083), .A2(n2254), .ZN(n1124) );
    CKND2D0BWP7T U1716 ( .A1(n2458), .A2(n601), .ZN(n2083) );
    NR4D0BWP7T U1717 ( .A1(n2459), .A2(n2460), .A3(n2461), .A4(n2462), .ZN(n2374)
         );
    AOI21D0BWP7T U1718 ( .A1(n817), .A2(n228), .B(n910), .ZN(n2462) );
    AOI21D0BWP7T U1719 ( .A1(n680), .A2(n875), .B(n1074), .ZN(n2461) );
    OAI22D0BWP7T U1720 ( .A1(n1979), .A2(n226), .B1(n1874), .B2(n196), .ZN(n2460)
         );
    NR2D0BWP7T U1721 ( .A1(n588), .A2(n32), .ZN(n1979) );
    OAI222D0BWP7T U1722 ( .A1(n1366), .A2(n1784), .B1(n1256), .B2(n1596), .C1(
         n726), .C2(n875), .ZN(n2459) );
    NR2D0BWP7T U1723 ( .A1(n1140), .A2(n774), .ZN(n726) );
    NR2D0BWP7T U1724 ( .A1(n438), .A2(n693), .ZN(n1256) );
    AOI22D0BWP7T U1725 ( .A1(n882), .A2(n216), .B1(n482), .B2(n518), .ZN(n1519)
          );
    AOI22D0BWP7T U1726 ( .A1(n773), .A2(n587), .B1(n34), .B2(n33), .ZN(n506) );
    AOI22D0BWP7T U1727 ( .A1(n438), .A2(n2136), .B1(n427), .B2(n599), .ZN(n1502)
          );
    IIND4D0BWP7T U1728 ( .A1(n1796), .A2(n1741), .B1(n59), .B2(n2463), .ZN(n2370)
         );
```

```
    INR4D0BWP7T U1729 ( .A1(n1786), .B1(n1554), .B2(n1175), .B3(n1775), .ZN(
        n2463) );
    OAI221D0BWP7T U1730 ( .A1(n1348), .A2(n2255), .B1(n1350), .B2(n2052), .C(
        n2464), .ZN(n1775) );
    AOI22D0BWP7T U1731 ( .A1(n602), .A2(n427), .B1(n2136), .B2(n460), .ZN(n2464)
        );
    CKND2D0BWP7T U1732 ( .A1(n2458), .A2(n482), .ZN(n2052) );
    OAI222D0BWP7T U1733 ( .A1(n875), .A2(n1529), .B1(n1391), .B2(n1240), .C1(
        n227), .C2(n319), .ZN(n1175) );
    OAI221D0BWP7T U1734 ( .A1(n966), .A2(n1449), .B1(n418), .B2(n2051), .C(n2465)
       , .ZN(n1554) );
    CKND0BWP7T U1735 ( .I(n1413), .ZN(n2465) );
    OAI221D0BWP7T U1736 ( .A1(n143), .A2(n1784), .B1(n309), .B2(n1099), .C(n2466)
       , .ZN(n1413) );
    AOI22D0BWP7T U1737 ( .A1(n605), .A2(n460), .B1(n2090), .B2(n291), .ZN(n2466)
        );
    NR2D0BWP7T U1738 ( .A1(n293), .A2(n294), .ZN(n966) );
    AOI222D0BWP7T U1739 ( .A1(n269), .A2(n606), .B1(n693), .B2(n599), .C1(n438),
        .C2(n1463), .ZN(n1786) );
    CKND0BWP7T U1740 ( .I(n1449), .ZN(n1463) );
    AN4D0BWP7T U1741 ( .A1(n2467), .A2(n2468), .A3(n2469), .A4(n2470), .Z(n59)
        );
    AOI211D0BWP7T U1742 ( .A1(n1101), .A2(n2471), .B(n2472), .C(n2473), .ZN(
        n2470) );
    AOI21D0BWP7T U1743 ( .A1(n1703), .A2(n875), .B(n406), .ZN(n2473) );
    NR2D0BWP7T U1744 ( .A1(n587), .A2(n483), .ZN(n1703) );
    AOI21D0BWP7T U1745 ( .A1(n222), .A2(n1134), .B(n226), .ZN(n2472) );
    CKND2D0BWP7T U1746 ( .A1(n509), .A2(n680), .ZN(n2471) );
    AOI22D0BWP7T U1747 ( .A1(n214), .A2(n587), .B1(n216), .B2(n1090), .ZN(n2469)
        );
    AOI22D0BWP7T U1748 ( .A1(n485), .A2(n773), .B1(n1007), .B2(n601), .ZN(n2468)
        );
    AOI21D0BWP7T U1749 ( .A1(n31), .A2(n242), .B(n1412), .ZN(n2467) );
    OAI21D0BWP7T U1750 ( .A1(n225), .A2(n1073), .B(n866), .ZN(n1412) );
    CKND2D0BWP7T U1751 ( .A1(n1140), .A2(n633), .ZN(n866) );
    NR2D0BWP7T U1752 ( .A1(n315), .A2(n631), .ZN(n225) );
    OAI221D0BWP7T U1753 ( .A1(n1170), .A2(n226), .B1(n1134), .B2(n313), .C(n2474)
       , .ZN(n1741) );
    AOI222D0BWP7T U1754 ( .A1(n640), .A2(n740), .B1(n182), .B2(n2475), .C1(n322),
        .C2(n1708), .ZN(n2474) );
    CKND2D0BWP7T U1755 ( .A1(n1528), .A2(n1099), .ZN(n1708) );
    CKND2D0BWP7T U1756 ( .A1(n223), .A2(n875), .ZN(n2475) );
    OAI211D0BWP7T U1757 ( .A1(n1240), .A2(n151), .B(n2476), .C(n2477), .ZN(n1796)
        );
    AOI221D0BWP7T U1758 ( .A1(n1773), .A2(n427), .B1(n691), .B2(n573), .C(n2478),
        .ZN(n2477) );
    AOI21D0BWP7T U1759 ( .A1(n613), .A2(n153), .B(n618), .ZN(n2478) );
    CKND0BWP7T U1760 ( .I(n1349), .ZN(n691) );
    AOI22D0BWP7T U1761 ( .A1(n599), .A2(n259), .B1(n1790), .B2(n601), .ZN(n2476)
        );
    INR4D0BWP7T U1762 ( .A1(n2479), .B1(n1499), .B2(n1092), .B3(n554), .ZN(n2346)
        );
    ND4D0BWP7T U1763 ( .A1(n74), .A2(n1557), .A3(n2480), .A4(n2481), .ZN(n554)
        );
```

```
    NR3D0BWP7T U1764 ( .A1(n2482), .A2(n1497), .A3(n382), .ZN(n2481) );
    OAI211D0BWP7T U1765 ( .A1(n614), .A2(n911), .B(n2483), .C(n2484), .ZN(n382)
          );
    AOI222D0BWP7T U1766 ( .A1(n293), .A2(n42), .B1(n288), .B2(n1443), .C1(n2414),
         .C2(n291), .ZN(n2484) );
    CKND0BWP7T U1767 ( .I(n2088), .ZN(n288) );
    CKND2D0BWP7T U1768 ( .A1(n2458), .A2(n583), .ZN(n2088) );
    AOI22D0BWP7T U1769 ( .A1(n2047), .A2(n259), .B1(n585), .B2(n296), .ZN(n2483)
          );
    OAI221D0BWP7T U1770 ( .A1(n512), .A2(n1118), .B1(n309), .B2(n510), .C(n2485),
         .ZN(n1497) );
    AOI22D0BWP7T U1771 ( .A1(n580), .A2(n438), .B1(n2047), .B2(n427), .ZN(n2485)
          );
    IND3D0BWP7T U1772 ( .A1(n1550), .B1(n1780), .B2(n1797), .ZN(n2482) );
    OA221D0BWP7T U1773 ( .A1(n318), .A2(n195), .B1(n435), .B2(n932), .C(n2486),
         .Z(n1797) );
    AOI22D0BWP7T U1774 ( .A1(n1773), .A2(n282), .B1(n585), .B2(n293), .ZN(n2486)
          );
    AOI22D0BWP7T U1775 ( .A1(n285), .A2(n264), .B1(n573), .B2(n2487), .ZN(n1780)
          );
    OAI222D0BWP7T U1776 ( .A1(n865), .A2(n1471), .B1(n912), .B2(n617), .C1(n892),
         .C2(n1240), .ZN(n1550) );
    NR2D0BWP7T U1777 ( .A1(n293), .A2(n693), .ZN(n912) );
    AOI211D0BWP7T U1778 ( .A1(n572), .A2(n1443), .B(n163), .C(n2488), .ZN(n2480)
          );
    AOI21D0BWP7T U1779 ( .A1(n189), .A2(n1118), .B(n614), .ZN(n2488) );
    IIND4D0BWP7T U1780 ( .A1(n1573), .A2(n871), .B1(n450), .B2(n2489), .ZN(n163)
          );
    AOI211D0BWP7T U1781 ( .A1(n1154), .A2(n1851), .B(n2490), .C(n270), .ZN(n2489)
        );
    NR2D0BWP7T U1782 ( .A1(n684), .A2(n820), .ZN(n270) );
    IND2D0BWP7T U1783 ( .A1(n1151), .B1(n647), .ZN(n2490) );
    CKND2D0BWP7T U1784 ( .A1(n42), .A2(n282), .ZN(n647) );
    NR2D0BWP7T U1785 ( .A1(n155), .A2(n683), .ZN(n1151) );
    CKND2D0BWP7T U1786 ( .A1(n2317), .A2(n573), .ZN(n450) );
    CKND0BWP7T U1787 ( .I(n419), .ZN(n2317) );
    OAI222D0BWP7T U1788 ( .A1(n1350), .A2(n2255), .B1(n1348), .B2(n2491), .C1(
         n143), .C2(n1449), .ZN(n871) );
    NR2D0BWP7T U1789 ( .A1(n2492), .A2(n2254), .ZN(n1573) );
    AOI22D0BWP7T U1790 ( .A1(n693), .A2(n1788), .B1(n427), .B2(n42), .ZN(n1557)
          );
    CKND0BWP7T U1791 ( .I(n431), .ZN(n42) );
    AOI22D0BWP7T U1792 ( .A1(n583), .A2(n1799), .B1(n482), .B2(n585), .ZN(n74)
          );
    OAI221D0BWP7T U1793 ( .A1(n346), .A2(n2010), .B1(n1086), .B2(n222), .C(n2493)
        , .ZN(n1092) );
    AOI222D0BWP7T U1794 ( .A1(n1534), .A2(n1102), .B1(n214), .B2(n322), .C1(n740)
        , .C2(n30), .ZN(n2493) );
    CKND0BWP7T U1795 ( .I(n311), .ZN(n214) );
    CKND0BWP7T U1796 ( .I(n1489), .ZN(n1102) );
    NR2D0BWP7T U1797 ( .A1(n31), .A2(n34), .ZN(n1489) );
    CKND0BWP7T U1798 ( .I(n892), .ZN(n1534) );
    CKND2D0BWP7T U1799 ( .A1(n2494), .A2(n269), .ZN(n222) );
    OAI211D0BWP7T U1800 ( .A1(n408), .A2(n613), .B(n1332), .C(n2495), .ZN(n1499)
```

```
      );
    AOI221D0BWP7T U1801 ( .A1(n1122), .A2(n294), .B1(n605), .B2(n285), .C(n2496),
         .ZN(n2495) );
    AOI31D0BWP7T U1802 ( .A1(n1120), .A2(n431), .A3(n1156), .B(n49), .ZN(n2496)
      );
    CKND2D0BWP7T U1803 ( .A1(n2497), .A2(n1978), .ZN(n431) );
    CKND0BWP7T U1804 ( .I(n1118), .ZN(n605) );
    AOI22D0BWP7T U1805 ( .A1(n601), .A2(n1788), .B1(n1443), .B2(n2414), .ZN(
         n1332) );
    CKND0BWP7T U1806 ( .I(n203), .ZN(n2414) );
    CKND2D0BWP7T U1807 ( .A1(n2458), .A2(n269), .ZN(n203) );
    INR4D0BWP7T U1808 ( .A1(n1408), .B1(n1719), .B2(n1334), .B3(n1404), .ZN(
         n2479) );
    ND4D0BWP7T U1809 ( .A1(n2498), .A2(n2499), .A3(n2500), .A4(n2501), .ZN(n1404)
        );
    NR4D0BWP7T U1810 ( .A1(n2502), .A2(n2503), .A3(n2504), .A4(n2505), .ZN(n2501)
        );
    AOI31D0BWP7T U1811 ( .A1(n312), .A2(n326), .A3(n1529), .B(n313), .ZN(n2505)
      );
    AOI21D0BWP7T U1812 ( .A1(n1528), .A2(n1529), .B(n223), .ZN(n2504) );
    AOI21D0BWP7T U1813 ( .A1(n345), .A2(n311), .B(n404), .ZN(n2503) );
    AOI21D0BWP7T U1814 ( .A1(n151), .A2(n347), .B(n234), .ZN(n2502) );
    AOI221D0BWP7T U1815 ( .A1(n633), .A2(n891), .B1(n485), .B2(n2506), .C(n2507),
         .ZN(n2500) );
    OAI22D0BWP7T U1816 ( .A1(n1752), .A2(n772), .B1(n751), .B2(n319), .ZN(n2507)
      );
    NR2D0BWP7T U1817 ( .A1(n282), .A2(n296), .ZN(n751) );
    NR2D0BWP7T U1818 ( .A1(n316), .A2(n485), .ZN(n1752) );
    CKND0BWP7T U1819 ( .I(n1100), .ZN(n2506) );
    NR2D0BWP7T U1820 ( .A1(n317), .A2(n280), .ZN(n1100) );
    CKND2D0BWP7T U1821 ( .A1(n320), .A2(n874), .ZN(n891) );
    AOI221D0BWP7T U1822 ( .A1(n774), .A2(n2236), .B1(n215), .B2(n322), .C(n2508),
         .ZN(n2499) );
    OAI22D0BWP7T U1823 ( .A1(n346), .A2(n1488), .B1(n1073), .B2(n1134), .ZN(
         n2508) );
    CKND2D0BWP7T U1824 ( .A1(n2509), .A2(n438), .ZN(n1134) );
    CKND0BWP7T U1825 ( .I(n1072), .ZN(n215) );
    CKND0BWP7T U1826 ( .I(n890), .ZN(n2236) );
    NR2D0BWP7T U1827 ( .A1(n316), .A2(n30), .ZN(n890) );
    AOI221D0BWP7T U1828 ( .A1(n630), .A2(n242), .B1(n608), .B2(n29), .C(n77),
         .ZN(n2498) );
    OAI211D0BWP7T U1829 ( .A1(n313), .A2(n224), .B(n869), .C(n2510), .ZN(n77) );
    AOI221D0BWP7T U1830 ( .A1(n1007), .A2(n438), .B1(n587), .B2(n2511), .C(n2512)
        , .ZN(n2510) );
    AOI21D0BWP7T U1831 ( .A1(n874), .A2(n326), .B(n150), .ZN(n2512) );
    AOI22D0BWP7T U1832 ( .A1(n638), .A2(n484), .B1(n322), .B2(n773), .ZN(n869)
      );
    ND4D0BWP7T U1833 ( .A1(n2513), .A2(n2514), .A3(n2515), .A4(n2516), .ZN(n1334)
        );
    OA222D0BWP7T U1834 ( .A1(n346), .A2(n2033), .B1(n345), .B2(n1728), .C1(n1087)
        , .C2(n1732), .Z(n2516) );
    NR2D0BWP7T U1835 ( .A1(n485), .A2(n633), .ZN(n1732) );
    NR2D0BWP7T U1836 ( .A1(n483), .A2(n322), .ZN(n1728) );
    NR2D0BWP7T U1837 ( .A1(n29), .A2(n727), .ZN(n2033) );
```

