

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



# Diseño y desarrollo de ADC cíclico con tecnología de 32 nm

Trabajo de Graduación presentado por

**Ennio Antonio Guzmán Colmenares**


para optar al grado académico de

Licenciado en Ingeniería Electrónica


Guatemala

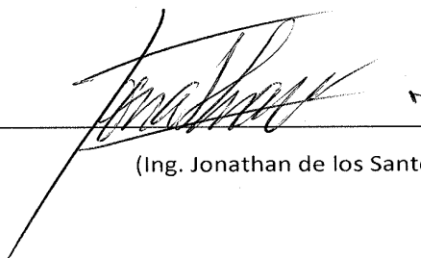
2018

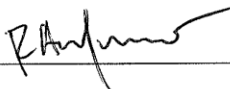
Vo. Bo. :

(f)   
\_\_\_\_\_  
(Msc. Carlos Esquit)

Tribunal Examinador:

(f)   
\_\_\_\_\_  
(Msc. Carlos Esquit)

(f)   
\_\_\_\_\_  
(Ing. Jonathan de los Santos)

(f)   
\_\_\_\_\_  
(Ing. Roberto Andrino)

Fecha de aprobación: Guatemala 14 de noviembre de 2018 / .

Diseño y desarrollo de ADC cíclico con  
tecnología de 32 nm

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



# Diseño y desarrollo de ADC cíclico con tecnología de 32 nm

Trabajo de Graduación presentado por

**Ennio Antonio Guzmán Colmenares**

para optar al grado académico de

Licenciado en Ingeniería Electrónica

Guatemala

2018

# ÍNDICE

	Página
LISTA DE GRÁFICOS .....	<b>viii</b>
LISTA DE TABLAS .....	<b>xi</b>
RESUMEN .....	<b>iv</b>

## Capítulos

I. Introducción .....	1
II. Objetivos.....	4
A. Objetivo general.....	4
B. Objetivos específicos.....	4
III. Justificación.....	5
IV. Marco teórico.....	6
A. Características de rendimiento de un ADC .....	6
1. Características generales .....	6
2. Tipos de errores en un ADC .....	10
3. Parámetros de ruido en ADCs .....	18
4. Sobre muestreo.....	21
5. Tiempos en un ADC.....	21
B. Otras arquitecturas de ADC .....	22
1. ADC Flash .....	22
2. ADC de aproximaciones sucesivas (SAR) .....	23

3.	ADC rampa o integrador .....	24
4.	ADC tipo Pipeline .....	24
C.	ADC cíclico.....	25
V.	Delimitación y alcances del diseño.....	28
VI.	Metodología.....	29
VII.	Diseño del sistema.....	32
A.	Contador digital .....	32
1.	Circuitos que componen este módulo.....	32
2.	Contador digital de tres bits .....	43
B.	Comparador digital .....	46
C.	Circuito Shift Register y memoria.....	49
D.	Comparador analógico.....	50
E.	Circuito sujetador de señales .....	54
F.	Circuito ajustador de señales .....	57
VIII.	Resultados.....	60
IX.	Discusión de resultados .....	68
X.	Conclusiones .....	74
XI.	Recomendaciones .....	75
XII.	Bibliografía.....	76
XIII.	Apéndice.....	79
A.	Especificaciones de transistores MOSFET de tecnología de 50 nm utilizados en este proyecto.....	79
B.	Especificaciones de transistores MOSFET de tecnología de 32 nm utilizados en este proyecto.....	81

C. Resultados de corrida de los 256 posibles valores de conversión.....	84
D. Programa en python para revisión resultados de ADC.....	91
XIV. Glosario .....	96

# LISTA DE GRÁFICOS

	Página
<b>Figura No.1.</b> Diagrama de bloques del ADC cíclico .....	2
<b>Figura No.2.</b> Diagrama de bloque de un ADC.....	6
<b>Figura No.3.</b> ADC de entrada única.....	7
<b>Figura No.4.</b> ADC de entrada diferencial.....	8
<b>Figura No.5.</b> Función de transferencia de un ADC ideal .....	9
<b>Figura No.6.</b> ADC perfecto de una entrada única.....	10
<b>Figura No.7.</b> Error de offset positivo.....	11
<b>Figura No.8.</b> Error de offset negativo .....	12
<b>Figura No.9.</b> Error de ganancia positivo .....	13
<b>Figura No.10.</b> Error de ganancia negativo.....	13
<b>Figura No.11.</b> Error de escala total .....	14
<b>Figura No.12.</b> No linealidad diferencial (DNL).....	15
<b>Figura No.13.</b> No linealidad integral (INL).....	16
<b>Figura No.14.</b> Ejemplo de ADC con códigos faltantes.....	17
<b>Figura No.15.</b> Espurios en una señal de entrada a 10 MHz .....	20
<b>Figura No.16.</b> Diagrama de ADC tipo Flash .....	23
<b>Figura No.17.</b> Arquitectura ADC SAR.....	24
<b>Figura No.18.</b> Diagrama de flujo de ADC cíclico .....	26
<b>Figura No.19.</b> Diagrama de bloques ADC cíclico convencional .....	27
<b>Figura No.20.</b> Diagrama de flujo de elaboración de proyecto.....	29
<b>Figura No.21.</b> Esquemática compuerta not.....	33



<b>Figura No.22.</b>	Resultados compuerta NOT en LTspice .....	34
<b>Figura No.23.</b>	Resultados de compuerta not en Hspice. En negro entra del sistema, en rojo salida del sistema. ....	34
<b>Figura No.24.</b>	Conexión de varios circuitos de entrada a través de puertas de transmisión, diagrama .....	36
<b>Figura No.25.</b>	Gráfica de ejemplo de conexión de varios circuitos de entrada a través de puertas de transmisión.....	37
<b>Figura No.26.</b>	Esquemático de circuito puerta de transmisión.....	37
<b>Figura No.27.</b>	Resultados, en verde la señal de control, en azul la señal de entrada y en rojo la señal de salida.....	38
<b>Figura No.28.</b>	Error por dimensionamiento de los transistores en puertas de transmisión, en azul señal de entrada, en rojo señal de salida.....	39
<b>Figura No.29.</b>	Correcto dimensionamiento de transistores en puertas de transmisión. En azul señal de entrada, en rojo señal de salida.....	39
<b>Figura No.30.</b>	Esquemático flip-flop tipo D.....	40
<b>Figura No.31.</b>	Esquemático de un flip flop tipo D a partir de latches.....	41
<b>Figura No.32.</b>	Gráfica de funcionamiento de un latch .....	41
<b>Figura No.33.</b>	Funcionamiento correcto de flip-flop tipo D. En rojo señal de CLK, en negro señal D y en azul señal Q.....	42
<b>Figura No.34.</b>	Funcionamiento incorrecto de flip-flop tipo D. En azul señal de CLK, en verde señal D y en rojo señal de Q.....	43
<b>Figura No.35.</b>	Circuito contador de 3 bits.....	44
<b>Figura No.36.</b>	Resultados circuito contador. En corinto señal clk, en rosado bit 0, en azul bit 1 y en rojo bit 2 .....	44
<b>Figura No.37.</b>	Resultados de circuito contador con señal de reloj desfasada .....	45

