

---

Desarrollo de diseño nanoelectrónico en celda de 160x100 micrómetros a través del proyecto open-source Tiny TapeOut y elaboración de material académico sobre el diseño de ASIC para el programa de ingeniería electrónica de la Universidad del Valle de Guatemala

---

Gabriela Maria Alfaro Contreras





UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería




**Desarrollo de diseño nanoelectrónico en celda de 160x100 micrómetros a través del proyecto open-source Tiny TapeOut y elaboración de material académico sobre el diseño de ASIC para el programa de ingeniería electrónica de la Universidad del Valle de Guatemala**

Trabajo de graduación presentado por Gabriela Maria Alfaro Contreras para optar al grado académico de Licenciada en Ingeniería Electrónica


Guatemala,

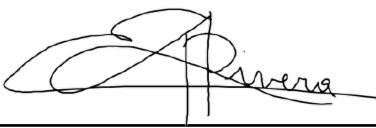
02 de Diciembre del 2024

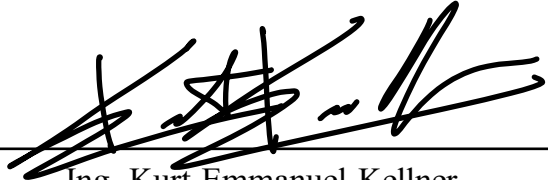
Vo.Bo.:

(f)   
M. Sc. Carlos Esquit

Tribunal Examinador:

(f)   
M.Sc. Carlos Esquit

(f)   
Dr. Luis Alberto Rivera Estrada

(f)   
Ing. Kurt Emmanuel Kellner

Fecha de aprobación: Guatemala, 02 de diciembre de 2024.

Quisiera expresar un gran agradecimiento a mi familia por proveer un apoyo incondicional, brindarme las oportunidades que me han permitido llegar hasta aquí y nunca dejar de confiar en mis capacidades. Agradezco especialmente a mi hermana, quien siempre ha estado a mi lado, ofreciendo su apoyo y motivándome en cada paso del proceso de desarrollo de este trabajo. También quiero mencionar a mi gato Teo, quien brindó mucha felicidad en mi vida en los momentos que más lo necesitaba. Finalmente, quiero decir que sin el apoyo de todos, este logro no habría sido posible.

---

## Índice

---

<b>Prefacio</b>	<b>III</b>
<b>Índice</b>	<b>IV</b>
<b>Lista de figuras</b>	<b>VII</b>
<b>Lista de cuadros</b>	<b>X</b>
<b>Resumen</b>	<b>XII</b>
<b>Abstract</b>	<b>XIV</b>
<b>1 Introducción</b>	<b>1</b>
<b>2 Antecedentes</b>	<b>3</b>
2.1 Proyecto Tiny TapeOut y chips desarrollados	3
2.1.1 Chips TT01, TT02, TT03, TT04, TT05, TT06 y TT07	4
<b>3 Justificación</b>	<b>10</b>
<b>4 Objetivos</b>	<b>11</b>
4.1 Objetivos generales	11
4.2 Objetivos específicos	11
<b>5 Alcance</b>	<b>13</b>

<b>6</b>	<b>Marco teórico</b>	<b>14</b>
6.1	Verilog HDL	14
6.1.1	Principios básicos para programación en Verilog	15
6.2	Transistores MOSFET	16
6.3	Fabricación de Circuitos Integrados	17
6.4	Uso de Wokwi para el desarrollo de proyectos de ASIC	21
6.5	Uso de OpenLane para el desarrollo de proyectos de ASIC	22
6.6	Uso de Capcut para el desarrollo de material académico	23
<b>7</b>	<b>Emisión de diseño electrónico a proyecto Tiny Tapeout</b>	<b>25</b>
7.1	Desarrollo de proyecto ALU de 4 bits en Verilog	25
7.1.1	Acerca del diseño	25
7.1.2	Restricciones del diseño	26
7.2	Proceso de emisión del diseño de una ALU a Tiny Tapeout	26
7.2.1	Creación de repositorio y accesos a Github Actions	27
7.2.1.1	Modificaciones en los archivos del repositorio	27
7.2.1.2	Ejecución de OpenLane mediante Github Actions	29
7.2.2	Caracterización del diseño electrónico final	30
<b>8</b>	<b>Desarrollo de material académico</b>	<b>34</b>
8.1	Guía para emisión de proyectos para Tiny Tapeout	34
8.1.1	Modos de emisión de proyectos: Simulador gráfico Wokwi ó HDL	34
8.1.1.1	Simulador gráfico Wokwi	34
8.1.1.2	HDL	35
8.1.2	Uso de software OpenLane	35
8.2	Material Académico para curso de Nanoelectrónica	36
<b>9</b>	<b>Conclusiones</b>	<b>37</b>

<b>10 Recomendaciones</b>	<b>39</b>
<b>11 Referencias</b>	<b>41</b>
<b>12 Anexos</b>	<b>43</b>
12.1 Código de Verilog a emitir para proyecto <i>Tiny Tapeout</i>	44
12.2 Proyecto ALU de 4 bits en datasheet oficial del chip TT09	46
12.3 Guías académicas	48
12.3.1 Guías académicas para cursos de Electrónica Digital	48
12.3.1.1 Guía 1 - Introducción a simulador Wokwi	48
12.3.1.2 Guía 2 - Envío a Tiny Tapeout de diseño en Wokwi	52
12.3.2 Guías académicas para curso de Arquitectura de Computadoras	58
12.3.2.1 Guía 1 - Adaptación y verificación de diseños en HDL	58
12.3.2.2 Guía 2 - Emisión de diseños en HDL a Tiny Tapeout	61
12.3.3 Guías académicas para cursos de Nanoelectrónica	67
12.3.3.1 Guía 1 - Introducción a Siliwiz	67
12.3.3.2 Guía 2 - Transistores en Siliwiz	72
12.3.3.3 Guía 3 - Compuertas Lógicas en Siliwiz	77
12.3.3.4 Guía 4 - Introducción a Stixu	81
12.4 Videos académicos	85
<b>13 Glosario</b>	<b>87</b>

---

## Lista de figuras

---

Figura 1. Chip TT01 [3]	4
Figura 2. Chip TT02 [4]	5
Figura 3. Chip TT03 [5]	5
Figura 4. Chip TT04 [6]	6
Figura 5. Chip TT06 [7]	7
Figura 6. Chip TT06 [8]	8
Figura 7. Chip TT07 [9]	9
Figura 8. Ejemplo de código en Verilog de compuertas lógicas	15
Figura 9. Ejemplo de código en Verilog de un flip flop	16
Figura 10. Composición de un MOSFET tipo N [15]	17
Figura 11. Fabricacion de circuitos integrados [16]	18
Figura 12. Boule [20]	18
Figura 13. Proceso de implantación iónica	19
Figura 14. Proceso de deposición química de vapor[23]	20
Figura 15. Fitolitografía [24]	20
Figura 16. Chip empaquetado [21]	21
Figura 17. Interfaz gráfica de Wokwi con proyecto ALU de 1 bit [26]	22
Figura 18. Diagrama de flujo funcionamiento OpenLane [27]	23
Figura 19. Interfaz de aplicación de escritorio Capcut	23
Figura 20. Repositorio ALU 4 bits en Github	26

Figura 21. Archivo realizado para ALU 4 bits en TT09	28
Figura 22. Flujo de trabajo seguido por OpenLane	29
Figura 23. Render GDS del diseño obtenido en OpenLane	32
Figura 24. Render GDS del diseño oficial del chip TT09	32
Figura 25. Visualización 3D del diseño de la ALU de 4 bits	33
Figura 26. Render del chip TT09 y ubicación de ALU diseñada	33
Figura 27. Código de ALU 4 bits a enviar	44
Figura 28. Testbench de ALU 4 bits a enviar a TT09	45
Figura 29. Datasheet TT09 del proyecto ALU 4 bits - Página 1	46
Figura 30. Datasheet TT09 del proyecto ALU 4 bits - Página 2	47
Figura 31. Guía 1- Introducción a Wokwi - Página 1	48
Figura 32. Guía 1- Introducción a Wokwi - Página 2	49
Figura 33. Guía 1- Introducción a Wokwi - Página 3	50
Figura 34. Guía 1- Introducción a Wokwi - Página 4	51
Figura 35. Guía 2 - Emisión desde diseño en Wokwi - Página 1	52
Figura 36. Guía 2 - Emisión desde diseño en Wokwi - Página 2	53
Figura 37. Guía 2 - Emisión desde diseño en Wokwi - Página 3	54
Figura 38. Guía 2 - Emisión desde diseño en Wokwi - Página 4	55
Figura 39. Guía 2 - Emisión desde diseño en Wokwi - Página 5	56
Figura 40. Guía 1 - Adaptación y verificación de diseños en HDL - Página 1	58
Figura 41. Guía 1 - Adaptación y verificación de diseños en HDL - Página 2	59
Figura 42. Guía 1 - Adaptación y verificación de diseños en HDL - Página 3	60
Figura 43. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 1	61
Figura 44. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 2	62
Figura 45. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 3	63
Figura 46. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 4	64
Figura 47. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 5	65

Figura 48. Guía 1 - Introducción a Siliwiz - Página 1	67
Figura 49. Guía 1 - Introducción a Siliwiz - Página 2	68
Figura 50. Guía 1 - Introducción a Siliwiz - Página 3	69
Figura 51. Guía 1 - Introducción a Siliwiz - Página 4	70
Figura 52. Guía 1 - Introducción a Siliwiz - Página 5	71
Figura 53. Guía 2 - Transistores en Siliwiz - Página 1	72
Figura 54. Guía 2 - Transistores en Siliwiz - Página 2	73
Figura 55. Guía 2 - Transistores en Siliwiz - Página 3	74
Figura 56. Guía 2 - Transistores en Siliwiz - Página 4	75
Figura 57. Guía 2 - Transistores en Siliwiz - Página 5	76
Figura 58. Guía 3 - Compuertas Lógicas en Siliwiz - Página 1	77
Figura 59. Guía 3 - Compuertas Lógicas en Siliwiz - Página 2	78
Figura 60. Guía 3 - Compuertas Lógicas en Siliwiz - Página 3	79
Figura 61. Guía 3 - Compuertas Lógicas en Siliwiz - Página 4	80
Figura 62. Guía 4 - Introducción a Stixu - Página 1	81
Figura 63. Guía 4 - Introducción a Stixu - Página 2	82
Figura 64. Guía 4 - Introducción a Stixu - Página 3	83
Figura 65. Guía 4 - Introducción a Stixu - Página 4	84

---

## Lista de cuadros

---

Cuadro 1. Recursos utilizados en el diseño de la ALU de 4 bits	31
Cuadro 2. Compuertas lógicas utilizadas en el diseño de la ALU de 4 bits	31

Este trabajo de graduación se centró en el diseño y la manufactura de un circuito digital, específicamente, una unidad aritmética y lógica (ALU) de 4 bits, implementada en Verilog y manufacturada en una celda de 160x100  $\mu\text{m}$  mediante la plataforma open-source Tiny Tapeout. El objetivo fue desarrollar un diseño digital funcional y realizar el proceso de manufacturación, con la elaboración de material de aprendizaje sobre las áreas básicas, intermedias y avanzadas de diseño ASIC mediante las herramientas Wokwi y OpenLane.

El diseño de la ALU de 4 bits fue sintetizado y optimizado en OpenLane para su manufactura, generando un archivo GDSII que cumplió con los estándares requeridos para su fabricación en Tiny Tapeout. Esta demostró un funcionamiento adecuado en la realización de operaciones aritméticas y lógicas básicas, lo que comprueba la viabilidad del flujo de trabajo propuesto para el diseño y la manufactura de ASIC en celdas de tamaño reducido.

Como parte de este proyecto, se desarrolló material académico incluyendo guías y videos educativos editados en la plataforma Capcut. Estos recursos cubrieron los aspectos básicos, intermedios y avanzados del diseño de ASIC, facilitando el aprendizaje autodirigido. Permitieron la comprensión de cada fase del proceso, desde la creación del diseño hasta su manufactura. Además, los videos proporcionaron una comprensión visual de conceptos en electrónica digital, arquitectura de computadoras y nanoelectrónica.

En conclusión, se alcanzó el objetivo de diseñar y manufacturar un ASIC funcional en una plataforma accesible de código abierto, además de proporcionar materiales educativos que complementan la formación práctica de los estudiantes de ingeniería electrónica. Este proyecto fue considerado una contribución significativa al programa académico que permite que los estudiantes adquirieran conocimientos aplicables en el diseño y la manufactura de circuitos integrados.

Se recomienda el desarrollo de más proyectos con la plataforma de Tiny Tapeout con

diseños más complejos con el fin de aprovechar al máximo los recursos educativos que ofrecen. Se sugiere evaluar cómo otras plataformas de simulación y síntesis se comparan con las herramientas open-source y si podrían complementarse en futuros proyectos.

Palabras clave: ASIC, TinyTapeout, nanoelectrónica, ALU

This graduation project focused on the design and manufacturing of a digital circuit, specifically, a 4-bit Arithmetic and Logic Unit (ALU), implemented in Verilog and fabricated in a 160x100  $\mu\text{m}$  cell using the open-source Tiny Tapeout platform. The main objective was to develop a functional digital design and carry out the manufacturing process, along with the creation of educational material for learning on basic, intermediate, and advanced ASIC design using the Wokwi and OpenLane tools.

The design of the 4-bit ALU was successfully synthesized and optimized in OpenLane for manufacturing, generating a GDSII file that met the required standards for fabrication on Tiny Tapeout. The designed ALU demonstrated proper functionality in performing basic arithmetic and logical operations, verifying the feasibility of the proposed workflow for designing and manufacturing ASICs in reduced-size cells.

Furthermore, as part of this project, academic material was developed, including detailed guides and educational videos edited on the Capcut platform. These resources covered the basic, intermediate, and advanced aspects of ASIC design, facilitating self-directed learning. The guides enabled the understanding of each phase of the process, from the creation of the digital design to its final manufacturing. Additionally, the videos provided a visual understanding of key concepts in digital electronics, computer architecture, and nanoelectronics, enhancing the learning experience and accessibility to integrated circuit design.

In conclusion, the objective of designing and manufacturing a functional ASIC on an accessible, open-source platform was achieved, as well as providing educational materials that complemented students' practical training in electronic engineering. This project was considered a significant contribution to the academic program, allowing students to acquire applicable knowledge in integrated circuit design and manufacturing.

It is recommended to develop more projects using the Tiny Tapeout platform, but with

more complex designs in order to fully take advantage of the educational resources it offers. It is suggested to evaluate how other simulation and synthesis platforms compare to the open-source tools and whether they could be integrated into future projects.

Keywords: ASIC, TinyTapeout, nanoelectronics, ALU

Este trabajo de graduación se centra en el desarrollo y manufactura de un diseño digital específico: una unidad aritmética y lógica (ALU) de 4 bits, implementada en Verilog y diseñada para una celda de 160x100  $\mu\text{m}$ . Utilizando la plataforma open-source Tiny Tapeout (TT09), este proyecto se propone no solo para demostrar la viabilidad de crear un circuito digital funcional con herramientas accesibles, sino también explorar el proceso de manufactura de un ASIC (*application-specific integrated circuit*) desde el diseño hasta la fabricación y entrega del chip físico.

El diseño de la ALU de 4 bits permite ejecutar operaciones aritméticas y lógicas básicas a través del lenguaje descriptivo de hardware (HDL) denominado Verilog. Por medio del uso de la herramienta OpenLane, se logra realizar el proceso de síntesis del diseño y la generación del archivo GDSII, indispensable en el flujo de manufactura.

Este trabajo aborda la problemática educativa en torno al acceso limitado a la manufactura de ASIC para estudiantes y profesionales interesados en la industria de semiconductores. Los altos costos y la complejidad de los procesos tradicionales dificultan el acceso para diseñar y fabricar circuitos integrados. Por ello este trabajo busca aprovechar el proyecto Tiny Tapeout para superar estas limitaciones mediante el uso de herramientas accesibles como Wokwi y OpenLane, que permiten una aproximación práctica y asequible al diseño digital.

El trabajo incluye la creación de material académico para el programa de ingeniería electrónica con el fin de cubrir los aspectos básicos, intermedios y avanzados del diseño de ASIC. Este material, permite aprender, de manera progresiva, el flujo de diseño, conceptos de electrónica digital, arquitectura de computadoras y nanoelectrónica.

Los primeros capítulos introducen el diseño de la ALU de 4 bits en Verilog, las he-

ramientas de simulación y síntesis utilizadas, así como el flujo de trabajo en TT09. Los capítulos intermedios se enfocan en el desarrollo de material académico que complementa el aprendizaje práctico, creados para reforzar el conocimiento de ASIC. El último capítulo presenta las conclusiones y recomendaciones para expandir el acceso al diseño y manufactura de circuitos integrados en el entorno académico.

### **2.1. Proyecto Tiny TapeOut y chips desarrollados**

La industria de manufactura de chips desde sus principios ha sido inalcanzable para usuarios independientes debido a los altos costos que estos implican. Por ello, Tiny TapeOut es un proyecto educacional que ha permitido, de forma accesible y sencilla, convertir diseños digitales a circuitos integrados de aplicación específica (ASIC (Application-specific integrated circuit):). Este proyecto está diseñado para personas con conocimientos básicos de diseño de electrónica digital hasta aquellas con dominio de lenguaje de descripción de hardware. Tiny TapeOut ha otorgado mayor accesibilidad a usuarios fuera de grandes empresas de chips, a manufacturar sus propios diseños dentro de una celda de área de  $150 \times 170 \mu\text{m}$  en el caso del chip más próximo a lanzar TT06 [1].

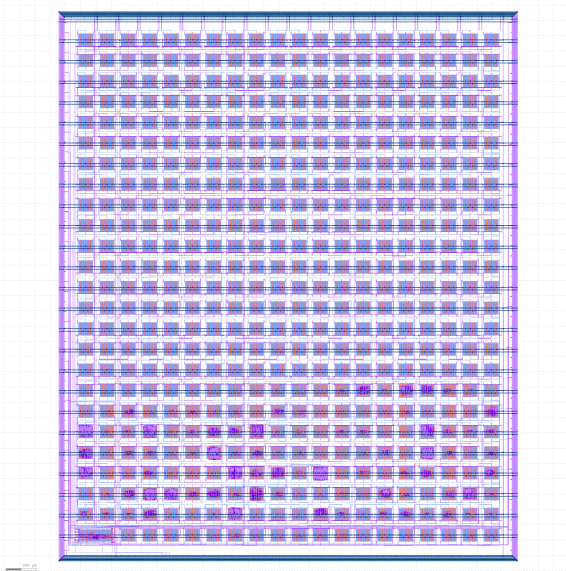
El proyecto logra alcanzar el desarrollo de los chips, se inicia a partir de emisiones de distintos diseños realizados por estudiantes o entusiastas del diseño electrónico. Posteriormente, administradores de Tiny TapeOut, elaboran un microcontrolador que contiene el chip embebido al mismo y permite el acceso a los múltiples proyectos emitidos que se encuentran dentro del chip. Para poder participar en Tiny TapeOut, se debe pagar \$ 150.00 por emisión de diseño y para recibir un chip en un PCB, si se desean agregar pines analógicos o celdas extra se pagan \$ 40.00 ó \$ 50.00 respectivamente [1].

### 2.1.1. Chips TT01, TT02, TT03, TT04, TT05, TT06 y TT07

Desde del año 2022, se han diseñado 7 distintos chips (TT01-TT07), de los cuales se esperan recibir en sus correspondientes PCBs, a lo largo del año 2024. Cada uno de ellos fueron realizados con la herramienta de Skywater open source PDK, la cual es una colaboración entre Google y SkyWater Technology Foundry para proporcionar un kit de diseño de procesos de código abierto y recursos relacionados, que se pueden utilizar para crear diseños fabricables con tecnología de 130nm. Sin embargo, cada uno de los proyectos emitidos para estos chips fueron diseñados desde el simulador Wokwi, a través de lenguajes de descripción de hardware (Verilog, Chisel, myhdl) ó a partir de lenguaje de lógica de circuitos desde python como lo es Amaranth [2]. A continuación se encuentra un poco acerca de cada uno de ellos:

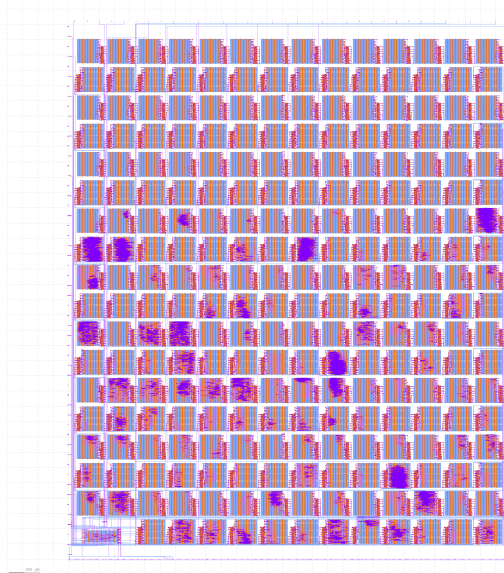
- **TT01**: emitido a Efabless en septiembre del 2022, contiene 152 proyectos (cada uno de  $100 \times 100 \mu\text{m}$ ), donde 115 personas realizaron sus diseños en el editor gráfico Wokwi, 31 en Verilog, 3 XLS, 2 Chisel y 1 Amaranth. Contiene una longitud de cableado de 772mm y utilizó 15 mil celdas estándar en total utilizadas en todos los proyectos. Es el único proyecto que no fue entregado a todo público [3].