```
    CKND0BWP7T U1838 ( .I(n1087), .ZN(n29) );
    CKND2D0BWP7T U1839 ( .A1(n601), .A2(n2509), .ZN(n1087) );
    AOI22D0BWP7T U1840 ( .A1(n324), .A2(n483), .B1(n31), .B2(n1101), .ZN(n2515)
          );
    CKND0BWP7T U1841 ( .I(n347), .ZN(n1101) );
    CKND2D0BWP7T U1842 ( .A1(n2509), .A2(n268), .ZN(n347) );
    CKND0BWP7T U1843 ( .I(n1099), .ZN(n324) );
    AOI22D0BWP7T U1844 ( .A1(n315), .A2(n638), .B1(n631), .B2(n640), .ZN(n2514)
          );
    AOI21D0BWP7T U1845 ( .A1(n630), .A2(n484), .B(n1714), .ZN(n2513) );
    OAI22D0BWP7T U1846 ( .A1(n313), .A2(n151), .B1(n1072), .B2(n680), .ZN(n1714)
          );
    OAI221D0BWP7T U1847 ( .A1(n150), .A2(n892), .B1(n326), .B2(n226), .C(n2517),
         .ZN(n1719) );
    AOI22D0BWP7T U1848 ( .A1(n518), .A2(n269), .B1(n216), .B2(n775), .ZN(n2517)
          );
    OA211D0BWP7T U1849 ( .A1(n1170), .A2(n404), .B(n462), .C(n2518), .Z(n1408)
          );
    AOI22D0BWP7T U1850 ( .A1(n640), .A2(n882), .B1(n182), .B2(n30), .ZN(n2518)
          );
    CKND0BWP7T U1851 ( .I(n510), .ZN(n182) );
    CKND2D0BWP7T U1852 ( .A1(n33), .A2(n608), .ZN(n462) );
    INR4D0BWP7T U1853 ( .A1(n2519), .B1(n1340), .B2(n1407), .B3(n64), .ZN(n2345)
          );
    ND4D0BWP7T U1854 ( .A1(n2520), .A2(n2521), .A3(n2522), .A4(n2523), .ZN(n64)
          );
    AOI222D0BWP7T U1855 ( .A1(n827), .A2(n515), .B1(n1894), .B2(n1683), .C1(n844)
        , .C2(n2524), .ZN(n2523) );
    AOI22D0BWP7T U1856 ( .A1(n477), .A2(n355), .B1(n1898), .B2(n828), .ZN(n2522)
          );
    CKND0BWP7T U1857 ( .I(n2015), .ZN(n1898) );
    CKND2D0BWP7T U1858 ( .A1(n2265), .A2(n282), .ZN(n2015) );
    AOI22D0BWP7T U1859 ( .A1(n714), .A2(n268), .B1(n583), .B2(n1200), .ZN(n2521)
          );
    AOI22D0BWP7T U1860 ( .A1(n1859), .A2(n285), .B1(n718), .B2(n269), .ZN(n2520)
          );
    OAI211D0BWP7T U1861 ( .A1(n824), .A2(n825), .B(n2525), .C(n2526), .ZN(n1407)
          );
    AOI211D0BWP7T U1862 ( .A1(n844), .A2(n693), .B(n1219), .C(n1057), .ZN(n2526)
          );
    OAI22D0BWP7T U1863 ( .A1(n512), .A2(n1237), .B1(n435), .B2(n843), .ZN(n1057)
          );
    OAI22D0BWP7T U1864 ( .A1(n432), .A2(n819), .B1(n753), .B2(n475), .ZN(n1219)
          );
    CKND0BWP7T U1865 ( .I(n1967), .ZN(n753) );
    AOI22D0BWP7T U1866 ( .A1(n518), .A2(n438), .B1(n1894), .B2(n517), .ZN(n2525)
          );
    ND4D0BWP7T U1867 ( .A1(n712), .A2(n1052), .A3(n2527), .A4(n2528), .ZN(n1340)
          );
    AOI221D0BWP7T U1868 ( .A1(n532), .A2(n828), .B1(n844), .B2(n482), .C(n2529),
         .ZN(n2528) );
    OAI211D0BWP7T U1869 ( .A1(n229), .A2(n1892), .B(n2530), .C(n2531), .ZN(n2529)
        );
    IOA21D0BWP7T U1870 ( .A1(n738), .A2(n1817), .B(n601), .ZN(n2531) );
```

```
    NR2D0BWP7T U1871 ( .A1(n1200), .A2(n43), .ZN(n1817) );
    OAI21D0BWP7T U1872 ( .A1(n1678), .A2(n1198), .B(n268), .ZN(n2530) );
    CKND0BWP7T U1873 ( .I(n1822), .ZN(n844) );
    CKND0BWP7T U1874 ( .I(n837), .ZN(n532) );
    AOI211D0BWP7T U1875 ( .A1(n1223), .A2(n1851), .B(n2532), .C(n747), .ZN(n2527)
        );
    NR2D0BWP7T U1876 ( .A1(n408), .A2(n1237), .ZN(n747) );
    AOI21D0BWP7T U1877 ( .A1(n1874), .A2(n1596), .B(n35), .ZN(n2532) );
    NR2D0BWP7T U1878 ( .A1(n717), .A2(n210), .ZN(n1874) );
    CKND0BWP7T U1879 ( .I(n1117), .ZN(n1851) );
    NR2D0BWP7T U1880 ( .A1(n583), .A2(n693), .ZN(n1117) );
    CKND0BWP7T U1881 ( .I(n47), .ZN(n1223) );
    OA221D0BWP7T U1882 ( .A1(n435), .A2(n236), .B1(n143), .B2(n1815), .C(n2533),
         .Z(n1052) );
    AOI22D0BWP7T U1883 ( .A1(n516), .A2(n515), .B1(n459), .B2(n583), .ZN(n2533)
        );
    CKND0BWP7T U1884 ( .I(n817), .ZN(n459) );
    AOI22D0BWP7T U1885 ( .A1(n1683), .A2(n1996), .B1(n296), .B2(n2170), .ZN(n712)
        );
    CKND0BWP7T U1886 ( .I(n815), .ZN(n1996) );
    CKND2D0BWP7T U1887 ( .A1(n259), .A2(n2103), .ZN(n815) );
    INR4D0BWP7T U1888 ( .A1(n166), .B1(n1861), .B2(n1403), .B3(n63), .ZN(n2519)
        );
    OAI211D0BWP7T U1889 ( .A1(n618), .A2(n1394), .B(n2534), .C(n2535), .ZN(n63)
        );
    AOI211D0BWP7T U1890 ( .A1(n1792), .A2(n296), .B(n1335), .C(n1420), .ZN(n2535)
        );
    OAI222D0BWP7T U1891 ( .A1(n408), .A2(n1367), .B1(n143), .B2(n1121), .C1(n510)
        , .C2(n313), .ZN(n1420) );
    CKND2D0BWP7T U1892 ( .A1(n2494), .A2(n693), .ZN(n510) );
    OAI221D0BWP7T U1893 ( .A1(n196), .A2(n1471), .B1(n2254), .B2(n419), .C(n2536)
        , .ZN(n1335) );
    AOI221D0BWP7T U1894 ( .A1(n1122), .A2(n583), .B1(n2047), .B2(n266), .C(n2537)
        , .ZN(n2536) );
    AOI21D0BWP7T U1895 ( .A1(n613), .A2(n155), .B(n143), .ZN(n2537) );
    AOI21D0BWP7T U1896 ( .A1(n650), .A2(n427), .B(n2538), .ZN(n2534) );
    AOI21D0BWP7T U1897 ( .A1(n1135), .A2(n1118), .B(n820), .ZN(n2538) );
    IND4D0BWP7T U1898 ( .A1(n76), .B1(n1798), .B2(n2539), .B3(n2540), .ZN(n1403)
        );
    AOI211D0BWP7T U1899 ( .A1(n1792), .A2(n285), .B(n2541), .C(n1783), .ZN(n2540)
        );
    OAI22D0BWP7T U1900 ( .A1(n143), .A2(n684), .B1(n432), .B2(n932), .ZN(n1783)
        );
    OAI22D0BWP7T U1901 ( .A1(n408), .A2(n617), .B1(n818), .B2(n1471), .ZN(n2541)
        );
    NR2D0BWP7T U1902 ( .A1(n282), .A2(n266), .ZN(n818) );
    NR2D0BWP7T U1903 ( .A1(n1551), .A2(n1496), .ZN(n2539) );
    OAI22D0BWP7T U1904 ( .A1(n1816), .A2(n613), .B1(n2542), .B2(n196), .ZN(n1496)
        );
    NR2D0BWP7T U1905 ( .A1(n692), .A2(n606), .ZN(n2542) );
    OAI221D0BWP7T U1906 ( .A1(n910), .A2(n1156), .B1(n196), .B2(n1121), .C(n2543)
        , .ZN(n1551) );
    OAI21D0BWP7T U1907 ( .A1(n2090), .A2(n971), .B(n573), .ZN(n2543) );
    CKND0BWP7T U1908 ( .I(n2491), .ZN(n2090) );
```

```
    CKND2D0BWP7T U1909 ( .A1(n2458), .A2(n259), .ZN(n2491) );
    AOI222D0BWP7T U1910 ( .A1(n268), .A2(n2136), .B1(n289), .B2(n290), .C1(n774),
         .C2(n1095), .ZN(n1798) );
    CKND0BWP7T U1911 ( .I(n1320), .ZN(n774) );
    CKND0BWP7T U1912 ( .I(n2310), .ZN(n290) );
    OAI221D0BWP7T U1913 ( .A1(n2254), .A2(n2253), .B1(n1348), .B2(n419), .C(
         n2544), .ZN(n76) );
    AOI22D0BWP7T U1914 ( .A1(n599), .A2(n438), .B1(n585), .B2(n427), .ZN(n2544)
          );
    CKND0BWP7T U1915 ( .I(n1784), .ZN(n585) );
    CKND2D0BWP7T U1916 ( .A1(n2458), .A2(n294), .ZN(n419) );
    OAI221D0BWP7T U1917 ( .A1(n865), .A2(n1237), .B1(n49), .B2(n236), .C(n2545),
         .ZN(n1861) );
    AOI22D0BWP7T U1918 ( .A1(n1062), .A2(n1683), .B1(n827), .B2(n517), .ZN(n2545)
        );
    CKND0BWP7T U1919 ( .I(n1971), .ZN(n1062) );
    AN4D0BWP7T U1920 ( .A1(n985), .A2(n1401), .A3(n2546), .A4(n2547), .Z(n166)
          );
    NR4D0BWP7T U1921 ( .A1(n1018), .A2(n1159), .A3(n1583), .A4(n870), .ZN(n2547)
          );
    OAI22D0BWP7T U1922 ( .A1(n1348), .A2(n1349), .B1(n1357), .B2(n1118), .ZN(
         n870) );
    CKND2D0BWP7T U1923 ( .A1(n2548), .A2(n2103), .ZN(n1118) );
    NR2D0BWP7T U1924 ( .A1(n269), .A2(n259), .ZN(n1357) );
    CKND2D0BWP7T U1925 ( .A1(n2458), .A2(n293), .ZN(n1349) );
    OAI221D0BWP7T U1926 ( .A1(n618), .A2(n2043), .B1(n1348), .B2(n2253), .C(
         n2549), .ZN(n1583) );
    MAOI22D0BWP7T U1927 ( .A1(n1370), .A2(n259), .B1(n2051), .B2(n2254), .ZN(
         n2549) );
    CKND0BWP7T U1928 ( .I(n289), .ZN(n2254) );
    CKND2D0BWP7T U1929 ( .A1(n2458), .A2(n285), .ZN(n2051) );
    CKND2D0BWP7T U1930 ( .A1(n2458), .A2(n296), .ZN(n2253) );
    ND4D0BWP7T U1931 ( .A1(n2550), .A2(n2551), .A3(n2552), .A4(n2553), .ZN(n1159)
        );
    AOI22D0BWP7T U1932 ( .A1(n2047), .A2(n285), .B1(n2138), .B2(n573), .ZN(n2553)
        );
    AOI22D0BWP7T U1933 ( .A1(n1107), .A2(n427), .B1(n1788), .B2(n482), .ZN(n2552)
        );
    AOI22D0BWP7T U1934 ( .A1(n1154), .A2(n460), .B1(n1792), .B2(n693), .ZN(n2551)
        );
    AOI22D0BWP7T U1935 ( .A1(n692), .A2(n583), .B1(n606), .B2(n266), .ZN(n2550)
          );
    ND4D0BWP7T U1936 ( .A1(n2554), .A2(n2555), .A3(n2556), .A4(n2557), .ZN(n1018)
        );
    AOI211D0BWP7T U1937 ( .A1(n1095), .A2(n727), .B(n2558), .C(n1397), .ZN(n2557)
        );
    OAI221D0BWP7T U1938 ( .A1(n475), .A2(n2043), .B1(n910), .B2(n1121), .C(n2559)
       , .ZN(n1397) );
    AOI221D0BWP7T U1939 ( .A1(n1154), .A2(n293), .B1(n1788), .B2(n268), .C(n461),
         .ZN(n2559) );
    NR2D0BWP7T U1940 ( .A1(n2314), .A2(n1348), .ZN(n461) );
    CKND0BWP7T U1941 ( .I(n1443), .ZN(n1348) );
    CKND0BWP7T U1942 ( .I(n1394), .ZN(n1788) );
    CKND0BWP7T U1943 ( .I(n1471), .ZN(n1154) );
```

```
    OAI22D0BWP7T U1944 ( .A1(n196), .A2(n617), .B1(n865), .B2(n1352), .ZN(n2558)
          );
    AOI22D0BWP7T U1945 ( .A1(n580), .A2(n2335), .B1(n2047), .B2(n438), .ZN(n2556)
        );
    CKND2D0BWP7T U1946 ( .A1(n910), .A2(n475), .ZN(n2335) );
    OAI21D0BWP7T U1947 ( .A1(n1171), .A2(n33), .B(n630), .ZN(n2555) );
    OAI21D0BWP7T U1948 ( .A1(n1381), .A2(n1370), .B(n427), .ZN(n2554) );
    CKND0BWP7T U1949 ( .I(n1156), .ZN(n1370) );
    CKND2D0BWP7T U1950 ( .A1(n2560), .A2(n2265), .ZN(n1156) );
    AOI211D0BWP7T U1951 ( .A1(n574), .A2(n266), .B(n735), .C(n1587), .ZN(n2546)
          );
    OAI22D0BWP7T U1952 ( .A1(n311), .A2(n1240), .B1(n618), .B2(n1120), .ZN(n1587)
        );
    OAI22D0BWP7T U1953 ( .A1(n49), .A2(n1242), .B1(n820), .B2(n1394), .ZN(n735)
          );
    AOI22D0BWP7T U1954 ( .A1(n631), .A2(n630), .B1(n438), .B2(n582), .ZN(n1401)
          );
    OA221D0BWP7T U1955 ( .A1(n2042), .A2(n910), .B1(n49), .B2(n617), .C(n2561),
         .Z(n985) );
    OAI21D0BWP7T U1956 ( .A1(n650), .A2(n1799), .B(n296), .ZN(n2561) );
    CKND0BWP7T U1957 ( .I(n155), .ZN(n650) );
    NR2D0BWP7T U1958 ( .A1(n599), .A2(n1381), .ZN(n2042) );
    CKND0BWP7T U1959 ( .I(n153), .ZN(n1381) );
    CKND0BWP7T U1960 ( .I(n1135), .ZN(n599) );
    AN3D0BWP7T U1961 ( .A1(n2562), .A2(n2563), .A3(n2564), .Z(n1933) );
    NR4D0BWP7T U1962 ( .A1(n2565), .A2(n1913), .A3(n1059), .A4(n1597), .ZN(n2564)
        );
    OAI221D0BWP7T U1963 ( .A1(n227), .A2(n819), .B1(n865), .B2(n236), .C(n2566),
         .ZN(n1597) );
    AOI22D0BWP7T U1964 ( .A1(n1005), .A2(n296), .B1(n266), .B2(n639), .ZN(n2566)
          );
    OAI211D0BWP7T U1965 ( .A1(n229), .A2(n2567), .B(n2568), .C(n2569), .ZN(n1059)
        );
    AOI22D0BWP7T U1966 ( .A1(n583), .A2(n1680), .B1(n717), .B2(n266), .ZN(n2569)
          );
    CKND0BWP7T U1967 ( .I(n1849), .ZN(n1680) );
    CKND2D0BWP7T U1968 ( .A1(n2265), .A2(n517), .ZN(n1849) );
    OAI21D0BWP7T U1969 ( .A1(n639), .A2(n210), .B(n294), .ZN(n2568) );
    CKND0BWP7T U1970 ( .I(n1679), .ZN(n2567) );
    OAI211D0BWP7T U1971 ( .A1(n227), .A2(n236), .B(n272), .C(n421), .ZN(n1913)
          );
    CKND2D0BWP7T U1972 ( .A1(n995), .A2(n515), .ZN(n421) );
    CKND2D0BWP7T U1973 ( .A1(n1681), .A2(n269), .ZN(n272) );
    IND3D0BWP7T U1974 ( .A1(n65), .B1(n821), .B2(n1884), .ZN(n2565) );
    AOI22D0BWP7T U1975 ( .A1(n515), .A2(n1887), .B1(n355), .B2(n374), .ZN(n1884)
          );
    CKND0BWP7T U1976 ( .I(n328), .ZN(n1887) );
    AOI22D0BWP7T U1977 ( .A1(n460), .A2(n1198), .B1(n259), .B2(n1461), .ZN(n821)
          );
    CKND0BWP7T U1978 ( .I(n1050), .ZN(n1461) );
    CKND0BWP7T U1979 ( .I(n194), .ZN(n1198) );
    OAI221D0BWP7T U1980 ( .A1(n196), .A2(n437), .B1(n143), .B2(n1596), .C(n2570),
         .ZN(n65) );
    AOI222D0BWP7T U1981 ( .A1(n604), .A2(n438), .B1(n601), .B2(n2571), .C1(n828),
```