<b>Figura No.38.</b>	Concepto comparador implementado .....	46
<b>Figura No.39.</b>	Esquemático comparador digital .....	47
<b>Figura No.40.</b>	Esquemático AND de tres entradas .....	47
<b>Figura No.41.</b>	Esquemático AND de dos entradas .....	48
<b>Figura No.42.</b>	Esquemático de circuito shift register y memoria .....	50
<b>Figura No.43.</b>	Esquemático de circuito ideal de comparador analógico .....	51
<b>Figura No.44.</b>	Comparador implementando opamp con transistores de 50 nm.....	52
<b>Figura No.45.</b>	Circuito comparador usando doble inversor.....	53
<b>Figura No.46.</b>	Análisis AC inversor con error de corrimiento y pendiente en Hspice .....	54
<b>Figura No.47.</b>	Salida de (a) circuito S/H ideal y (b) circuito track and hold (T/H).....	56
<b>Figura No.48.</b>	Esquemático circuito muestreador.....	57
<b>Figura No.49.</b>	Circuito restador.....	58
<b>Figura No.50.</b>	Esquemático de circuito ajustador de señales.....	59
<b>Figura No.51.</b>	Simulación en LTspice de ADC cíclico con circuitos analógicos ideales, conversión de 800 mV @ 125 kHz.....	60
<b>Figura No.52.</b>	Simulación en LTspice de ADC cíclico con amplificadores operacionales ideales, conversión de 800 mV @ 125 kHz.....	61
<b>Figura No.53.</b>	Resultados de máxima tasa de conversión, señal de salida del circuito S/H 61	
<b>Figura No.54.</b>	Contador digital de tres bits, en rojo señal de reloj, en amarillo señales de bit 0, bit 1 y bit 2 del contador.....	62
<b>Figura No.55.</b>	Salida de comparador analógico .....	62
<b>Figura No.56.</b>	Curva DC de inversores utilizados como comparador analógico.....	63
<b>Figura No.57.</b>	Salida de comparador digital; en amarillo señales de contador; en azul señal para indicar nuevo muestreo; en rojo señal para copiar del shift register a la memoria	

.....	63
<b>Figura No.58.</b> Resultados de conversión en subcircuito memoria, siendo el LSB la gráfica de hasta arriba y el MSB la gráfica de hasta abajo. ....	64
<b>Figura No.59.</b> Consumo de potencia de circuito ADC cíclico.....	65
<b>Figura No.60.</b> Simulación de ADC cíclico implementando tecnología de 32 nm al diseño actual, señal de salida de circuito S/H .....	65
<b>Figura No.61.</b> Gráfica conversión de función aleatoria, resultados obtenidos vs resultados esperados .....	67
<b>Figura No.62.</b> Etapas de conversión en salida del S/H (café) con error por tasa de conversión inadecuada, señal de nueva conversión en morado y señal de reloj en amarillo .....	68
<b>Figura No.63.</b> Análisis DC de comparador e inversores que lo componen, en amarillo análisis DC de primer inversor; en morado análisis DC de segundo inversor; en verde análisis DC de los dos inversores en serie.....	70

# LISTA DE TABLAS

	Página
<b>Tabla No.1.</b> Tabla de verdad de compuerta NOT.....	33
<b>Tabla No.2.</b> Tabla de verdad circuito de puerta de transmisión.....	35
<b>Tabla No.3.</b> Resultados de simulación de ADC cíclico con una entrada de 800mV @ 125 Ksps .....	66
<b>Tabla No.4.</b> Resultados de corrida de los 256 valores posibles de conversión .....	84

## Listado de abreviaciones

<b>ADC</b>	Convertidor analógico a Digital (Analog to digital converter)
<b>DAC</b>	Convertidor digital a analógico (Digital to analog converter)
<b>CMOS</b>	Complementary Metal oxide Semiconductor
<b>DNL</b>	Diferencial no lineal
<b>ENOB</b>	Número efectivo de bits
<b>INL</b>	No linealidad integral
<b>LSB</b>	Bit menos significativo
<b>MOSFET</b>	Metal Oxide Semiconductor Field-Effective Transistor
<b>MSB</b>	Bit más significativo
<b>NMOS</b>	n-channel MOSFET
<b>PMOS</b>	p-channel MOSFET
<b>S/H</b>	Sample and hold (muestrador)
<b>SAR</b>	Registro sucesivo de aproximaciones
<b>SINAD</b>	Señal a ruido y distorsión de radio
<b>SNR</b>	Radio de señal a ruido
<b>THD</b>	Distorsión total armónica
<b>Opamp</b>	Amplificador operacional
<b>V<sub>in</sub></b>	Voltaje de entrada
<b>V<sub>out</sub></b>	Voltaje de salida

## RESUMEN

Los convertidores de señales de analógicas a digitales (ADC) han sido muy importantes desde el desarrollo de las computadoras, estos son fundamentales en la interconexión de una computadora con el mundo exterior, sin estos una computadora no tendría noción del entorno. Los ADC son desarrollados utilizando otros subcircuitos, como lo son los amplificadores operacionales, comparadores, flip-flops, contadores, etc., fabricados con transistores, capacitores y resistencias.

En este trabajo de graduación se presenta el desarrollo y estudio de un convertidor A/D (analógico a digital) o ADC de tipo cíclico, junto a un diseño en VLSI del ADC. Un aspecto importante de este trabajo consiste en caracterizar el rendimiento de este circuito tomando como medidas de rendimiento más importantes la potencia del circuito, tasa de conversión máxima y precisión.

La idea fundamental de un ADC cíclico consiste en comparar si la señal de entrada es mayor o menor a un valor preestablecido (en este caso 0.5V), el sistema envía un valor de un uno o cero lógico respectivamente a un circuito shift register, luego la señal se reajusta y se retroalimenta (de forma recursiva), el procedimiento es repetido, correspondiendo a un bit menos significativo, hasta alcanzarse la cantidad de veces (n veces) que se hubiese designado previamente. Al final el valor de conversión en binario pasa a una memoria en donde se retiene como salida del ADC durante el tiempo que dure la siguiente conversión.

La finalidad del trabajo consiste en el desarrollo e implementación de una topología de ADC, con una tecnología de 50 nm, que compruebe los alcances de un ADC cíclico y que pueda servir para futuras investigaciones con diferentes tecnologías o con mejoras que hagan la topología más robusta en diferentes aspectos.

El proyecto se dividió en siete módulos para abstraer y concentrarse mejor en el funcionamiento de cada bloque por individual, estos módulos son los siguientes: Circuito contador, circuito comparador de conteos, circuito sujetador de señales, circuito comparador, circuito shift register, circuito de memoria (banco de memoria) y circuito ajustador de señales.

# I. Introducción

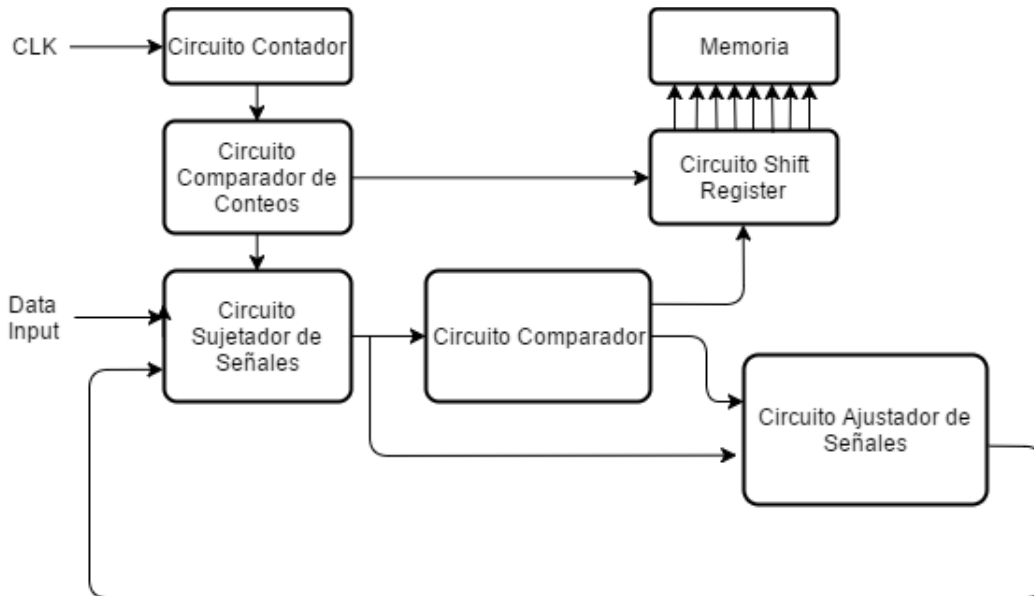
Los convertidores de señales analógicas a señales digitales ADCs actúan como interfaz entre el mundo real "analógico" y el mundo digital. Siempre serán necesarios e irán evolucionando para poder estar adaptados a las tecnologías digitales que a su vez evolucionan rápidamente. Existe una gran diversidad de tipos de ADCs, para determinar cuál implementar se observa los requerimientos de rendimiento de la aplicación, como potencia, tasa de conversión y resolución. El presente trabajo consiste en el diseño y simulación de un ADC tipo cíclico con tecnología de 50 nm, que cuente con una resolución de 8 bits.