**Figura 1.** *Chip TT01 [3]*



- **TT02**: emitido a Efabless en diciembre del 2022, contiene 165 proyectos (cada uno de  $150 \times 170 \mu\text{m}$ ), donde 64 personas realizaron sus diseños en el editor gráfico Wokwi, 82 en Verilog, 5 Amaranth y 1 Chisel. Contiene una longitud de cableado de 2024mm y utilizó 39,935 celdas estándar entre todos los proyectos [4].

Figura 2. Chip TT02 [4]



- **TT03**: emitido a Efabless en mayo del 2023, contiene 100 proyectos nuevos y 149 de TT02, donde 94 personas realizaron sus diseños en el editor gráfico Wokwi, 135 en Verilog, 7 Amaranth, 1 myhdl, 1 spade, 2 xls, 1 migen, 3 systemverilog, 1 mrccs y 1 Chisel. Contiene una longitud de cableado de 4079mm y utilizó 79,674 celdas estándar entre todos los proyectos [5].

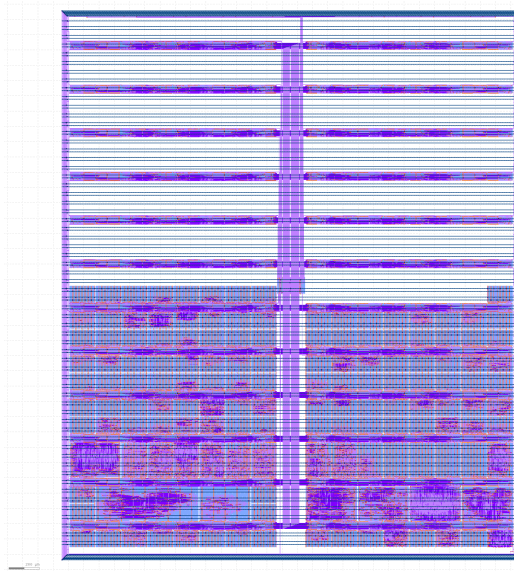
Figura 3. Chip TT03 [5]



- **TT04**: emitido a Efabless en septiembre del 2023, contiene 143 proyectos nuevos y 149 de TT02, donde 46 personas realizaron sus diseños en el editor gráfico Wokwi, 92 en Verilog, 1 spade, 1 systemverilog, 1 mrccs, 1 SpinalHDL y 1 RustHDL. Contiene

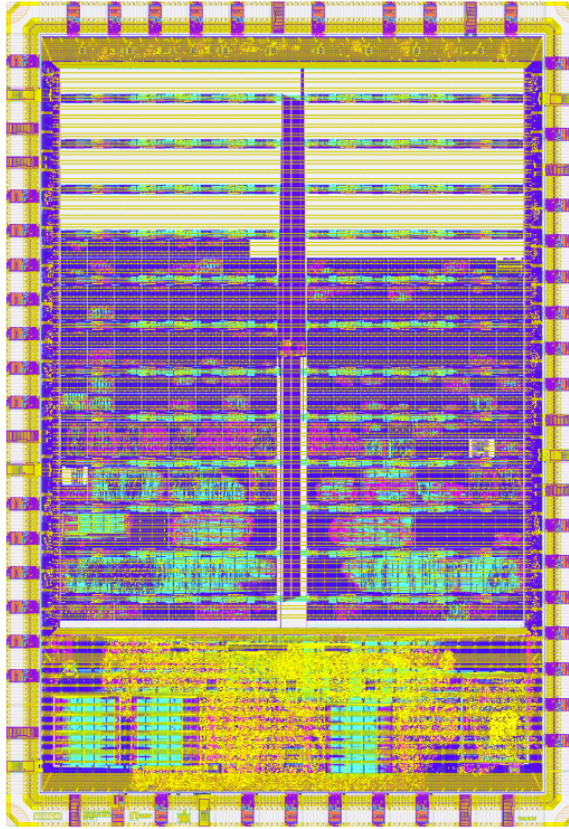
una longitud de cableado de 2607mm y utilizó 82,126 celdas estándar entre todos los proyectos [6].

**Figura 4.** *Chip TT04* [6]



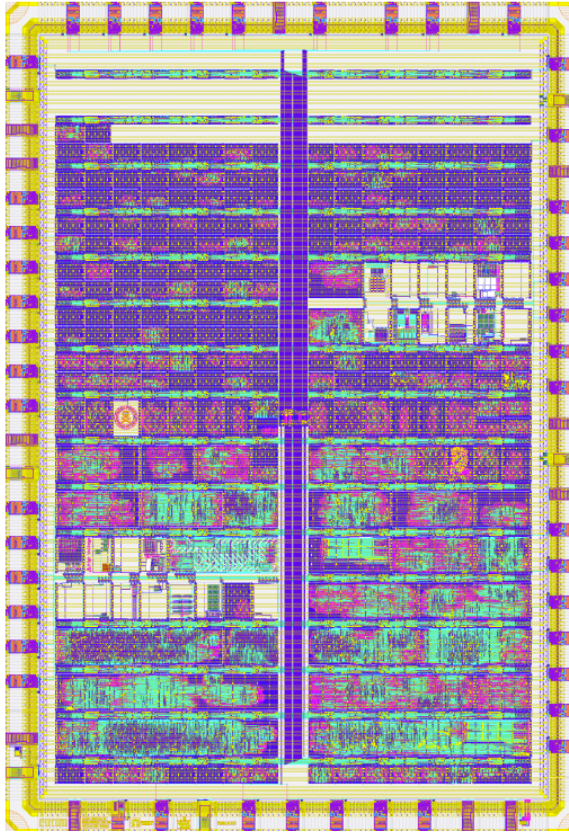
- **TT05:** emitido Efabless en noviembre del 2023, los chips se esperan en abril de 2024 y los PCBs en junio de 2024 [7].

**Figura 5.** *Chip TT06 [7]*



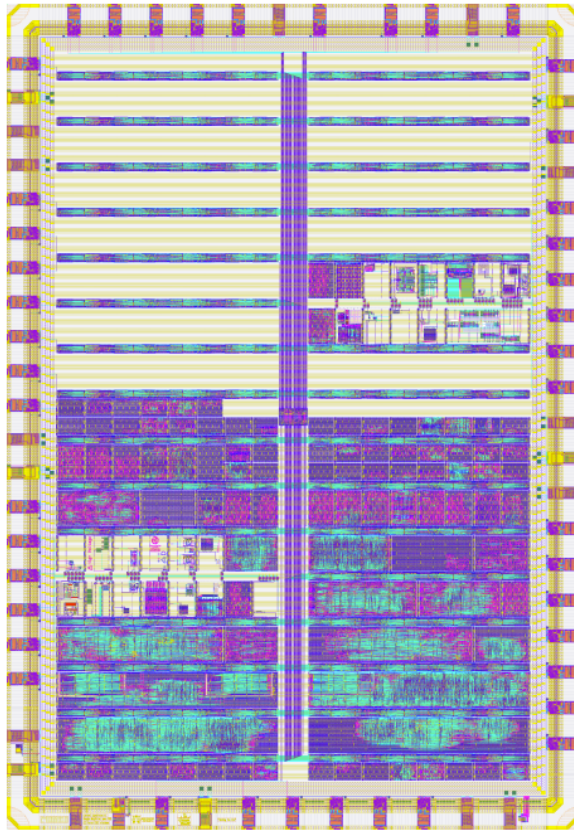
- **TT06:** emitido Efabless en abril del 2024, los chips se esperan en octubre de 2024 y los PCBs en diciembre de 2024 [8].

**Figura 6.** *Chip TT06 [8]*



- **TT07:** emitido Efabless en junio del 2024, los chips se esperan en noviembre de 2024 y los PCBs en diciembre de 2024 [9].

Figura 7. *Chip TT07 [9]*



El desarrollo de esta investigación busca realizar una contribución al avance del conocimiento en el diseño de circuitos integrados a nivel de sistema (ASIC). Mediante el uso de la plataforma de código abierto Wokwi y OpenLane que proporciona el proyecto Tiny TapeOut, se desarrollará un diseño digital específico en una celda de tamaño reducido, lo que no solo fomentará la transparencia y la colaboración en el proceso de diseño de un chip, sino que también proporcionará recursos educativos para estudiantes y profesionales interesados en este campo dentro de la Universidad del Valle de Guatemala. Además, se espera que el diseño eficiente propuesto tenga un impacto significativo que motive a continuar el desarrollo de diseños digitales.

Es importante recalcar que esta investigación busca fomentar la innovación y brindar oportunidades para que los estudiantes de ingeniería y científicos, que no cuentan con acceso a tecnologías de fabricación de circuitos integrados, exploren el potencial del silicio personalizado en aplicaciones de aprendizaje. Pues se busca aprovechar el proyecto de Tiny TapeOut como una plataforma de lanzamiento que permitirá a los futuros líderes tecnológicos lograr un impacto significativo en el campo de desarrollo de ASIC y semiconductores.

### 4.1. Objetivos generales

- Desarrollar un diseño digital funcional y realizar el proceso de manufacturación en una celda de 160x100  $\mu\text{m}$  en el proyecto open-source Tiny TapeOut.
- Elaborar material de aprendizaje universitario de las áreas básicas, intermedias y avanzadas de diseño ASIC mediante herramientas open-source Wokwi y OpenLane.

### 4.2. Objetivos específicos

- Investigar y comprender el uso de las herramientas open-source Wokwi y OpenLane.
- Investigar y comprender el proceso de emisión de diseño, manufactura, pago y entrega del proyecto a realizar en el proyecto open-source Tiny TapeOut.
- Elaborar un diseño digital en una celda de 160x100  $\mu\text{m}$  para su manufactura en el proyecto open-source Tiny TapeOut.
- Desarrollar documentación académica para cursos universitarios básicos sobre diseño de circuitos integrados para aplicaciones específicas mediante la herramienta open-source Wokwi.
- Desarrollar documentación académica para cursos universitarios avanzados sobre diseño de circuitos integrados para aplicaciones específicas (ASIC) mediante la herramienta open-source OpenLane.

- Editar y desarrollar videos intuitivos sobre electrónica digital y arquitectura de computadoras, en la plataforma CapCut.

Este trabajo de graduación tiene como alcance realizar un diseño nanoelectrónico, dentro del proyecto open-source Tiny Tapeout, que cumpla con los requisitos para emisión establecidos para su manufactura. Entre ellos, se encuentran las restricciones de tamaño, donde el diseño se encuentra dentro de una celda de 160x100 micrómetros. Además, los proyectos se realizan mediante un HDL (*Hardware descriptive language*) ó la herramienta de simulación visual Wokwi. Independientemente del método de emisión, el uso de la herramienta OpenLane es clave para la generación de los archivos GDS, los cuales son necesarios para el proceso de fabricación.

Este proyecto busca que el diseño sea manufacturado, por lo que finalizar al momento de ser recibido y aprobado por Tiny Tapeout. Además será autofinanciado por su desarrolladora.

Debido a que Tiny Tapeout es un proyecto que se encuentra ligado con la enseñanza, aprendizaje y comprensión del desarrollo de ASIC, se busca la elaboración de material académico para 3 cursos de la carrera de Ingeniería Electrónica dentro de la Universidad del Valle de Guatemala. Es importante recalcar que este material está orientado al desarrollo, emisión y manufacturación de diseños dentro el proyecto Tiny Tapeout para los cursos de Electrónica Digital 1 y Arquitectura de Computadoras. Para el curso de Nanoelectrónica 1, se enfoca a que el estudiante tenga comprensión de los conceptos fundamentales del diseño de sistemas electrónicos a nanoescala. Cabe mencionar que el material académico consiste en elaboración de guías y videos breves de un máximo de 10 minutos para asegurar la retención y la atención del estudiante que lo utilice. Finalmente, es importante recalcar que el proyecto se realizó desde enero hasta noviembre de 2024.

## 6.1. Verilog HDL

Verilog: es un lenguaje de descripción de hardware (HDL (Hardware description language):) que se utiliza para modelar sistemas electrónicos. Es un lenguaje poderoso y flexible que se puede utilizar para describir una amplia gama de sistemas, desde simples circuitos digitales hasta complejos procesadores de computadora [10]. Se utiliza principalmente para:

- **Diseñar sistemas electrónicos:** verilog se puede utilizar para describir la arquitectura y el comportamiento de un sistema electrónico. Esta descripción se puede utilizar para simular el sistema, verificar su funcionamiento y generar el código necesario para fabricarlo [11].
- **Sintetizar circuitos:** verilog se puede utilizar para sintetizar circuitos digitales a partir de una descripción de alto nivel. Esto significa que puedes describir el comportamiento deseado del circuito en Verilog y luego usar una herramienta de síntesis para generar automáticamente el circuito digital correspondiente. Este proceso es indispensable para la optimización del diseño ya que permite reducir la cantidad de hardware utilizado [11].
- **Simulación y verificación de diseños:** verilog se puede utilizar para verificar el diseño de un sistema electrónico. Esto se puede hacer mediante la simulación del sistema y la comprobación de que su comportamiento cumple con los requisitos especificados en el diseño [11].

### 6.1.1. Principios básicos para programación en Verilog

- **Módulos:** debido a que el lenguaje de Verilog es de descripción de hardware, existen bloques de hardware con entradas y salidas llamados módulos. Una compuerta lógica AND ó un multiplexor, son ejemplos de módulos de hardware, pues estos siempre tendrán una función específica que buscan realizar[12].

Existen dos formas generales para describir la funcionalidad de un módulo: estructural y procedimental. Mientras el primero busca describir cómo un módulo está construido de piezas más sencillas, el segundo describe únicamente lo que el módulo realiza.

- **Lógica Combinacional:** las salidas de la lógica combinacional dependen únicamente en las entradas actuales del módulo. Si se observa el Cuadro 1, se pueden identificar los símbolos como  $\&$  y  $|$ , estos son ejemplos de operadores. Mientras que  $a$ ,  $b$  y  $y1$ , son ejemplo de operandos. La combinación de operadores y operandos, son llamados expresiones. Las asignaciones continuas terminan con un  $;$  y cualquier momento en que la entrada que se encuentre a la derecha del  $=$  cambie, la salida será recalculada. Es por ello que las asignaciones continuas, describen la lógica combinacional a la perfección [12].

Figura 8. Ejemplo de código en Verilog de compuertas lógicas

```
1 module gates (  
2     input logic [3:0] a, b  
3     output logic [3:0] y1, y2, y3, y4, y5);  
4  
5     assign y1 = a & b;           // AND  
6     assign y2 = a | b;         // OR  
7     assign y3 = a ^ b;         // XOR  
8     assign y4 = ~ (a & b);     // NAND  
9     assign y5 = ~ (a | b);     // NOR  
10  
11 endmodule
```

**Nota.** Adaptado de [12]

- **Lógica Secuencial:** un sistema secuencial dispone de elementos de memoria cuyo contenido puede cambiar a lo largo del tiempo. El estado de un sistema secuencial viene dado por el contenido de sus elementos de memoria. La salida en un instante concreto viene dada por la entrada y por el estado anterior del sistema. El estado actual del sistema, junto con la entrada, determinará el estado en el instante siguiente (retroalimentación). Un ejemplo de lógica secuencial son los *flip flops*, como el observado en el Cuadro 2. También se puede evidenciar enunciados como *always*, los cuales ejecu-

tan instrucciones siempre que se cumpla alguna condición especificada, en el ejemplo inferior sería el cambio positivo de la señal de reloj [12].

**Figura 9.** *Ejemplo de código en Verilog de un flip flop*

```
1
2     module flipflop (
3         input logic clk
4         input logic [3:0] d
5         output logic[3:0] q );
6
7         always_ff @ (posedge clk)
8             q <= d;
9     endmodule
10
```

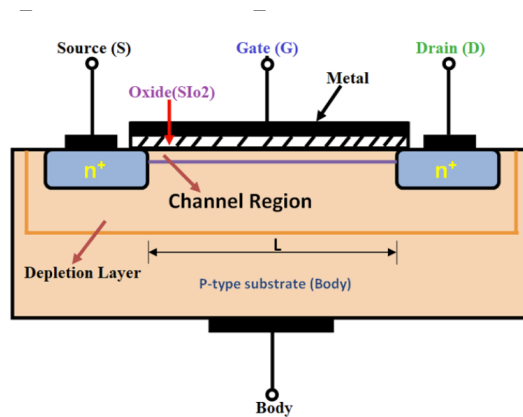
**Nota.** *Adaptado de [12]*

## 6.2. Transistores MOSFET

Un transistor MOSFET (Metal-oxide-semiconductor field-effect transistor): (*Metal-oxide-semiconductor field-effect transistor*) es un tipo de transistor utilizado en circuitos electrónicos para amplificación o conmutación de señales. Su estructura básica consiste en tres regiones: la fuente (source), el drenaje (drain) y la compuerta (gate). Estas regiones están conectadas por un sustrato semiconductor [13].

La estructura interna de un MOSFET típico consiste en una capa de óxido de silicio ( $SiO_2$ ) entre la compuerta y el canal de conducción, que está formado en el sustrato semiconductor. La compuerta está aislada eléctricamente del canal por este óxido. Hay dos tipos principales de MOSFETs: los de canal N y los de canal P, que se diferencian en el tipo de dopaje del sustrato semiconductor, en la Figura 5 se muestra un MOSFET tipo N [14].

**Figura 10.** Composición de un MOSFET tipo N [15]



El funcionamiento de un MOSFET se basa en la modulación del campo eléctrico a través del canal semiconductor inducido por la tensión aplicada a la compuerta. Por ello, existen tres posibles modos de operación dentro de un MOSFET:

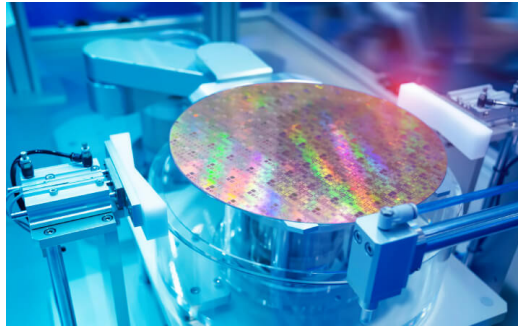
- **Lineal:** en este modo, el MOSFET está parcialmente activado y actúa como un resistor variable. Se produce cuando se aplica una tensión intermedia a la compuerta, lo que resulta en un canal conductor parcial entre el drenaje y la fuente. En este estado, la corriente entre el drenaje y la fuente está controlada por el voltaje aplicado a la compuerta y la resistencia del dispositivo es variable. Este modo se utiliza típicamente en aplicaciones de amplificación de señales, donde se requiere una respuesta lineal a la entrada de señales analógicas [13].
- **Saturación:** en este modo, el MOSFET está completamente activado y actúa como un interruptor cerrado. Se produce cuando se aplica un voltaje adecuado a la compuerta, lo que crea un canal conductor entre el drenaje y la fuente. En este estado, hay un flujo máximo de corriente entre el drenaje y la fuente, y la resistencia entre estos dos terminales es mínima [13].
- **Corte:** en este modo, el MOSFET está apagado y no hay flujo de corriente entre el drenaje y la fuente. Se produce cuando la tensión aplicada a la compuerta no es suficiente para crear un canal conductor entre el drenaje y la fuente. En este estado, el MOSFET actúa como un interruptor abierto y tiene una alta impedancia entre el drenaje y la fuente [13].

### 6.3. Fabricación de Circuitos Integrados

La litografía es una técnica utilizada para crear circuitos integrados depositando finas capas de distintos materiales sobre una oblea de silicio. Estos materiales se eligen cuida-

dosamente en función de sus cualidades eléctricas e incluyen semiconductores como silicio, germanio y arseniuro de galio [16]. La preparación de las obleas, la deposición de las capas, el diseño de los circuitos y las pruebas son sólo algunas de las fases del proceso de fabricación. Los circuitos integrados son los componentes básicos de la electrónica moderna y hacen funcionar desde teléfonos móviles hasta ordenadores y satélites [17].

**Figura 11.** *Fabricacion de circuitos integrados [16]*



Los circuitos integrados se fabrican mediante un proceso complejo y preciso de creación de componentes electrónicos y sus interconexiones sobre un fino sustrato de silicio. Estos son algunos de los pasos necesarios para desarrollar un circuito integrado [18]:

- **Preparación de Obleas:** consiste en cortar, dar forma y pulir una fina lámina de silicio monocristalino denominada oblea. Los cristales de silicio utilizados en la fabricación de obleas son muy puros y se obtienen a partir de una masa fundida de silicio en forma cilíndrica denominada boule [19].