```
      .C2(n1893), .ZN(n2570) );
    CKND2D0BWP7T U1982 ( .A1(n817), .A2(n47), .ZN(n2571) );
    CKND2D0BWP7T U1983 ( .A1(n2572), .A2(n1978), .ZN(n47) );
    CKND2D0BWP7T U1984 ( .A1(n1683), .A2(n2573), .ZN(n817) );
    CKND0BWP7T U1985 ( .I(n409), .ZN(n604) );
    AOI211D0BWP7T U1986 ( .A1(n1967), .A2(n693), .B(n2574), .C(n2575), .ZN(n2563)
        );
    AOI21D0BWP7T U1987 ( .A1(n409), .A2(n1822), .B(n49), .ZN(n2575) );
    CKND2D0BWP7T U1988 ( .A1(n2573), .A2(n2576), .ZN(n1822) );
    CKND2D0BWP7T U1989 ( .A1(n2577), .A2(n2103), .ZN(n409) );
    AOI21D0BWP7T U1990 ( .A1(n819), .A2(n434), .B(n910), .ZN(n2574) );
    CKND2D0BWP7T U1991 ( .A1(n2265), .A2(n1683), .ZN(n434) );
    NR2D0BWP7T U1992 ( .A1(n329), .A2(n1910), .ZN(n1967) );
    AOI222D0BWP7T U1993 ( .A1(n1859), .A2(n296), .B1(n1679), .B2(n828), .C1(
        n1894), .C2(n515), .ZN(n2562) );
    NR2D0BWP7T U1994 ( .A1(n143), .A2(n1910), .ZN(n1894) );
    NR2D0BWP7T U1995 ( .A1(n432), .A2(n1910), .ZN(n1679) );
    CKND0BWP7T U1996 ( .I(n849), .ZN(n1859) );
    CKND2D0BWP7T U1997 ( .A1(n2265), .A2(n515), .ZN(n849) );
    ND3D0BWP7T U1998 ( .A1(n2578), .A2(n2579), .A3(n2580), .ZN(n903) );
    INR4D0BWP7T U1999 ( .A1(n2581), .B1(n2582), .B2(n2583), .B3(n2584), .ZN(
        n2580) );
    OAI222D0BWP7T U2000 ( .A1(n874), .A2(n223), .B1(n1073), .B2(n311), .C1(n865),
        .C2(n749), .ZN(n2584) );
    CKND2D0BWP7T U2001 ( .A1(n2494), .A2(n296), .ZN(n311) );
    OAI222D0BWP7T U2002 ( .A1(n910), .A2(n1449), .B1(n318), .B2(n617), .C1(n346),
        .C2(n1099), .ZN(n2583) );
    CKND2D0BWP7T U2003 ( .A1(n2494), .A2(n427), .ZN(n1099) );
    OAI222D0BWP7T U2004 ( .A1(n816), .A2(n635), .B1(n2160), .B2(n234), .C1(n1702)
       , .C2(n752), .ZN(n2582) );
    NR2D0BWP7T U2005 ( .A1(n583), .A2(n601), .ZN(n1702) );
    NR2D0BWP7T U2006 ( .A1(n32), .A2(n740), .ZN(n2160) );
    INR4D0BWP7T U2007 ( .A1(n2585), .B1(n1908), .B2(n40), .B3(n1246), .ZN(n2581)
        );
    NR2D0BWP7T U2008 ( .A1(n820), .A2(n1596), .ZN(n1246) );
    NR2D0BWP7T U2009 ( .A1(n328), .A2(n229), .ZN(n40) );
    CKND2D0BWP7T U2010 ( .A1(n293), .A2(n2265), .ZN(n328) );
    NR2D0BWP7T U2011 ( .A1(n843), .A2(n910), .ZN(n1908) );
    CKND0BWP7T U2012 ( .I(n2078), .ZN(n843) );
    NR2D0BWP7T U2013 ( .A1(n327), .A2(n1910), .ZN(n2078) );
    OAI21D0BWP7T U2014 ( .A1(n2019), .A2(n827), .B(n828), .ZN(n2585) );
    AN2D0BWP7T U2015 ( .A1(n2265), .A2(n269), .Z(n827) );
    CKND0BWP7T U2016 ( .I(n1689), .ZN(n2019) );
    CKND2D0BWP7T U2017 ( .A1(n583), .A2(n2103), .ZN(n1689) );
    INR4D0BWP7T U2018 ( .A1(n89), .B1(n2586), .B2(n375), .B3(n1341), .ZN(n2579)
        );
    OAI221D0BWP7T U2019 ( .A1(n1508), .A2(n437), .B1(n199), .B2(n412), .C(n2587),
        .ZN(n1341) );
    AOI22D0BWP7T U2020 ( .A1(n639), .A2(n427), .B1(n717), .B2(n285), .ZN(n2587)
        );
    CKND0BWP7T U2021 ( .I(n236), .ZN(n717) );
    NR2D0BWP7T U2022 ( .A1(n269), .A2(n285), .ZN(n1508) );
    OAI211D0BWP7T U2023 ( .A1(n227), .A2(n752), .B(n758), .C(n2588), .ZN(n375)
        );
```

```
    AOI221D0BWP7T U2024 ( .A1(n2457), .A2(n1193), .B1(n583), .B2(n716), .C(n2589)
        , .ZN(n2588) );
    AOI21D0BWP7T U2025 ( .A1(n236), .A2(n738), .B(n820), .ZN(n2589) );
    CKND2D0BWP7T U2026 ( .A1(n1815), .A2(n635), .ZN(n716) );
    CKND0BWP7T U2027 ( .I(n1853), .ZN(n1193) );
    CKND0BWP7T U2028 ( .I(n186), .ZN(n2457) );
    CKND2D0BWP7T U2029 ( .A1(n460), .A2(n2103), .ZN(n186) );
    AOI22D0BWP7T U2030 ( .A1(n294), .A2(n718), .B1(n828), .B2(n516), .ZN(n758)
          );
    CKND0BWP7T U2031 ( .I(n2232), .ZN(n516) );
    CKND2D0BWP7T U2032 ( .A1(n2265), .A2(n601), .ZN(n2232) );
    OAI221D0BWP7T U2033 ( .A1(n229), .A2(n2284), .B1(n875), .B2(n1072), .C(n2590)
        , .ZN(n2586) );
    AOI22D0BWP7T U2034 ( .A1(n1063), .A2(n755), .B1(n1007), .B2(n259), .ZN(n2590)
         );
    CKND0BWP7T U2035 ( .I(n928), .ZN(n755) );
    NR2D0BWP7T U2036 ( .A1(n517), .A2(n515), .ZN(n928) );
    CKND0BWP7T U2037 ( .I(n230), .ZN(n1063) );
    CKND2D0BWP7T U2038 ( .A1(n693), .A2(n2103), .ZN(n230) );
    CKND2D0BWP7T U2039 ( .A1(n2509), .A2(n482), .ZN(n1072) );
    CKND0BWP7T U2040 ( .I(n1079), .ZN(n2284) );
    NR2D0BWP7T U2041 ( .A1(n614), .A2(n1910), .ZN(n1079) );
    AOI22D0BWP7T U2042 ( .A1(n517), .A2(n1682), .B1(n296), .B2(n481), .ZN(n89)
          );
    CKND0BWP7T U2043 ( .I(n526), .ZN(n481) );
    CKND2D0BWP7T U2044 ( .A1(n2576), .A2(n1978), .ZN(n526) );
    CKND0BWP7T U2045 ( .I(n757), .ZN(n1682) );
    INR4D0BWP7T U2046 ( .A1(n1763), .B1(n2591), .B2(n1084), .B3(n1742), .ZN(
         n2578) );
    OAI211D0BWP7T U2047 ( .A1(n313), .A2(n1320), .B(n2592), .C(n2593), .ZN(n1742)
         );
    AOI222D0BWP7T U2048 ( .A1(n315), .A2(n587), .B1(n280), .B2(n316), .C1(n630),
         .C2(n588), .ZN(n2593) );
    CKND0BWP7T U2049 ( .I(n1074), .ZN(n588) );
    CKND0BWP7T U2050 ( .I(n320), .ZN(n315) );
    CKND2D0BWP7T U2051 ( .A1(n285), .A2(n2509), .ZN(n320) );
    AOI22D0BWP7T U2052 ( .A1(n1095), .A2(n453), .B1(n633), .B2(n773), .ZN(n2592)
          );
    CKND0BWP7T U2053 ( .I(n1170), .ZN(n773) );
    CKND2D0BWP7T U2054 ( .A1(n259), .A2(n2494), .ZN(n1170) );
    CKND2D0BWP7T U2055 ( .A1(n285), .A2(n2494), .ZN(n1320) );
    OAI22D0BWP7T U2056 ( .A1(n1073), .A2(n325), .B1(n221), .B2(n892), .ZN(n1084)
          );
    CKND2D0BWP7T U2057 ( .A1(n266), .A2(n2494), .ZN(n892) );
    IND3D0BWP7T U2058 ( .A1(n1293), .B1(n1206), .B2(n505), .ZN(n2591) );
    AOI22D0BWP7T U2059 ( .A1(n518), .A2(n293), .B1(n453), .B2(n31), .ZN(n505) );
    CKND0BWP7T U2060 ( .I(n1391), .ZN(n453) );
    CKND2D0BWP7T U2061 ( .A1(n2509), .A2(n296), .ZN(n1391) );
    CKND0BWP7T U2062 ( .I(n319), .ZN(n518) );
    AOI22D0BWP7T U2063 ( .A1(n43), .A2(n293), .B1(n438), .B2(n1005), .ZN(n1206)
          );
    CKND0BWP7T U2064 ( .I(n1237), .ZN(n1005) );
    ND4D0BWP7T U2065 ( .A1(n2594), .A2(n2595), .A3(n2596), .A4(n2597), .ZN(n1293)
         );
```

```
    NR4D0BWP7T U2066 ( .A1(n2598), .A2(n2599), .A3(n1907), .A4(n2600), .ZN(n2597)
        );
    AOI21D0BWP7T U2067 ( .A1(n837), .A2(n1603), .B(n329), .ZN(n2600) );
    CKND2D0BWP7T U2068 ( .A1(n427), .A2(n2103), .ZN(n1603) );
    CKND2D0BWP7T U2069 ( .A1(n601), .A2(n2103), .ZN(n837) );
    NR2D0BWP7T U2070 ( .A1(n1050), .A2(n227), .ZN(n1907) );
    CKND2D0BWP7T U2071 ( .A1(n2601), .A2(n2103), .ZN(n1050) );
    OAI22D0BWP7T U2072 ( .A1(n2602), .A2(n435), .B1(n2603), .B2(n846), .ZN(n2599)
        );
    CKND0BWP7T U2073 ( .I(n1200), .ZN(n846) );
    NR2D0BWP7T U2074 ( .A1(n2604), .A2(n1910), .ZN(n1200) );
    NR3D0BWP7T U2075 ( .A1(n714), .A2(n1681), .A3(n43), .ZN(n2602) );
    CKND0BWP7T U2076 ( .I(n437), .ZN(n43) );
    CKND2D0BWP7T U2077 ( .A1(n2573), .A2(n2577), .ZN(n437) );
    NR2D0BWP7T U2078 ( .A1(n1910), .A2(n824), .ZN(n1681) );
    CKND0BWP7T U2079 ( .I(n752), .ZN(n714) );
    CKND2D0BWP7T U2080 ( .A1(n2572), .A2(n2103), .ZN(n752) );
    OAI221D0BWP7T U2081 ( .A1(n327), .A2(n1892), .B1(n1853), .B2(n1819), .C(
        n2605), .ZN(n2598) );
    AOI22D0BWP7T U2082 ( .A1(n718), .A2(n44), .B1(n298), .B2(n355), .ZN(n2605)
        );
    CKND0BWP7T U2083 ( .I(n1257), .ZN(n44) );
    NR2D0BWP7T U2084 ( .A1(n482), .A2(n460), .ZN(n1257) );
    CKND0BWP7T U2085 ( .I(n1596), .ZN(n718) );
    CKND2D0BWP7T U2086 ( .A1(n2573), .A2(n517), .ZN(n1596) );
    CKND0BWP7T U2087 ( .I(n229), .ZN(n517) );
    CKND2D0BWP7T U2088 ( .A1(n266), .A2(n2103), .ZN(n1819) );
    NR2D0BWP7T U2089 ( .A1(n828), .A2(n1683), .ZN(n1853) );
    CKND2D0BWP7T U2090 ( .A1(n293), .A2(n2103), .ZN(n1892) );
    NR4D0BWP7T U2091 ( .A1(n1864), .A2(n823), .A3(n1825), .A4(n66), .ZN(n2596)
        );
    OAI221D0BWP7T U2092 ( .A1(n229), .A2(n825), .B1(n512), .B2(n1815), .C(n2606),
        .ZN(n66) );
    AOI22D0BWP7T U2093 ( .A1(n2170), .A2(n285), .B1(n1893), .B2(n1683), .ZN(
        n2606) );
    CKND0BWP7T U2094 ( .I(n329), .ZN(n1683) );
    NR2D0BWP7T U2095 ( .A1(n618), .A2(n1910), .ZN(n1893) );
    CKND0BWP7T U2096 ( .I(n749), .ZN(n2170) );
    CKND2D0BWP7T U2097 ( .A1(n828), .A2(n2265), .ZN(n749) );
    OAI222D0BWP7T U2098 ( .A1(n618), .A2(n194), .B1(n329), .B2(n750), .C1(n820),
        .C2(n819), .ZN(n1825) );
    CKND2D0BWP7T U2099 ( .A1(n268), .A2(n2103), .ZN(n750) );
    CKND2D0BWP7T U2100 ( .A1(n2265), .A2(n2572), .ZN(n194) );
    OAI222D0BWP7T U2101 ( .A1(n318), .A2(n1815), .B1(n229), .B2(n1971), .C1(n408)
       , .C2(n236), .ZN(n823) );
    CKND2D0BWP7T U2102 ( .A1(n2265), .A2(n2576), .ZN(n236) );
    CKND2D0BWP7T U2103 ( .A1(n2265), .A2(n427), .ZN(n1971) );
    CKND2D0BWP7T U2104 ( .A1(n2576), .A2(n2103), .ZN(n1815) );
    OAI22D0BWP7T U2105 ( .A1(n229), .A2(n188), .B1(n196), .B2(n1237), .ZN(n1864)
        );
    CKND2D0BWP7T U2106 ( .A1(n2601), .A2(n1978), .ZN(n1237) );
    CKND0BWP7T U2107 ( .I(n995), .ZN(n188) );
    NR2D0BWP7T U2108 ( .A1(n227), .A2(n1910), .ZN(n995) );
    CKND2D0BWP7T U2109 ( .A1(n6329), .A2(n2607), .ZN(n229) );
```

```
    AOI22D0BWP7T U2110 ( .A1(n2110), .A2(n515), .B1(n615), .B2(n1567), .ZN(n2595)
        );
    CKND0BWP7T U2111 ( .I(n1507), .ZN(n1567) );
    NR2D0BWP7T U2112 ( .A1(n583), .A2(n266), .ZN(n1507) );
    CKND0BWP7T U2113 ( .I(n738), .ZN(n615) );
    CKND2D0BWP7T U2114 ( .A1(n2265), .A2(n2577), .ZN(n738) );
    NR2D0BWP7T U2115 ( .A1(n1910), .A2(n318), .ZN(n2110) );
    AOI22D0BWP7T U2116 ( .A1(n1678), .A2(n294), .B1(n639), .B2(n482), .ZN(n2594)
        );
    CKND0BWP7T U2117 ( .I(n635), .ZN(n1678) );
    CKND2D0BWP7T U2118 ( .A1(n2601), .A2(n2573), .ZN(n635) );
    AOI22D0BWP7T U2119 ( .A1(n291), .A2(n2057), .B1(n285), .B2(n582), .ZN(n1763)
        );
    CKND0BWP7T U2120 ( .I(n684), .ZN(n582) );
    INR4D0BWP7T U2121 ( .A1(n1406), .B1(n1264), .B2(n1383), .B3(n96), .ZN(n2305)
        );
    IND4D0BWP7T U2122 ( .A1(n395), .B1(n1337), .B2(n2608), .B3(n2609), .ZN(n96)
        );
    AOI211D0BWP7T U2123 ( .A1(n580), .A2(n482), .B(n2610), .C(n572), .ZN(n2609)
        );
    OAI22D0BWP7T U2124 ( .A1(n618), .A2(n1135), .B1(n432), .B2(n153), .ZN(n2610)
        );
    CKND2D0BWP7T U2125 ( .A1(n2548), .A2(n2265), .ZN(n1135) );
    CKND0BWP7T U2126 ( .I(n195), .ZN(n580) );
    CKND2D0BWP7T U2127 ( .A1(n2560), .A2(n2573), .ZN(n195) );
    NR2D0BWP7T U2128 ( .A1(n937), .A2(n1398), .ZN(n2608) );
    OAI221D0BWP7T U2129 ( .A1(n1350), .A2(n2310), .B1(n865), .B2(n1367), .C(
        n2611), .ZN(n1398) );
    AOI211D0BWP7T U2130 ( .A1(n1773), .A2(n438), .B(n2612), .C(n2613), .ZN(n2611)
        );
    AOI31D0BWP7T U2131 ( .A1(n1352), .A2(n153), .A3(n932), .B(n512), .ZN(n2613)
        );
    CKND2D0BWP7T U2132 ( .A1(n2497), .A2(n2265), .ZN(n153) );
    AOI21D0BWP7T U2133 ( .A1(n2043), .A2(n578), .B(n49), .ZN(n2612) );
    CKND0BWP7T U2134 ( .I(n617), .ZN(n1773) );
    CKND2D0BWP7T U2135 ( .A1(n2458), .A2(n693), .ZN(n2310) );
    CKND0BWP7T U2136 ( .I(n573), .ZN(n1350) );
    OAI222D0BWP7T U2137 ( .A1(n35), .A2(n1120), .B1(n49), .B2(n1471), .C1(n143),
        .C2(n577), .ZN(n937) );
    CKND2D0BWP7T U2138 ( .A1(n2614), .A2(n1443), .ZN(n1471) );
    OA221D0BWP7T U2139 ( .A1(n1447), .A2(n1449), .B1(n318), .B2(n189), .C(n2615),
        .Z(n1337) );
    AOI22D0BWP7T U2140 ( .A1(n602), .A2(n285), .B1(n692), .B2(n427), .ZN(n2615)
        );
    CKND0BWP7T U2141 ( .I(n1120), .ZN(n692) );
    CKND2D0BWP7T U2142 ( .A1(n2616), .A2(n2265), .ZN(n1120) );
    NR2D0BWP7T U2143 ( .A1(n693), .A2(n296), .ZN(n1447) );
    OAI221D0BWP7T U2144 ( .A1(n35), .A2(n1394), .B1(n143), .B2(n2043), .C(n2617),
        .ZN(n395) );
    MAOI22D0BWP7T U2145 ( .A1(n2057), .A2(n1443), .B1(n2255), .B2(n418), .ZN(
        n2617) );
    CKND0BWP7T U2146 ( .I(n291), .ZN(n418) );
    CKND2D0BWP7T U2147 ( .A1(n2458), .A2(n427), .ZN(n2255) );
    CKND0BWP7T U2148 ( .I(n1772), .ZN(n2057) );
```