El proyecto inició con una etapa de investigación y una serie de etapas de diseño para obtener un diseño que tuviera los requerimientos necesarios.

La primera etapa de diseño fue el desarrollo de un diagrama de bloques en donde se modulariza el proyecto y se indica el funcionamiento general de cada bloque, se expresa su modelo matemático, si lo tuviera, y se determina la interconexión de los diferentes bloques. A continuación se pueden observar los diferentes bloques del presente trabajo de graduación



**Figura No.1.** Diagrama de bloques del ADC cíclico



El sistema posee cuatro entradas, siendo estas VDD, VSS, Data input, CLK y 8 bits de salida, correspondientes al valor convertido y estando estos almacenados en la memoria. Consta de siete bloques, de los cuales se puede diferenciar que 4 son digitales y 3 son analógicos. Los bloques que conforman el ADC cíclico son los siguientes:

- **Circuito contador**, es un circuito digital, recibe la señal de reloj, (llamada CLK), una señal cuadrada que va a una cierta frecuencia que determina la tasa de conversión de cada bit del sistema. Este sistema cuenta cuantos ciclos de la señal de reloj han transcurrido durante cada conversión.
- **Circuito comparador de conteos**, es un circuito digital, busca levantar una bandera para indicar y ayudar al buen funcionamiento de otros módulos, como es el caso del circuito sujetador de señales que necesita una señal para indicar cuando se inicia la conversión de una nueva señal. Esto a partir de los valores de salida del circuito contador.
- **Circuito sujetador de señales**, es un circuito analógico, este muestrea y retiene una señal de entrada analógica por cierto tiempo definido, luego de eso vuelve a muestrear una nueva señal.
- **Circuito comparador**, es un circuito analógico, este circuito compara si la señal es

mayor o menor que cierto voltaje. Su salida es una señal digital que indica si es mayor o menor respectivamente.

- **Circuito Shift Register**, es un circuito digital. Este circuito recibe los datos digitales del comparador. Corre un bit con cada señal de reloj, para ir formando la palabra que corresponde al valor digital de la señal inicial a convertir.
- **Memoria**, es un circuito digital. Recibe la palabra del valor de la conversión del circuito shift register. Conecta la salida del ADC con otros circuitos externos.
- **Circuito Ajustador de Señales**, es un circuito analógico. Opera la señal de entrada según indique el circuito comparador para retroalimentar al circuito ajustador de señales.

## **II. Objetivos**

### **A. Objetivo general**

Desarrollar un convertidor analógico a digital tipo cíclico en VLSI utilizando una tecnología de 50 nm

### **B. Objetivos específicos**

- Caracterizar circuito ADC cíclico para obtención de métricas para tecnología de 50 nm
- Utilizar las herramientas de software de simulación Hspice para el diseño e implementación en VLSI en tecnología de 50 nm.
- Desarrollar un primer circuito VLSI analógico en el departamento de ingeniería electrónica en la universidad del Valle de Guatemala.
- Realizar un ADC cíclico de 8 bits de resolución en tecnología de 50 nm.

### **III. Justificación**

En la actualidad cualquier sistema que procese información del mundo físico requiere un convertidor analógico a digital. Estos sistemas necesitan diferentes medidas de rendimiento para satisfacer los requisitos mínimos y cumplir con una funcionalidad respectiva. El uso de estos sistemas se ha incrementado con el avance exponencial que presenta la tecnología actualmente. Estos sistemas de procesamiento se ubican en una gran serie de aplicaciones, como lo es la biomédica, aplicaciones del internet de las cosas, instrumentación industrial, robótica, etc. La alta demanda, los avances en dichas áreas y las múltiples aplicaciones en los que se necesitan estos convertidores hace que sea necesario un constante avance tecnológico en este circuito en particular.

El hecho de probar nuevas alternativas de arquitectura da lugar a encontrar particularidades inesperadas, en este proyecto se pretende buscar medidas de rendimiento de dicho circuito y de este modo poder sugerir de forma más sólida en que aplicación se aprovecharía mejor.

En la Universidad del Valle de Guatemala, en el departamento de ingeniería electrónica, se ha estado realizando investigación sobre el VLSI para diferentes tipos de aplicaciones. Sin embargo, en el diseño de circuitos analógicos en VLSI se encuentra muy poco contenido. Por lo que con este trabajo de graduación se busca colocar a Guatemala y a la Universidad del Valle de Guatemala en dicho ámbito de investigación científica.

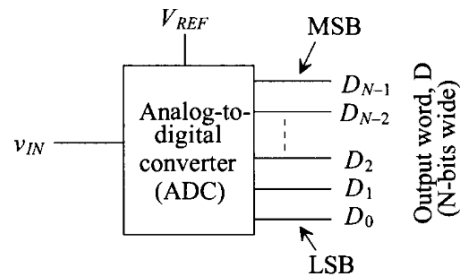
## IV. Marco teórico

### A. Características de rendimiento de un ADC

#### 1. Características generales

Un ADC es un sistema o módulo que posee una entrada analógica, un voltaje de referencia de entrada ( $V_{REF}$ ) y salidas digitales. Un ADC convierte una señal de entrada analógica en un valor digital, como salida, que representa la magnitud o valor del voltaje de entrada en comparación del voltaje de referencia ( $V_{REF}$ ). Muestra el voltaje analógico de entrada y produce un código digital como salida para cada muestra realizada.

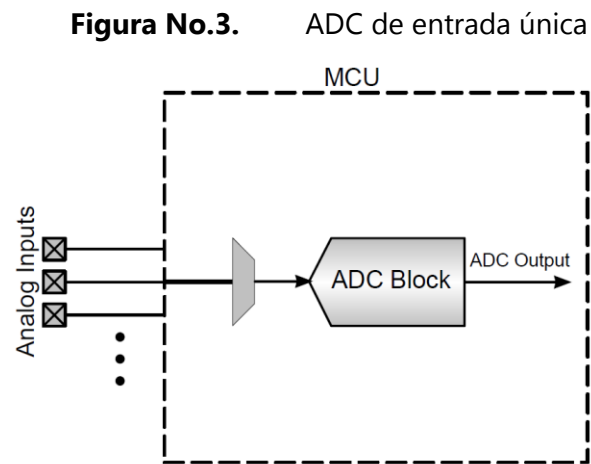
**Figura No.2.** Diagrama de bloque de un ADC



El voltaje de referencia ( $V_{REF}$ ) es un voltaje que determina el rango de conversión con el cual en voltaje de entrada analógico tiene que ser medido.  $V_{REF}$  puede ser una entrada de voltaje proveniente del exterior. Algunos ADCs son capaces de generar su propio  $V_{REF}$ .

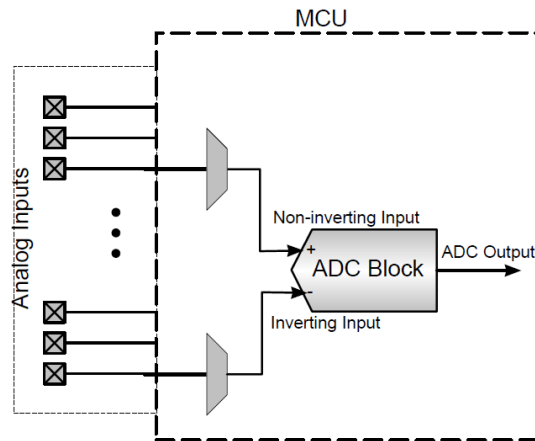
El voltaje analógico de entrada ( $V_{IN}$ ) es el voltaje por medir y convertir a un valor digital. Esta señal siempre debe ser menor o igual a  $V_{REF}$  para evitar que se sature el ADC. El rango de voltaje es llamado rango de conversión. Dependiendo del número de entradas analógicas se puede clasificar a los ADCs como de entrada única o entrada diferencial.