**Figura 12.** *Boule [20]*

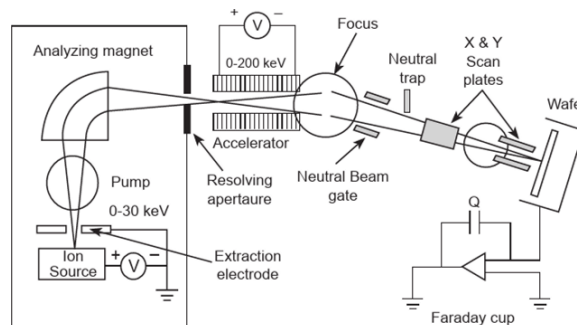


- **Oxidación:** el proceso de oxidación puede describirse como un proceso destinado a aportar oxígeno a la oblea de silicio para generar una capa de óxido de silicio en la superficie de los circuitos integrados. El proceso puede tener lugar en hornos a temperaturas muy elevadas. La oxidación puede ser húmeda o seca, dependiendo de si

se utiliza vapor de agua u oxígeno gaseoso como agente oxidante. La oxidación húmeda es más rápida, mientras que la seca tiene mejores propiedades eléctricas. La capa de  $SiO_2$  actúa como aislante y como máscara para las fases posteriores de procesamiento [21].

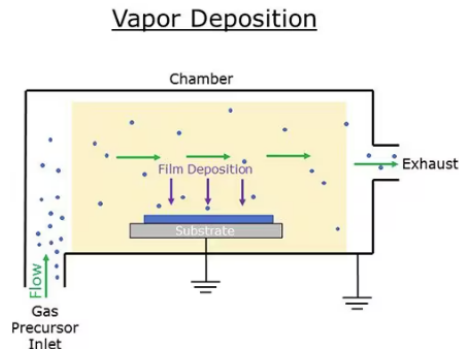
- Difusión:** la difusión es el proceso de introducir impurezas en zonas seleccionadas de la oblea de silicio para modificar sus características eléctricas. Las impurezas se denominan dopantes y pueden ser de tipo n o de tipo p, según donen o acepten electrones en la red de silicio. La difusión se realiza calentando la oblea en un horno con una fuente de gas dopante, como la fosfina ( $PH_3$ ) para el tipo n o el trifluoruro de boro ( $BF_3$ ) para el tipo p. Los átomos dopantes se difunden en el interior de la oblea. Los átomos dopantes se difunden en el silicio a través de aberturas en la capa de  $SiO_2$  creadas por fotolitografía [22].
- Implantación Iónica:** la implantación iónica es un método alternativo para dopar la oblea de silicio bombardeándola con un haz de iones de alta energía del elemento dopante deseado. Los iones penetran en el silicio y crean regiones de material tipo n o tipo p. La implantación de iones controla la concentración y distribución de dopantes con más precisión que la difusión. Sin embargo, también provoca más daños en la estructura cristalina del silicio que deben repararse mediante recocido [21].

**Figura 13.** *Proceso de implantación iónica*



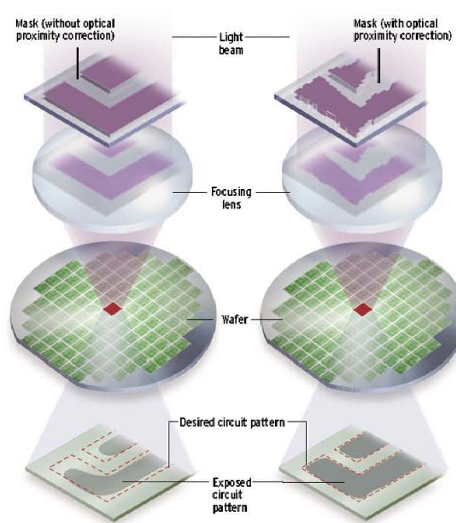
- Deposición Química de Vapor (CVD):** el método de formación de películas finas de diversos materiales sobre una oblea de silicio exponiéndola a un gas que contiene el elemento o compuesto deseado se conoce como deposición química en fase vapor (CVD). El gas reacciona o se descompone en la superficie de la oblea, generando una capa sólida. La CVD puede depositar materiales como polisilicio, nitruro de silicio ( $Si_3N_4$ ), siliciuros metálicos (como el siliciuro de wolframio,  $WSi_2$ ) y metales (como el aluminio,  $Al$ ).

Figura 14. Proceso de deposición química de vapor[23]



- **Fotolitografía:** la fotolitografía es el proceso de transferir un patrón de una máscara a la superficie de una oblea utilizando materiales sensibles a la luz llamados fotorresistentes. El material fotorresistente se recubre sobre la oblea y luego se expone a la luz ultravioleta a través de una máscara que contiene el patrón deseado. La máscara puede ser positiva o negativa, dependiendo de si bloquea o deja pasar la luz a través de sus aberturas. Las zonas expuestas de la fotorresistencia se vuelven solubles o insolubles en una solución reveladora, dependiendo de si se trata de una fotorresistente positivo o negativo. El fotorresistente desarrollado forma una capa estampada que puede utilizarse para otros pasos de procesamiento, como el grabado, el dopaje o la metalización [24].

Figura 15. Fotolitografía [24]

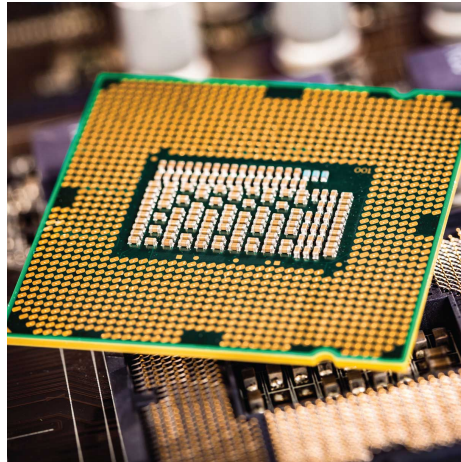


- **Metalización:** la metalización es el proceso de formación de interconexiones metálicas entre diferentes regiones o capas de un circuito integrado. El metal suele ser aluminio o cobre y se deposita en la superficie mediante técnicas de CVD o evaporación. A continuación, la capa metálica se modela mediante fotolitografía y grabado para crear

cables y contactos que conecten los distintos componentes y terminales del circuito integrado.

- **Embalaje:** consiste en encerrar el chip de CI acabado en una carcasa protectora que proporciona soporte mecánico, conexiones eléctricas y protección medioambiental. El embalaje puede ser de plástico o de cerámica y tener diferentes formas y tamaños en función de la aplicación y los requisitos de rendimiento del CI. El embalaje también contiene pines o pistas (leads) que conectan el chip de CI a circuitos o dispositivos externos [21].

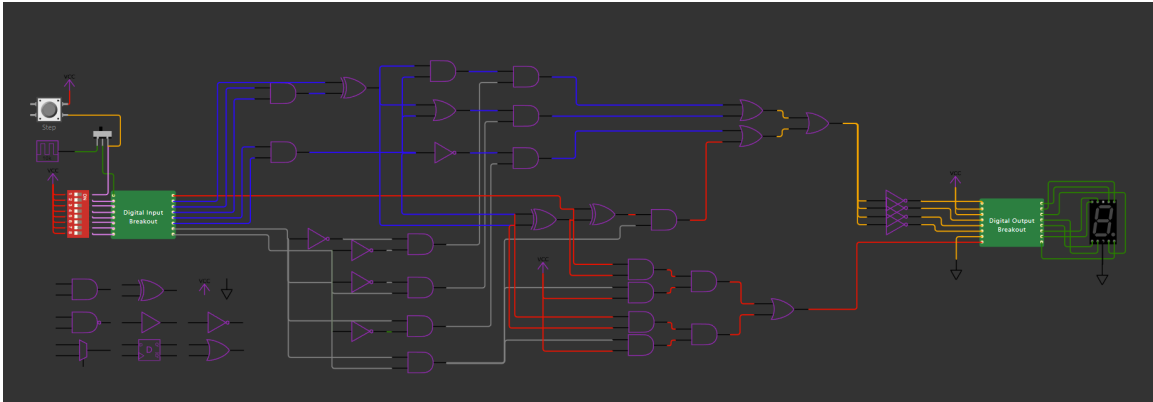
**Figura 16.** *Chip empaquetado [21]*



## 6.4. Uso de Wokwi para el desarrollo de proyectos de ASIC

Wokwi es un simulador de electrónica en línea. Wokwi compila el código en un *firmware* binario y luego este se ejecuta una instrucción a la vez, como lo haría un microcontrolador real. Contiene un editor de circuitos interactivo que proporciona una alternativa conveniente para editar el archivo diagrama *.json* directamente. Esta herramienta permite desarrollar proyectos también con lenguaje de programación C/C++ y cualquier lenguaje que compile a *WebAssembly* (*Rust*, *AssemblyScript*, etc.) [25].

**Figura 17.** Interfaz gráfica de Wokwi con proyecto ALU de 1 bit [26]

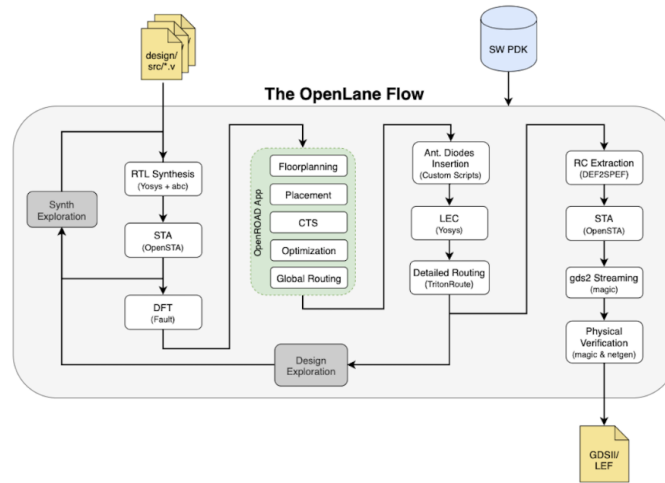


Para el caso de desarrollo de ASICs en el proyecto de Tiny TapeOut se ha aprovechado de las utilidades del software posteriormente discutido OpenLane, el cual permite obtener los documentos GDS2 necesarios para la manufactura de los chips. Un ejemplo claro de su uso se observa en la Figura 6, donde se muestra el diseño de una ALU (Unidad Aritmética Lógica) de 1 bit, encontrado en el primer chip TT01. Es importante recalcar que a partir de simulaciones interactivas es posible emitir diseños al proyecto Tiny TapeOut [25].

## 6.5. Uso de OpenLane para el desarrollo de proyectos de ASIC

*OpenLane*: es una herramienta open-source para el desarrollo de ASICs que permite convertir un archivo en lenguaje de descripción de hardware (HDL por sus siglas en inglés) a los archivos GDS2: necesarios para la manufactura de un ASIC. La Figura 13, muestra el flujo de funcionamiento de esta herramienta al recibir los documentos en HDL hasta el resultado final de recibir los archivos GDS2. Los pasos fundamentales que ejecuta OpenLane es el proceso de síntesis, *floorplanning* (área a utilizar para para poder distribuir el diseño en las celdas), colocación y ruteo (proceso automatizado para la conversión de los archivos a GDS2). Además, es importante mencionar que realiza un *Static Timing Analysis* (STA) y el chequeo de reglas en el diseño (DRC) [27].

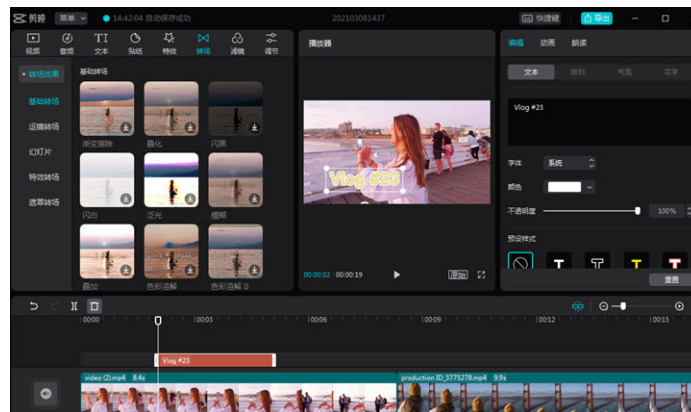
**Figura 18.** Diagrama de flujo funcionamiento OpenLane [27]



## 6.6. Uso de Capcut para el desarrollo de material académico

CapCut es una aplicación de edición de video desarrollada por *ByteDance*. Es conocida por su interfaz intuitiva y sus numerosas funciones, lo que la convierte en una herramienta popular tanto para principiantes como para editores de video más avanzados. Capcut es una herramienta versátil que puede ser muy útil en el contexto educativo, permitiendo a los educadores crear contenido atractivo y efectivo para sus estudiantes. Su facilidad de uso y amplias funciones hacen que sea una opción excelente para mejorar el aprendizaje mediante el uso de recursos multimedia.

**Figura 19.** Interfaz de aplicación de escritorio Capcut



Permite realizar ediciones básicas como cortar, dividir y ajustar la velocidad del video, además de incluir filtros, transiciones y efectos especiales. Ofrece herramientas para agregar texto y subtítulos con diversos estilos y animaciones, y facilita la incorporación de música,

efectos de sonido y narraciones de voz. También cuenta con herramientas de corrección de color y autocolores, así como plantillas predefinidas y una biblioteca de recursos como stickers y clips de video. Estas características la convierten en una herramienta versátil para aplicar en los tutoriales a elaborar dentro del proyecto.

---

## Emisión de diseño electrónico a proyecto Tiny Tapeout

---

En este proyecto, inicialmente se buscó el desarrollo de un diseño electrónico para ser emitido al proyecto de open-source Tiny Tapeout. Para ello, se optó por desarrollar un proyecto en el lenguaje de descripción de hardware Verilog, específicamente, una *Arithmetic logic unit* (ALU) de 4 bits.

### 7.1. Desarrollo de proyecto ALU de 4 bits en Verilog

#### 7.1.1. Acerca del diseño

El diseño de la ALU que se decidió realizar es de 4 bits con las operaciones de suma, resta, multiplicación, división y 4 operaciones lógicas: AND, OR, NOT y XOR. El diseño cuenta con prevención de divisiones por cero para evitar errores no deseados. El proyecto cuenta con 2 entradas de 4 bits para los operandos, 1 entrada de 3 bits para indicar la operación deseada a realizar, 1 salida de 8 bits para el resultado de las operaciones y 2 salidas de un bit que funcionan como indicadores de cero y *carry out* en caso sucedan. El código en Verilog del diseño y su respectivo *testbench* realizado se puede observar en los Anexos, donde se encuentra comentado el funcionamiento para mejor comprensión.

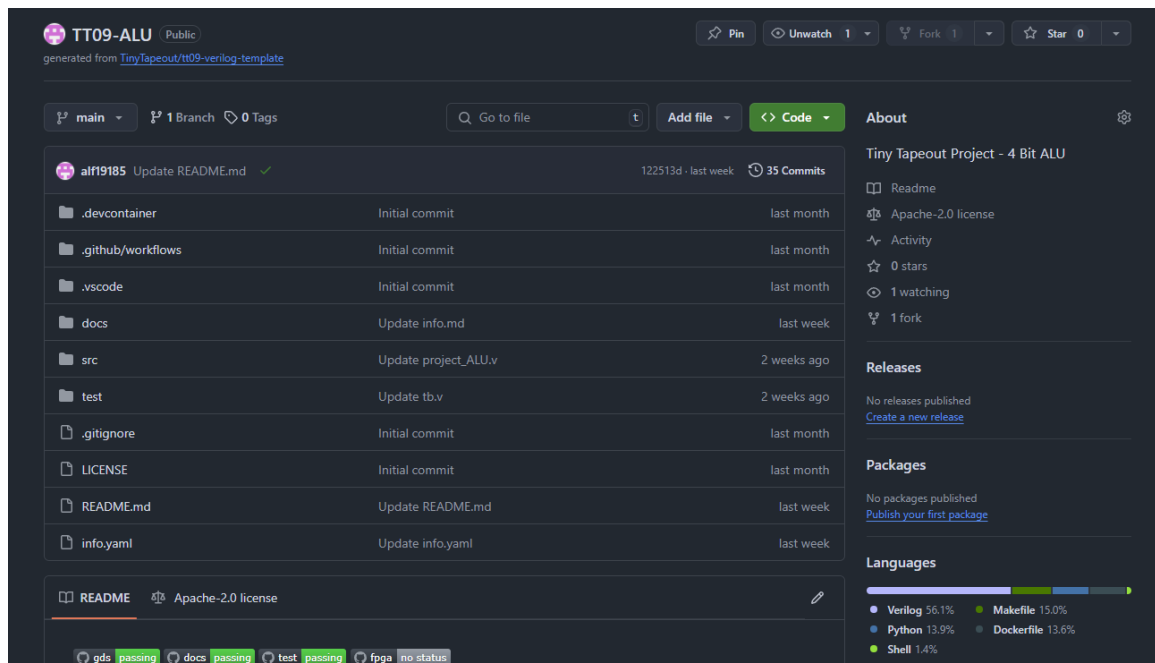
### 7.1.2. Restricciones del diseño

Como diseño a emitir dentro del proyecto de Tiny Tapeout se decidió realizar una ALU de 4 bits. Detrás de la elección de este diseño, se encuentran ciertas restricciones presentadas por Tiny Tapeout que todo diseño debe cumplir. Entre ellas están los límites en las entradas y salidas a utilizar, las cuales se limitan a 8 pines de entradas, 8 pines de salidas y 8 pines bidireccionales I/O. Además, se encuentra la restricción de la frecuencia de reloj, con un valor mínimo de 50MHz. Finalmente, se cuenta con una restricción de una celda de 160x100 micrómetros, la cual es suficiente para aproximadamente unas 1000 compuertas lógicas digitales dependiendo del tamaño. Es importante recalcar que se puede pagar un monto extra para añadir más celdas, sin embargo, se busca aprovechar y optimizar el diseño para evitar más gastos al momento de realizar el proyecto.

## 7.2. Proceso de emisión del diseño de una ALU a Tiny Tapeout

Para poder enviar el diseño electrónico de una ALU sencilla en Verilog, se siguieron una serie de pasos para realizar los procesos síntesis, verificación de reglas de diseño (DRC), ruteo, colocación de cada elementos, hasta obtener los archivos GDSII necesarios para manufactura. Estos pasos se encuentran integrados en OpenLane el cual se ejecuta dentro de Github Actions, al crear un repositorio en el sitio web de Github a partir de una plantilla que provee Tiny Tapeout. A continuación se muestran los pasos que se siguieron para poder enviar el diseño electrónico.

Figura 20. Repositorio ALU 4 bits en Github



### 7.2.1. Creación de repositorio y accesos a Github Actions

Para poder utilizar la herramienta de OpenLane es necesario realizar un repositorio en Github por proyecto, derivado de una plantilla que se encuentra dada por Tiny Tapeout. Esta plantilla cuenta con una serie de archivos específicos y necesarios para diseños que se realizan a partir de cualquier variedad de HDL, en este caso Verilog.

Para permitir que OpenLane logre acceder a los documentos y realice los pasos de síntesis, verificación de reglas de diseño (DRC), ruteo y colocación de elementos, es necesario habilitar Github Actions en el repositorio del proyecto. Para ello se debe acceder a los ajustes y dirigirse a la sección de páginas, donde se habilita que la fuente de ejecución de flujos sea realizado por medio Github Actions en vez del uso de ramas con el que son creados por defecto los repositorios.

#### 7.2.1.1. *Modificaciones en los archivos del repositorio*

Cuando se cuenta con el diseño en Verilog, es necesario adaptarlo a la plantilla que provee el sitio de *Tiny Tapeout* en su sitio web. Primero se debe editar el archivo `info.yaml` y actualizar la información sobre el proyecto, prestando atención a las propiedades `source_files` y `top_module`.