```
    CKND2D0BWP7T U2149 ( .A1(n2458), .A2(n438), .ZN(n1772) );
    OAI211D0BWP7T U2150 ( .A1(n1528), .A2(n313), .B(n1598), .C(n2618), .ZN(n1383)
        );
    AOI221D0BWP7T U2151 ( .A1(n608), .A2(n1490), .B1(n878), .B2(n638), .C(n1711),
         .ZN(n2618) );
    NR2D0BWP7T U2152 ( .A1(n151), .A2(n226), .ZN(n1711) );
    AOI22D0BWP7T U2153 ( .A1(n216), .A2(n878), .B1(n1171), .B2(n633), .ZN(n1598)
         );
    CKND0BWP7T U2154 ( .I(n312), .ZN(n1171) );
    ND3D0BWP7T U2155 ( .A1(n71), .A2(n1622), .A3(n2619), .ZN(n1264) );
    AOI22D0BWP7T U2156 ( .A1(n353), .A2(n805), .B1(n340), .B2(n530), .ZN(n2619)
         );
    AOI22D0BWP7T U2157 ( .A1(n764), .A2(n1427), .B1(n355), .B2(n337), .ZN(n1622)
         );
    AOI222D0BWP7T U2158 ( .A1(n204), .A2(n854), .B1(n299), .B2(n297), .C1(n701),
         .C2(n686), .ZN(n71) );
    AN4D0BWP7T U2159 ( .A1(n2620), .A2(n2621), .A3(n2622), .A4(n2623), .Z(n1406)
         );
    AOI222D0BWP7T U2160 ( .A1(n183), .A2(n638), .B1(n34), .B2(n2511), .C1(n33),
         .C2(n322), .ZN(n2623) );
    CKND0BWP7T U2161 ( .I(n150), .ZN(n322) );
    CKND2D0BWP7T U2162 ( .A1(n573), .A2(n2624), .ZN(n150) );
    CKND2D0BWP7T U2163 ( .A1(n724), .A2(n312), .ZN(n2511) );
    CKND2D0BWP7T U2164 ( .A1(n294), .A2(n2494), .ZN(n312) );
    CKND0BWP7T U2165 ( .I(n234), .ZN(n34) );
    CKND2D0BWP7T U2166 ( .A1(n289), .A2(n2625), .ZN(n234) );
    CKND0BWP7T U2167 ( .I(n309), .ZN(n638) );
    CKND2D0BWP7T U2168 ( .A1(n1443), .A2(n2626), .ZN(n309) );
    CKND0BWP7T U2169 ( .I(n772), .ZN(n183) );
    AOI222D0BWP7T U2170 ( .A1(n1140), .A2(n485), .B1(n31), .B2(n1490), .C1(n317),
         .C2(n640), .ZN(n2622) );
    CKND0BWP7T U2171 ( .I(n1255), .ZN(n1490) );
    CKND0BWP7T U2172 ( .I(n151), .ZN(n1140) );
    CKND2D0BWP7T U2173 ( .A1(n2494), .A2(n438), .ZN(n151) );
    AOI211D0BWP7T U2174 ( .A1(n484), .A2(n30), .B(n75), .C(n1172), .ZN(n2621) );
    OAI222D0BWP7T U2175 ( .A1(n772), .A2(n313), .B1(n1749), .B2(n226), .C1(n875),
         .C2(n224), .ZN(n1172) );
    NR2D0BWP7T U2176 ( .A1(n242), .A2(n727), .ZN(n1749) );
    CKND0BWP7T U2177 ( .I(n1529), .ZN(n727) );
    CKND2D0BWP7T U2178 ( .A1(n2494), .A2(n268), .ZN(n1529) );
    CKND0BWP7T U2179 ( .I(n345), .ZN(n242) );
    CKND2D0BWP7T U2180 ( .A1(n2509), .A2(n427), .ZN(n345) );
    CKND2D0BWP7T U2181 ( .A1(n583), .A2(n2509), .ZN(n772) );
    ND3D0BWP7T U2182 ( .A1(n2627), .A2(n2628), .A3(n2629), .ZN(n75) );
    AOI222D0BWP7T U2183 ( .A1(n485), .A2(n882), .B1(n631), .B2(n316), .C1(n280),
         .C2(n630), .ZN(n2629) );
    CKND0BWP7T U2184 ( .I(n1488), .ZN(n280) );
    CKND2D0BWP7T U2185 ( .A1(n2509), .A2(n269), .ZN(n1488) );
    CKND0BWP7T U2186 ( .I(n404), .ZN(n316) );
    CKND2D0BWP7T U2187 ( .A1(n291), .A2(n2630), .ZN(n404) );
    CKND0BWP7T U2188 ( .I(n221), .ZN(n485) );
    CKND2D0BWP7T U2189 ( .A1(n289), .A2(n2630), .ZN(n221) );
    AOI22D0BWP7T U2190 ( .A1(n740), .A2(n633), .B1(n32), .B2(n483), .ZN(n2628)
         );
```

```
    CKND0BWP7T U2191 ( .I(n874), .ZN(n32) );
    CKND0BWP7T U2192 ( .I(n1086), .ZN(n633) );
    CKND2D0BWP7T U2193 ( .A1(n291), .A2(n2624), .ZN(n1086) );
    CKND0BWP7T U2194 ( .I(n407), .ZN(n740) );
    AOI22D0BWP7T U2195 ( .A1(n1007), .A2(n269), .B1(n212), .B2(n775), .ZN(n2627)
        );
    CKND0BWP7T U2196 ( .I(n313), .ZN(n212) );
    CKND0BWP7T U2197 ( .I(n509), .ZN(n30) );
    CKND2D0BWP7T U2198 ( .A1(n289), .A2(n2624), .ZN(n509) );
    INR3D0BWP7T U2199 ( .A1(n1744), .B1(n1520), .B2(n1715), .ZN(n2620) );
    OAI211D0BWP7T U2200 ( .A1(n143), .A2(n319), .B(n2631), .C(n2632), .ZN(n1715)
        );
    AOI22D0BWP7T U2201 ( .A1(n484), .A2(n640), .B1(n33), .B2(n31), .ZN(n2632) );
    CKND0BWP7T U2202 ( .I(n880), .ZN(n31) );
    CKND2D0BWP7T U2203 ( .A1(n291), .A2(n2625), .ZN(n880) );
    CKND0BWP7T U2204 ( .I(n224), .ZN(n33) );
    CKND2D0BWP7T U2205 ( .A1(n282), .A2(n2494), .ZN(n224) );
    CKND0BWP7T U2206 ( .I(n875), .ZN(n640) );
    CKND0BWP7T U2207 ( .I(n1528), .ZN(n484) );
    CKND2D0BWP7T U2208 ( .A1(n293), .A2(n2509), .ZN(n1528) );
    OAI21D0BWP7T U2209 ( .A1(n1078), .A2(n775), .B(n630), .ZN(n2631) );
    CKND0BWP7T U2210 ( .I(n1240), .ZN(n630) );
    CKND2D0BWP7T U2211 ( .A1(n289), .A2(n2626), .ZN(n1240) );
    CKND0BWP7T U2212 ( .I(n724), .ZN(n775) );
    CKND2D0BWP7T U2213 ( .A1(n266), .A2(n2509), .ZN(n724) );
    CKND0BWP7T U2214 ( .I(n406), .ZN(n1078) );
    CKND2D0BWP7T U2215 ( .A1(n282), .A2(n2509), .ZN(n406) );
    CKND2D0BWP7T U2216 ( .A1(n2573), .A2(n515), .ZN(n319) );
    CKND0BWP7T U2217 ( .I(n824), .ZN(n515) );
    CKND2D0BWP7T U2218 ( .A1(n2633), .A2(n6329), .ZN(n824) );
    OAI221D0BWP7T U2219 ( .A1(n874), .A2(n875), .B1(n1074), .B2(n313), .C(n2634),
        .ZN(n1520) );
    AOI22D0BWP7T U2220 ( .A1(n1095), .A2(n882), .B1(n216), .B2(n317), .ZN(n2634)
        );
    CKND0BWP7T U2221 ( .I(n879), .ZN(n317) );
    CKND2D0BWP7T U2222 ( .A1(n583), .A2(n2494), .ZN(n879) );
    CKND0BWP7T U2223 ( .I(n405), .ZN(n882) );
    CKND2D0BWP7T U2224 ( .A1(n259), .A2(n2509), .ZN(n405) );
    CKND2D0BWP7T U2225 ( .A1(n291), .A2(n2626), .ZN(n313) );
    CKND2D0BWP7T U2226 ( .A1(n294), .A2(n2509), .ZN(n1074) );
    CKND2D0BWP7T U2227 ( .A1(n573), .A2(n2630), .ZN(n875) );
    CKND2D0BWP7T U2228 ( .A1(n460), .A2(n2494), .ZN(n874) );
    AOI22D0BWP7T U2229 ( .A1(n587), .A2(n878), .B1(n483), .B2(n1090), .ZN(n1744)
        );
    CKND0BWP7T U2230 ( .I(n326), .ZN(n1090) );
    CKND2D0BWP7T U2231 ( .A1(n460), .A2(n2509), .ZN(n326) );
    CKND0BWP7T U2232 ( .I(n1073), .ZN(n483) );
    CKND2D0BWP7T U2233 ( .A1(n1443), .A2(n2625), .ZN(n1073) );
    CKND0BWP7T U2234 ( .I(n325), .ZN(n878) );
    CKND2D0BWP7T U2235 ( .A1(n2494), .A2(n482), .ZN(n325) );
    CKND0BWP7T U2236 ( .I(n223), .ZN(n587) );
    CKND2D0BWP7T U2237 ( .A1(n1443), .A2(n2624), .ZN(n223) );
    NR4D0BWP7T U2238 ( .A1(n2635), .A2(n2636), .A3(n904), .A4(n1785), .ZN(n2179)
        );
```

```
    OAI211D0BWP7T U2239 ( .A1(n614), .A2(n613), .B(n1399), .C(n2637), .ZN(n1785)
        );
    AOI22D0BWP7T U2240 ( .A1(n606), .A2(n2048), .B1(n574), .B2(n268), .ZN(n2637)
        );
    CKND0BWP7T U2241 ( .I(n189), .ZN(n574) );
    CKND2D0BWP7T U2242 ( .A1(n2616), .A2(n2103), .ZN(n189) );
    CKND2D0BWP7T U2243 ( .A1(n820), .A2(n318), .ZN(n2048) );
    CKND0BWP7T U2244 ( .I(n1367), .ZN(n606) );
    AOI222D0BWP7T U2245 ( .A1(n460), .A2(n1122), .B1(n289), .B2(n2138), .C1(
        n1443), .C2(n971), .ZN(n1399) );
    CKND0BWP7T U2246 ( .I(n2492), .ZN(n971) );
    CKND2D0BWP7T U2247 ( .A1(n2458), .A2(n460), .ZN(n2492) );
    CKND0BWP7T U2248 ( .I(n1351), .ZN(n2138) );
    CKND2D0BWP7T U2249 ( .A1(n2458), .A2(n266), .ZN(n1351) );
    ND4D0BWP7T U2250 ( .A1(n2638), .A2(n2639), .A3(n2640), .A4(n2641), .ZN(n904)
        );
    AOI211D0BWP7T U2251 ( .A1(n477), .A2(n204), .B(n1359), .C(n1355), .ZN(n2641)
        );
    NR2D0BWP7T U2252 ( .A1(n1358), .A2(n148), .ZN(n1355) );
    NR2D0BWP7T U2253 ( .A1(n802), .A2(n1112), .ZN(n1359) );
    AOI211D0BWP7T U2254 ( .A1(n586), .A2(n687), .B(n1655), .C(n80), .ZN(n2640)
        );
    OAI22D0BWP7T U2255 ( .A1(n1296), .A2(n802), .B1(n149), .B2(n682), .ZN(n80)
        );
    OAI22D0BWP7T U2256 ( .A1(n681), .A2(n1321), .B1(n1137), .B2(n728), .ZN(n1655)
       );
    NR3D0BWP7T U2257 ( .A1(n1291), .A2(n165), .A3(n1036), .ZN(n2639) );
    OAI211D0BWP7T U2258 ( .A1(n1365), .A2(n1017), .B(n2642), .C(n2643), .ZN(
        n1036) );
    OA21D0BWP7T U2259 ( .A1(n927), .A2(n1321), .B(n1324), .Z(n2643) );
    AOI22D0BWP7T U2260 ( .A1(n479), .A2(n340), .B1(n478), .B2(n205), .ZN(n1324)
        );
    CKND0BWP7T U2261 ( .I(n423), .ZN(n2642) );
    NR2D0BWP7T U2262 ( .A1(n201), .A2(n148), .ZN(n423) );
    NR2D0BWP7T U2263 ( .A1(n687), .A2(n1938), .ZN(n1365) );
    IIND4D0BWP7T U2264 ( .A1(n659), .A2(n791), .B1(n1210), .B2(n2644), .ZN(n165)
        );
    NR3D0BWP7T U2265 ( .A1(n394), .A2(n2645), .A3(n344), .ZN(n2644) );
    OAI22D0BWP7T U2266 ( .A1(n410), .A2(n802), .B1(n1137), .B2(n1321), .ZN(n344)
        );
    AOI21D0BWP7T U2267 ( .A1(n1247), .A2(n1296), .B(n1248), .ZN(n2645) );
    IND3D0BWP7T U2268 ( .A1(n1211), .B1(n628), .B2(n2646), .ZN(n394) );
    AOI222D0BWP7T U2269 ( .A1(n350), .A2(n351), .B1(n477), .B2(n297), .C1(n586),
        .C2(n841), .ZN(n2646) );
    MAOI22D0BWP7T U2270 ( .A1(n340), .A2(n829), .B1(n2647), .B2(n1247), .ZN(n628)
       );
    OAI211D0BWP7T U2271 ( .A1(n146), .A2(n1247), .B(n2648), .C(n2649), .ZN(n1211)
       );
    AOI22D0BWP7T U2272 ( .A1(n338), .A2(n353), .B1(n586), .B2(n853), .ZN(n2649)
        );
    OAI21D0BWP7T U2273 ( .A1(n356), .A2(n372), .B(n805), .ZN(n2648) );
    OA221D0BWP7T U2274 ( .A1(n1248), .A2(n760), .B1(n927), .B2(n802), .C(n2650),
        .Z(n1210) );
    AOI22D0BWP7T U2275 ( .A1(n855), .A2(n204), .B1(n352), .B2(n374), .ZN(n2650)
```

```
      );
    OAI211D0BWP7T U2276 ( .A1(n157), .A2(n440), .B(n567), .C(n2651), .ZN(n791)
      );
    AOI222D0BWP7T U2277 ( .A1(n338), .A2(n841), .B1(n764), .B2(n2652), .C1(n339),
      .C2(n1937), .ZN(n2651) );
    CKND2D0BWP7T U2278 ( .A1(n803), .A2(n926), .ZN(n1937) );
    CKND2D0BWP7T U2279 ( .A1(n201), .A2(n231), .ZN(n2652) );
    AOI22D0BWP7T U2280 ( .A1(n1129), .A2(n855), .B1(n356), .B2(n696), .ZN(n567)
      );
    OAI211D0BWP7T U2281 ( .A1(n1638), .A2(n440), .B(n2653), .C(n566), .ZN(n659)
      );
    AOI22D0BWP7T U2282 ( .A1(n299), .A2(n764), .B1(n205), .B2(n853), .ZN(n566)
      );
    OAI21D0BWP7T U2283 ( .A1(n764), .A2(n455), .B(n853), .ZN(n2653) );
    CKND2D0BWP7T U2284 ( .A1(n682), .A2(n1017), .ZN(n455) );
    IND4D0BWP7T U2285 ( .A1(n1639), .B1(n1221), .B2(n1660), .B3(n2654), .ZN(
      n1291) );
    AOI211D0BWP7T U2286 ( .A1(n584), .A2(n1186), .B(n79), .C(n1033), .ZN(n2654)
      );
    OAI22D0BWP7T U2287 ( .A1(n759), .A2(n1671), .B1(n1280), .B2(n146), .ZN(n1033)
     );
    OAI222D0BWP7T U2288 ( .A1(n410), .A2(n440), .B1(n237), .B2(n802), .C1(n232),
      .C2(n1279), .ZN(n79) );
    OA221D0BWP7T U2289 ( .A1(n1248), .A2(n804), .B1(n147), .B2(n682), .C(n2655),
      .Z(n1660) );
    AOI211D0BWP7T U2290 ( .A1(n841), .A2(n297), .B(n2656), .C(n2657), .ZN(n2655)
      );
    AOI21D0BWP7T U2291 ( .A1(n1637), .A2(n2064), .B(n440), .ZN(n2657) );
    AOI21D0BWP7T U2292 ( .A1(n1980), .A2(n1112), .B(n232), .ZN(n2656) );
    AOI22D0BWP7T U2293 ( .A1(n530), .A2(n696), .B1(n297), .B2(n855), .ZN(n1221)
      );
    OAI21D0BWP7T U2294 ( .A1(n1637), .A2(n1321), .B(n2658), .ZN(n1639) );
    OAI21D0BWP7T U2295 ( .A1(n358), .A2(n855), .B(n205), .ZN(n2658) );
    AN3D0BWP7T U2296 ( .A1(n1944), .A2(n1627), .A3(n1662), .Z(n2638) );
    OA221D0BWP7T U2297 ( .A1(n1248), .A2(n157), .B1(n410), .B2(n682), .C(n2659),
      .Z(n1662) );
    AOI22D0BWP7T U2298 ( .A1(n337), .A2(n696), .B1(n764), .B2(n371), .ZN(n2659)
      );
    CKND0BWP7T U2299 ( .I(n2660), .ZN(n1627) );
    IND4D0BWP7T U2300 ( .A1(n1284), .B1(n392), .B2(n2661), .B3(n2662), .ZN(n2660)
     );
    AOI211D0BWP7T U2301 ( .A1(n374), .A2(n686), .B(n2663), .C(n1932), .ZN(n2662)
      );
    OAI22D0BWP7T U2302 ( .A1(n759), .A2(n1980), .B1(n1690), .B2(n414), .ZN(n1932)
     );
    NR2D0BWP7T U2303 ( .A1(n586), .A2(n1129), .ZN(n1690) );
    OAI22D0BWP7T U2304 ( .A1(n802), .A2(n1358), .B1(n1113), .B2(n1114), .ZN(
      n2663) );
    INR2D0BWP7T U2305 ( .A1(n497), .B1(n559), .ZN(n2661) );
    OAI221D0BWP7T U2306 ( .A1(n1113), .A2(n1358), .B1(n728), .B2(n149), .C(n2664)
    , .ZN(n559) );
    AOI22D0BWP7T U2307 ( .A1(n1075), .A2(n351), .B1(n586), .B2(n763), .ZN(n2664)
      );
    AOI211D0BWP7T U2308 ( .A1(n355), .A2(n763), .B(n1286), .C(n2665), .ZN(n497)
```