En un ADC de entrada única, el muestreo y la conversión pueden ser realizados únicamente en una señal de analógica a la vez. Se muestra esto en el siguiente diagrama de bloques en la Figura No.3.



En el de entrada diferencial, la diferencia en voltaje de las dos (o más) entradas analógicas es procesado en el módulo del ADC. Esto puede ser realizado directamente o después de una etapa de amplificación. Por ejemplo, en el caso dos entradas analógicas,  $A_{in1} = 1.5\text{ V}$  y  $A_{in2} = 0.7\text{ V}$ , el valor diferencial corresponde a  $0.8\text{ V}$  ( $A_{in1} - A_{in2} = 0.8\text{ V}$ ), el cual es posteriormente convertido a un valor digital.

**Figura No.4.** ADC de entrada diferencial



La mayoría de ADCs pueden operar solamente usando voltajes positivos de entrada. Los ADCs que permiten rangos positivos, ( $0 < A_{in} < V_{REF}$ ) son llamados ADCs unipolares. Los ADCs capaces de aceptar voltajes tanto positivos como negativos son llamados ADCs bipolares. Por ejemplo, en un ADC unipolar, si el  $V_{REF} = 2V$ , entonces  $0 < A_{in} < 2V$ . En este proyecto se desarrolla un ADC unipolar de entrada única.

El rango total de entrada ( $0V$  a  $V_{REF}$ ) está dividido en un número de sub-rangos llamados estados o pasos. Un estado es llamado LSB (bit menos significativo, least significant bit), el número de estados es definido por  $2^N$ . Donde  $N$  es llamado la resolución del ADC. Para un  $V_{REF}$  específico, el estado es determinado por la resolución ( $\frac{V_{REF}}{2^N}$ ). Por ejemplo, un ADC con resolución de 3 bits y un  $V_{REF} = 10V$ , tiene un número total de 8 estados y el tamaño de cada estado es de 1.25.

La cuantificación es un proceso donde la señal de entrada analógica muestreada va a ser reemplazada, empleando una aproximación, por un número finito de valores discretos. Se hace un redondeo de la señal. El LSB se determina si la tensión analógica de entrada se encuentra en el escalón más bajo del rango de tensión de entrada.

Por ejemplo, cuando  $V_{REF} = 2V$ , la resolución ( $N$ ) = 3 bits, el rango total es dividido en ocho

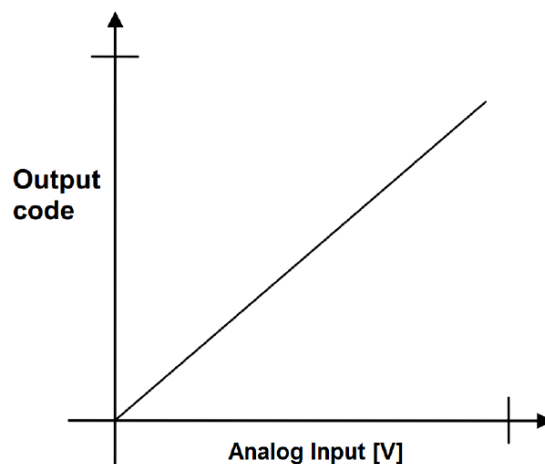
pasos. La entrada analógica de 0V a 250 mV es asignada a la misma salida digital con el código 000 y el voltaje de 251 mV es asignado a 001 y así sucesivamente con los otros valores digitales. Este proceso se llama cuantificación.

El paso de cuantificación ( $q$ ) está dado de la siguiente forma:

$$q = \frac{V_{REF}}{2^N} = LSB$$

Cuando la resolución de un ADC específico es infinita, se le dice que es un ADC ideal. En otras palabras, la resolución de un ADC ideal es igual al número efectivo de bits (ENOB). En un ADC ideal, todos los posibles valores de entrada proveen una única salida digital del ADC con un rango específico de conversión. Esto es considerado un concepto teórico que no puede ser realizado. Un ADC ideal puede ser descrito matemáticamente usando una función de transferencia lineal, como se muestra en la Figura No.6.

**Figura No.5.** Función de transferencia de un ADC ideal

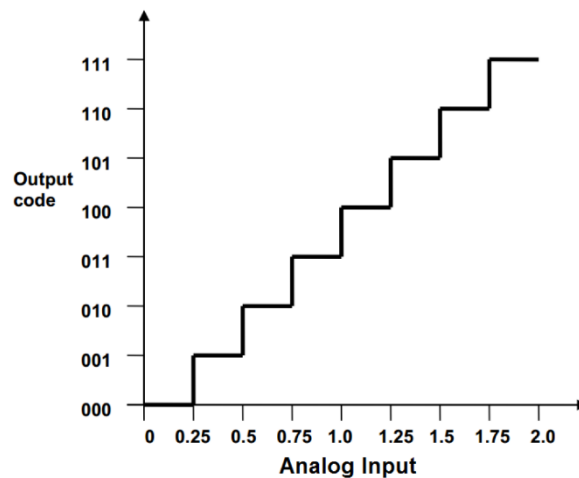




Un ADC genera una salida digital, diciendo eso, se puede decir que no le es posible generar una salida continua de valores, como en el caso del ADC ideal.

El ADC perfecto (por definición) realiza el proceso de cuantificación de forma exacta (perfecta). Esto se refiere a una función tipo escalera como función de transferencia, donde cada etapa es un LSB sin ningún tipo de error.

**Figura No.6.** ADC perfecto de una entrada única



## 2. Tipos de errores en un ADC

Para poder entender de mejor forma el rendimiento de un ADC, es necesario comprender como este es afectado por varios tipos de errores en su salida. A continuación, se explicarán varios de estos.

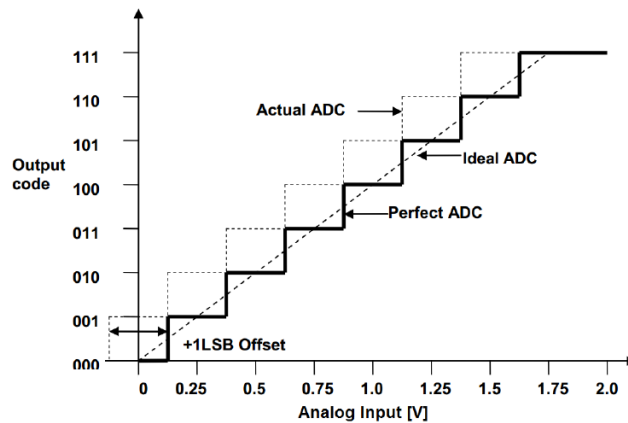
### a. Error de offset (compensación)

El error por compensación u offset es definido como la variación que existen entre la función de transferencia del actual ADC y el ADC perfecto en el punto cero, medido en

el bit LSB. Este puede ser medido en LSB o en voltios. Cuando la transición del valor de salida de 0 a 1 no ocurren en un valor de entrada de 0.5 LSB, entonces se dice que existe un error de compensación (u offset).

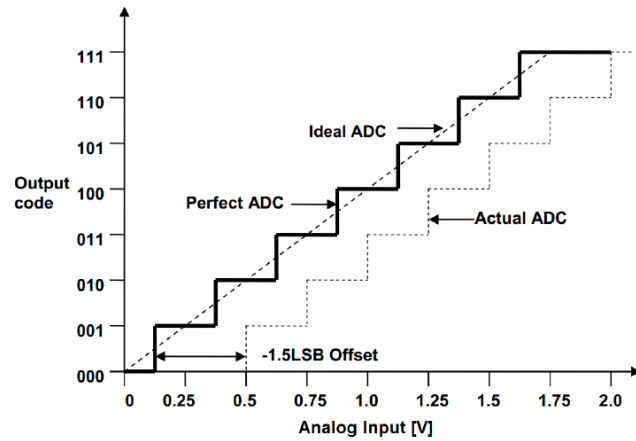
Con un error por compensación positivo, el valor de salida es mayor que 0 cuando la entrada es menor que 0.5 LSB. Con un error por compensación negativo, la entrada es mayor que 0.5 LSB cuando el primer valor de salida ocurre. Se puede explicar de forma más sencilla diciendo que si la función de transferencia actual está por debajo de la del ADC ideal, es un error por compensación negativo, y vice versa. Esto se puede observar mejor en las siguientes gráficas.