Figura 21. Archivo realizado para ALU 4 bits en TT09

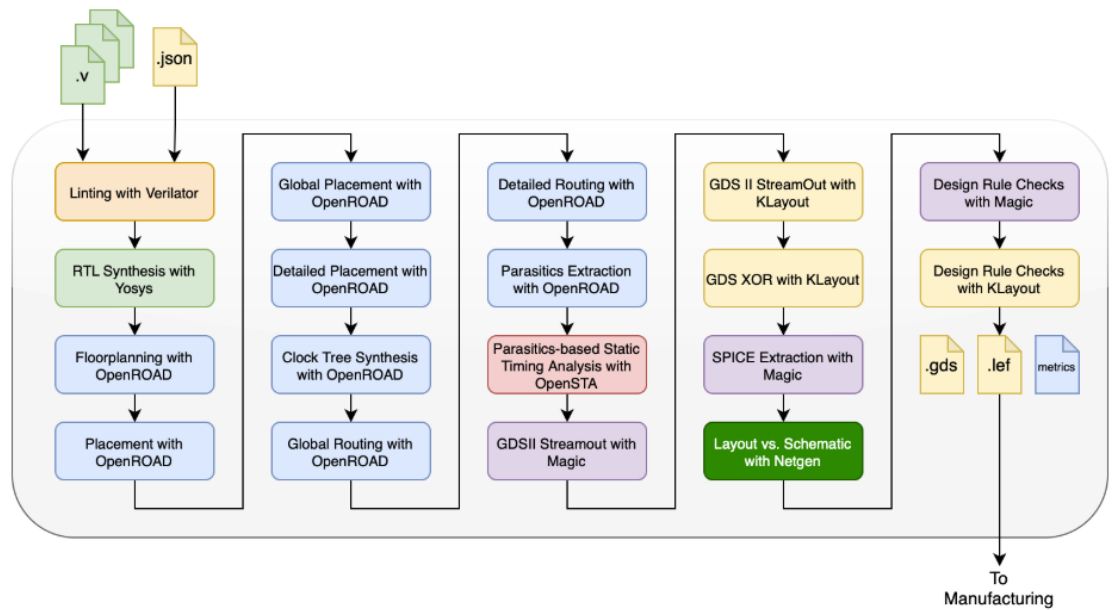
```
1      # Tiny Tapeout project information
2      project:
3      title:      "4 bit ALU "
4      author:     "Gabriela Alfaro"
5      discord:    "N/A"
6      description: "A simple design of an Arithmetic Logic Unit
7                  capable of basic operations: addition, subtraction, multiplication,
8                  division and some logic operations."
9      language:   "Verilog"
10
11     tiles: "1x1"
12     top_module: "tt_um_alf19185_ALU"
13
14     source_files:
15         - "project_ALU.v"
16
17     # Inputs
18     ui[0]: "A[0]"
19     ui[1]: "A[1]"
20     ui[2]: "A[2]"
21     ui[3]: "A[3]"
22     ui[4]: "B[0]"
23     ui[5]: "B[1]"
24     ui[6]: "B[2]"
25     ui[7]: "B[3]"
26
27     # Outputs
28     uo[0]: "ALU_Out[0]"
29     uo[1]: "ALU_Out[1]"
30     uo[2]: "ALU_Out[2]"
31     uo[3]: "ALU_Out[3]"
32     uo[4]: "ALU_Out[4]"
33     uo[5]: "ALU_Out[5]"
34     uo[6]: "ALU_Out[6]"
35     uo[7]: "ALU_Out[7]"
36
37     # Bidirectional pins
38     uio[0]: "ZeroFlag"
39     uio[1]: "CarryOut"
40     uio[2]: ""
41     uio[3]: ""
42     uio[4]: ""
43     uio[5]: ""
44     uio[6]: ""
45     uio[7]: ""
46
47     yaml_version: 6
48
49
50
```

Nota. Adaptación propia

### 7.2.1.2. Ejecución de OpenLane mediante Github Actions

Al adaptarse el programa de la ALU de forma adecuada al formato, se ejecutan las *Github Actions* permitiendo que *OpenLane* realice los procesos de síntesis, *floorplanning*, colocación de elementos, ruteo, DRC, entre otros. Al finalizar este proceso de forma correcta, se logra obtener una verificación simulada del funcionamiento del diseño elaborado, los archivos de manufactura GDSII y LEF, junto a imágenes visuales del render final, la hoja de datos del proyecto y estadísticas del diseño. A continuación en la Figura 16 se muestran con detalle los procesos que ejecuta OpenLane y las herramientas que utiliza para realizar cada uno de los pasos mencionados anteriormente.

**Figura 22.** Flujo de trabajo seguido por OpenLane



*OpenLane* es un flujo automatizado de diseño de circuitos integrados que convierte una descripción de hardware en RTL (Register Transfer Level) en un archivo *GDSII* listo para fabricación. Este proceso se divide en varias etapas clave, cada una utilizando herramientas específicas para garantizar un diseño óptimo y funcional. A continuación, se presenta un resumen de cada etapa:

- **Síntesis:** Se transforma el código RTL en una netlist de puertas lógicas. Herramientas como *Yosys* y *ABC* realizan la síntesis y mapeo tecnológico, mientras que *OpenSTA* lleva a cabo análisis de temporización estática para generar informes de temporización.
- **Floorplanning:** Se define el área del núcleo del chip, las filas para la colocación de celdas y las pistas para el ruteo. Las herramientas *Init\_fp*, *ioplacer*, *pdngen* y *tapcell* se encargan de estas tareas, incluyendo la colocación de puertos de entrada y salida,

la generación de la red de distribución de energía y la inserción de celdas de tap y decoupling.

■ **Colocación:**

- *Colocación Global: RePLace* realiza la colocación inicial de las celdas en el diseño.
- *Optimización: Resizer* lleva a cabo optimizaciones opcionales en el diseño.
- *Colocación Detallada: OpenDP* legaliza la colocación global, asegurando que las celdas estén correctamente posicionadas.

■ **Clock Tree Synthesis - CTS:** *TritonCTS* sintetiza la red de distribución del reloj, garantizando una distribución equilibrada y eficiente del mismo.

■ **Ruteo:**

- *Ruteo Global: FastRoute* genera una guía para el ruteo detallado.
- *Ruteo Detallado: TritonRoute* realiza el ruteo preciso de las conexiones.
- *Extracción de Parámetros Eléctricos: OpenRCX* extrae los parámetros eléctricos del diseño para análisis posteriores.

■ **Generación de GDSII (*Tapeout*):** *Magic* y *KLayout* generan el archivo *GDSII* final, que es el formato estándar para la fabricación de circuitos integrados.

■ **Verificación Final:**

- *Verificación de Reglas de Diseño (DRC): Magic* y *KLayout* verifican que el diseño cumpla con las reglas de fabricación.
- *Verificación de Conectividad (LVS): Netgen* asegura que la netlist coincide con el diseño físico.
- *Verificación de Antena: Magic* verifica posibles problemas de antena en el diseño.
- *Verificación de Validez de Circuito: CVC* realiza comprobaciones adicionales para garantizar la funcionalidad del circuito.

### 7.2.2. Caracterización del diseño electrónico final

El diseño descrito en las secciones anteriores, se envía a fabricar a Efabless utilizando una tecnología de 130nm en el chip TT09 de Tiny Tapeout el día 11 de noviembre del 2024. A continuación, se detallan las estadísticas clave y las características del diseño obtenido tras el proceso que sigue *OpenLane*. Para más información del diseño en el chip, también se puede acceder al envío oficial en *Tiny Tapeout* aquí.

## Recursos Utilizados

A continuación se muestra en el Cuadro 3, el porcentaje de utilización de la celda y el largo de cable necesario para el diseño de la ALU de 4 bits. Es importante mencionar que estas estadísticas se obtienen sin considerar las celdas necesarias para alimentación tipo *Tap* ni aquellas tipo *Fill*. Estas últimas son celdas de capacitancia de desacople que se colocan como relleno luego del proceso de colocación y ruteo, por lo que no son consideradas dentro de las estadísticas.

**Cuadro 1.** Recursos utilizados en el diseño de la ALU de 4 bits

Porcentaje de utilización	Largo del cable utilizado
9.715 %	4199 $\mu\text{m}$

**Cuadro 2.** Compuertas lógicas utilizadas en el diseño de la ALU de 4 bits

Categoría de Celda	Cantidad
AND	13
NOT	8
MISC	18
BUFFER	21
OR	25
NAND	27
NOR	27
Tap (Alimentación)	225
Combo Logic	68
Fill	1444
<b>Total (Excluyendo Fill y Tap)</b>	<b>207</b>

Figura 23. Render GDS del diseño obtenido en OpenLane

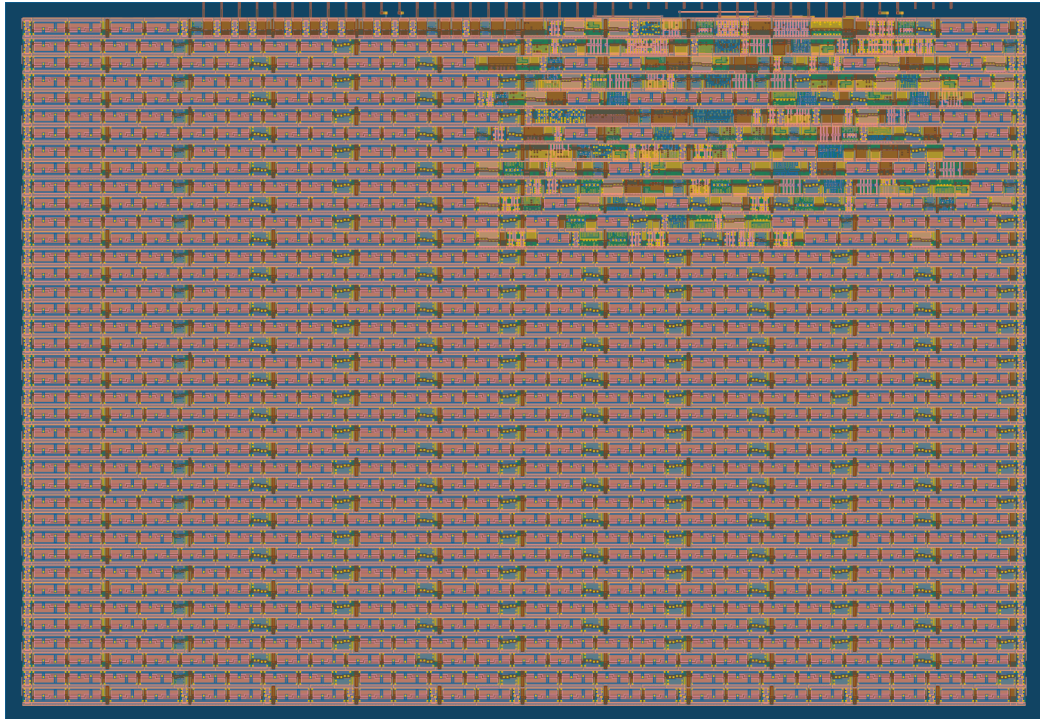


Figura 24. Render GDS del diseño oficial del chip TT09

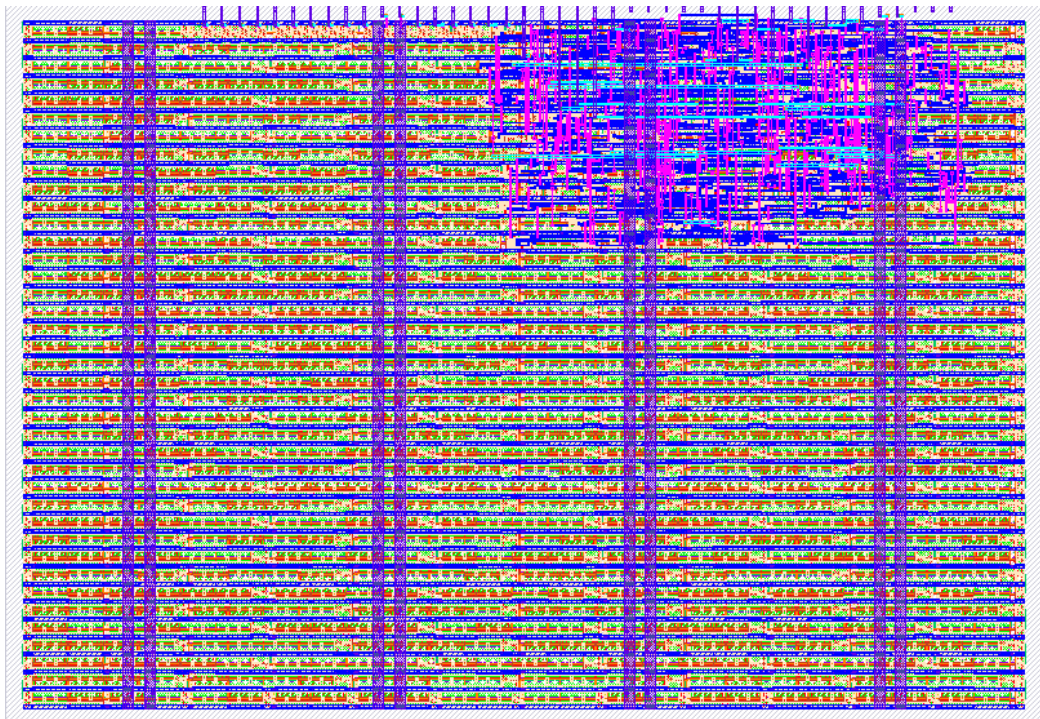


Figura 25. Visualización 3D del diseño de la ALU de 4 bits

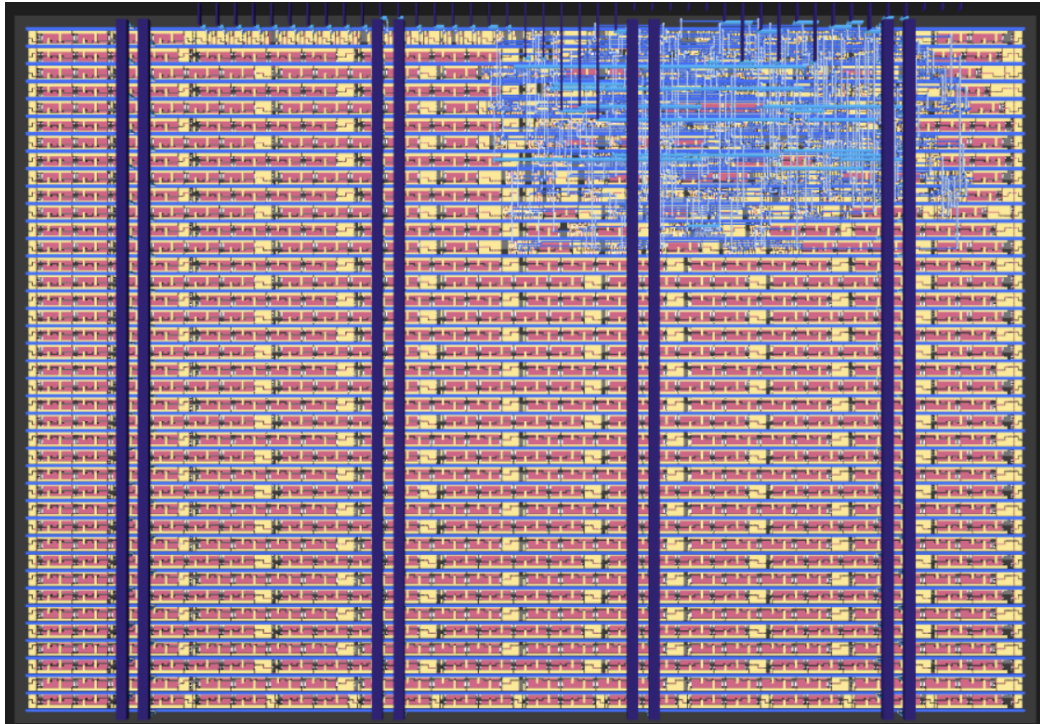
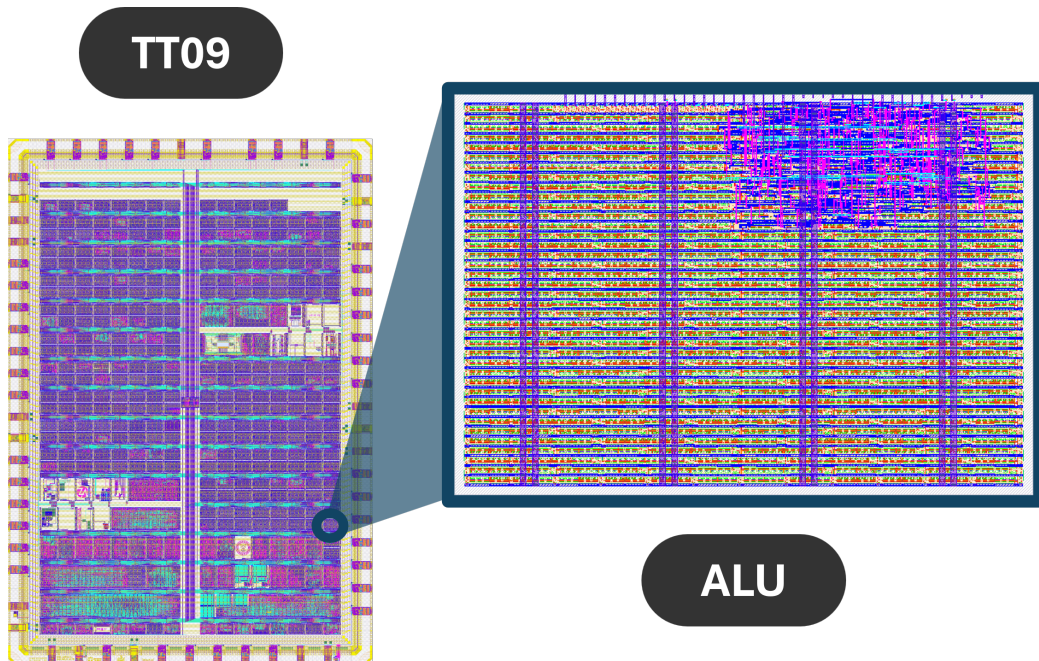


Figura 26. Render del chip TT09 y ubicación de ALU diseñada



## 8.1. Guía para emisión de proyectos para Tiny Tapeout

El presente capítulo detalla la elaboración de material académico dirigido a facilitar el proceso de emisión de proyectos de diseño de circuitos digitales en el entorno de Tiny Tapeout. Con el objetivo de mejorar la accesibilidad al diseño de ASICs para estudiantes y profesionales, se han desarrollado guías que explican, paso a paso, los métodos de emisión mediante dos enfoques principales: el uso del simulador gráfico Wokwi para las guías de Electrónica Digital y el diseño de circuitos mediante HDL (Verilog) para las guías de Arquitectura de Computadoras. Además, se ofrece en ambas un entendimiento sobre la integración de estos diseños con el software OpenLane para la síntesis y generación de archivos necesarios para la manufactura.

### 8.1.1. Modos de emisión de proyectos: Simulador gráfico Wokwi ó HDL

#### 8.1.1.1. *Simulador gráfico Wokwi*

Las primeras guías están orientadas al método de emisión mediante Wokwi, una herramienta que permite a los usuarios diseñar y simular circuitos digitales a través de una interfaz gráfica intuitiva. Esta guía ha sido elaborada pensando en los estudiantes que inician en el ámbito del diseño digital y prefieren un enfoque visual como aquellos que se encuentran en el curso de Electrónica Digital. En ellas, se describe el proceso completo para crear circuitos

en Wokwi, obtener el ID del proyecto y configurar los archivos necesarios para la emisión en Tiny Tapeout. La explicación incluye capturas de pantalla y descripciones detalladas de cada paso, permitiendo que los usuarios sigan el proceso de manera autónoma y logren emitir sus primeros diseños en un entorno profesional. Para comprender con mayor detalle, se pueden chequear las guías elaboradas para el curso de Electrónica Digital en la sección de Anexos.

#### *8.1.1.2. HDL*

La segunda sección de guías se centra en la emisión de diseños mediante HDL, específicamente en Verilog, y está dirigida a estudiantes y profesionales que poseen un conocimiento intermedio o avanzado en lenguajes de descripción de hardware, como aquellos que se encuentran en el curso de Arquitectura de Computadoras. En estas guías, se detalla la configuración de un proyecto en GitHub, y las modificaciones necesarias dentro de los archivos del proyecto para obtener correctamente los archivos de manufactura, GDSII.

Estas permiten a los estudiantes a familiarizarse con el flujo de trabajo de OpenLane, lo cual resulta crucial para quienes buscan dominar las herramientas de diseño digital y optimizar sus proyectos. Estas guías muestran el proceso descrito a en el capítulo anterior, adaptado para el entorno académico y permitiendo a los estudiantes a desarrollar y comprender con claridad el proceso de diseños electrónicos elaborados en código de descripción de hardware. Para comprender con mayor detalle, se pueden chequear las guías elaboradas para el curso de Arquitectura de Computadoras en la sección de Anexos 12.3.2.

#### **8.1.2. Uso de software OpenLane**

Se incluye tanto en las guías de Electrónica Digital como en las de Arquitectura de computadoras, acerca del uso de OpenLane mediante Github Actions, que abarca desde la verificación inicial de reglas de diseño (DRC) hasta la generación del archivo GDSII, necesario para la manufactura del ASIC. Esta sección describe cada paso de manera práctica y técnica, incluyendo los pasos a tomar y los flujos de trabajo que permiten sintetizar y optimizar el diseño para la manufactura. Las guías incluyen el como OpenLane representa un recurso fundamental para comprender cómo los diseños digitales pueden ser preparados para la fabricación en una plataforma accesible como Tiny Tapeout, constituyendo una experiencia educativa valiosa en el campo de la ingeniería electrónica.

## 8.2. Material Académico para curso de Nanoelectrónica

Las guías y los videos desarrollados para el curso de Nanoelectrónica se encuentran orientados a la comprensión de los fundamentos de los componentes a un nivel nanoelectrónico y su implementación a nivel físico de semiconductores. Mediante ellas se busca que los estudiantes aprendan sobre el comportamiento de los MOSFETs en escalas pequeñas y cómo su estructura afecta el funcionamiento de compuertas lógicas en diseño de ASICs. Estos videos también muestran cómo utilizar las herramientas que ofrece Tiny Tapeout como lo es Siliwiz, Stixu y un visualizador de diseños en 3D, proporcionando una experiencia interactiva que permite a los estudiantes explorar el diseño físico de sus circuitos y comprender con facilidad estos temas. Para observar con mayor detalle este material, se pueden chequear las guías elaboradas para el curso de Nanoelectrónica en la sección de Anexos 12.3.3 .