```
      );
    AO222D0BWP7T U2309 ( .A1(n529), .A2(n764), .B1(n352), .B2(n2666), .C1(n687),
         .C2(n357), .Z(n2665) );
    CKND2D0BWP7T U2310 ( .A1(n441), .A2(n1114), .ZN(n2666) );
    NR2D0BWP7T U2311 ( .A1(n358), .A2(n350), .ZN(n441) );
    OAI21D0BWP7T U2312 ( .A1(n232), .A2(n1638), .B(n2667), .ZN(n1286) );
    OAI21D0BWP7T U2313 ( .A1(n829), .A2(n529), .B(n352), .ZN(n2667) );
    OA211D0BWP7T U2314 ( .A1(n198), .A2(n146), .B(n2668), .C(n2669), .Z(n392) );
    AOI222D0BWP7T U2315 ( .A1(n529), .A2(n340), .B1(n205), .B2(n854), .C1(n352),
         .C2(n853), .ZN(n2669) );
    AOI22D0BWP7T U2316 ( .A1(n358), .A2(n2670), .B1(n348), .B2(n339), .ZN(n2668)
      );
    OAI221D0BWP7T U2317 ( .A1(n694), .A2(n926), .B1(n803), .B2(n1671), .C(n2671),
         .ZN(n1284) );
    AOI222D0BWP7T U2318 ( .A1(n348), .A2(n1040), .B1(n529), .B2(n2672), .C1(n350)
        , .C2(n764), .ZN(n2671) );
    CKND2D0BWP7T U2319 ( .A1(n199), .A2(n1017), .ZN(n2672) );
    INR2D0BWP7T U2320 ( .A1(n2673), .B1(n2674), .ZN(n1944) );
    IND4D0BWP7T U2321 ( .A1(n2675), .B1(n390), .B2(n1661), .B3(n1656), .ZN(n2674)
      );
    AOI22D0BWP7T U2322 ( .A1(n763), .A2(n338), .B1(n700), .B2(n353), .ZN(n1656)
      );
    CKND2D0BWP7T U2323 ( .A1(n926), .A2(n146), .ZN(n700) );
    AOI21D0BWP7T U2324 ( .A1(n584), .A2(n205), .B(n1285), .ZN(n1661) );
    OAI22D0BWP7T U2325 ( .A1(n410), .A2(n148), .B1(n413), .B2(n1296), .ZN(n1285)
      );
    AOI221D0BWP7T U2326 ( .A1(n355), .A2(n701), .B1(n357), .B2(n372), .C(n852),
         .ZN(n390) );
    NR2D0BWP7T U2327 ( .A1(n199), .A2(n1137), .ZN(n852) );
    OAI211D0BWP7T U2328 ( .A1(n413), .A2(n147), .B(n1612), .C(n2676), .ZN(n2675)
      );
    AOI22D0BWP7T U2329 ( .A1(n348), .A2(n349), .B1(n337), .B2(n686), .ZN(n2676)
      );
    CKND2D0BWP7T U2330 ( .A1(n763), .A2(n204), .ZN(n1612) );
    NR4D0BWP7T U2331 ( .A1(n2677), .A2(n244), .A3(n164), .A4(n1035), .ZN(n2673)
      );
    OAI211D0BWP7T U2332 ( .A1(n232), .A2(n760), .B(n2678), .C(n2679), .ZN(n1035)
      );
    AOI221D0BWP7T U2333 ( .A1(n348), .A2(n353), .B1(n205), .B2(n687), .C(n2680),
         .ZN(n2679) );
    OAI22D0BWP7T U2334 ( .A1(n803), .A2(n237), .B1(n1278), .B2(n413), .ZN(n2680)
      );
    NR2D0BWP7T U2335 ( .A1(n855), .A2(n529), .ZN(n1278) );
    AOI22D0BWP7T U2336 ( .A1(n1075), .A2(n764), .B1(n652), .B2(n696), .ZN(n2678)
      );
    OR4D0BWP7T U2337 ( .A1(n2681), .A2(n1212), .A3(n790), .A4(n705), .Z(n164) );
    OAI21D0BWP7T U2338 ( .A1(n1247), .A2(n802), .B(n2682), .ZN(n705) );
    OAI21D0BWP7T U2339 ( .A1(n357), .A2(n805), .B(n584), .ZN(n2682) );
    ND3D0BWP7T U2340 ( .A1(n1288), .A2(n391), .A3(n2683), .ZN(n790) );
    AOI222D0BWP7T U2341 ( .A1(n371), .A2(n351), .B1(n586), .B2(n854), .C1(n696),
         .C2(n1040), .ZN(n2683) );
    AOI22D0BWP7T U2342 ( .A1(n764), .A2(n356), .B1(n1075), .B2(n686), .ZN(n391)
      );
    AOI222D0BWP7T U2343 ( .A1(n353), .A2(n586), .B1(n297), .B2(n354), .C1(n372),
```

```
      .C2(n352), .ZN(n1288) );
    OAI221D0BWP7T U2344 ( .A1(n198), .A2(n440), .B1(n232), .B2(n681), .C(n1289),
      .ZN(n1212) );
    OA21D0BWP7T U2345 ( .A1(n1137), .A2(n1113), .B(n2684), .Z(n1289) );
    OAI21D0BWP7T U2346 ( .A1(n298), .A2(n652), .B(n348), .ZN(n2684) );
    AO21D0BWP7T U2347 ( .A1(n529), .A2(n204), .B(n626), .Z(n2681) );
    OAI211D0BWP7T U2348 ( .A1(n760), .A2(n682), .B(n2685), .C(n2686), .ZN(n626)
      );
    AOI222D0BWP7T U2349 ( .A1(n352), .A2(n687), .B1(n297), .B2(n701), .C1(n652),
      .C2(n204), .ZN(n2686) );
    AOI22D0BWP7T U2350 ( .A1(n337), .A2(n764), .B1(n354), .B2(n1129), .ZN(n2685)
      );
    ND4D0BWP7T U2351 ( .A1(n1290), .A2(n2687), .A3(n2688), .A4(n2689), .ZN(n244)
      );
    AOI211D0BWP7T U2352 ( .A1(n297), .A2(n1040), .B(n2690), .C(n2691), .ZN(n2689)
     );
    AOI21D0BWP7T U2353 ( .A1(n1671), .A2(n410), .B(n728), .ZN(n2691) );
    OAI22D0BWP7T U2354 ( .A1(n412), .A2(n802), .B1(n2064), .B2(n1321), .ZN(n2690)
     );
    AOI22D0BWP7T U2355 ( .A1(n805), .A2(n299), .B1(n358), .B2(n340), .ZN(n2688)
      );
    CKND2D0BWP7T U2356 ( .A1(n354), .A2(n338), .ZN(n2687) );
    MAOI22D0BWP7T U2357 ( .A1(n687), .A2(n297), .B1(n2127), .B2(n2064), .ZN(
      n1290) );
    NR2D0BWP7T U2358 ( .A1(n338), .A2(n351), .ZN(n2127) );
    IND3D0BWP7T U2359 ( .A1(n1640), .B1(n1628), .B2(n368), .ZN(n2677) );
    OA211D0BWP7T U2360 ( .A1(n201), .A2(n682), .B(n794), .C(n2692), .Z(n368) );
    AOI22D0BWP7T U2361 ( .A1(n352), .A2(n584), .B1(n1129), .B2(n479), .ZN(n2692)
      );
    AOI22D0BWP7T U2362 ( .A1(n351), .A2(n477), .B1(n696), .B2(n374), .ZN(n794)
      );
    CKND0BWP7T U2363 ( .I(n804), .ZN(n374) );
    AN2D0BWP7T U2364 ( .A1(n2693), .A2(n2694), .Z(n1628) );
    NR4D0BWP7T U2365 ( .A1(n2695), .A2(n2696), .A3(n2697), .A4(n2698), .ZN(n2694)
     );
    IAO21D0BWP7T U2366 ( .A1(n1042), .A2(n355), .B(n1671), .ZN(n2698) );
    CKND2D0BWP7T U2367 ( .A1(n232), .A2(n1248), .ZN(n1042) );
    AOI21D0BWP7T U2368 ( .A1(n231), .A2(n237), .B(n926), .ZN(n2697) );
    OAI22D0BWP7T U2369 ( .A1(n232), .A2(n149), .B1(n1137), .B2(n148), .ZN(n2696)
      );
    OAI222D0BWP7T U2370 ( .A1(n1296), .A2(n440), .B1(n804), .B2(n1017), .C1(n413)
    , .C2(n1114), .ZN(n2695) );
    NR4D0BWP7T U2371 ( .A1(n2699), .A2(n1287), .A3(n989), .A4(n387), .ZN(n2693)
      );
    OAI221D0BWP7T U2372 ( .A1(n232), .A2(n147), .B1(n413), .B2(n681), .C(n2700),
      .ZN(n387) );
    AOI222D0BWP7T U2373 ( .A1(n298), .A2(n764), .B1(n854), .B2(n2670), .C1(n340),
      .C2(n478), .ZN(n2700) );
    CKND2D0BWP7T U2374 ( .A1(n682), .A2(n926), .ZN(n2670) );
    OAI211D0BWP7T U2375 ( .A1(n1637), .A2(n802), .B(n2701), .C(n2702), .ZN(n989)
      );
    AOI22D0BWP7T U2376 ( .A1(n338), .A2(n1040), .B1(n352), .B2(n763), .ZN(n2702)
      );
    OAI21D0BWP7T U2377 ( .A1(n478), .A2(n354), .B(n805), .ZN(n2701) );
```

```
    ND4D0BWP7T U2378 ( .A1(n2703), .A2(n2704), .A3(n2705), .A4(n2706), .ZN(n1287)
        );
    AOI22D0BWP7T U2379 ( .A1(n352), .A2(n478), .B1(n701), .B2(n480), .ZN(n2706)
        );
    CKND0BWP7T U2380 ( .I(n411), .ZN(n480) );
    AOI22D0BWP7T U2381 ( .A1(n687), .A2(n204), .B1(n338), .B2(n350), .ZN(n2705)
        );
    AOI22D0BWP7T U2382 ( .A1(n337), .A2(n205), .B1(n477), .B2(n1129), .ZN(n2704)
        );
    AOI22D0BWP7T U2383 ( .A1(n855), .A2(n340), .B1(n354), .B2(n355), .ZN(n2703)
        );
    ND3D0BWP7T U2384 ( .A1(n793), .A2(n498), .A3(n2707), .ZN(n2699) );
    AOI22D0BWP7T U2385 ( .A1(n340), .A2(n372), .B1(n855), .B2(n764), .ZN(n2707)
        );
    AOI22D0BWP7T U2386 ( .A1(n351), .A2(n855), .B1(n805), .B2(n1938), .ZN(n498)
        );
    AOI222D0BWP7T U2387 ( .A1(n299), .A2(n348), .B1(n338), .B2(n829), .C1(n586),
        .C2(n354), .ZN(n793) );
    OAI211D0BWP7T U2388 ( .A1(n1321), .A2(n1671), .B(n2708), .C(n2709), .ZN(
        n1640) );
    AOI211D0BWP7T U2389 ( .A1(n805), .A2(n2710), .B(n2711), .C(n2712), .ZN(n2709)
        );
    AOI21D0BWP7T U2390 ( .A1(n1454), .A2(n1358), .B(n440), .ZN(n2712) );
    AOI21D0BWP7T U2391 ( .A1(n1280), .A2(n1358), .B(n1248), .ZN(n2711) );
    CKND0BWP7T U2392 ( .I(n2133), .ZN(n2710) );
    NR2D0BWP7T U2393 ( .A1(n1040), .A2(n298), .ZN(n2133) );
    AOI22D0BWP7T U2394 ( .A1(n1427), .A2(n1186), .B1(n586), .B2(n1075), .ZN(
        n2708) );
    IND3D0BWP7T U2395 ( .A1(n1295), .B1(n947), .B2(n1329), .ZN(n2636) );
    CKND0BWP7T U2396 ( .I(n2713), .ZN(n1329) );
    IND4D0BWP7T U2397 ( .A1(n1644), .B1(n2714), .B2(n2715), .B3(n2716), .ZN(
        n2713) );
    NR4D0BWP7T U2398 ( .A1(n2717), .A2(n2718), .A3(n158), .A4(n1609), .ZN(n2716)
        );
    NR2D0BWP7T U2399 ( .A1(n232), .A2(n1296), .ZN(n1609) );
    NR2D0BWP7T U2400 ( .A1(n413), .A2(n198), .ZN(n158) );
    AOI21D0BWP7T U2401 ( .A1(n760), .A2(n147), .B(n926), .ZN(n2718) );
    OAI222D0BWP7T U2402 ( .A1(n156), .A2(n804), .B1(n1456), .B2(n1321), .C1(n148)
        , .C2(n1671), .ZN(n2717) );
    NR2D0BWP7T U2403 ( .A1(n687), .A2(n841), .ZN(n1456) );
    NR3D0BWP7T U2404 ( .A1(n1626), .A2(n1653), .A3(n70), .ZN(n2715) );
    OAI221D0BWP7T U2405 ( .A1(n1638), .A2(n1248), .B1(n414), .B2(n682), .C(n2719)
        , .ZN(n70) );
    AOI221D0BWP7T U2406 ( .A1(n854), .A2(n357), .B1(n1938), .B2(n351), .C(n2720),
        .ZN(n2719) );
    AOI21D0BWP7T U2407 ( .A1(n1017), .A2(n148), .B(n927), .ZN(n2720) );
    OAI222D0BWP7T U2408 ( .A1(n146), .A2(n1279), .B1(n1113), .B2(n1637), .C1(
        n803), .C2(n412), .ZN(n1653) );
    ND3D0BWP7T U2409 ( .A1(n796), .A2(n1019), .A3(n2721), .ZN(n1626) );
    AOI22D0BWP7T U2410 ( .A1(n841), .A2(n355), .B1(n338), .B2(n358), .ZN(n2721)
        );
    AOI22D0BWP7T U2411 ( .A1(n357), .A2(n1040), .B1(n204), .B2(n358), .ZN(n1019)
        );
    AOI22D0BWP7T U2412 ( .A1(n696), .A2(n763), .B1(n340), .B2(n349), .ZN(n796)
```