**Figura No.7.** Error de offset positivo



En la Figura No.7 se puede observar como la primera transición ocurre en 0.5 LSB y la transición es del valor 001 al valor 010 (de 1 a 2). Esta transición debería de ocurrir en 1.5 LSB para el caso del ADC perfecto. La diferencia (Perfecto – Actual = 1.5LSB – 0.5LSB = +1LSB) es el error por compensación.

**Figura No.8.** Error de offset negativo

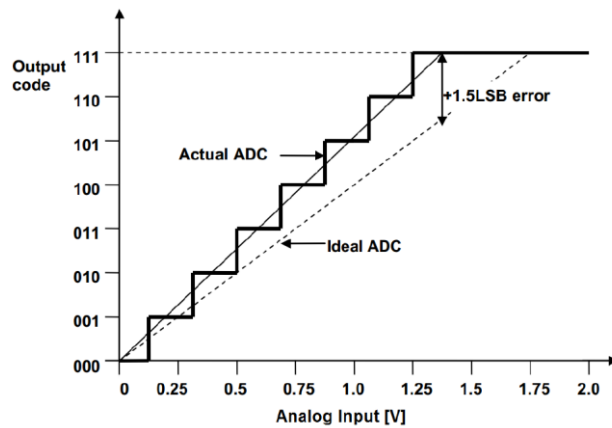


En la Figura No.8 se puede observar que la primera transición ocurre en 2 LSB y esta es del valor 000 al valor 001 (de 0 a 1). Esta transición debería de ocurrir en 0.5 LSB para el caso perfecto. Entonces la diferencia (Perfecto – Actual = 0.5 LSB – 2 LSB = -1.5 LSB) es el error por compensación.

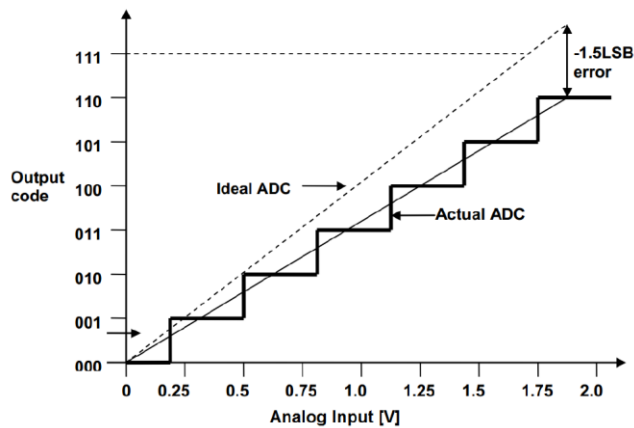
### **b. Error de ganancia**

El error de ganancia es definido por la desviación del punto medio del último paso, del actual ADC, con respecto al punto medio del último paso del ADC ideal, después de haber hecho una compensación por el error de offset. Luego de haber descartado el error por offset, introduciendo un voltaje de 0V siempre va a devolver una salida de 000. El error de ganancia causa que la pendiente de la actual función de transferencia se desvíe de la pendiente del ADC ideal. Este error puede ser medido y compensado usando varios valores de salida.

**Figura No.9.** Error de ganancia positivo



**Figura No.10.** Error de ganancia negativo



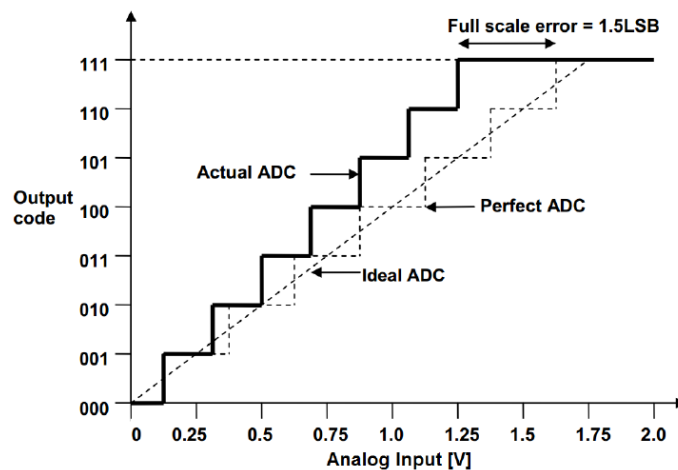
Si la función de transferencia del ADC actual ocurre debajo de la ideal, entonces se produce un error de ganancia positivo y vice versa. El error de ganancia es calculado en términos de LSB de la línea vertical que se produce entre la pendiente ideal y la actual en el último paso de ambas, como se puede observar en la Figura No.9 y Figura No.10.

En la Figura No.9 se puede observar como el valor de salida se satura antes de que la entrada encuentre su valor máximo.

### c. Error de escala total

Este error es una desviación de la última transición del actual ADC con respecto a la transición del ADC perfecto. Con transición se refiere al tamaño del paso, del cual varía de LSB. Esta medido en términos de LSB o voltios. El error de escala total se debe al error de offset y al de ganancia.

**Figura No.11.** Error de escala total



### d. No linealidad

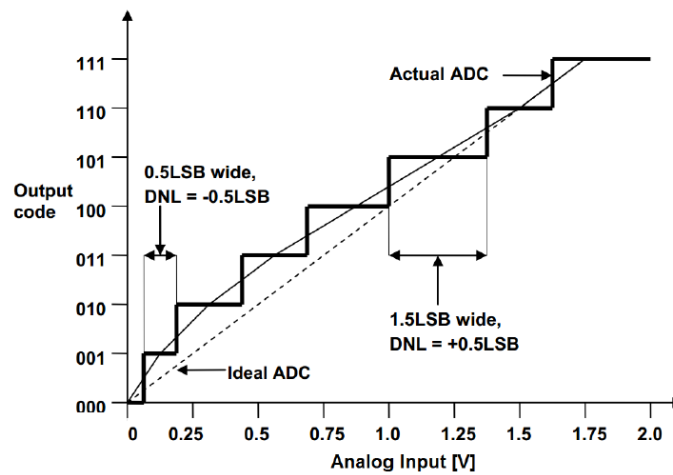
Cuando los errores de offset y ganancia son compensados por completo, la actual función de transferencia debería ser igual a la función de transferencia de un ADC perfecto. Sin embargo, la no linealidad de un ADC podría causar que la actual curva se desviara de la curva perfecta, aun si las dos curvas son iguales en cierto punto donde el error de ganancia es medido.

En general existen dos tipos de no linealidades que degradan el rendimiento de un ADC. Estos son la no linealidad diferencial (DNL, differential non-linearity) y la no linealidad integral (INL, integral non-linearity).

- **No linealidad diferencial (DNL)**

La no linealidad diferencial es definida como la máxima y la mínima diferencia de anchura de un paso entre la actual función de transferencia y la función de transferencia del ADC perfecto. Las no linealidades producen que en la cuantificación de los pasos varié el ancho de estos.

**Figura No.12.** No linealidad diferencial (DNL)



Para el caso del ADC perfecto, el ancho del paso es de 1 LSB. Sin embargo, en un ADC con DNL se puede observar como el ancho de los pasos no es exactamente 1 LSB. En la Figura No.12, en el máximo caso el ancho de un paso llega a ser de 1.5 LSB, para el valor de salida de 101, el DNL en este caso sería de 0.5 LSB. En el caso mínimo, opuesto, el ancho del paso es de 0.5 LSB, para el valor de salida de 001, aquí el DNL es de -0.5 LSB. Por lo que se puede decir de la función de transferencia de la Figura No. 12 que su DNL es de  $\pm 0.5$  LSB.