---

### Conclusiones

---

- Se logró implementar un flujo de trabajo efectivo para el diseño y la manufactura de circuitos integrados mediante el uso de herramientas de Tiny Tapeout.
- Se diseñaron y simulon circuitos de lógica combinacional dentro de las restricciones de una celda de 160x100 micrómetros, accediendo a la manufactura real de un chip a través de una plataforma de bajo costo.
- Las guías elaboradas permiten que tanto estudiantes como profesionales accedan a procesos de manufactura de semiconductores sin necesidad de costosas herramientas o infraestructura especializada.
- Se crearon guías académicas paso a paso, detallando el proceso desde la simulación en Wokwi y la descripción en HDL (Verilog) hasta la emisión del diseño a Tiny Tapeout.
- Los materiales académicos desarrollados constituyen un recurso valioso para facilitar la comprensión del flujo de diseño, promoviendo el aprendizaje práctico sobre la construcción de circuitos lógicos y su implementación en chips físicos.
- Se logró la emisión exitosa del diseño HDL dentro de Tiny Tapeout, utilizando los flujos de trabajo automáticos de OpenLane para verificar el diseño y generar archivos GDSII listos para la manufactura en fábricas de semiconductores.
- Se documentó el proceso de verificación y optimización mediante OpenLane, lo que demostró que los circuitos diseñados pueden ser manufacturados cumpliendo con los estándares de la industria.
- Las simulaciones permitieron identificar y corregir posibles errores en el diseño antes de enviarlo a manufactura, demostrando la efectividad del flujo de trabajo y reduciendo

el riesgo de fallos en la etapa de fabricación.

- Se elaboraron guías que permiten desarrollar competencias clave en el diseño de circuitos integrados, incluyendo el uso de lenguajes de descripción de hardware (Verilog) y la comprensión de conceptos de nanoelectrónica.

---

## Recomendaciones

---

En primer lugar, sería recomendable profundizar en la creación de proyectos más complejos que integren una mayor cantidad de componentes lógicos, siempre ajustándose a las limitaciones físicas de la celda de 160x100 micrómetros. Esto podría incluir el diseño de módulos combinatoriales y secuenciales más avanzados que puedan llevarse a manufactura, explorando cómo optimizar el espacio disponible para diseños más sofisticados.

Sería beneficioso extender la documentación y guías prácticas. Aunque se ha generado un material educativo útil, podría ampliarse para cubrir más áreas del diseño nanoelectrónico, como la optimización de temporización y consumo energético en los circuitos al contar con el chip físico. Las guías podrían incluir ejemplos más avanzados, como el uso de técnicas de minimización de área y rutas de señal, lo que ofrecería a los estudiantes y profesionales una comprensión más profunda del diseño eficiente de ASIC.

Un área clave de mejora sería explorar las limitaciones de las plataformas open-source en términos de fabricación de circuitos integrados más complejos y la posibilidad de superar estos desafíos. Esto incluiría investigar formas de escalar los proyectos a diseños de mayor complejidad o encontrar maneras de optimizar el uso del espacio físico disponible. También, sería interesante evaluar cómo otras plataformas de simulación y síntesis se comparan con las herramientas open-source y si podrían complementarse en proyectos futuros.

Finalmente, se sugiere explorar otras herramientas open-source más allá de Tiny apeout para diversificar las aplicaciones del flujo de diseño hacia una línea de investigación avanzada. Herramientas como Magic VLSI, Qflow, KLayout y la combinación de Yosys con OpenROAD ofrecen funcionalidades más amplias para la personalización de layouts, síntesis lógica y optimización de diseño físico. Estas plataformas permitirían investigar metodologías más

complejas de diseño y manufactura de circuitos integrados, impulsando así la innovación y contribuyendo al desarrollo de nuevos enfoques en el diseño de semiconductores.

- [1] T. Tapeout, *From idea to design in minutes!* Dirección: <https://tinytapeout.com/>.
- [2] T. Tapeout, *Tiny Tapeout Chips*. dirección: <https://tinytapeout.com/runs/>.
- [3] T. Tapeout, *Tiny Tapeout 1*. dirección: <https://tinytapeout.com/runs/tt01/>.
- [4] T. Tapeout, *Tiny Tapeout 2*. dirección: <https://tinytapeout.com/runs/tt02/>.
- [5] T. Tapeout, *Tiny Tapeout 3*. dirección: <https://tinytapeout.com/runs/tt03/>.
- [6] T. Tapeout, *Tiny Tapeout 4*. dirección: <https://tinytapeout.com/runs/tt04/>.
- [7] T. Tapeout, *Tiny Tapeout 5*. dirección: <https://tinytapeout.com/runs/tt05/>.
- [8] T. Tapeout, *Tiny Tapeout 6*. dirección: <https://tinytapeout.com/runs/tt06/>.
- [9] T. Tapeout, *Tiny Tapeout 7*. dirección: <https://tinytapeout.com/runs/tt07/>.
- [10] W.-L. Chin, “Principles of Verilog Digital Design,” CRC Press, 2022.
- [11] D. H. Sarah Harris, “Digital Design and Computer Architecture,” Elsevier Science, 2021.
- [12] J. L. H. David A. Patterson, “Computer Organization and Design MIPS Edition: The Hardware/Software Interface,” Elsevier Science, 2013.
- [13] B. Razavi, “Fundamentals of Microelectronics,” University of California, Los Angeles, 2016.
- [14] Z. to ASIC, *MOSFET*. dirección: <https://www.zerotoasiccourse.com/terminology/mosfet/>.
- [15] C. 101, *Fundamentals of MOSFETs and Their Working*, T. M. Telegraph, ed. dirección: <https://components101.com/articles/mosfet-symbol-working-operation-types-and-applications>.

- [16] A. W. Services, *¿Qué es un Chip de Computadora?* A. W. Services, ed. dirección: <https://aws.amazon.com/es/what-is/computer-chip/>.
- [17] U. de los Andes, *¿Cómo Fabricar Chips?* Dirección: <https://electricayelectronica.uniandes.edu.co/es/preg%C3%BAntale-a-un-ingeniero-iele-como-fabricar-chip>.
- [18] E. Corporation, *Chip Design, Fabrication and Bring-up for Product Companies*, E. Corporation, ed. dirección: <https://efabless.com/chipignite>.
- [19] Z. to ASIC, *Wafer*. dirección: <https://www.zerotoasiccourse.com/terminology/wafer/>.
- [20] PAM-XIAMEN, *Silicon Ingot*. dirección: <https://www.powerwaywafer.com/it/silicon-ingot.html>.
- [21] A. LI, *Semiconductor Manufacturing Process Steps*. dirección: <https://www.asml.com/en/news/stories/2021/semiconductor-manufacturing-process-steps>.
- [22] Z. to ASIC, *Wafer*. dirección: <https://www.zerotoasiccourse.com/terminology/doping/>.
- [23] Merck, *Solution and Vapor Deposition Precursors*. dirección: <https://www.sigmaaldrich.com/IT/it/products/materials-science/energy-materials/solution-and-vapor-deposition-precursors>.
- [24] Z. to ASIC, *Photolithography*. dirección: <https://www.zerotoasiccourse.com/terminology/photolithography/>.
- [25] C. LTD, *Wokwi*. dirección: <https://docs.wokwi.com/>.
- [26] A. S. Tanenbaum, "Structured Computer Organization," Pearson, 2013.
- [27] T. Edwards y K. Farid, *OpenLane*. dirección: <https://openlane.readthedocs.io/en/latest/index.html>.

## 12.1. Código de Verilog a emitir para proyecto *Tiny Tapeout*

Figura 27. Código de ALU 4 bits a enviar

```

1
2     'default_nettype none
3
4 module tt_um_alf19185_ALU (
5     input wire [7:0] ui_in,    // Dedicated inputs
6     output wire [7:0] uo_out,  // Dedicated outputs
7     input wire [7:0] uio_in,   // IOs: Input path
8     output wire [7:0] uio_out, // IOs: Output path
9     output wire [7:0] uio_oe,  // IOs: Enable path (active high: 0=input, 1=output)
10    input wire    ena,         // Always 1 when the design is powered, so you can
    ignore it
11    input wire    clk,        // Clock (not used for combinational ALU)
12    input wire    rst_n      // Reset (not used for combinational ALU) );
13
14    // Map the ALU inputs and outputs to the provided ports
15    wire [3:0] A = ui_in[3:0]; // Lower 4 bits of ui_in are A
16    wire [3:0] B = ui_in[7:4]; // Upper 4 bits of ui_in are B
17    wire [2:0] Opcode = uio_in[2:0]; // Use lower 3 bits of uio_in for Opcode
18    reg [7:0] ALU_Result;
19    reg Carry;
20    reg Zero;
21
22    // Assign ALU outputs to uo_out
23    assign uo_out[7:0] = {Zero, Carry, ALU_Result[5:0]}; // Concatenate Zero, Carry,
    and the lower 6 bits of ALU_Result
24    // Set unused outputs and IOs to 0
25    assign uio_out = 8'b0;
26    assign uio_oe = 8'b0;
27    // ALU Logic
28    always @(*) begin
29        // Default flags
30        Carry = 1'b0;
31        Zero = 1'b0;
32        ALU_Result = 8'b0;
33        case (Opcode)
34            3'b000: begin // Addition

```

Figura 28. Testbench de ALU 4 bits a enviar a TT09

```
1
2     'default_nettype none
3     'timescale 1ns / 1ps
4     module tb ();
5         // Declare continuous signals for power and ground
6         wire VPWR = 1'b1; // Power supply
7         wire VGND = 1'b0; // Ground
8         // Dump the signals to a VCD file. You can view it with gtkwave.
9         initial begin
10            $dumpfile("tb.vcd");
11            $dumpvars(0, tb);
12            #1;
13        end
14        // Wire up the inputs and outputs:
15        reg clk;
16        reg rst_n;
17        reg ena;
18        reg [7:0] ui_in;
19        reg [7:0] uio_in;
20        wire [7:0] uo_out;
21        wire [7:0] uio_out;
22        wire [7:0] uio_oe;
23        // Clock generation
24        initial begin
25            clk = 0;
26            forever #5 clk = ~clk; // 10 ns clock period
27        end
28        // Reset and enable signals
29        initial begin
30            rst_n = 0;
31            ena = 0;
32            #10 rst_n = 1; // Release reset after 10ns
33            #10 ena = 1; // Enable after 20ns
34        end
35        // Instantiate your ALU module
36        tt_um_alf19185_ALU U1 (
37            'ifdef GL_TEST
38                .VPWR(VPWR), // Connect to power using wire
39                .VGND(VGND), // Connect to ground using wire
40            'endif
41                .ui_in (ui_in),
42                .uo_out (uo_out),
43                .uio_in (uio_in),
44                .uio_out(uio_out),
45                .uio_oe (uio_oe),
46                .ena (ena),
47                .clk (clk),
48                .rst_n (rst_n) );
49        endmodule
50
```

Nota. Adaptación propia

## 12.2. Proyecto ALU de 4 bits en datasheet oficial del chip TT09

**Figura 29.** *Datasheet TT09 del proyecto ALU 4 bits - Página 1*

### 4 bit ALU

- Author: Gabriela Alfaro
- Description: A simple design of an Arithmetic Logic Unit capable of basic operations: addition, subtraction, multiplication, division and some logic operations.
- Language: Verilog

#### How it works?

The 4-bit ALU (Arithmetic Logic Unit) is designed to perform a range of arithmetic and logical operations on two 4-bit inputs, A and B. The operation is determined by a 3-bit control signal, Opcode, which specifies the function to execute, such as addition, subtraction, multiplication, division, and bitwise operations (AND, OR, NOT, XOR).

When an arithmetic operation like addition is selected, the ALU outputs an 8-bit result, ALU\_Result, to accommodate larger sums or products, and it sets a Carry flag if there's an overflow. For logical operations like AND or OR, the ALU applies the operation bit-by-bit between A and B. The Zero flag is activated when the result is zero, providing a useful condition for further logic. This flexibility allows the ALU to handle various computational tasks, making it a crucial part of digital systems that require multi-functional data processing.

#### How to test?

To test the design, the operation codes are:

- Addition (000)
- Subtraction (001)
- Multiplication (010)
- Division (011)
- Logic AND (100)
- Logic OR (101)
- Logic NOT (110)
- Logic XOR (111)

#### Pinout

**Figura 30.** *Datasheet TT09 del proyecto ALU 4 bits - Página 2*

#	Input	Output	Bidirectional
0	A[0]	ALU_Out[0]	ZeroFlag
1	A[1]	ALU_Out[1]	CarryOut
2	A[2]	ALU_Out[2]	
3	A[3]	ALU_Out[3]	
4	B[0]	ALU_Out[4]	
5	B[1]	ALU_Out[5]	
6	B[2]	ALU_Out[6]	
7	B[3]	ALU_Out[7]	

## 12.3. Guías académicas

### 12.3.1. Guías académicas para cursos de Electrónica Digital

#### 12.3.1.1. Guía 1 - Introducción a simulador Wokwi

**Figura 31.** *Guía 1- Introducción a Wokwi - Página 1*

---

Universidad del Valle de Guatemala  
Electrónica Digital 1  
Ingeniería Electrónica



### Guía a Tiny Tapeout: Uso del simulador de Wokwi

24 de noviembre de 2024

---

#### ¿Qué es el proyecto Tiny Tapeout?

Tiny Tapeout es un proyecto open-source que permite a través de diseños digitales, introducir a estudiantes a la fabricación de circuitos integrados para aplicaciones específicas (ASICs). Este proyecto acepta propuestas de diseños electrónicos para poder integrar desde un diseño simple a partir de un simulador visual de circuitos hasta un diseño más complejo elaborado desde un Lenguaje de Descripción de Hardware (HDL), como lo es Verilog.

Para la introducción a estudiantes con conocimiento básicos de Electrónica Digital 1, se introduce el proyecto de Tiny Tapeout con el simulador de circuitos en línea Wokwi. Ya que permite elaborar circuitos digitales y observar errores de diseño de forma sencilla e intuitiva. Logrando identificar la influencia e importancia de las compuertas lógicas dentro de circuitos integrados.

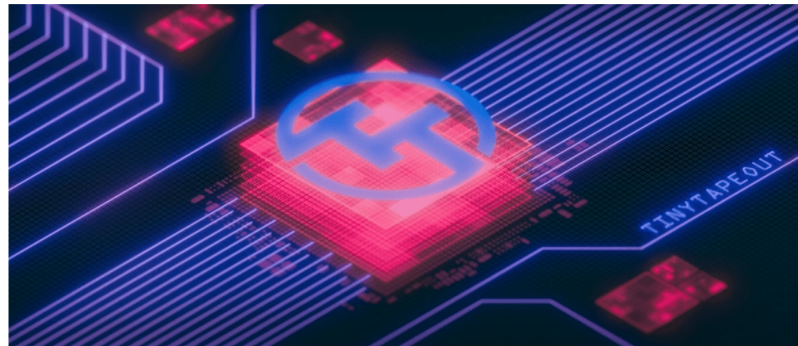


Figura 1: Tiny Tapeout

#### Objetivos

- Comprender el uso de Wokwi dentro del proyecto Tiny Tapeout.
- Familiarizar al estudiante con las herramientas del simulador Wokwi.
- Elaborar un circuito digital funcional con el simulador Wokwi.

Figura 32. Guía 1- Introducción a Wokwi - Página 2

## Parte 1: Acerca de Wokwi y creación de un proyecto

### Simulador de circuitos en línea Wokwi

Para inicializar en el simulador, es necesario ingresar a la página web de Wokwi [aquí](#). Al encontrarse dentro de la página de inicio, se recomienda inicializar sesión en la esquina superior derecha, con el correo de la universidad, como se observa en la Figura 2.



Figura 2: Página de inicio de simulador en línea Wokwi

Posteriormente al inicio de sesión, se puede tener un acceso completo a la herramienta. Permitiendo la capacidad de crear nuevos proyectos, acceder a diseños pasados y la revisión de proyectos marcados de interés elaborados por personas de la comunidad de Wokwi, como se aprecia en la Figura 3.

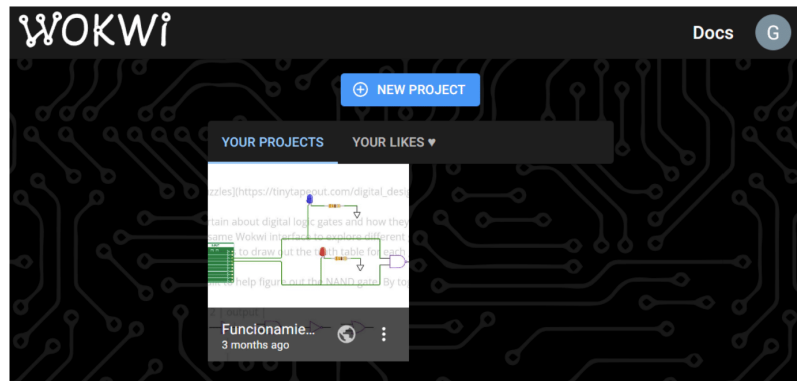


Figura 3: Página de inicio de simulador en línea Wokwi

Para crear el primer proyecto, se debe ingresar en el botón de **NEW PROJECT**. Lo cual redirige a una nueva página y se despliegan una variedad de microcontroladores con los que se pueden trabajar a través de este simulador, incluyendo variedades de ESP32, STM32, Arduino y Raspberry Pi. En este caso, debido a que el proyecto de Tiny Tapeout se orienta a la introducción de ASICs, se abrirá un proyecto en blanco para enfocarse en el uso de compuertas lógicas.

## Parte 2: Herramientas de simulación en Wokwi

### Simulación de circuitos en Wokwi

A partir del proyecto en blanco creado en la sección anterior, se puede observar que la página se encuentra dividida en 2, el espacio de simulación y un editor de texto. Iniciando por las herramientas de simulación, se encuentran 3 de ellas en la esquina superior izquierda del área de simulación como se ve en la Figura 4. El botón verde, sirve para dar inicio y pausar la simulación cuando el diseño electrónico se encuentre completo y listo para verificar su funcionamiento.



Figura 4: Herramientas de simulación.

Si se accede al botón de +, este permite añadir componentes electrónicos al diseño. En esta pestaña, se encuentran altas variedades de componentes básicos, displays, elementos de entrada y salida, compuertas lógicas, sensores, breadboards, entre otros. También se pueden buscar elementos específicos en la parte superior del listado, como se muestra en la Figura 5.

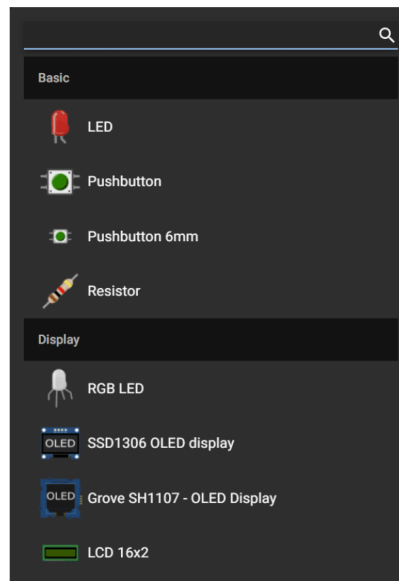


Figura 5: Añadir componentes electrónicos en la simulación.

Finalmente, se encuentra el botón gris que permite realizar ajustes de visualización dentro del área de simulación. Este permite añadir cuadrícula, realizar zoom, deshacer y rehacer elementos dentro del diseño, como se observa en la Figura 6.

Es importante recordar que existe un editor de texto que permite añadir una breve descripción de la simulación, para que la persona que lo consulte conozca el funcionamiento del diseño.

## Parte 2: Herramientas de simulación en Wokwi

### Simulación de circuitos en Wokwi

A partir del proyecto en blanco creado en la sección anterior, se puede observar que la página se encuentra dividida en 2, el espacio de simulación y un editor de texto. Iniciando por las herramientas de simulación, se encuentran 3 de ellas en la esquina superior izquierda del área de simulación como se ve en la Figura 4. El botón verde, sirve para dar inicio y pausar la simulación cuando el diseño electrónico se encuentre completo y listo para verificar su funcionamiento.



Figura 4: Herramientas de simulación.

Si se accede al botón de +, este permite añadir componentes electrónicos al diseño. En esta pestaña, se encuentran altas variedades de componentes básicos, displays, elementos de entrada y salida, compuertas lógicas, sensores, breadboards, entre otros. También se pueden buscar elementos específicos en la parte superior del listado, como se muestra en la Figura 5.

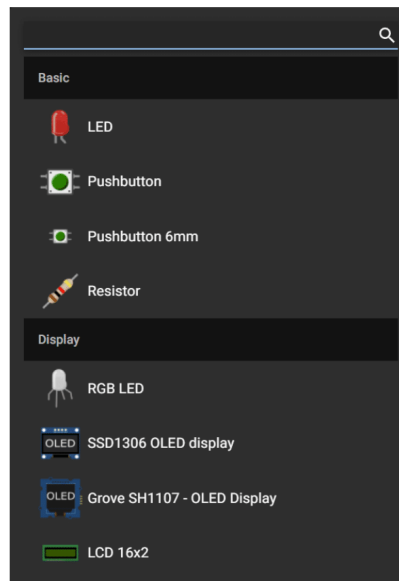


Figura 5: Añadir componentes electrónicos en la simulación.

Finalmente, se encuentra el botón gris que permite realizar ajustes de visualización dentro del área de simulación. Este permite añadir cuadrícula, realizar zoom, deshacer y rehacer elementos dentro del diseño, como se observa en la Figura 6.