```
      );
    AOI22D0BWP7T U2413 ( .A1(n340), .A2(n299), .B1(n348), .B2(n371), .ZN(n2714)
      );
    OAI22D0BWP7T U2414 ( .A1(n1112), .A2(n148), .B1(n237), .B2(n759), .ZN(n1644)
      );
    CKND0BWP7T U2415 ( .I(n2722), .ZN(n947) );
    IND4D0BWP7T U2416 ( .A1(n78), .B1(n1631), .B2(n2723), .B3(n1038), .ZN(n2722)
      );
    AOI222D0BWP7T U2417 ( .A1(n340), .A2(n1075), .B1(n351), .B2(n353), .C1(n371),
     .C2(n586), .ZN(n1038) );
    AOI22D0BWP7T U2418 ( .A1(n841), .A2(n351), .B1(n696), .B2(n479), .ZN(n2723)
      );
    AOI222D0BWP7T U2419 ( .A1(n204), .A2(n841), .B1(n297), .B2(n358), .C1(n299),
     .C2(n686), .ZN(n1631) );
    CKND0BWP7T U2420 ( .I(n149), .ZN(n841) );
    OAI221D0BWP7T U2421 ( .A1(n237), .A2(n682), .B1(n926), .B2(n1279), .C(n2724),
     .ZN(n78) );
    OAI21D0BWP7T U2422 ( .A1(n701), .A2(n372), .B(n764), .ZN(n2724) );
    CKND0BWP7T U2423 ( .I(n1296), .ZN(n701) );
    OAI211D0BWP7T U2424 ( .A1(n146), .A2(n412), .B(n2725), .C(n420), .ZN(n1295)
      );
    CKND2D0BWP7T U2425 ( .A1(n855), .A2(n696), .ZN(n420) );
    CKND0BWP7T U2426 ( .I(n1454), .ZN(n855) );
    OAI21D0BWP7T U2427 ( .A1(n530), .A2(n477), .B(n357), .ZN(n2725) );
    IIND4D0BWP7T U2428 ( .A1(n1767), .A2(n1498), .B1(n1930), .B2(n1405), .ZN(
     n2635) );
    CKND0BWP7T U2429 ( .I(n1766), .ZN(n1405) );
    OAI211D0BWP7T U2430 ( .A1(n143), .A2(n932), .B(n2726), .C(n2727), .ZN(n1766)
      );
    AOI211D0BWP7T U2431 ( .A1(n1122), .A2(n1445), .B(n2728), .C(n2729), .ZN(
     n2727) );
    AOI21D0BWP7T U2432 ( .A1(n577), .A2(n2043), .B(n35), .ZN(n2729) );
    AOI21D0BWP7T U2433 ( .A1(n684), .A2(n1394), .B(n408), .ZN(n2728) );
    CKND0BWP7T U2434 ( .I(n460), .ZN(n408) );
    CKND2D0BWP7T U2435 ( .A1(n910), .A2(n435), .ZN(n1445) );
    CKND0BWP7T U2436 ( .I(n1242), .ZN(n1122) );
    CKND2D0BWP7T U2437 ( .A1(n2614), .A2(n291), .ZN(n1242) );
    AOI22D0BWP7T U2438 ( .A1(n572), .A2(n291), .B1(n2047), .B2(n268), .ZN(n2726)
      );
    CKND0BWP7T U2439 ( .I(n578), .ZN(n2047) );
    CKND2D0BWP7T U2440 ( .A1(n2560), .A2(n1978), .ZN(n578) );
    AN2D0BWP7T U2441 ( .A1(n2458), .A2(n268), .Z(n572) );
    AN3D0BWP7T U2442 ( .A1(n2730), .A2(n2731), .A3(n2732), .Z(n1930) );
    INR4D0BWP7T U2443 ( .A1(n798), .B1(n2733), .B2(n2734), .B3(n1422), .ZN(n2732)
     );
    OAI22D0BWP7T U2444 ( .A1(n1638), .A2(n1321), .B1(n1112), .B2(n1248), .ZN(
     n1422) );
    OAI22D0BWP7T U2445 ( .A1(n411), .A2(n410), .B1(n156), .B2(n1454), .ZN(n2734)
      );
    NR2D0BWP7T U2446 ( .A1(n586), .A2(n348), .ZN(n156) );
    NR2D0BWP7T U2447 ( .A1(n1129), .A2(n204), .ZN(n411) );
    ND4D0BWP7T U2448 ( .A1(n648), .A2(n699), .A3(n1611), .A4(n2735), .ZN(n2733)
      );
    AOI22D0BWP7T U2449 ( .A1(n686), .A2(n2736), .B1(n478), .B2(n351), .ZN(n2735)
```

```
      );
    CKND0BWP7T U2450 ( .I(n1137), .ZN(n478) );
    CKND2D0BWP7T U2451 ( .A1(n583), .A2(n2737), .ZN(n1137) );
    CKND2D0BWP7T U2452 ( .A1(n412), .A2(n1663), .ZN(n2736) );
    CKND2D0BWP7T U2453 ( .A1(n337), .A2(n351), .ZN(n1611) );
    CKND2D0BWP7T U2454 ( .A1(n340), .A2(n356), .ZN(n699) );
    CKND0BWP7T U2455 ( .I(n1247), .ZN(n356) );
    CKND0BWP7T U2456 ( .I(n1677), .ZN(n648) );
    NR2D0BWP7T U2457 ( .A1(n728), .A2(n237), .ZN(n1677) );
    AOI22D0BWP7T U2458 ( .A1(n763), .A2(n348), .B1(n764), .B2(n1938), .ZN(n798)
      );
    CKND0BWP7T U2459 ( .I(n1980), .ZN(n763) );
    AN4D0BWP7T U2460 ( .A1(n369), .A2(n145), .A3(n1323), .A4(n246), .Z(n2731) );
    AN4D0BWP7T U2461 ( .A1(n2738), .A2(n2739), .A3(n2740), .A4(n2741), .Z(n246)
      );
    AOI222D0BWP7T U2462 ( .A1(n1427), .A2(n351), .B1(n529), .B2(n297), .C1(n340),
      .C2(n339), .ZN(n2741) );
    CKND0BWP7T U2463 ( .I(n1112), .ZN(n339) );
    CKND0BWP7T U2464 ( .I(n927), .ZN(n1427) );
    MAOI22D0BWP7T U2465 ( .A1(n298), .A2(n696), .B1(n201), .B2(n1651), .ZN(n2740)
     );
    NR2D0BWP7T U2466 ( .A1(n696), .A2(n357), .ZN(n1651) );
    OAI21D0BWP7T U2467 ( .A1(n338), .A2(n1186), .B(n372), .ZN(n2739) );
    CKND2D0BWP7T U2468 ( .A1(n1113), .A2(n146), .ZN(n1186) );
    CKND0BWP7T U2469 ( .I(n1017), .ZN(n338) );
    OAI21D0BWP7T U2470 ( .A1(n1938), .A2(n350), .B(n1129), .ZN(n2738) );
    AN4D0BWP7T U2471 ( .A1(n1668), .A2(n1029), .A3(n2742), .A4(n2743), .Z(n1323)
      );
    AOI221D0BWP7T U2472 ( .A1(n354), .A2(n204), .B1(n348), .B2(n337), .C(n2744),
      .ZN(n2743) );
    OAI221D0BWP7T U2473 ( .A1(n695), .A2(n232), .B1(n1248), .B2(n149), .C(n2745),
      .ZN(n2744) );
    OAI21D0BWP7T U2474 ( .A1(n829), .A2(n652), .B(n297), .ZN(n2745) );
    NR2D0BWP7T U2475 ( .A1(n529), .A2(n687), .ZN(n695) );
    CKND0BWP7T U2476 ( .I(n414), .ZN(n529) );
    CKND0BWP7T U2477 ( .I(n146), .ZN(n204) );
    AOI211D0BWP7T U2478 ( .A1(n477), .A2(n340), .B(n1650), .C(n848), .ZN(n2742)
      );
    OAI22D0BWP7T U2479 ( .A1(n199), .A2(n1247), .B1(n413), .B2(n1279), .ZN(n848)
      );
    OAI22D0BWP7T U2480 ( .A1(n2064), .A2(n148), .B1(n146), .B2(n157), .ZN(n1650)
      );
    AOI22D0BWP7T U2481 ( .A1(n477), .A2(n352), .B1(n353), .B2(n1129), .ZN(n1029)
      );
    CKND0BWP7T U2482 ( .I(n760), .ZN(n477) );
    AOI221D0BWP7T U2483 ( .A1(n584), .A2(n686), .B1(n298), .B2(n586), .C(n704),
      .ZN(n1668) );
    OAI22D0BWP7T U2484 ( .A1(n413), .A2(n197), .B1(n1358), .B2(n682), .ZN(n704)
      );
    CKND0BWP7T U2485 ( .I(n1663), .ZN(n298) );
    CKND2D0BWP7T U2486 ( .A1(n693), .A2(n2737), .ZN(n1663) );
    CKND0BWP7T U2487 ( .I(n237), .ZN(n584) );
    AOI22D0BWP7T U2488 ( .A1(n355), .A2(n854), .B1(n297), .B2(n1075), .ZN(n145)
      );
```

```
    CKND0BWP7T U2489 ( .I(n1638), .ZN(n1075) );
    CKND0BWP7T U2490 ( .I(n1113), .ZN(n297) );
    AOI22D0BWP7T U2491 ( .A1(n357), .A2(n479), .B1(n355), .B2(n1938), .ZN(n369)
          );
    CKND0BWP7T U2492 ( .I(n1358), .ZN(n1938) );
    CKND2D0BWP7T U2493 ( .A1(n482), .A2(n2746), .ZN(n1358) );
    CKND0BWP7T U2494 ( .I(n1280), .ZN(n479) );
    NR4D0BWP7T U2495 ( .A1(n561), .A2(n1034), .A3(n1670), .A4(n1648), .ZN(n2730)
          );
    OAI211D0BWP7T U2496 ( .A1(n802), .A2(n1671), .B(n2747), .C(n2748), .ZN(n1648)
         );
    OA222D0BWP7T U2497 ( .A1(n760), .A2(n1017), .B1(n927), .B2(n2647), .C1(n728),
         .C2(n1280), .Z(n2748) );
    CKND2D0BWP7T U2498 ( .A1(n294), .A2(n2746), .ZN(n1280) );
    NR2D0BWP7T U2499 ( .A1(n348), .A2(n352), .ZN(n2647) );
    AOI22D0BWP7T U2500 ( .A1(n340), .A2(n371), .B1(n352), .B2(n354), .ZN(n2747)
          );
    CKND0BWP7T U2501 ( .I(n201), .ZN(n354) );
    CKND2D0BWP7T U2502 ( .A1(n601), .A2(n2737), .ZN(n201) );
    OAI211D0BWP7T U2503 ( .A1(n1980), .A2(n1321), .B(n2749), .C(n2750), .ZN(
         n1670) );
    AOI211D0BWP7T U2504 ( .A1(n205), .A2(n299), .B(n2751), .C(n2752), .ZN(n2750)
          );
    AOI21D0BWP7T U2505 ( .A1(n1017), .A2(n413), .B(n1247), .ZN(n2752) );
    OAI22D0BWP7T U2506 ( .A1(n232), .A2(n1454), .B1(n926), .B2(n414), .ZN(n2751)
          );
    CKND2D0BWP7T U2507 ( .A1(n285), .A2(n2737), .ZN(n414) );
    CKND2D0BWP7T U2508 ( .A1(n259), .A2(n2746), .ZN(n1454) );
    CKND0BWP7T U2509 ( .I(n198), .ZN(n299) );
    AOI22D0BWP7T U2510 ( .A1(n358), .A2(n764), .B1(n372), .B2(n351), .ZN(n2749)
          );
    CKND0BWP7T U2511 ( .I(n803), .ZN(n351) );
    CKND2D0BWP7T U2512 ( .A1(n2630), .A2(n2753), .ZN(n803) );
    CKND0BWP7T U2513 ( .I(n694), .ZN(n372) );
    CKND2D0BWP7T U2514 ( .A1(n269), .A2(n2737), .ZN(n694) );
    CKND0BWP7T U2515 ( .I(n681), .ZN(n358) );
    CKND2D0BWP7T U2516 ( .A1(n438), .A2(n2737), .ZN(n1980) );
    OAI211D0BWP7T U2517 ( .A1(n1638), .A2(n682), .B(n499), .C(n2754), .ZN(n1034)
          );
    AOI22D0BWP7T U2518 ( .A1(n337), .A2(n340), .B1(n853), .B2(n686), .ZN(n2754)
          );
    CKND0BWP7T U2519 ( .I(n1279), .ZN(n853) );
    AOI22D0BWP7T U2520 ( .A1(n357), .A2(n652), .B1(n355), .B2(n1040), .ZN(n499)
          );
    CKND0BWP7T U2521 ( .I(n1659), .ZN(n1040) );
    CKND0BWP7T U2522 ( .I(n199), .ZN(n355) );
    IIND4D0BWP7T U2523 ( .A1(n1649), .A2(n389), .B1(n2755), .B2(n2756), .ZN(n561)
         );
    NR4D0BWP7T U2524 ( .A1(n2757), .A2(n2758), .A3(n2759), .A4(n2760), .ZN(n2756)
         );
    AOI21D0BWP7T U2525 ( .A1(n1638), .A2(n1112), .B(n199), .ZN(n2760) );
    CKND2D0BWP7T U2526 ( .A1(n2633), .A2(n2626), .ZN(n199) );
    CKND2D0BWP7T U2527 ( .A1(n2746), .A2(n268), .ZN(n1112) );
    AOI21D0BWP7T U2528 ( .A1(n197), .A2(n412), .B(n1248), .ZN(n2759) );
```

```
    AOI31D0BWP7T U2529 ( .A1(n804), .A2(n1296), .A3(n927), .B(n759), .ZN(n2758)
          );
    CKND2D0BWP7T U2530 ( .A1(n293), .A2(n2746), .ZN(n927) );
    CKND2D0BWP7T U2531 ( .A1(n2746), .A2(n296), .ZN(n1296) );
    CKND2D0BWP7T U2532 ( .A1(n427), .A2(n2746), .ZN(n804) );
    OAI222D0BWP7T U2533 ( .A1(n1136), .A2(n149), .B1(n2132), .B2(n1017), .C1(
         n146), .C2(n1659), .ZN(n2757) );
    CKND2D0BWP7T U2534 ( .A1(n282), .A2(n2746), .ZN(n1659) );
    CKND2D0BWP7T U2535 ( .A1(n2607), .A2(n2625), .ZN(n146) );
    NR2D0BWP7T U2536 ( .A1(n652), .A2(n371), .ZN(n2132) );
    CKND0BWP7T U2537 ( .I(n197), .ZN(n371) );
    CKND2D0BWP7T U2538 ( .A1(n460), .A2(n2737), .ZN(n197) );
    CKND0BWP7T U2539 ( .I(n1671), .ZN(n652) );
    CKND2D0BWP7T U2540 ( .A1(n583), .A2(n2746), .ZN(n1671) );
    CKND2D0BWP7T U2541 ( .A1(n266), .A2(n2737), .ZN(n149) );
    NR2D0BWP7T U2542 ( .A1(n352), .A2(n686), .ZN(n1136) );
    CKND0BWP7T U2543 ( .I(n148), .ZN(n686) );
    CKND0BWP7T U2544 ( .I(n440), .ZN(n352) );
    CKND2D0BWP7T U2545 ( .A1(n2624), .A2(n2633), .ZN(n440) );
    AOI221D0BWP7T U2546 ( .A1(n349), .A2(n357), .B1(n337), .B2(n805), .C(n245),
         .ZN(n2755) );
    OAI221D0BWP7T U2547 ( .A1(n1247), .A2(n148), .B1(n198), .B2(n1321), .C(n2761)
        , .ZN(n245) );
    AOI22D0BWP7T U2548 ( .A1(n854), .A2(n805), .B1(n586), .B2(n350), .ZN(n2761)
          );
    CKND0BWP7T U2549 ( .I(n147), .ZN(n350) );
    CKND0BWP7T U2550 ( .I(n802), .ZN(n586) );
    CKND0BWP7T U2551 ( .I(n2064), .ZN(n854) );
    CKND2D0BWP7T U2552 ( .A1(n266), .A2(n2746), .ZN(n2064) );
    CKND2D0BWP7T U2553 ( .A1(n2737), .A2(n296), .ZN(n198) );
    CKND2D0BWP7T U2554 ( .A1(n2762), .A2(n2626), .ZN(n148) );
    CKND0BWP7T U2555 ( .I(n232), .ZN(n805) );
    CKND2D0BWP7T U2556 ( .A1(n2624), .A2(n2607), .ZN(n232) );
    CKND0BWP7T U2557 ( .I(n1637), .ZN(n337) );
    CKND2D0BWP7T U2558 ( .A1(n268), .A2(n2737), .ZN(n1637) );
    CKND0BWP7T U2559 ( .I(n413), .ZN(n357) );
    CKND2D0BWP7T U2560 ( .A1(n2626), .A2(n2607), .ZN(n413) );
    OAI211D0BWP7T U2561 ( .A1(n681), .A2(n802), .B(n1667), .C(n2763), .ZN(n389)
          );
    AOI22D0BWP7T U2562 ( .A1(n829), .A2(n764), .B1(n349), .B2(n696), .ZN(n2763)
          );
    CKND0BWP7T U2563 ( .I(n926), .ZN(n696) );
    CKND2D0BWP7T U2564 ( .A1(n2626), .A2(n2753), .ZN(n926) );
    CKND0BWP7T U2565 ( .I(n728), .ZN(n764) );
    CKND2D0BWP7T U2566 ( .A1(n2624), .A2(n2753), .ZN(n728) );
    CKND0BWP7T U2567 ( .I(n412), .ZN(n829) );
    AOI22D0BWP7T U2568 ( .A1(n1129), .A2(n349), .B1(n340), .B2(n353), .ZN(n1667)
          );
    CKND0BWP7T U2569 ( .I(n1114), .ZN(n353) );
    CKND2D0BWP7T U2570 ( .A1(n282), .A2(n2737), .ZN(n1114) );
    CKND0BWP7T U2571 ( .I(n759), .ZN(n340) );
    CKND0BWP7T U2572 ( .I(n157), .ZN(n349) );
    CKND2D0BWP7T U2573 ( .A1(n285), .A2(n2746), .ZN(n157) );
    CKND0BWP7T U2574 ( .I(n1321), .ZN(n1129) );
```