Se puede usar la siguiente ecuación para determinar cada DNL

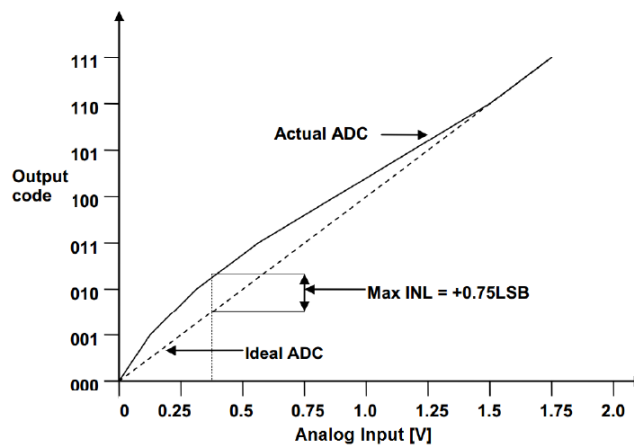
$$DNL = (D_{cx} - 1)LSBs$$

Donde,  $D_{cx}$  es el tamaño actual de cada paso en LSBs.

- **No linealidad integral (INL)**

La no linealidad integral es definida como la diferencia máxima vertical entre la curva actual y la ideal, lo cual indica que tanto se ha desviado la curva actual de la ideal. El INL puede ser interpretado como la suma de los DNLs. Por ejemplo, en la Figura No.13 se puede observar que existen múltiples DNLs que generan un INL total positivo. Un INL negativo indica a su vez que está por debajo de la curva ideal. Esto significa que la distorsión de los múltiples DNLs determina la linealidad integral del ADC.

**Figura No.13.** No linealidad integral (INL)

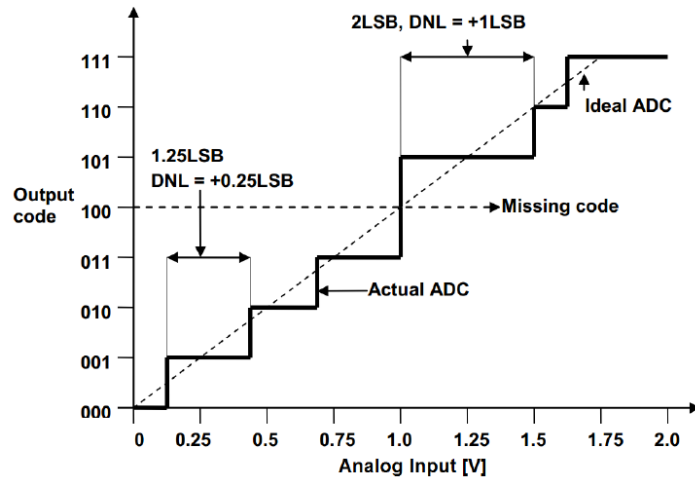


El INL puede ser medido conectando los puntos medios de los pasos de salida de la curva actual y encontrando la desviación máxima de la curva ideal en términos de LSBs. En la Figura No.13 se puede observar como el INL es de 0.75 LSB.

- **Falta de códigos**

Existen casos en los que no le es posible al ADC generar en su salida algunos códigos, que se dice que son códigos faltantes, esto se puede observar en la mejor en la Figura No.14.

**Figura No.14.** Ejemplo de ADC con códigos faltantes



En la Figura No. 14 se puede observar lo siguiente, el primer código (de 000 a 001) ocurre en los valores de entrada de 250 mV, esto es como lo esperado. En la segunda transición, de 001 a 010 existen entradas que cambian 1.25 LSB, esto es 0.25 LSB más de lo esperado (un DNL de 0.25 LSB). En el siguiente cambio de entrada, en la tercera transición, ocurre que el valor digital de salida se mantiene constante cuando el valor de entrada cambia de 1000 mV a 1500 mV y el código 100 nunca aparece en la salida. Se dice que se pierde ese código.

Los ADCs de alta resolución en varias ocasiones pierden códigos. Un ADC con un DNL menor de  $\pm 1$  LSB garantiza que no se pierdan códigos.

**e. Error total de desajuste (TUE)**

El error total de desajuste representa la máxima desviación entre la función de transferencia del ADC actual y la ideal. Es decir, incluye los errores de cuantificación, offset, ganancia y no linealidad. Este es expresado en LSBs.

$$TUE = \sqrt{e_{offset}^2 + e_{ganancia}^2 + e_{DNL}^2 + e_{INL}^2}$$



### 3. Parámetros de ruido en ADCs

#### a. Relación señal a ruido (SNR)

SNR es definido como la relación entre niveles de voltaje, en RMS, de la señal de salida con respecto del ruido que esta posee. Este es medido en dB. La siguiente fórmula describe la ecuación general para obtener el SNR:

$$SNR(dB) = 20 \log \left( \frac{V_{RMS(Signal)}}{V_{RMS(Noise)}} \right)$$

Además, el valor de SNR de un ADC perfecto es calculado usando la siguiente formula:

$$SNR(dB) = 6.02N + 1.76 (dB)$$

Donde N es la resolución del ADC.

Por ejemplo, un ADC perfecto de 10 bits tendría un SNR de 62 dB. En aplicaciones prácticas, en función de obtener un mayor rendimiento, el SNR del ADC debe ser mayor.

#### b. Distorsión armónica total (THD)

Siempre que una señal de entrada con una frecuencia particular pasa por un sistema no lineal contenido adicional es agregado a los armónicos de la frecuencia original. Por ejemplo, si una señal de entrada contuviera una frecuencia f, los armónicos de esa frecuencia serían 2f, 3f, 4f, etc. El sistema no lineal en el convertidor produciría armónicos que no estuvieran presentes en la señal original. Estas frecuencias armónicas usualmente distorsionan la señal de salida por lo cual se degrada el rendimiento del sistema. El termino de distorsión armónica total describe este efecto.

El THD es definido como la relación de la suma de las potencias de las frecuencias armónicas y la potencia de la frecuencia fundamental, como se puede observar a continuación.

$$THD = \frac{\sum \text{Potencia de los armónicos}}{\text{Potencia de la frecuencia fundamental}} = \frac{P_1 + P_2 + P_3 + \dots + P_N}{P_0}$$

En términos de voltaje, en RMS, el THD es de la siguiente forma.

$$THD = \frac{\sqrt{V_1^2 + V_2^2 + \dots + V_n^2}}{V_0}$$

El THD incrementa con el incremento en frecuencia, de la señal de entrada. Mientras la amplitud de la señal de entrada incrementa, la distorsión incrementa. Para obtener mejor rendimiento, es necesario que el valor de THD sea menor.

**c. Distorsión señal a ruido (SINAD)**

La distorsión señal a ruido (SINAD) es una combinación de SNR y de THD. Este es definido como la relación entre el valor RMS de la amplitud de la señal y el valor RMS de todos los demás componentes espectrales, incluyendo armónicos. Contiene componentes de ruido y distorsión. Se puede determinar de la siguiente forma:

$$SINAD = -10 \log \left( 10^{-\frac{SNR}{10}} + 10^{-\frac{THD}{10}} \right)$$

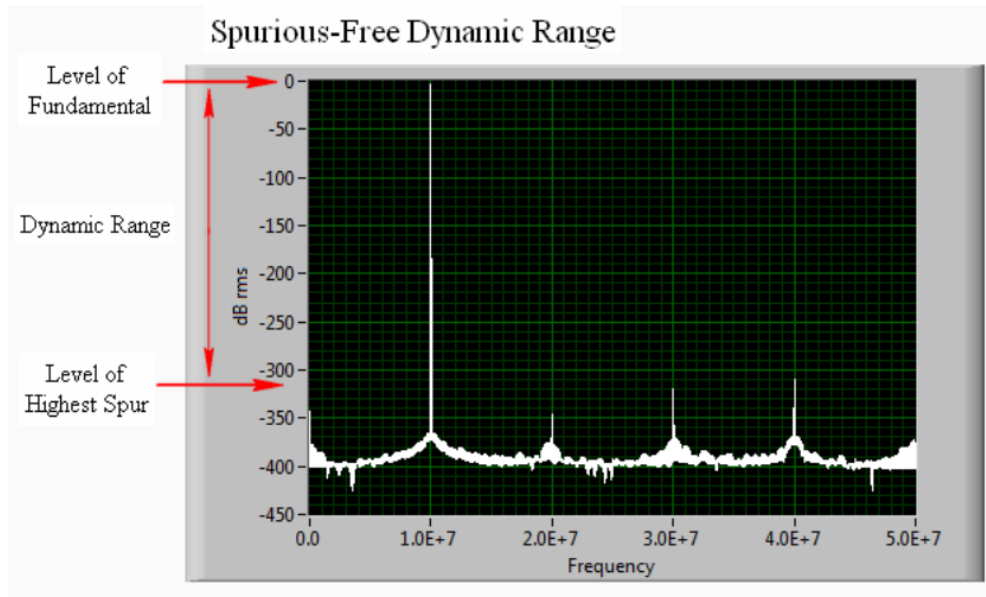
Para obtener un mayor rendimiento, el valor de SINAD de un ADC debería ser mayor.