Es importante recordar que existe un editor de texto que permite añadir una breve descripción de la simulación, para que la persona que lo consulte conozca el funcionamiento del diseño.

### 12.3.1.2. Guía 2 - Envío a Tiny Tapeout de diseño en Wokwi

**Figura 35.** Guía 2 - Emisión desde diseño en Wokwi - Página 1

---

Universidad del Valle de Guatemala  
Electrónica Digital 1  
Ingeniería Electrónica



## Guía a Tiny Tapeout: Entrega de un diseño electrónico desde simulador Wokwi para fabricación

24 de noviembre de 2024

---

### Introducción

Esta guía te mostrará cómo subir un diseño desde *Wokwi*, una plataforma de simulación de circuitos electrónicos, a *GitHub* y cómo conectar este diseño con *Tiny Tapeout* usando el flujo de trabajo de *OpenLane*. *Tiny Tapeout* es una plataforma que permite a los usuarios diseñar y manufacturar sus propios chips de manera sencilla y asequible, utilizando herramientas open-source.



Figura 1: Enlace Github y Tiny Tapeout

### Objetivos

- Subir un diseño simulado en *Wokwi* a un repositorio de GitHub.
- Configurar los archivos clave para la integración con *OpenLane*.
- Realizar la síntesis y generación del archivo GDSII para la manufactura del diseño.

### Parte 1: Diseñar en *Wokwi* y guardar la simulación

- **Crear un diseño en *Wokwi*:** Dirígete a [Wokwi](#) y empieza un nuevo proyecto de simulación de circuitos digitales. Diseña tu circuito usando compuertas lógicas y otros componentes electrónicos proporcionados por *Wokwi*. **Simula** tu diseño para asegurarte de que funciona como esperas.
- **Guardar el proyecto y obtener el ID:** Una vez que hayas finalizado el diseño, guarda tu proyecto. Al hacerlo, *Wokwi* generará un ID único del proyecto. Copia este ID ya que lo necesitarás en los siguientes pasos.

Figura 36. Guía 2 - Emisión desde diseño en Wokwi - Página 2

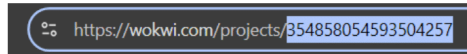


Figura 2: ID del diseño en Wokwi

## Parte 2: Crear un repositorio en GitHub con la plantilla de Tiny Tapeout

- **Clonar la plantilla de Tiny Tapeout:** Accede al repositorio de plantilla de Tiny Tapeout en [GitHub](#). Haz clic en **Fork** para crear una copia del repositorio en tu cuenta personal.

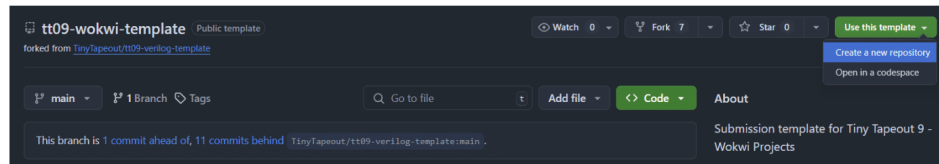


Figura 3: Usar template de Wokwi proveído por Tiny Tapeout

- **Modificar el archivo info.yaml:** Dentro de tu repositorio, busca el archivo `info.yaml`. Este archivo contiene la configuración de tu proyecto. Actualiza el campo `wokwi_id` con el ID de tu proyecto de Wokwi.
- **Agregar detalles del proyecto:** Puedes añadir información sobre tu proyecto, como su nombre, descripción y autor, para que esté bien documentado.

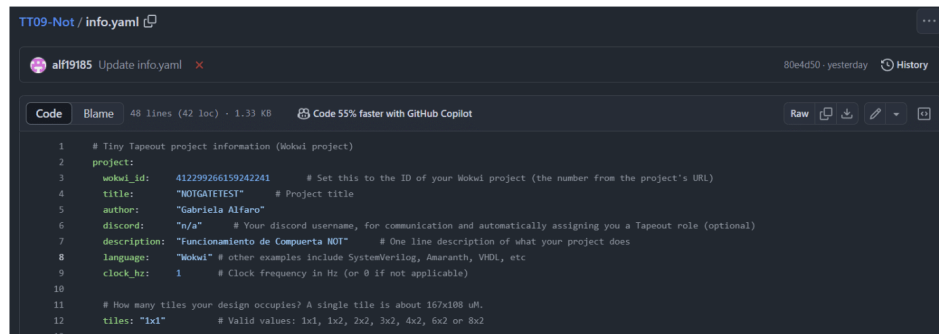


Figura 4: Detalles a modificar en archivo info.yaml

## Parte 3: Configuración y ejecución de GitHub Actions para OpenLane

- **Activar GitHub Actions:** GitHub Actions es una herramienta que ejecuta automáticamente un flujo de trabajo cuando actualizas tu repositorio. Esto incluirá la síntesis del diseño usando OpenLane. Verifica que GitHub Actions esté habilitado en la pestaña **Actions** de tu repositorio.

Figura 37. Guía 2 - Emisión desde diseño en Wokwi - Página 3

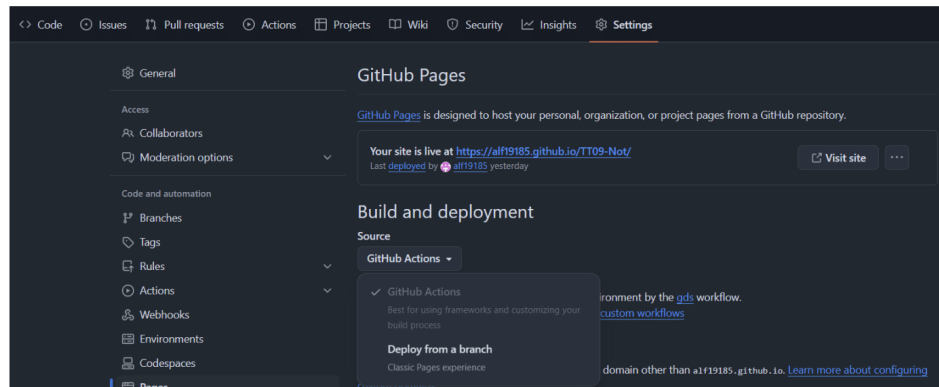


Figura 5: Habilitar Github Actions

- **Iniciar la acción para generar el archivo GDSII:** Cada vez que realices cambios en tu diseño o en el archivo `info.yaml`, GitHub Actions ejecutará los siguientes pasos:
  - **Verificación de diseño (DRC):** Se comprobará que tu diseño cumpla con las restricciones físicas y de pines.

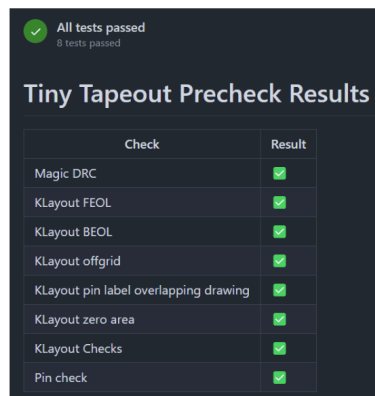
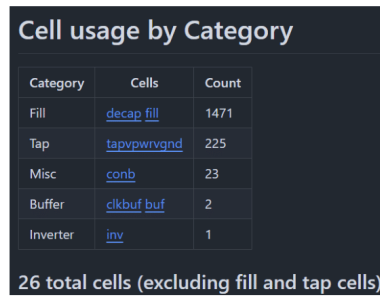


Figura 6: Revisión de reglas de diseño

- **Síntesis:** El diseño se sintetizará, convirtiéndose en un netlist de puertas lógicas.

Figura 38. Guía 2 - Emisión desde diseño en Wokwi - Página 4



Category	Cells	Count
Fill	<a href="#">decap fill</a>	1471
Tap	<a href="#">tapypwrvqnd</a>	225
Misc	<a href="#">conb</a>	23
Buffer	<a href="#">clkbuf buf</a>	2
Inverter	<a href="#">inv</a>	1

26 total cells (excluding fill and tap cells)

Figura 7: Celdas que conforman el diseño

- **Generación del archivo GDSII:** OpenLane generará el archivo GDSII, el formato necesario para la manufactura del chip.
- **Monitorear el proceso:** Se puede monitorear el progreso de la acción desde la pestaña **Actions** de GitHub. Si todo funciona correctamente, debería de completarse la generación de los archivos de manufactura, incluyendo el GDSII.

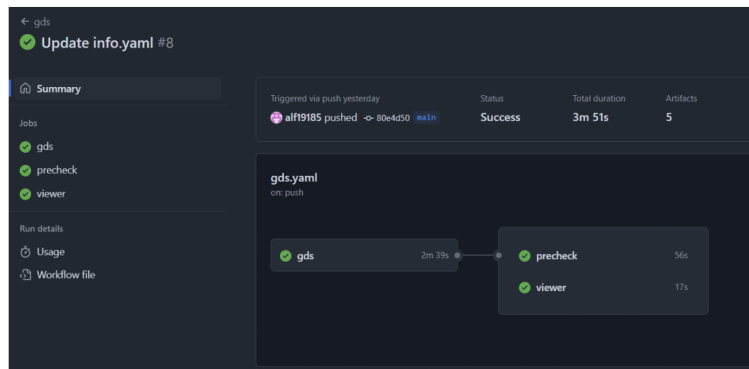


Figura 8: Monitoreo de acción GDS

#### Parte 4: Enviar el diseño para manufactura

- **Verificar los archivos:** Asegurarse que el archivo GDSII y la documentación asociada estén correctamente generados en el repositorio.

Figura 39. Guía 2 - Emisión desde diseño en Wokwi - Página 5

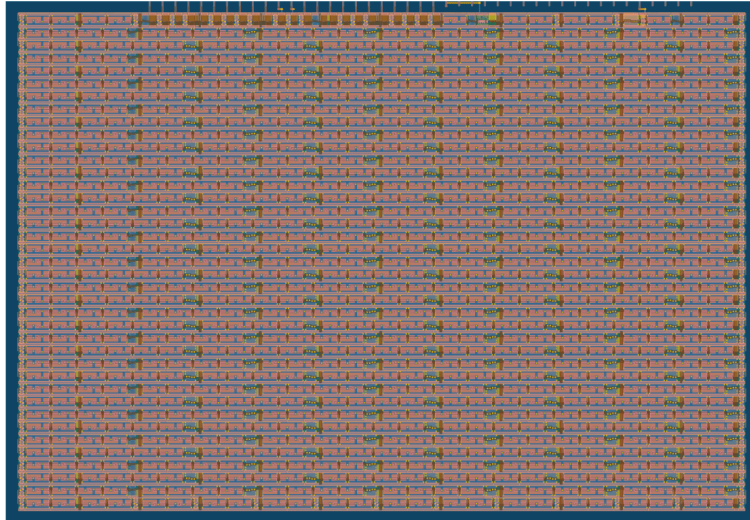


Figura 9: Vista 2D del diseño obtenido al finalizar proceso

- **Enviar el diseño a Tiny Tapeout:** Una vez completado el flujo de trabajo, puedes enviar tu diseño para su manufactura a través de la plataforma de Tiny Tapeout. Siguiendo las indicaciones del sitio web para el proceso de pago por la fabricación del chip, y se recibirá el ASIC una vez esté listo.

## 12.3.2. Guías académicas para curso de Arquitectura de Computadoras

### 12.3.2.1. Guía 1 - Adaptación y verificación de diseños en HDL

Figura 40. Guía 1 - Adaptación y verificación de diseños en HDL - Página 1

---

Universidad del Valle de Guatemala  
Arquitectura de Computadoras  
Ingeniería Electrónica



## Guía a Tiny Tapeout: Adaptación y Verificación de Diseños en HDL

---

### Introducción

Esta guía describe el proceso para adaptar y verificar diseños digitales utilizando Verilog, con un enfoque específico en proyectos de Tiny Tapeout. Este proyecto es una plataforma de código abierto, colaborativa para la fabricación de diseños digitales personalizados y orientados para la educación. Los principios descritos aquí ayudan a cumplir con los requisitos técnicos y estructurales para enviar un diseño electrónico en HDL para su fabricación mediante Tiny Tapeout.

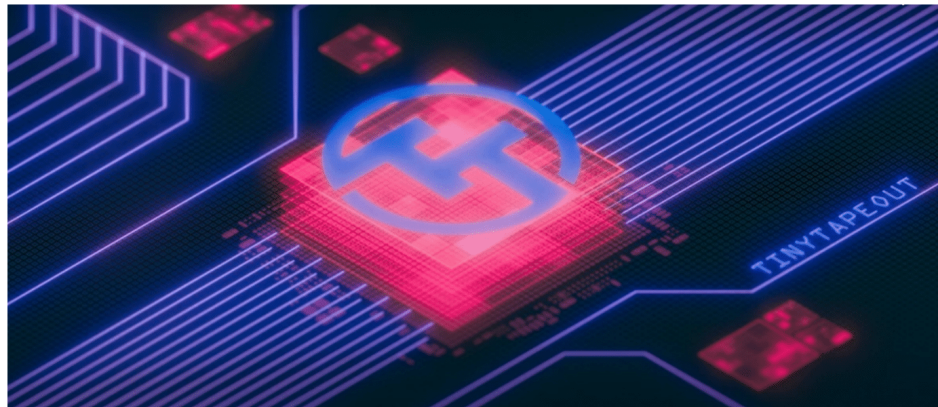


Figura 1: Tiny Tapeout

### Objetivos

- Configurar un proyecto HDL en Verilog usando la plantilla de TinyTapeout.
- Crear y ajustar el archivo `info.yaml` e `info.md` con los detalles del proyecto.
- Diseñar y adaptar módulos HDL que cumplan con las especificaciones de TinyTapeout.

## Antes de iniciar: Requisitos del Diseño

- **Nombre del módulo:** El módulo principal debe comenzar con `tt_um_` seguido de un identificador único.
- **Entradas y salidas:** Las entradas deben mapearse a `ui_in` y `uio_in`, mientras que las salidas deben asignarse a `uo_out` y `uio_out`.
- **Power y Ground:** Los módulos deben incluir conexiones a `VPWR` y `VGND`.

## Parte 1: Creación del Proyecto en GitHub

- **Clonar la plantilla de Tiny Tapeout:** Primero, acceder a la página oficial de la plantilla para diseños en Verilog en GitHub, se puede encontrar el repositorio en [aquí](#). Es importante revisar la plantilla más actual dependiendo el número de chip que se encuentre en lanzamiento ya que pueden existir cambios en ella. Utilizando la opción de Fork para crear una copia del repositorio en una cuenta personal de GitHub.
- **Agregar los archivos de Verilog:** Dirígete a la carpeta `src` en el repositorio clonado y sube los archivos `.v` de Verilog que definen tu diseño. Es esencial que el archivo principal contenga el módulo superior que define el comportamiento de tu circuito y su correspondiente testbench.

## Parte 2: Configuración del archivo `info.yaml`

El archivo `info.yaml` es crucial para describir tu diseño y especificar los archivos necesarios para la compilación.

**Estructura del Archivo `info.yaml`:**

- **title:** Título del proyecto.
- **author:** Nombre del autor.
- **source\_files:** Lista de archivos Verilog que contienen tu diseño.
- **top\_module:** Nombre del módulo superior en tu diseño.

## Parte 3: Configuración del archivo `info.md` en la carpeta `docs`

El archivo `info.md` es una parte esencial en los proyectos enviados a *Tiny Tapeout*, ya que proporciona información detallada sobre el diseño y su funcionalidad. Este archivo está ubicado dentro de la carpeta `docs` del repositorio y debe incluir información clara y concisa sobre el proyecto.

**Estructura del archivo `info.md`**

El contenido del archivo `info.md` debe estructurarse en las siguientes secciones:

- **Título del Proyecto:** Proporcione un título claro que identifique el diseño. Por ejemplo:

```
# ALU de 4 bits
```

- **Descripción del Proyecto:** Incluya una breve descripción que explique la funcionalidad del diseño. Por ejemplo:

**Figura 42.** *Guía 1 - Adaptación y verificación de diseños en HDL - Página 3*

Esta es una unidad aritmética y lógica (ALU) de 4 bits capaz de realizar operaciones aritméticas (suma, resta) y lógicas (AND, OR, XOR). Fue diseñada en Verilog y optimizada para su manufactura en Tiny Tapeout.

- **ID de Wokwi (opcional):** Si el proyecto fue creado y simulado en *Wokwi*, incluya el ID del proyecto:

```
## Wokwi Project ID
123456
```

- **Detalles Técnicos:** Proporcione información técnica sobre las entradas, salidas y funcionalidad del circuito. Por ejemplo:

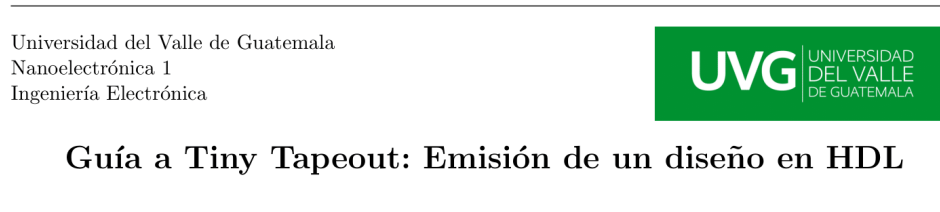
```
## Technical Details
- Entradas: 4 bits (A y B), 3 bits para seleccionar operación.
- Salida: 8 bits (resultado de la operación).
- Operaciones soportadas:
  - Suma
  - Resta
  - Multiplicación
  - División
  - AND
  - OR
  - NOT
  - XOR
```

- **Instrucciones para usar el proyecto :** Añada cualquier instrucción relevante para entender o usar el diseño. Por ejemplo:

```
## How to Use
Este diseño requiere que las entradas A y B estén definidas en los pines 0 a 8.
Use los pines 9 a 11 para seleccionar la operación deseada.
```

### 12.3.2.2. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout

Figura 43. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 1



## Introducción

El proceso de transformar un diseño HDL en un chip físico comienza con la síntesis y culmina con la fabricación del chip. OpenLane es el conjunto de herramientas utilizadas para convertir tu diseño Verilog en un formato manufacturable (archivo GDSII). En esta guía, aprenderás cómo tomar tu diseño de HDL y generar el archivo GDSII paso a paso.

## Objetivos

- Enviar un proyecto HDL a Tiny Tapeout utilizando GitHub.
- Comprender cómo se utiliza OpenLane para la síntesis del diseño y generación del archivo GDSII.
- Ejecutar y verificar el diseño paso a paso hasta la etapa de fabricación.

## Parte 1: Envío del Proyecto a Tiny Tapeout

- Conectar con Tiny Tapeout : Accede al formulario de envío en [Tiny Tapeout](#) y sigue los pasos para conectar tu repositorio de GitHub. Esto permitirá a Tiny Tapeout acceder a tu diseño Verilog automáticamente desde el repositorio. Es importante verificar que al adaptar el template dado por ellos en la página, se habiliten las acciones dentro de Github.

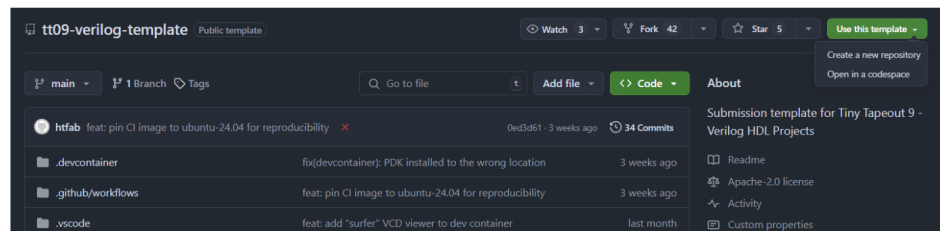


Figura 1: Realizar una copia al repositorio

- Verificación automática : Tiny Tapeout ejecuta una verificación automática del diseño, comprobando que cumple con las restricciones de pines (8 entradas, 8 salidas), tamaño (160x100 micrómetros), frecuencia de reloj mínima (50 MHz), entre otros. Habilitar las acciones, permiten que el proceso de OpenLane se ejecute automáticamente cada vez que se realiza un commit dentro del repositorio.

Figura 44. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 2

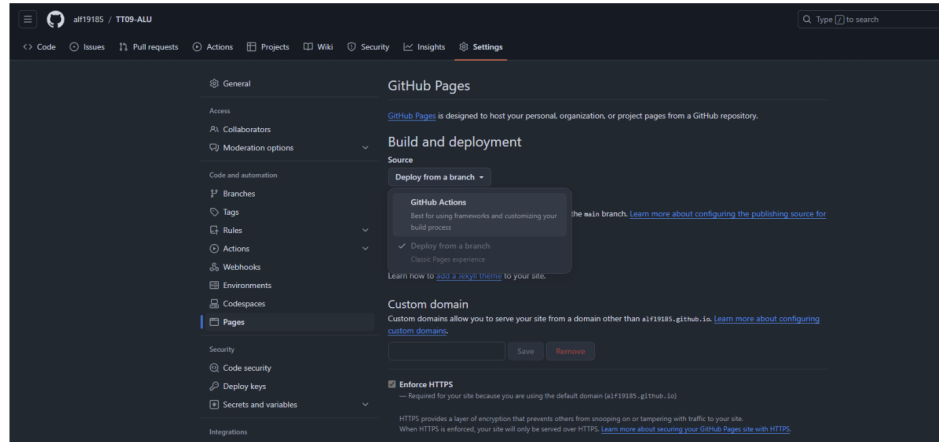


Figura 2: Habilitar las acciones permite que OpenLane opere correctamente

## Parte 2: ¿Que proceso realiza OpenLane?

OpenLane es una suite de herramientas automatizadas que transforma tu diseño HDL en un archivo GDSII listo para ser fabricado. Es importante conocer que este proceso puede llegar a tomar varios minutos dependiendo de la complejidad del diseño. A continuación, se describe paso a paso cómo se utiliza para generar el archivo final.