```
    CKND2D0BWP7T U2575 ( .A1(n438), .A2(n2746), .ZN(n681) );
    OAI221D0BWP7T U2576 ( .A1(n759), .A2(n1279), .B1(n802), .B2(n760), .C(n2764),
         .ZN(n1649) );
    AOI22D0BWP7T U2577 ( .A1(n348), .A2(n687), .B1(n205), .B2(n530), .ZN(n2764)
          );
    CKND0BWP7T U2578 ( .I(n410), .ZN(n530) );
    CKND2D0BWP7T U2579 ( .A1(n601), .A2(n2746), .ZN(n410) );
    CKND0BWP7T U2580 ( .I(n1248), .ZN(n205) );
    CKND2D0BWP7T U2581 ( .A1(n2762), .A2(n2630), .ZN(n1248) );
    CKND0BWP7T U2582 ( .I(n231), .ZN(n687) );
    CKND2D0BWP7T U2583 ( .A1(n259), .A2(n2737), .ZN(n231) );
    CKND0BWP7T U2584 ( .I(n682), .ZN(n348) );
    CKND2D0BWP7T U2585 ( .A1(n2762), .A2(n2624), .ZN(n682) );
    CKND2D0BWP7T U2586 ( .A1(n293), .A2(n2737), .ZN(n760) );
    CKND2D0BWP7T U2587 ( .A1(n2633), .A2(n2630), .ZN(n802) );
    CKND2D0BWP7T U2588 ( .A1(n294), .A2(n2737), .ZN(n1279) );
    OAI211D0BWP7T U2589 ( .A1(n910), .A2(n1367), .B(n2765), .C(n2766), .ZN(n1498)
        );
    AOI211D0BWP7T U2590 ( .A1(n1095), .A2(n631), .B(n939), .C(n956), .ZN(n2766)
          );
    OAI22D0BWP7T U2591 ( .A1(n435), .A2(n1784), .B1(n614), .B2(n1352), .ZN(n956)
          );
    CKND2D0BWP7T U2592 ( .A1(n2767), .A2(n573), .ZN(n1784) );
    OAI222D0BWP7T U2593 ( .A1(n865), .A2(n1449), .B1(n475), .B2(n617), .C1(n1255)
       , .C2(n680), .ZN(n939) );
    CKND2D0BWP7T U2594 ( .A1(n601), .A2(n2494), .ZN(n1255) );
    CKND2D0BWP7T U2595 ( .A1(n2767), .A2(n1443), .ZN(n617) );
    CKND2D0BWP7T U2596 ( .A1(n2497), .A2(n2573), .ZN(n1449) );
    CKND0BWP7T U2597 ( .I(n2010), .ZN(n631) );
    CKND2D0BWP7T U2598 ( .A1(n293), .A2(n2494), .ZN(n2010) );
    AN2D0BWP7T U2599 ( .A1(n2768), .A2(n2769), .Z(n2494) );
    CKND0BWP7T U2600 ( .I(n680), .ZN(n1095) );
    CKND2D0BWP7T U2601 ( .A1(n573), .A2(n2626), .ZN(n680) );
    AOI21D0BWP7T U2602 ( .A1(n602), .A2(n581), .B(n2770), .ZN(n2765) );
    AOI21D0BWP7T U2603 ( .A1(n684), .A2(n613), .B(n435), .ZN(n2770) );
    CKND2D0BWP7T U2604 ( .A1(n2497), .A2(n2103), .ZN(n613) );
    AN2D0BWP7T U2605 ( .A1(n2771), .A2(n1443), .Z(n2497) );
    CKND2D0BWP7T U2606 ( .A1(n2614), .A2(n289), .ZN(n684) );
    CKND0BWP7T U2607 ( .I(n154), .ZN(n581) );
    CKND0BWP7T U2608 ( .I(n932), .ZN(n602) );
    CKND2D0BWP7T U2609 ( .A1(n2560), .A2(n2103), .ZN(n932) );
    AN2D0BWP7T U2610 ( .A1(n2771), .A2(n573), .Z(n2560) );
    CKND2D0BWP7T U2611 ( .A1(n2548), .A2(n1978), .ZN(n1367) );
    IND3D0BWP7T U2612 ( .A1(n1466), .B1(n1380), .B2(n1105), .ZN(n1767) );
    CKND2D0BWP7T U2613 ( .A1(n1107), .A2(n601), .ZN(n1105) );
    CKND0BWP7T U2614 ( .I(n1352), .ZN(n1107) );
    CKND2D0BWP7T U2615 ( .A1(n2767), .A2(n289), .ZN(n1352) );
    CKND2D0BWP7T U2616 ( .A1(n2487), .A2(n291), .ZN(n1380) );
    CKND0BWP7T U2617 ( .I(n2314), .ZN(n2487) );
    CKND2D0BWP7T U2618 ( .A1(n2458), .A2(n282), .ZN(n2314) );
    AN2D0BWP7T U2619 ( .A1(n2768), .A2(n2103), .Z(n2458) );
    NR2D0BWP7T U2620 ( .A1(n1394), .A2(n614), .ZN(n1466) );
    CKND2D0BWP7T U2621 ( .A1(n2548), .A2(n2573), .ZN(n1394) );
    AN2D0BWP7T U2622 ( .A1(n2771), .A2(n289), .Z(n2548) );
```

```
    NR4D0BWP7T U2623 ( .A1(n2772), .A2(n1792), .A3(n210), .A4(n1790), .ZN(n2021)
        );
    CKND0BWP7T U2624 ( .I(n1121), .ZN(n1790) );
    CKND2D0BWP7T U2625 ( .A1(n2614), .A2(n573), .ZN(n1121) );
    AN2D0BWP7T U2626 ( .A1(n2768), .A2(n1978), .Z(n2614) );
    CKND0BWP7T U2627 ( .I(n819), .ZN(n210) );
    CKND2D0BWP7T U2628 ( .A1(n2265), .A2(n2601), .ZN(n819) );
    AN2D0BWP7T U2629 ( .A1(n2762), .A2(n6329), .Z(n2601) );
    CKND0BWP7T U2630 ( .I(n2043), .ZN(n1792) );
    CKND2D0BWP7T U2631 ( .A1(n2616), .A2(n1978), .ZN(n2043) );
    CKND0BWP7T U2632 ( .I(n1910), .ZN(n1978) );
    CKND2D0BWP7T U2633 ( .A1(n2624), .A2(n2773), .ZN(n1910) );
    ND4D0BWP7T U2634 ( .A1(n757), .A2(n1282), .A3(n159), .A4(n1226), .ZN(n2772)
        );
    CKND0BWP7T U2635 ( .I(n1693), .ZN(n1226) );
    NR2D0BWP7T U2636 ( .A1(n825), .A2(n329), .ZN(n1693) );
    CKND2D0BWP7T U2637 ( .A1(n2607), .A2(contador[0]), .ZN(n329) );
    CKND2D0BWP7T U2638 ( .A1(n2265), .A2(n460), .ZN(n825) );
    CKND2D0BWP7T U2639 ( .A1(n1007), .A2(n482), .ZN(n159) );
    CKND0BWP7T U2640 ( .I(n889), .ZN(n1007) );
    CKND2D0BWP7T U2641 ( .A1(n828), .A2(n2573), .ZN(n889) );
    CKND0BWP7T U2642 ( .I(n327), .ZN(n828) );
    CKND2D0BWP7T U2643 ( .A1(n2633), .A2(contador[0]), .ZN(n327) );
    CKND0BWP7T U2644 ( .I(n800), .ZN(n1282) );
    NR2D0BWP7T U2645 ( .A1(n1321), .A2(n412), .ZN(n800) );
    CKND2D0BWP7T U2646 ( .A1(n427), .A2(n2737), .ZN(n412) );
    CKND2D0BWP7T U2647 ( .A1(n282), .A2(n2103), .ZN(n757) );
    AN2D0BWP7T U2648 ( .A1(n2625), .A2(n2773), .Z(n2103) );
    NR4D0BWP7T U2649 ( .A1(n2774), .A2(n2775), .A3(n1610), .A4(n2776), .ZN(n2020)
       );
    AOI21D0BWP7T U2650 ( .A1(n154), .A2(n910), .B(n155), .ZN(n2776) );
    CKND2D0BWP7T U2651 ( .A1(n2616), .A2(n2573), .ZN(n155) );
    AN2D0BWP7T U2652 ( .A1(n2771), .A2(n291), .Z(n2616) );
    NR2D0BWP7T U2653 ( .A1(n266), .A2(n482), .ZN(n154) );
    NR2D0BWP7T U2654 ( .A1(n1017), .A2(n1638), .ZN(n1610) );
    CKND2D0BWP7T U2655 ( .A1(n460), .A2(n2746), .ZN(n1638) );
    CKND2D0BWP7T U2656 ( .A1(n2753), .A2(n2625), .ZN(n1017) );
    OAI21D0BWP7T U2657 ( .A1(n2212), .A2(n407), .B(n2777), .ZN(n2775) );
    OAI21D0BWP7T U2658 ( .A1(n1799), .A2(n639), .B(n460), .ZN(n2777) );
    CKND0BWP7T U2659 ( .I(n228), .ZN(n639) );
    CKND2D0BWP7T U2660 ( .A1(n2572), .A2(n2573), .ZN(n228) );
    AN2D0BWP7T U2661 ( .A1(n2762), .A2(contador[0]), .Z(n2572) );
    CKND0BWP7T U2662 ( .I(n577), .ZN(n1799) );
    CKND2D0BWP7T U2663 ( .A1(n2767), .A2(n291), .ZN(n577) );
    AN2D0BWP7T U2664 ( .A1(n2265), .A2(n2768), .Z(n2767) );
    AN2D0BWP7T U2665 ( .A1(n2630), .A2(n2773), .Z(n2265) );
    CKND2D0BWP7T U2666 ( .A1(n693), .A2(n2509), .ZN(n407) );
    AN2D0BWP7T U2667 ( .A1(n2771), .A2(n2769), .Z(n2509) );
    NR2D0BWP7T U2668 ( .A1(contador[6]), .A2(n6338), .ZN(n2771) );
    NR2D0BWP7T U2669 ( .A1(n216), .A2(n608), .ZN(n2212) );
    CKND0BWP7T U2670 ( .I(n226), .ZN(n608) );
    CKND2D0BWP7T U2671 ( .A1(n573), .A2(n2625), .ZN(n226) );
    CKND0BWP7T U2672 ( .I(n346), .ZN(n216) );
    CKND2D0BWP7T U2673 ( .A1(n1443), .A2(n2630), .ZN(n346) );
```

```
    OAI222D0BWP7T U2674 ( .A1(n1247), .A2(n1113), .B1(n237), .B2(n1321), .C1(
        n759), .C2(n147), .ZN(n2774) );
    CKND2D0BWP7T U2675 ( .A1(n693), .A2(n2746), .ZN(n147) );
    CKND2D0BWP7T U2676 ( .A1(n2630), .A2(n2607), .ZN(n759) );
    CKND2D0BWP7T U2677 ( .A1(n2762), .A2(n2625), .ZN(n1321) );
    CKND2D0BWP7T U2678 ( .A1(n269), .A2(n2746), .ZN(n237) );
    AN2D0BWP7T U2679 ( .A1(n2769), .A2(contador[0]), .Z(n2746) );
    CKND2D0BWP7T U2680 ( .A1(n2633), .A2(n2625), .ZN(n1113) );
    NR2D0BWP7T U2681 ( .A1(contador[7]), .A2(n3527), .ZN(n2625) );
    CKND2D0BWP7T U2682 ( .A1(n482), .A2(n2737), .ZN(n1247) );
    AN2D0BWP7T U2683 ( .A1(n2769), .A2(n6329), .Z(n2737) );
    AN2D0BWP7T U2684 ( .A1(n6417), .A2(n2912), .Z(n2769) );
    NR4D0BWP7T U2685 ( .A1(n265), .A2(n970), .A3(n649), .A4(n175), .ZN(n1949) );
    IND3D0BWP7T U2686 ( .A1(n1807), .B1(n662), .B2(n2778), .ZN(n175) );
    AOI21D0BWP7T U2687 ( .A1(n263), .A2(n438), .B(n1342), .ZN(n2778) );
    NR2D0BWP7T U2688 ( .A1(n909), .A2(n318), .ZN(n1342) );
    CKND0BWP7T U2689 ( .I(n2779), .ZN(n662) );
    IND4D0BWP7T U2690 ( .A1(n2092), .B1(n2261), .B2(n2780), .B3(n2781), .ZN(
        n2779) );
    NR4D0BWP7T U2691 ( .A1(n2782), .A2(n2783), .A3(n466), .A4(n1922), .ZN(n2781)
        );
    AOI31D0BWP7T U2692 ( .A1(n1347), .A2(n432), .A3(n143), .B(n46), .ZN(n1922)
        );
    NR2D0BWP7T U2693 ( .A1(n474), .A2(n49), .ZN(n466) );
    AOI21D0BWP7T U2694 ( .A1(n1803), .A2(n816), .B(n458), .ZN(n2783) );
    NR2D0BWP7T U2695 ( .A1(n285), .A2(n438), .ZN(n816) );
    OAI22D0BWP7T U2696 ( .A1(n1366), .A2(n1575), .B1(n2784), .B2(n187), .ZN(
        n2782) );
    AN4D0BWP7T U2697 ( .A1(n1347), .A2(n2285), .A3(n2603), .A4(n1241), .Z(n2784)
        );
    NR2D0BWP7T U2698 ( .A1(n294), .A2(n693), .ZN(n1241) );
    CKND0BWP7T U2699 ( .I(n435), .ZN(n693) );
    NR2D0BWP7T U2700 ( .A1(n482), .A2(n296), .ZN(n2603) );
    NR2D0BWP7T U2701 ( .A1(n427), .A2(n268), .ZN(n2285) );
    NR2D0BWP7T U2702 ( .A1(n293), .A2(n282), .ZN(n1347) );
    CKND0BWP7T U2703 ( .I(n35), .ZN(n282) );
    NR2D0BWP7T U2704 ( .A1(n285), .A2(n266), .ZN(n1366) );
    AOI22D0BWP7T U2705 ( .A1(n600), .A2(n1897), .B1(n970), .B2(n268), .ZN(n2780)
        );
    CKND2D0BWP7T U2706 ( .A1(n49), .A2(n614), .ZN(n1897) );
    IINR4D0BWP7T U2707 ( .A1(n2785), .A2(n2786), .B1(n504), .B2(n1311), .ZN(
        n2261) );
    OAI21D0BWP7T U2708 ( .A1(n618), .A2(n975), .B(n2787), .ZN(n1311) );
    OAI21D0BWP7T U2709 ( .A1(n259), .A2(n460), .B(n286), .ZN(n2787) );
    CKND0BWP7T U2710 ( .I(n579), .ZN(n286) );
    OAI22D0BWP7T U2711 ( .A1(n618), .A2(n36), .B1(n196), .B2(n417), .ZN(n504) );
    CKND0BWP7T U2712 ( .I(n266), .ZN(n618) );
    AOI211D0BWP7T U2713 ( .A1(n264), .A2(n427), .B(n2788), .C(n381), .ZN(n2786)
        );
    OAI221D0BWP7T U2714 ( .A1(n227), .A2(n975), .B1(n435), .B2(n612), .C(n2789),
        .ZN(n381) );
    AOI22D0BWP7T U2715 ( .A1(n264), .A2(n294), .B1(n2136), .B2(n269), .ZN(n2789)
        );
    CKND0BWP7T U2716 ( .I(n436), .ZN(n2136) );
```

```
    IND3D0BWP7T U2717 ( .A1(n468), .B1(n934), .B2(n2790), .ZN(n2788) );
    AO31D0BWP7T U2718 ( .A1(n1816), .A2(n614), .A3(n35), .B(n612), .Z(n2790) );
    NR2D0BWP7T U2719 ( .A1(n294), .A2(n482), .ZN(n1816) );
    CKND2D0BWP7T U2720 ( .A1(n261), .A2(n482), .ZN(n934) );
    NR2D0BWP7T U2721 ( .A1(n909), .A2(n35), .ZN(n468) );
    NR3D0BWP7T U2722 ( .A1(n1939), .A2(n1213), .A3(n917), .ZN(n2785) );
    OAI22D0BWP7T U2723 ( .A1(n512), .A2(n909), .B1(n614), .B2(n187), .ZN(n917)
          );
    CKND0BWP7T U2724 ( .I(n438), .ZN(n614) );
    CKND0BWP7T U2725 ( .I(n294), .ZN(n512) );
    OAI222D0BWP7T U2726 ( .A1(n435), .A2(n911), .B1(n1470), .B2(n909), .C1(n820),
         .C2(n476), .ZN(n1213) );
    CKND0BWP7T U2727 ( .I(n293), .ZN(n820) );
    NR2D0BWP7T U2728 ( .A1(n427), .A2(n259), .ZN(n1470) );
    ND3D0BWP7T U2729 ( .A1(n2791), .A2(n2792), .A3(n2793), .ZN(n1939) );
    NR4D0BWP7T U2730 ( .A1(n1562), .A2(n950), .A3(n552), .A4(n1806), .ZN(n2793)
          );
    OAI211D0BWP7T U2731 ( .A1(n839), .A2(n36), .B(n2794), .C(n2795), .ZN(n1806)
          );
    AOI22D0BWP7T U2732 ( .A1(n268), .A2(n969), .B1(n295), .B2(n427), .ZN(n2795)
          );
    CKND0BWP7T U2733 ( .I(n227), .ZN(n427) );
    OAI21D0BWP7T U2734 ( .A1(n976), .A2(n295), .B(n269), .ZN(n2794) );
    CKND0BWP7T U2735 ( .I(n909), .ZN(n976) );
    NR2D0BWP7T U2736 ( .A1(n285), .A2(n259), .ZN(n839) );
    OAI211D0BWP7T U2737 ( .A1(n475), .A2(n476), .B(n379), .C(n2796), .ZN(n552)
          );
    AOI22D0BWP7T U2738 ( .A1(n914), .A2(n1925), .B1(n600), .B2(n460), .ZN(n2796)
          );
    CKND0BWP7T U2739 ( .I(n36), .ZN(n600) );
    CKND0BWP7T U2740 ( .I(n2344), .ZN(n1925) );
    AOI21D0BWP7T U2741 ( .A1(n438), .A2(n261), .B(n1543), .ZN(n379) );
    NR2D0BWP7T U2742 ( .A1(n46), .A2(n318), .ZN(n1543) );
    ND4D0BWP7T U2743 ( .A1(n2797), .A2(n2798), .A3(n2799), .A4(n2800), .ZN(n950)
          );
    AOI211D0BWP7T U2744 ( .A1(n603), .A2(n2801), .B(n465), .C(n2802), .ZN(n2800)
          );
    AOI21D0BWP7T U2745 ( .A1(n49), .A2(n435), .B(n579), .ZN(n2802) );
    NR2D0BWP7T U2746 ( .A1(n975), .A2(n865), .ZN(n465) );
    CKND2D0BWP7T U2747 ( .A1(n432), .A2(n196), .ZN(n2801) );
    CKND0BWP7T U2748 ( .I(n268), .ZN(n196) );
    AOI21D0BWP7T U2749 ( .A1(n295), .A2(n287), .B(n2803), .ZN(n2799) );
    IAO21D0BWP7T U2750 ( .A1(n283), .A2(n261), .B(n227), .ZN(n2803) );
    CKND2D0BWP7T U2751 ( .A1(n436), .A2(n36), .ZN(n283) );
    CKND2D0BWP7T U2752 ( .A1(n2804), .A2(n2762), .ZN(n36) );
    CKND2D0BWP7T U2753 ( .A1(n2805), .A2(n1443), .ZN(n436) );
    NR2D0BWP7T U2754 ( .A1(contador[0]), .A2(contador[5]), .ZN(n1443) );
    CKND0BWP7T U2755 ( .I(n683), .ZN(n287) );
    NR2D0BWP7T U2756 ( .A1(n438), .A2(n268), .ZN(n683) );
    AOI22D0BWP7T U2757 ( .A1(n264), .A2(n259), .B1(n914), .B2(n293), .ZN(n2798)
          );
    CKND0BWP7T U2758 ( .I(n417), .ZN(n914) );
    AOI22D0BWP7T U2759 ( .A1(n292), .A2(n583), .B1(n261), .B2(n266), .ZN(n2797)
          );
```