**d. Rango dinámico libre de espurios (SFDR)**

Permite caracterizar el desempeño dinámico de una señal. El SFDR describe la relación entre amplitud de la frecuencia dominante y la amplitud de la armónica principal.

Por ejemplo, cómo se puede observar en la Figura No.15, en una señal de entrada con 10 MHz como frecuencia dominante, que hubiera componentes armónicos a 20 MHz, 30 MHz. Estos armónicos se les conoce como espurios.

**Figura No.15.** Espurios en una señal de entrada a 10 MHz



El rango dinámico entre el armónico dominante y el mayor espurio se le conoce como rango dinámico libre de espurios (SFDR).

El SFDR se puede describir con la siguiente ecuación, en dB

$$SFDR = \text{Amplitud frecuencia dominante (dB)} - \text{Amplitud de espurio más grande (dB)}$$

Del mismo modo se puede describir en términos de voltaje, en RMS.

$$SFDR = 20 \log \left( \frac{\text{Amplitud frecuencia fundamental (RMS)}}{\text{Amplitud de espurio más grande (RMS)}} \right)$$

**e. Número efectivo de bits (ENOB)**

El número efectivo de bits es el número con el cual el ADC se comporta como un ADC perfecto. Es otra forma de describir el SINAD y se describe con la siguiente ecuación

$$ENOB = \frac{SINAD - 1.76}{6.02}$$

Para tener un mejor rendimiento, el valor de ENOB de un ADC debería ser lo más cercano posible a la resolución del ADC.

#### **4. Sobre muestreo**

El sobre muestreo es un proceso que consiste en muestrear a mayor frecuencia que la frecuencia de Nyquist. Entre las diferentes ventajas que conlleva esto se mencionan las siguientes:

- Evita problemas de aliasing
- Provee un modo de incrementar la resolución de un ADC.
- El número de muestreos requiere tener un adicional de  $n$  bits es de  $2^{2n}$
- Mejora el SNR de un ADC

Por ejemplo, para implementar un conversor de 14 bits, es suficiente tener un conversor de 14 bits que puede muestrear 256 veces la frecuencia de muestreo con la que contaba previamente el ADC.

#### **5. Tiempos en un ADC**

En el momento de hacer una conversión, se caracteriza a la conversión con ciertas medidas de tiempo que son importantes, por lo cual serán mencionadas a continuación.

##### **a. Tiempo de inicio**

Contiene el tiempo mínimo que se requiere para hacer la primera conversión desde que el ADC ha sido inicializado.

##### **b. Tiempo de retención y muestreo**

Luego de iniciar la conversión, se requiere que un capacitor mantenga los valores de entrada estables en orden de poder ser convertidos posteriormente. Este es llamado tiempo de muestreo. El número de señales de reloj que requiere para convertir el voltaje muestreado se llama tiempo de retención.

### **c. Tiempo de asentamiento**

Es un tiempo observable en el conversor diferencial. Cuando múltiples canales son usados, hay escenarios donde cada canal de ADC tiene diferente ganancia y offset configurados. Cambiar entre estos canales requiere una cierta cantidad de tiempo, antes de empezar la fase de muestreo y retención, en orden de tener resultados acertados puesto si este comenzara antes, se producirían resultados erróneos en la salida del ADC.

### **d. Tiempo de conversión**

La suma de los tiempos de muestreo y retención se le conoce como tiempo de conversión. Este es representado en número de tiempos de reloj. Este es el parámetro más importante al decidir la tasa de conversión del ADC.

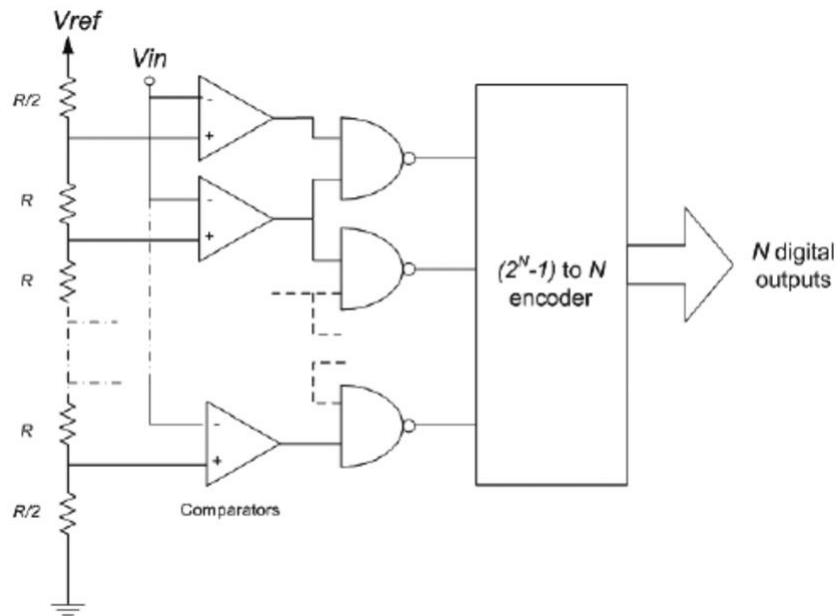
## **B. Otras arquitecturas de ADC**

En esta sección se mencionan varios tipos de arquitecturas de ADC. Se describirán las características principales haciendo un énfasis en las ventajas y desventajas de estos. Se proporcionará una caracterización general de cada tipo de arquitectura.

### **1. ADC Flash**

Este ADC es uno de los más comunes. Es la arquitectura de ADC más rápida que existe. Se basa en una serie de comparadores, como se puede apreciar en la Figura No.16, estos comparan todos los niveles de voltaje, estos a su vez pasan a un codificador que proporciona la salida de N bits.

**Figura No.16.** Diagrama de ADC tipo Flash



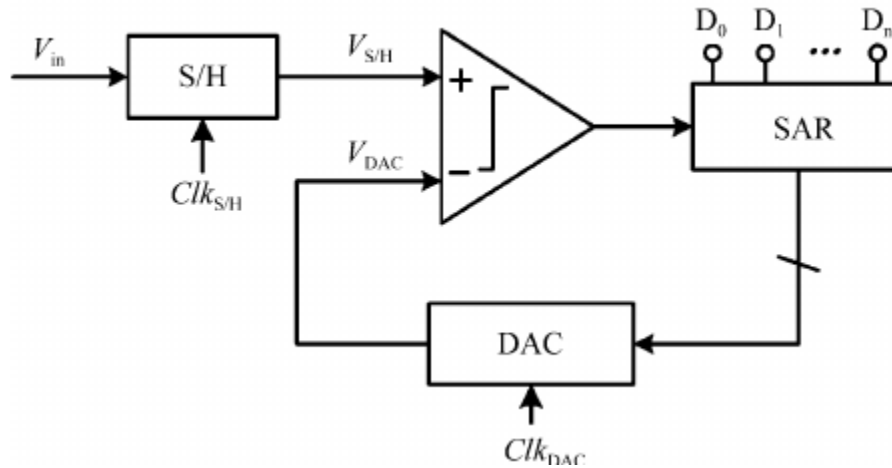
Entre las características destacadas de esta arquitectura es que solo necesita un tiempo de reloj para hacer la conversión. Por otro lado, necesita  $2^n - 1$  comparadores, lo cual hace que el circuito tenga un área superficial y consumo de potencia muy elevados.