### Paso 1: Sintetización

1. Convertir Verilog a puertas lógicas : El primer paso en OpenLane es la sintetización , que toma tu diseño HDL en Verilog y lo transforma en una red de compuertas lógicas, utilizando una librería de celdas estándar. Esto se realiza con la herramienta Yosys , que identifica los componentes digitales y optimiza el diseño.

2. Optimización del diseño : Durante la síntesis, OpenLane optimiza el uso del área, minimizando la cantidad de puertas lógicas y reduciendo el consumo energético, todo sin alterar la funcionalidad del diseño.

3. Comprobación : Una vez que se ha sintetizado el diseño, se realiza una comprobación de lógica para asegurar que la salida del diseño coincide con las entradas originales, validando que no haya errores en la conversión.

### Paso 2: Floorplanning (Planificación del área física)

1. Definir el área disponible : OpenLane asigna un espacio físico de 160x100 micrómetros en el que se colocarán los componentes del diseño. Esto se conoce como el floorplanning y define cómo se distribuyen las compuertas lógicas en el espacio físico disponible del chip.

**Figura 45.** Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 3

Utilisation (%)	Wire length (um)
9.715 %	4199

Cell usage by Category		
Category	Cells	Count
Fill	<a href="#">decap fill</a>	1444
Tap	<a href="#">tapvprvrgnd</a>	225
Combo Logic	<a href="#">a21o a221o a2111o a211o o211a o31aj a31o a21oj o21a and4b o2bb2a a2bb2oj a2bb2o or3b o22a a32o a22o o2111aj o221a o21aj o21bai o2111a a21bo o32a o31a or4bb or4b a31oj a32oi</a>	68
NOR	<a href="#">nor2 nor3 xnor2</a>	27
NAND	<a href="#">nand2 nand3</a>	27
OR	<a href="#">or2 or3 xor2 or4</a>	25
Buffer	<a href="#">buf clkbuf</a>	21
Misc	<a href="#">conb dlymetal6s2s</a>	18
AND	<a href="#">and3 and2 and4 a21boj</a>	13
Inverter	<a href="#">inv</a>	8

Figura 3: Cantidad de compuertas que conforman el diseño son mostradas junto con estadísticas de ruteo

2. Separación de áreas críticas : En esta fase, también se identifican áreas críticas, como el espacio reservado para los pines de entrada y salida, así como las zonas de conexión para la alimentación (VDD y VSS). Esto garantiza que el diseño sea viable para la fabricación.

### Paso 3: Colocación y Ruteo (Place and Route)

1. Colocación de componentes : Una vez que se ha definido el área del diseño, OpenLane organiza las compuertas lógicas de forma eficiente en el espacio disponible. Esta colocación garantiza que los componentes no se superpongan y estén distribuidos de manera óptima dentro del área del chip.


2. Ruteo de las señales : Después de colocar los componentes, OpenLane realiza el ruteo , conectando todas las señales entre las compuertas lógicas, minimizando el tiempo de propagación de las señales y optimizando el uso de energía.

3. Optimización del ruteo : OpenLane se asegura de que las rutas de señal sean lo más cortas posible, lo que reduce el retardo en la propagación de las señales a través del chip, mejorando el rendimiento global.

### Paso 4: Verificación de Reglas de Diseño (DRC) y Temporización

1. Verificación DRC : La verificación de reglas de diseño (DRC) asegura que el diseño cumple con los estándares de la fábrica, como el espaciado mínimo entre componentes, el tamaño de las interconexiones y la colocación correcta de los pines de entrada/salida. Si se encuentran violaciones de las reglas de diseño, se notifica para realizar las correcciones necesarias.

Figura 46. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 4



Check	Result
MagiC DRC	✓
KLayout FEOL	✓
KLayout BEOL	✓
KLayout offgrid	✓
KLayout pin label overlapping drawing	✓
KLayout zero area	✓
KLayout Checks	✓
Pin check	✓

Figura 4: Chequeo de reglas de diseño

2. Análisis de tiempos : Se ejecuta un análisis de tiempos para comprobar que el diseño cumple con los requisitos de temporización. Esto asegura que las señales llegan a sus destinos dentro del tiempo permitido por la frecuencia de reloj especificada.

#### Paso 5: Generación del archivo GDSII

1. Exportar el diseño físico : Tras completar la colocación, ruteo y verificación de tiempos, OpenLane genera el archivo GDSII , que es la representación física completa del diseño. Este archivo es el estándar de la industria para la fabricación de chips.



Figura 5: Visualización 2D del render del diseño final

Figura 47. Guía 2 - Emisión de diseños en HDL a Tiny Tapeout - Página 5

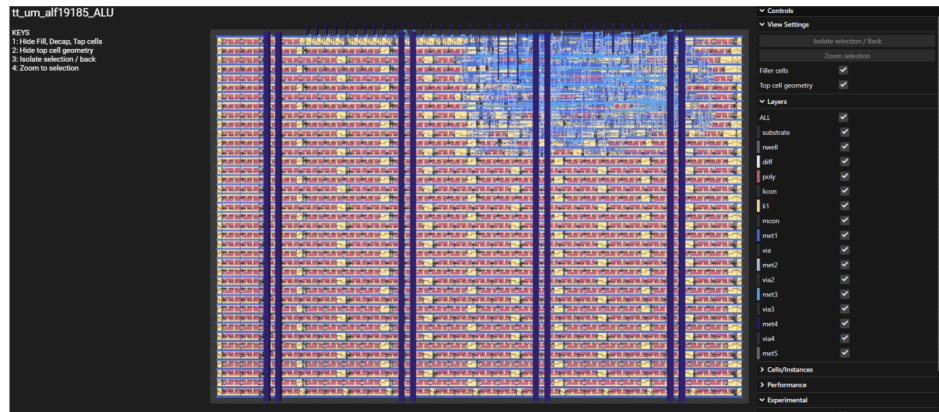


Figura 6: Visualización 3D del render del diseño final

2. Revisión final : Antes de enviar el archivo GDSII a la fábrica, se realiza una revisión final para asegurar que no haya errores críticos y que el diseño esté completamente optimizado, este se realiza en Tiny Tapeout al momento de cerrar el lote de diseños a fabricar en un solo chip.

### Parte 3: Envío a Fabricación y Recepción del Chip

- Revisión y Pago : Después de generar el archivo GDSII, Tiny Tapeout te solicitará que realices el pago correspondiente para que el diseño sea incluido en el próximo lote de fabricación. El pago por diseño cubre los costos de manufactura y envío del chip.
- Fabricación y envío : Una vez fabricado, recibirás tu chip montado en una PCB para facilitar las pruebas y el desarrollo posterior. Este proceso puede tardar varios meses, ya que los chips se fabrican en lotes ("shuttles") junto con otros proyectos.
- Pruebas y validación del chip físico : Tras recibir el chip, puedes realizar pruebas en un entorno físico utilizando herramientas como osciloscopios y analizadores lógicos para comprobar que el diseño funciona como se esperaba en hardware real.

### 12.3.3. Guías académicas para cursos de Nanoelectrónica

#### 12.3.3.1. Guía 1 - Introducción a Siliwiz

Figura 48. Guía 1 - Introducción a Siliwiz - Página 1

---

Universidad del Valle de Guatemala  
Nanoelectrónica 1  
Ingeniería Electrónica



## Guía a Nanoelectrónica: Introducción a Siliwiz

1 de octubre de 2024

---

### Introducción

**\*\*SiliWiz\*\*** es una herramienta educativa en línea gratuita diseñada para enseñar los fundamentos de los semiconductores y cómo se fabrican a nivel básico. Desarrollada como parte del proyecto Tiny Tapeout, SiliWiz permite dibujar componentes simples como resistencias, capacitores y MOSFETs, y simular su comportamiento. En esta guía, te mostraremos cómo usar SiliWiz para aprender los conceptos clave de los semiconductores.



Figura 1: Logo Siliwiz

### Objetivos

- Familiarizarse con la interfaz de SiliWiz.
- Comprender cómo se fabrican los componentes semiconductores a nivel básico.
- Realizar simulaciones de componentes electrónicos para observar su comportamiento.

### Parte 1: Acceso y navegación en SiliWiz

- Para comenzar, accede a SiliWiz [aquí](#).

Figura 49. Guía 1 - Introducción a Siliwiz - Página 2

- Una vez dentro, verás un lienzo donde podrás dibujar los componentes semiconductores, junto con herramientas para seleccionar diferentes capas y materiales. A continuación en la Figura 2 se logra observar la interfaz y herramientas útiles.

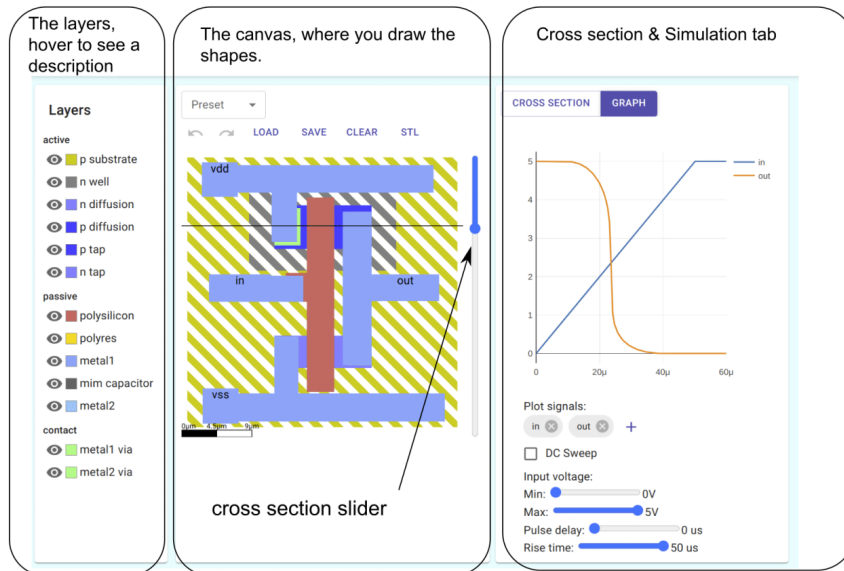


Figura 2: Interfaz Siliwiz

- Familiarízate con las capas, ya que estas son fundamentales en la fabricación de chips. Podrás dibujar elementos como capas de difusión de tipo N y P, contacto con metal y la capa de polisilicio, que forman parte del proceso de fabricación de semiconductores. Al colocar los distintos tipos de materiales, se pueden editar en ancho, largo y nombrar para colocar señales para simular, como se observa en la Figura 3.

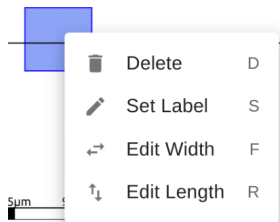


Figura 3: Edición de materiales en Siliwiz

## Parte 2: Dibujar y simular un componente

### Dibujo de una resistencia

- Selecciona la capa de **\*\*Poliresistivo (Polyres)\*\*** y dibuja una forma que represente una resistencia en el lienzo.
- Conecta los extremos de la resistencia utilizando contactos metálicos. En la Figura 4, se muestra un ejemplo de cómo se observaría de forma transversal.

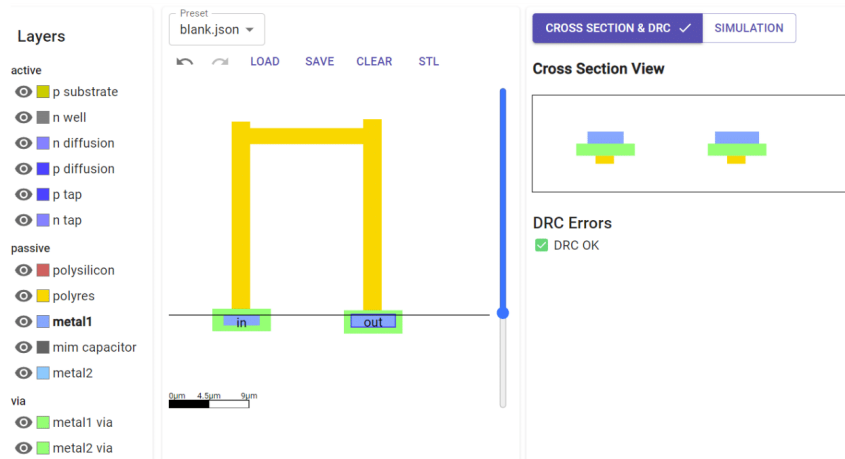


Figura 4: Resistencia en Siliwiz

- Usa la simulación integrada para observar cómo la resistencia afecta al flujo de corriente en el circuito. Ya estando listo el diseño, se puede observar la simulación de la señal en el tiempo como se muestra a continuación en la Figura 5:

Figura 51. Guía 1 - Introducción a Siliwiz - Página 4

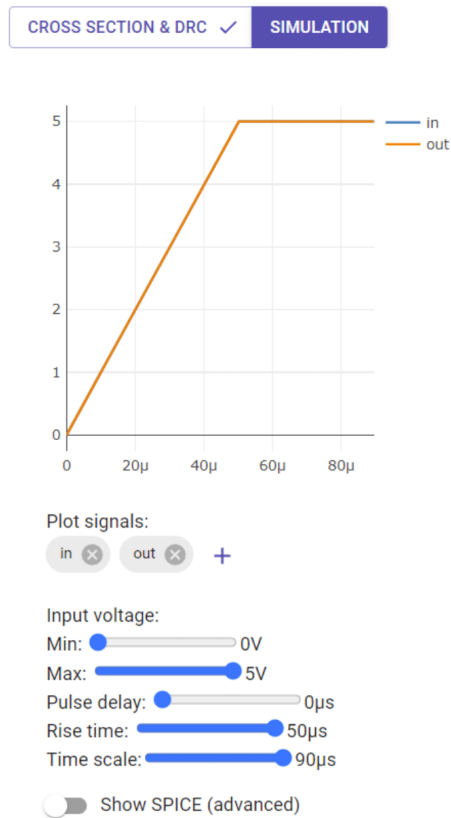


Figura 5: Simulación de una resistencia en Siliwiz

- Si ahora se habilita la opción de SHOW SPICE, se verá en la parte inferior el código SPICE de la simulación, como se muestra en la Figura 6. Es importante notar que se puede modificar el código y la simulación se modifica en tiempo real.

Figura 52. Guía 1 - Introducción a Siliwiz - Página 5

### SPICE Code (advanced)

 [DOWNLOAD MAGIC](#)  [TECH FILE](#)  [DOWNLOAD SPICE](#)  Edit SPICE

```
* SiliWiz Simulation (app rev 4823f4f)

*signals: in out

Vdd vdd 0 5 ; power supply: 5V
Vss vss 0 0 ; ground

* Input pulse: ramp the `in` signal
Vin in 0 pulse (0 5 0u 50u 50u 1 1)

* Extracted circuit:
R0 in out 11080.000000
C0 a_78_43# out 0.01fF
C1 a_78_43# in 0.01fF
C2 out SUB 0.08fF
C3 in SUB 0.08fF
C4 a_78_43# SUB 0.34fF $ **FLOATING

* Models:
.model nmos nmos (vto=1 tox=15n uo=600 cbd=20f cbs=20f gamma=0.37)
.model pmos pmos (vto=-1 tox=15n uo=230 cbd=20f cbs=20f gamma=0.37)

* Simulation parameters:
.tran 746.14n 89.54u

.end
```

Figura 6: SPICE de la resistencia creada en Siliwiz

## Guía a Nanoelectrónica: Transistores en Siliwiz

6 de octubre de 2024

### Introducción

El **MOSFET** (Transistor de Efecto de Campo de Semiconductor de Óxido Metálico) es uno de los componentes más importantes en la electrónica moderna, usado principalmente para amplificar o conmutar señales electrónicas. Los MOSFETs tienen la capacidad de controlar el flujo de corriente entre el **drenador** (drain) y la **fuentes** (source) mediante la aplicación de un voltaje en la **puerta** (gate). Esta estructura les permite funcionar como interruptores electrónicos o amplificadores de señal.

Existen dos tipos principales de MOSFETs:

- **MOSFET de tipo N**: Los electrones son los principales portadores de carga, y el canal se forma cuando se aplica un voltaje positivo en la puerta.
- **MOSFET de tipo P**: Los agujeros son los portadores de carga mayoritarios, y el canal se forma cuando se aplica un voltaje negativo en la puerta.

El MOSFET de tipo N es el más común y eficiente debido a que los electrones tienen una mayor movilidad que los agujeros, lo que resulta en tiempos de conmutación más rápidos. Además, los MOSFETs se utilizan ampliamente en circuitos integrados para construir puertas lógicas y otros dispositivos fundamentales en sistemas digitales.

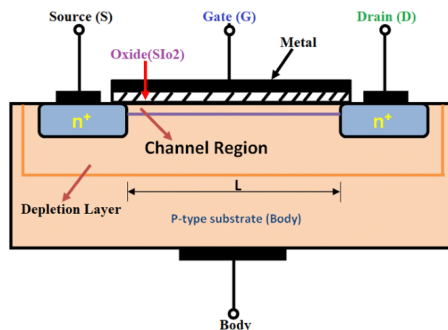


Figura 1: Estructura de un MOSFET

## Objetivos

- Dibujar un MOSFET tipo N utilizando las capas disponibles en SiliWiz.
- Comprender cómo se forma el canal de conducción en un MOSFET y cómo el voltaje aplicado en la puerta controla este proceso.
- Simular el comportamiento del MOSFET y trazar la curva VGS (Voltaje entre puerta y fuente) para medir el voltaje de umbral ( $V_{th}$ ).

## Parte 1: Dibujar un MOSFET tipo N en SiliWiz

Para visualizar un MOSFET de tipo N, sigue estos pasos en SiliWiz:

- **\*\*Seleccionar la capa de sustrato P (p-substrate):\*\*** Dibuja un rectángulo grande que cubra todo el lienzo. Este sustrato representa la base del MOSFET, que está hecho de silicio dopado con material tipo P. Es importante recordar que el sustrato p debe estar conectado a vss . Para ello, utilizamos un área p ligeramente dopada, denominada p-tap. En la esquina, dibujar un pequeño cuadrado de p-tap. Luego, lo conectamos a través de una vía de metal 1 a un contacto de metal 1 y etiquetarlo como vss.



Figura 2: Simulación del sustrato junto p-tap

- **\*\*Añadir el área de difusión tipo N:\*\*** Cambia a la capa **\*\*N-Difusión\*\*** y dibuja sobre el sustrato P. Estos representan el **\*\*drenador\*\*** y la **\*\*fuente\*\*** del MOSFET. La separación entre estos dos rectángulos determinará la longitud del canal.

Figura 55. Guía 2 - Transistores en Siliwiz - Página 3

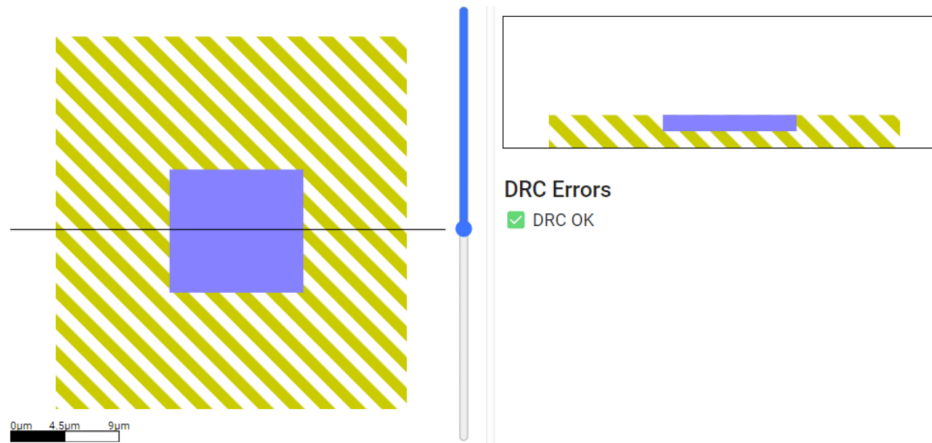


Figura 3: Añadir difusión en simulación.

- **\*\*Dibujar la puerta del MOSFET (Polisilicona):\*\*** Usa la capa **\*\*Polisilicona\*\*** y dibuja un rectángulo estrecho entre el drenador y la fuente. Este será la **\*\*puerta\*\*** del MOSFET, el componente que controlará el flujo de corriente en el canal.