```
    OAI21D0BWP7T U2760 ( .A1(n1803), .A2(n909), .B(n2807), .ZN(n1562) );
    OAI21D0BWP7T U2761 ( .A1(n264), .A2(n295), .B(n482), .ZN(n2807) );
    CKND0BWP7T U2762 ( .I(n911), .ZN(n264) );
    CKND2D0BWP7T U2763 ( .A1(n2808), .A2(n2762), .ZN(n909) );
    NR3D0BWP7T U2764 ( .A1(contador[10]), .A2(n6416), .A3(contador[6]), .ZN(
        n2762) );
    NR2D0BWP7T U2765 ( .A1(n266), .A2(n460), .ZN(n1803) );
    OA221D0BWP7T U2766 ( .A1(n579), .A2(n227), .B1(n417), .B2(n35), .C(n1225),
        .Z(n2792) );
    CKND2D0BWP7T U2767 ( .A1(n970), .A2(n460), .ZN(n1225) );
    ND3D0BWP7T U2768 ( .A1(n6420), .A2(contador[1]), .A3(n2809), .ZN(n35) );
    AOI22D0BWP7T U2769 ( .A1(n292), .A2(n259), .B1(n261), .B2(n285), .ZN(n2791)
        );
    CKND0BWP7T U2770 ( .I(n474), .ZN(n261) );
    CKND0BWP7T U2771 ( .I(n143), .ZN(n259) );
    OAI211D0BWP7T U2772 ( .A1(n318), .A2(n612), .B(n1915), .C(n2810), .ZN(n2092)
        );
    MAOI22D0BWP7T U2773 ( .A1(n969), .A2(n2524), .B1(n417), .B2(n2324), .ZN(
        n2810) );
    NR2D0BWP7T U2774 ( .A1(n269), .A2(n296), .ZN(n2324) );
    CKND0BWP7T U2775 ( .I(n49), .ZN(n269) );
    ND3D0BWP7T U2776 ( .A1(n6419), .A2(contador[2]), .A3(n2811), .ZN(n49) );
    CKND0BWP7T U2777 ( .I(n1115), .ZN(n2524) );
    NR2D0BWP7T U2778 ( .A1(n285), .A2(n294), .ZN(n1115) );
    CKND2D0BWP7T U2779 ( .A1(n975), .A2(n579), .ZN(n969) );
    OA221D0BWP7T U2780 ( .A1(n910), .A2(n417), .B1(n865), .B2(n579), .C(n2812),
        .Z(n1915) );
    AOI22D0BWP7T U2781 ( .A1(n292), .A2(n293), .B1(n603), .B2(n266), .ZN(n2812)
        );
    CKND0BWP7T U2782 ( .I(n612), .ZN(n603) );
    CKND2D0BWP7T U2783 ( .A1(n2805), .A2(n573), .ZN(n579) );
    NR2D0BWP7T U2784 ( .A1(n6416), .A2(n6329), .ZN(n573) );
    CKND0BWP7T U2785 ( .I(n482), .ZN(n865) );
    CKND2D0BWP7T U2786 ( .A1(n2804), .A2(n2607), .ZN(n417) );
    CKND2D0BWP7T U2787 ( .A1(n2805), .A2(n289), .ZN(n612) );
    NR2D0BWP7T U2788 ( .A1(contador[0]), .A2(n6416), .ZN(n289) );
    OAI211D0BWP7T U2789 ( .A1(n318), .A2(n458), .B(n2814), .C(n2815), .ZN(n1807)
        );
    OA211D0BWP7T U2790 ( .A1(n187), .A2(n143), .B(n2171), .C(n2237), .Z(n2815)
        );
    AOI22D0BWP7T U2791 ( .A1(n601), .A2(n292), .B1(n2368), .B2(n295), .ZN(n2237)
        );
    CKND0BWP7T U2792 ( .I(n476), .ZN(n295) );
    CKND2D0BWP7T U2793 ( .A1(n435), .A2(n432), .ZN(n2368) );
    ND3D0BWP7T U2794 ( .A1(n6421), .A2(contador[4]), .A3(n2809), .ZN(n435) );
    CKND0BWP7T U2795 ( .I(n975), .ZN(n292) );
    CKND2D0BWP7T U2796 ( .A1(n2808), .A2(n2753), .ZN(n975) );
    CKND0BWP7T U2797 ( .I(n432), .ZN(n601) );
    ND3D0BWP7T U2798 ( .A1(n6420), .A2(n6421), .A3(n2809), .ZN(n432) );
    AOI21D0BWP7T U2799 ( .A1(n607), .A2(n583), .B(n502), .ZN(n2171) );
    OAI22D0BWP7T U2800 ( .A1(n910), .A2(n187), .B1(n931), .B2(n476), .ZN(n502)
        );
    CKND2D0BWP7T U2801 ( .A1(n2804), .A2(n2753), .ZN(n476) );
    NR2D0BWP7T U2802 ( .A1(n266), .A2(n296), .ZN(n931) );
```

```
    CKND0BWP7T U2803 ( .I(n285), .ZN(n910) );
    CKND2D0BWP7T U2804 ( .A1(n474), .A2(n911), .ZN(n607) );
    CKND2D0BWP7T U2805 ( .A1(n2805), .A2(n291), .ZN(n911) );
    NR2D0BWP7T U2806 ( .A1(contador[5]), .A2(n6329), .ZN(n291) );
    AN2D0BWP7T U2807 ( .A1(n2768), .A2(n2573), .Z(n2805) );
    AN2D0BWP7T U2808 ( .A1(n2626), .A2(n2773), .Z(n2573) );
    NR2D0BWP7T U2809 ( .A1(contador[11]), .A2(n6417), .ZN(n2773) );
    NR2D0BWP7T U2810 ( .A1(n3527), .A2(n6347), .ZN(n2626) );
    NR2D0BWP7T U2811 ( .A1(n6338), .A2(n6418), .ZN(n2768) );
    CKND2D0BWP7T U2812 ( .A1(n2808), .A2(n2607), .ZN(n474) );
    NR3D0BWP7T U2813 ( .A1(contador[10]), .A2(n6418), .A3(contador[5]), .ZN(
        n2607) );
    CKND2D0BWP7T U2814 ( .A1(n2809), .A2(n2811), .ZN(n143) );
    NR2D0BWP7T U2815 ( .A1(n6419), .A2(n3528), .ZN(n2809) );
    AOI22D0BWP7T U2816 ( .A1(n970), .A2(n2816), .B1(n263), .B2(n460), .ZN(n2814)
        );
    CKND0BWP7T U2817 ( .I(n1575), .ZN(n263) );
    CKND2D0BWP7T U2818 ( .A1(n2344), .A2(n227), .ZN(n2816) );
    ND3D0BWP7T U2819 ( .A1(n3528), .A2(contador[3]), .A3(n2811), .ZN(n227) );
    NR2D0BWP7T U2820 ( .A1(n294), .A2(n583), .ZN(n2344) );
    CKND0BWP7T U2821 ( .I(n475), .ZN(n583) );
    ND3D0BWP7T U2822 ( .A1(n6419), .A2(n3528), .A3(n2811), .ZN(n475) );
    NR2D0BWP7T U2823 ( .A1(n6421), .A2(n6420), .ZN(n2811) );
    CKND2D0BWP7T U2824 ( .A1(n2817), .A2(n2577), .ZN(n458) );
    NR2D0BWP7T U2825 ( .A1(n1575), .A2(n318), .ZN(n649) );
    CKND0BWP7T U2826 ( .I(n296), .ZN(n318) );
    CKND2D0BWP7T U2827 ( .A1(n2817), .A2(n2576), .ZN(n1575) );
    NR2D0BWP7T U2828 ( .A1(n2818), .A2(n6329), .ZN(n2576) );
    AN3D0BWP7T U2829 ( .A1(n6417), .A2(contador[11]), .A3(n2624), .Z(n2817) );
    NR2D0BWP7T U2830 ( .A1(contador[8]), .A2(n6347), .ZN(n2624) );
    CKND0BWP7T U2831 ( .I(n46), .ZN(n970) );
    CKND2D0BWP7T U2832 ( .A1(n2804), .A2(n2633), .ZN(n46) );
    AN2D0BWP7T U2833 ( .A1(n2819), .A2(n6329), .Z(n2804) );
    CKND0BWP7T U2834 ( .I(n187), .ZN(n265) );
    CKND2D0BWP7T U2835 ( .A1(n2808), .A2(n2633), .ZN(n187) );
    NR3D0BWP7T U2836 ( .A1(n6416), .A2(n6418), .A3(contador[10]), .ZN(n2633) );
    AN2D0BWP7T U2837 ( .A1(n2819), .A2(contador[0]), .Z(n2808) );
    AN3D0BWP7T U2838 ( .A1(n2630), .A2(contador[11]), .A3(n6417), .Z(n2819) );
    CKND0BWP7T U2839 ( .I(reset), .ZN(n310) );
    INR2D0BWP7T U2840 ( .A1(N4361), .B1(n2820), .ZN(N4412) );
    INR2D0BWP7T U2841 ( .A1(N4360), .B1(n2820), .ZN(N4411) );
    INR2D0BWP7T U2842 ( .A1(N4359), .B1(n2820), .ZN(N4410) );
    INR2D0BWP7T U2843 ( .A1(N4358), .B1(n2820), .ZN(N4409) );
    INR2D0BWP7T U2844 ( .A1(N4357), .B1(n2820), .ZN(N4408) );
    INR2D0BWP7T U2845 ( .A1(N4356), .B1(n2820), .ZN(N4407) );
    INR2D0BWP7T U2846 ( .A1(N4355), .B1(n2820), .ZN(N4406) );
    INR2D0BWP7T U2847 ( .A1(N4354), .B1(n2820), .ZN(N4405) );
    INR2D0BWP7T U2848 ( .A1(N4353), .B1(n2820), .ZN(N4404) );
    INR2D0BWP7T U2849 ( .A1(N4352), .B1(n2820), .ZN(N4403) );
    INR2D0BWP7T U2850 ( .A1(N4351), .B1(n2820), .ZN(N4402) );
    INR2D0BWP7T U2851 ( .A1(N4350), .B1(n2820), .ZN(N4401) );
    AN3D0BWP7T U2852 ( .A1(n2821), .A2(contador[11]), .A3(n2822), .Z(n2820) );
    ND4D0BWP7T U2853 ( .A1(n6417), .A2(n2630), .A3(n6338), .A4(n2823), .ZN(n2822)
        );
```

```
  ND4D0BWP7T U2854 ( .A1(n1950), .A2(n2806), .A3(contador[5]), .A4(contador[6])
      , .ZN(n2823) );
  NR2D0BWP7T U2855 ( .A1(contador[8]), .A2(contador[7]), .ZN(n2630) );
  ND4D0BWP7T U2856 ( .A1(n5), .A2(n6417), .A3(n3527), .A4(n2824), .ZN(n2821)
       );
  OAI32D0BWP7T U2857 ( .A1(n2604), .A2(n2813), .A3(contador[4]), .B1(n2818),
       .B2(n2806), .ZN(n2824) );
  CKND2D0BWP7T U2858 ( .A1(n6419), .A2(n6420), .ZN(n2806) );
  CKND0BWP7T U2859 ( .I(n2577), .ZN(n2604) );
  NR2D0BWP7T U2860 ( .A1(n2818), .A2(contador[0]), .ZN(n2577) );
  CKND0BWP7T U2861 ( .I(n2753), .ZN(n2818) );
  NR3D0BWP7T U2862 ( .A1(contador[5]), .A2(contador[10]), .A3(contador[6]),
       .ZN(n2753) );
  CKND0BWP7T U2863 ( .I(n1950), .ZN(n5) );
  CKXOR2D0BWP7T U2864 ( .A1(select[1]), .A2(select[0]), .Z(n1950) );
endmodule


module circuit ( reset, clk, select, EN, q_out, clk_s );
  input [1:0] select;
  output [7:0] q_out;
  input reset, clk, EN;
  output clk_s;
  wire clk_s_w, reset_w, clk_w, EN_w, net8327, n2;
  wire [7:0] q_out_w;
  wire [1:0] select_w;
  tri reset;
  tri clk;
  tri [1:0] select;
  tri EN;
  tri [7:0] q_out;
  tri clk_s;

  chip_SP circuit_new ( .q_out(q_out_w), .reset(reset_w), .clk(clk_w), .EN(
       EN_w), .clk_s(clk_s_w), .select(select_w) );
  PDDW0204SCDG IN0 ( .I(net8327), .OEN(n2), .IE(n2), .PAD(reset), .DS(net8327),
       .PE(net8327), .C(reset_w) );
  PDDW0204SCDG IN1 ( .I(net8327), .OEN(n2), .IE(n2), .PAD(clk), .DS(net8327),
       .PE(net8327), .C(clk_w) );
  PDDW0204SCDG IN2 ( .I(net8327), .OEN(n2), .IE(n2), .PAD(EN), .DS(net8327),
       .PE(net8327), .C(EN_w) );
  PDDW0204SCDG IN3 ( .I(net8327), .OEN(n2), .IE(n2), .PAD(select[0]), .DS(
       net8327), .PE(net8327), .C(select_w[0]) );
  PDDW0204SCDG IN4 ( .I(net8327), .OEN(n2), .IE(n2), .PAD(select[1]), .DS(
       net8327), .PE(net8327), .C(select_w[1]) );
  PDDW0204SCDG OUT0 ( .I(clk_s_w), .OEN(net8327), .IE(net8327), .PAD(clk_s),
       .DS(net8327), .PE(net8327) );
  PDDW0204SCDG OUT1 ( .I(q_out_w[0]), .OEN(net8327), .IE(net8327), .PAD(
       q_out[0]), .DS(net8327), .PE(net8327) );
  PDDW0204SCDG OUT2 ( .I(q_out_w[1]), .OEN(net8327), .IE(net8327), .PAD(
       q_out[1]), .DS(net8327), .PE(net8327) );
  PDDW0204SCDG OUT3 ( .I(q_out_w[2]), .OEN(net8327), .IE(net8327), .PAD(
       q_out[2]), .DS(net8327), .PE(net8327) );
  PDDW0204SCDG OUT4 ( .I(q_out_w[3]), .OEN(net8327), .IE(net8327), .PAD(
```

```
        q_out[3]), .DS(net8327), .PE(net8327) );
    PDDW0204SCDG OUT5 ( .I(q_out_w[4]), .OEN(net8327), .IE(net8327), .PAD(
        q_out[4]), .DS(net8327), .PE(net8327) );
    PDDW0204SCDG OUT6 ( .I(q_out_w[5]), .OEN(net8327), .IE(net8327), .PAD(
        q_out[5]), .DS(net8327), .PE(net8327) );
    PDDW0204SCDG OUT7 ( .I(q_out_w[6]), .OEN(net8327), .IE(net8327), .PAD(
        q_out[6]), .DS(net8327), .PE(net8327) );
    PDDW0204SCDG OUT8 ( .I(net8327), .OEN(net8327), .IE(net8327), .PAD(q_out[7]),
        .DS(net8327), .PE(net8327) );
    TIELBWP7T U6 ( .ZN(net8327) );
    TIEHBWP7T U7 ( .Z(n2) );
  endmodule
```

Cuadro 134: Archivo nanochip_sintesis_final.v: *netlist* final obtenido de la síntesis lógica del circuito previamente sintetizado con pines de I/O