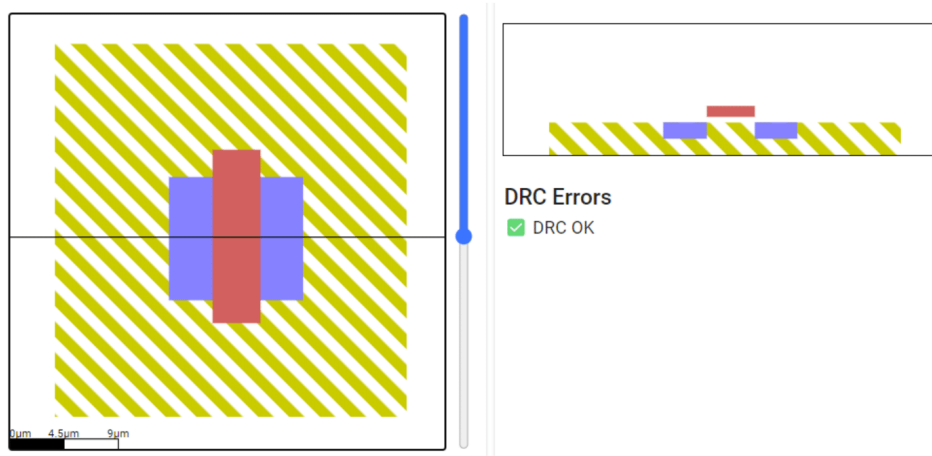


Figura 4: Añadir difusión en simulación.

- **\*\*Conectar las áreas de drenador, fuente y puerta:\*\*** Usa la capa **\*\*Metal 1\*\*** para conectar el drenador y la fuente a contactos metálicos. Conecta también la puerta a un contacto metálico, lo que permitirá aplicar voltaje y controlar el MOSFET.

Figura 56. Guía 2 - Transistores en Siliwiz - Página 4

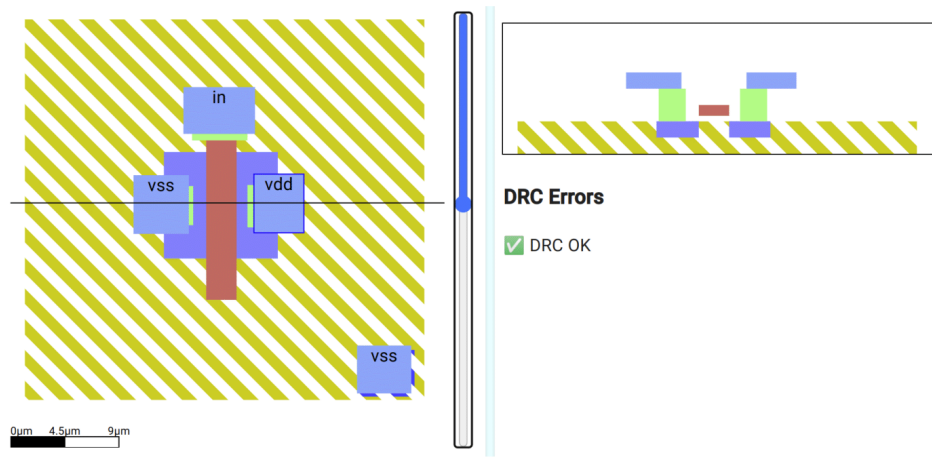


Figura 5: NMOS simulado en Siliwiz.

## Parte 2: Simulación del MOSFET y medición del voltaje de umbral

- **\*\*Realiza la simulación:\*\*** En el menú de simulación de SiliWiz, selecciona la opción para medir la curva **\*\*VGS\*\*** (Voltaje entre la puerta y la fuente) para observar cómo se comporta el MOSFET cuando varías el voltaje en la puerta.

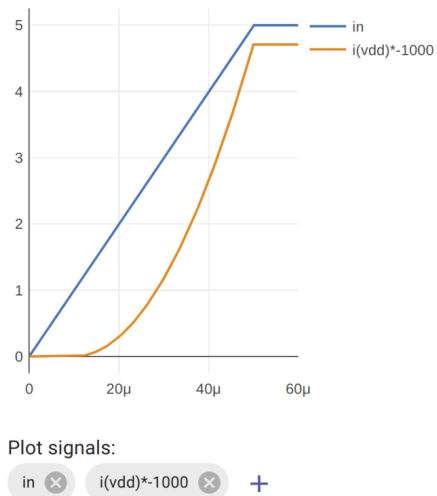


Figura 6: Medición corriente y voltaje de umbral en NMOS.

**Figura 57.** *Guía 2 - Transistores en Siliwiz - Página 5*

- **\*\*Trazado de la curva VGS:\*\*** La curva VGS muestra cómo el voltaje aplicado en la puerta afecta la formación del canal entre el drenador y la fuente. A medida que aumentas el voltaje de la puerta, el canal se forma y la corriente comienza a fluir entre el drenador y la fuente.
- **\*\*Medición del voltaje de umbral ( $V_{th}$ ):\*\*** El **\*\*voltaje de umbral\*\*** es el punto en el que el MOSFET comienza a conducir corriente. En la simulación, podrás observar el punto en que se forma el canal y el MOSFET empieza a permitir el paso de corriente, lo que te dará una idea del rendimiento del dispositivo.

Figura 58. Guía 3 - Compuertas Lógicas en Siliwiz - Página 1

---

Universidad del Valle de Guatemala  
Nanoelectrónica 1  
Ingeniería Electrónica



## Guía a Nanoelectrónica: Compuertas lógicas en Siliwiz

19 de octubre de 2024

---

### Introducción

El Inversor CMOS (Metal-Oxide-Semiconductor Field-Effect Transistor) es uno de los circuitos lógicos más básicos y fundamentales en la electrónica digital. Este componente es esencial para la construcción de puertas lógicas y circuitos más complejos. SiliWiz es una herramienta educativa en línea que facilita la visualización y el diseño de componentes semiconductores, permitiendo a los usuarios comprender mejor cómo funcionan y se fabrican estos dispositivos.

En esta guía, se aprenderá a diseñar un inversor CMOS utilizando SiliWiz, así como a simular y analizar su comportamiento. Esta guía proporcionará una comprensión práctica de cómo los transistores PMOS y NMOS trabajan juntos para invertir una señal lógica funcionando como un inversor.

### Objetivos

- Comprender la teoría detrás de los transistores MOSFET tipo N y tipo P.
- Aprender a utilizar SiliWiz para diseñar un inversor CMOS.
- Simular el comportamiento del inversor CMOS y analizar su funcionamiento.

### Parte 1: ¿Cómo es un inversor a nivel nanoelectrónico?

#### Transistor MOSFET tipo N (NMOS)

- Estructura: Compuesto por un canal de tipo N entre el drenador (Drain) y la fuente (Source).
- Funcionamiento: Cuando se aplica un voltaje positivo en la puerta (Gate) respecto al sustrato, se forma un canal que permite el flujo de electrones desde el drenador hacia la fuente.
- Aplicación en CMOS: El NMOS se utiliza para conectar la salida a tierra cuando la entrada es alta.

#### Transistor MOSFET tipo P (PMOS)

- Estructura: Compuesto por un canal de tipo P entre el drenador y la fuente.
- Funcionamiento: Cuando se aplica un voltaje negativo en la puerta respecto al sustrato, se forma un canal que permite el flujo de huecos desde el drenador hacia la fuente.
- Aplicación en CMOS: El PMOS se utiliza para conectar la salida a la alimentación (VDD) cuando la entrada es baja.

**Figura 59.** Guía 3 - Compuertas Lógicas en Siliwiz - Página 2

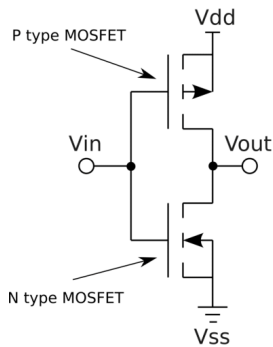


Figura 1: Esquemático de un inversor

## Parte 2: Creación del Inversor CMOS en SiliWiz

### Paso 1: Acceder a SiliWiz

- Dirígete al sitio web de SiliWiz en el [sitio oficial](#).
- Selecciona la opción para crear un Inversor CMOS. Esto te llevará al entorno de diseño donde podrás dibujar los componentes.

### Paso 2: Dibujar el Transistor PMOS

- Seleccionar la capa de difusión tipo P: En el menú de SiliWiz, selecciona la capa correspondiente para dibujar componentes tipo P.
- Dibujar el PMOS: Dibuja un rectángulo en la parte superior del lienzo que representará el transistor PMOS. Este transistor conectará la salida a la alimentación (VDD) cuando la entrada sea baja.
- Añadir terminales: Agrega los terminales de fuente (Source) y drenador (Drain) en los extremos del PMOS.

### Paso 3: Dibujar el Transistor NMOS

- Seleccionar la capa de difusión tipo N: Cambia a la capa correspondiente para dibujar componentes tipo N.
- Dibujar el NMOS: Dibuja un rectángulo en la parte inferior del lienzo que representará el transistor NMOS. Este transistor conectará la salida a tierra (VSS) cuando la entrada sea alta.
- Añadir terminales: Agrega los terminales de fuente y drenador en los extremos del NMOS.

### Paso 4: Dibujar la Puerta del Inversor

- Seleccionar la capa de polisilicona: En el menú de capas, selecciona la capa de polisilicona para dibujar la puerta.
- Dibujar la puerta: Dibuja una línea horizontal que conecte las puertas de ambos transistores (PMOS y NMOS). Esta línea representará la entrada (Gate) del inversor.

## Figura 60. Guía 3 - Compuertas Lógicas en Siliwiz - Página 3

- Agregar contactos metálicos: Usa la capa de metal para conectar la entrada a una fuente de señal externa y la salida a un nodo común donde se unen los drenadores del PMOS y NMOS.

### Paso 5: Conectar la Salida

- Definir el nodo de salida: El punto donde se unen los drenadores del PMOS y NMOS es la salida del inversor.
- Agregar contactos de salida: Usa la capa de metal para conectar el nodo de salida a un contacto metálico, permitiendo observar la señal invertida.

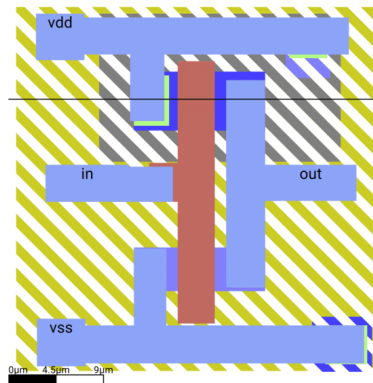


Figura 2: Diseño de un inversor en Siliwiz

## Parte 3: Simulación del Inversor CMOS

### Paso 1: Configurar la Simulación

- Seleccionar la herramienta de simulación: En SiliWiz, elige la opción para iniciar la simulación del circuito.
- Configurar la entrada: Aplica una señal de entrada baja (0 V) a la puerta del inversor.

### Paso 2: Observar el Comportamiento

- Entrada baja (0 V):
  - PMOS encendido: Conecta la salida a la alimentación (VDD).
  - NMOS apagado: No permite el flujo de corriente hacia tierra.
  - Resultado: Salida alta (VDD).
- Entrada alta (VDD):
  - PMOS apagado: No permite el flujo de corriente desde VDD.
  - NMOS encendido: Conecta la salida a tierra (VSS).
  - Resultado: Salida baja (0 V).

Figura 61. Guía 3 - Compuertas Lógicas en Siliwiz - Página 4

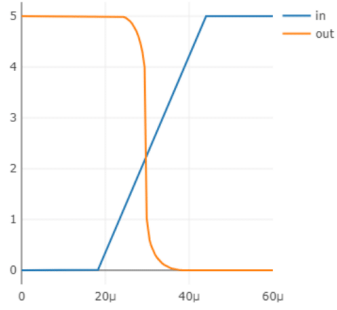


Figura 3: Simulación con entrada de 0V

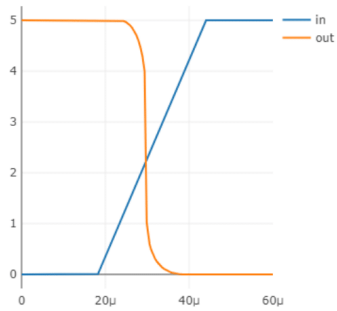


Figura 4: Simulación con entrada de 5V

Figura 62. Guía 4 - Introducción a Stixu - Página 1

Universidad del Valle de Guatemala  
Nanoelectrónica 1  
Ingeniería Electrónica



## Guía a Nanoelectrónica: Introducción a Stixu

22 de octubre de 2024

### Introducción

Stixu es una plataforma en línea especializada en la creación de stick diagrams, utilizados principalmente en el diseño de semiconductores a nivel físico. Los stick diagrams permiten representar los transistores y las conexiones en un circuito de manera simplificada antes de proceder con el diseño físico detallado. En esta guía, aprenderás cómo utilizar Stixu.io para crear un inversor CMOS, uno de los circuitos más básicos y fundamentales en la electrónica digital.

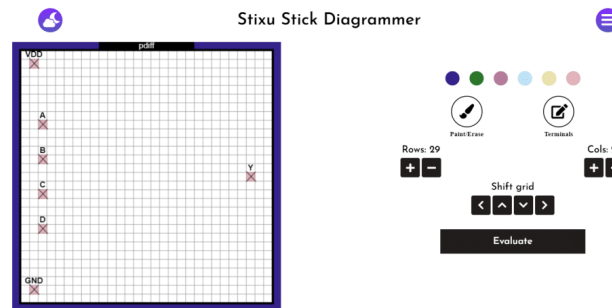


Figura 1: Interfaz gráfica Stixu

### Objetivos

- Aprender a utilizar las herramientas de Stixu.io para la creación de stick diagrams.
- Diseñar un inversor CMOS a nivel físico utilizando PMOS y NMOS.
- Entender cómo los stick diagrams ayudan a visualizar el layout de un inversor CMOS.

### Parte 1: Acceso y exploración de las herramientas de Stixu.io

- Acceder a Stixu.io : Dirígete al sitio web de [Stixu](https://stixu.io).

Figura 63. Guía 4 - Introducción a Stixu - Página 2

- Explorar el entorno de diseño : En Stixu, las herramientas principales te permiten seleccionar distintos materiales semiconductores, como N-difusión , P-difusión , polisilicio (que forma las puertas de los transistores), y las interconexiones metálicas. Además existen herramientas para dimensionar la cuadrícula donde se realiza el diseño y se pueden modificar las cantidades de entradas y salidas deseadas dentro del diseño.

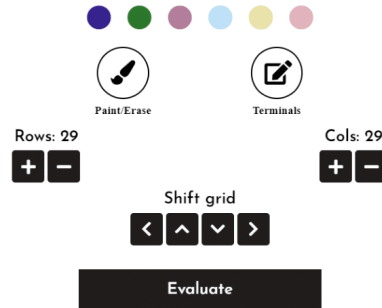


Figura 2: Herramientas para colocar materiales

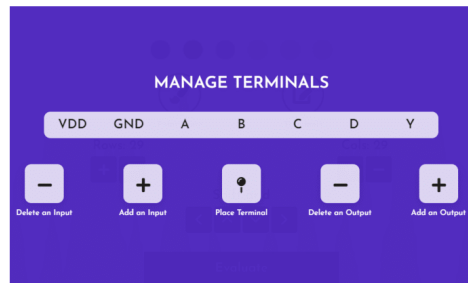


Figura 3: Herramientas para modificar entradas y salidas

## Parte 2: Creación del Inversor CMOS mediante Stick Diagrams

Un inversor CMOS se construye utilizando dos transistores: un PMOS y un NMOS . En Stixu.io, los stick diagrams te permiten visualizar cómo se distribuyen estos componentes en un diseño físico.

### Paso 1: Añadir las regiones de difusión para los transistores

- Dibuja la región de difusión tipo P (PMOS) :

### Figura 64. Guía 4 - Introducción a Stixu - Página 3

- Selecciona la herramienta de P-difusión y dibuja una región rectangular que representará el área del transistor PMOS.
- Conecta el extremo superior de esta región de difusión a VDD (la fuente del PMOS).
- Dibuja la región de difusión tipo N (NMOS) :
  - Cambia a la herramienta de N-difusión y dibuja otra región rectangular en la parte inferior de tu diseño. Esta será la región de difusión del NMOS.
  - Conecta el extremo inferior de la región de difusión a VSS (tierra).

#### Paso 2: Crear la puerta de los transistores con polisilicio

- Añadir la puerta de polisilicio :
  - Selecciona la herramienta de polisilicio y dibuja una línea horizontal que conecte las puertas del PMOS y NMOS. Esta línea representará la puerta del inversor, que estará conectada a la señal de entrada.
  - Asegúrate de que la polisilicio cruce las regiones de difusión de ambos transistores, ya que esto representa la conexión de las puertas del PMOS y NMOS a la entrada.

#### Paso 3: Conectar la salida del inversor CMOS

- Conexión de drenadores (salida) :
  - Dibuja una línea de interconexión metálica que conecte el drenador del PMOS con el drenador del NMOS. Este punto será la salida del inversor CMOS.

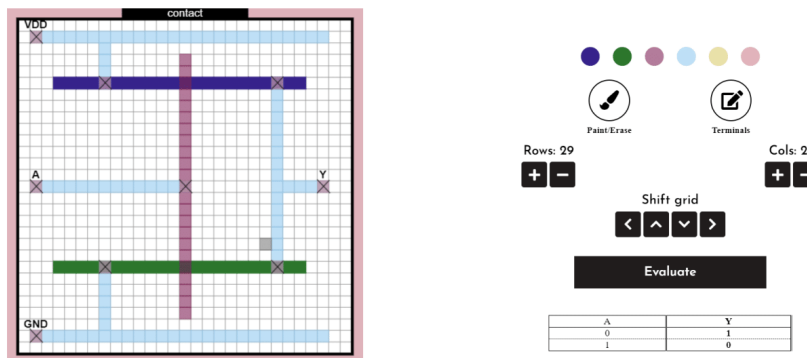


Figura 4: Inversor en Stixu

- Revisar el layout del stick diagram :
  - Asegúrate de que tu stick diagram tenga una representación clara del circuito:
    - Las puertas del PMOS y NMOS deben estar conectadas a la entrada del inversor (línea de polisilicio).
    - Los drenadores del PMOS y NMOS deben estar conectados entre sí, formando la salida.
    - El PMOS debe estar conectado a VDD y el NMOS a VSS.

### **Parte 3: Análisis del diseño y uso de Stick Diagrams**

El stick diagram es una forma simplificada de representar el diseño físico del inversor CMOS. A través de Stixu.io, este diagrama te permite visualizar cómo se distribuyen los transistores y las conexiones en el espacio físico del chip.

- Ventajas del uso de stick diagrams :
  - Ayudan a visualizar la topología del circuito antes de realizar el layout detallado.
  - Permiten simplificar el diseño inicial y prever posibles problemas de enrutamiento de señales o áreas conflictivas.
- Verificación del diseño :
  - Asegúrate de que el stick diagram cumple con las reglas básicas de diseño, como la correcta conexión de las puertas y la alineación de las regiones de difusión y las interconexiones.
  - A partir del stick diagram, se puede proceder con la creación del layout completo, respetando los tamaños físicos de cada componente.

## 12.4. Videos académicos

En la finalización del desarrollo de este trabajo, se entregó al Departamento de Ingeniería Electrónica de la Universidad del Valle de Guatemala una serie de videos académicos correspondientes a los temas tratados en las guías adjuntas en la sección anterior.

**ALU (Arithmetic logic unit):** unidad aritmética y lógica encargada de realizar operaciones matemáticas y lógicas en un procesador. 41

**ASIC (Application-specific integrated circuit):** circuito integrado diseñado para una tarea específica, en lugar de ser de propósito general. 3, 41

**Compuerta Lógica:** elemento básico en un circuito digital que realiza operaciones lógicas sobre una o más señales de entrada. 41

**Célula Estándar:** bloque de construcción básico utilizado en el diseño de circuitos integrados, que puede ser repetido para crear circuitos más grandes. 41

**Fotolitografía:** técnica utilizada para transferir patrones a la superficie de una oblea de silicio durante la fabricación de circuitos integrados. 41

**GDS2:** formato de archivo utilizado para la descripción física de un diseño de circuito integrado. 24, 41

**HDL (Hardware description language):** lenguaje de descripción de hardware utilizado para modelar circuitos electrónicos, como Verilog y VHDL. 17, 41

**MOSFET (Metal-oxide-semiconductor field-effect transistor):** tipo de transistor utilizado para amplificar o conmutar señales electrónicas. 19, 41

**OpenLane:** herramienta open-source que facilita la conversión de diseños en HDL a los archivos GDS2 necesarios para la manufactura de ASICs. 24, 41

**PCB (Printed Circuit Board):** tablero utilizado para conectar electrónicamente componentes mediante pistas conductoras grabadas en su superficie. 41

**Simulación:** proceso de modelado del comportamiento de un sistema electrónico mediante herramientas de software antes de la fabricación. 41

**Sintetización:** proceso de convertir una descripción de alto nivel en un diseño físico real de un circuito integrado. 41

**Tiny TapeOut:** proyecto educacional que permite convertir diseños digitales en circuitos integrados de aplicación específica (ASICs) de manera accesible. 41

**Verilog:** lenguaje de descripción de hardware utilizado para modelar y diseñar sistemas electrónicos. 17, 41

**Wokwi:** simulador de electrónica en línea que permite la creación y prueba de circuitos electrónicos. 41