## 2. ADC de aproximaciones sucesivas (SAR)

Esta arquitectura consiste en un ciclo cerrado que empieza con una muestra comparada, siguiendo por un shift register, este está conectado a un DAC y el DAC se compara con la señal comparada originalmente, esto se puede observar en la Figura No.17.

Para esta arquitectura es necesario  $N$  ciclos para poder llegar a hacer una conversión. El DAC que contiene incrementa considerablemente la potencia y el área superficial del circuito.

**Figura No.17.** Arquitectura ADC SAR



### 3. ADC rampa o integrador

Este tipo de arquitectura consiste en generar una señal tipo diente de sierra que tenga una amplitud de todo el rango de conversión. Seguido se compara la señal de entrada con esta señal de diente de sierra. El momento en que las dos señales son iguales, el comparador cambia su señal y se identifica el valor en digital que representa la señal. Por lo general la forma generar la señal de diente de sierra es a partir de un contador seguido de un DAC.

Esta arquitectura posee un tiempo de conversión variable, el cual puede ir de 1 ciclo hasta llegar a  $2^N$  ciclos, donde N es el número de bits del ADC.

### 4. ADC tipo Pipeline

Esta arquitectura consiste dividir la conversión por etapas secuenciales interconectadas. Todas las etapas se realizan en forma paralela, por lo que cada ciclo realiza una conversión.

### C. ADC cíclico

Se puede observar en la Figura No.18 el diagrama de flujo de un ADC cíclico convencional. Este especifica el algoritmo que implementa dicho sistema. Adicionalmente se puede observar en la Figura No.19 el diagrama de bloques del ADC cíclico. En el inicio este sistema muestrea una señal del exterior, llamada  $V(i)$  donde  $i=1$  para la primera iteración. Esta señal debe mantener ciertas especificaciones de voltaje y frecuencia para poder realizar la conversión de una forma efectiva. Una conversión tiene  $N$  etapas, se utiliza un contador que utiliza una variable, en este caso  $i$ , para saber en qué momento termina dicha conversión. Se establece un voltaje llamado  $V_{CM}$ , el cual se usa como voltaje de comparación en el comparador posteriormente, este debe de ser igual a  $V_{DD}/2$ . Otro dato importante en este diseño convencional es que la conversión llega a tener un rango de conversión de  $0\text{ V}$  a  $V_{REF}$ , donde  $V_{REF} = V_{DD}$ .

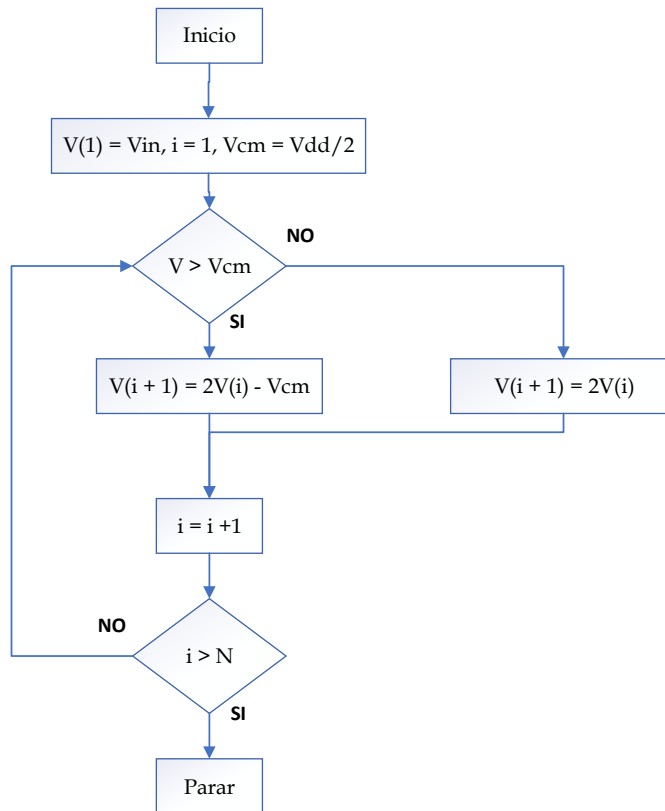
Luego de tener una señal inicial muestreada prosigue comparar si esta señal es mayor o menor a  $V_{CM}$ , implementando un comparador para esto. Si la señal de entrada del comparador es mayor que  $V_{CM}$ , entonces a la señal de salida del S/H se le resta  $V_{CM}$ . Si la señal de entrada del comparador es menor que  $V_{CM}$ , entonces no se le resta nada a la señal de salida de S/H, esto se puede observar en la Figura No.19, donde un circuito MUX determina como operar la salida del S/H. Posteriormente se multiplica por dos esta señal, para tener una señal  $V(i+1) = 2 V(i) - 2 V_{CM}$  si fuera el caso que  $V(i)$  es mayor que  $V_{CM}$  y  $V(i+1) = 2 V(i)$  si fuera el caso que  $V(i)$  fuera menor que  $V_{CM}$ . Se adiciona un valor al contador de ciclos del sistema ( $i = i + 1$ ). Luego se compara si  $i < N$ , si sigue siendo menor a  $N$  se prosigue al siguiente ciclo de conversión y se establece que  $V(i) = V(i+1)$  antes de empezar este nuevo ciclo.

De forma simultánea en cada ciclo de conversión, la señal de salida del comparador llega a un circuito shift register, donde se va corriendo cada ciclo. En el momento de terminar la totalidad de la conversión una señal le indica a este circuito que pase su

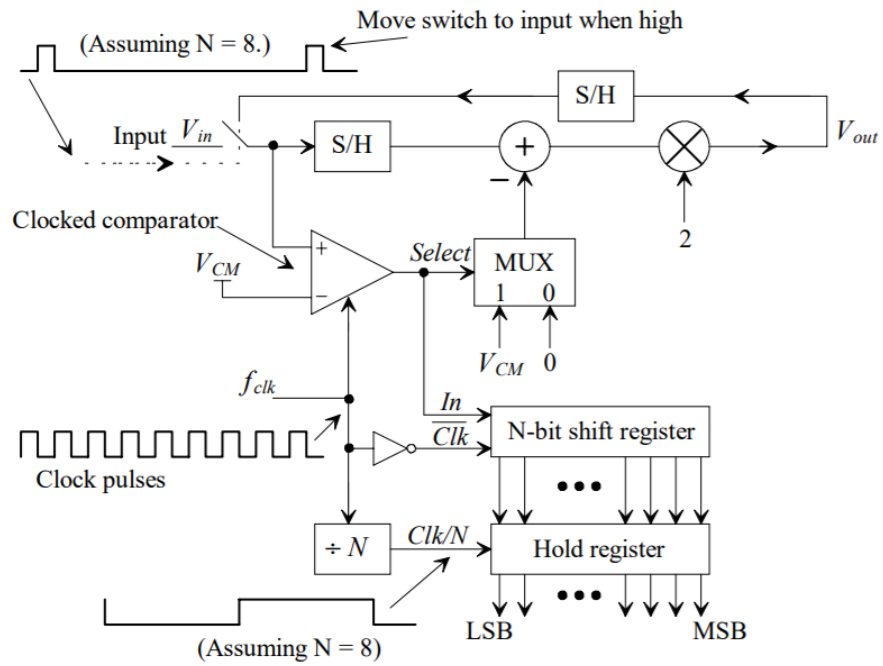


resultado a otro llamado "Hold register" o "Memoria", el cual se comunica con el exterior para dar un resultado de conversión.

**Figura No.18.** Diagrama de flujo de ADC cíclico



**Figura No.19.** Diagrama de bloques ADC cíclico convencional



## **V. Delimitación y alcances del diseño**

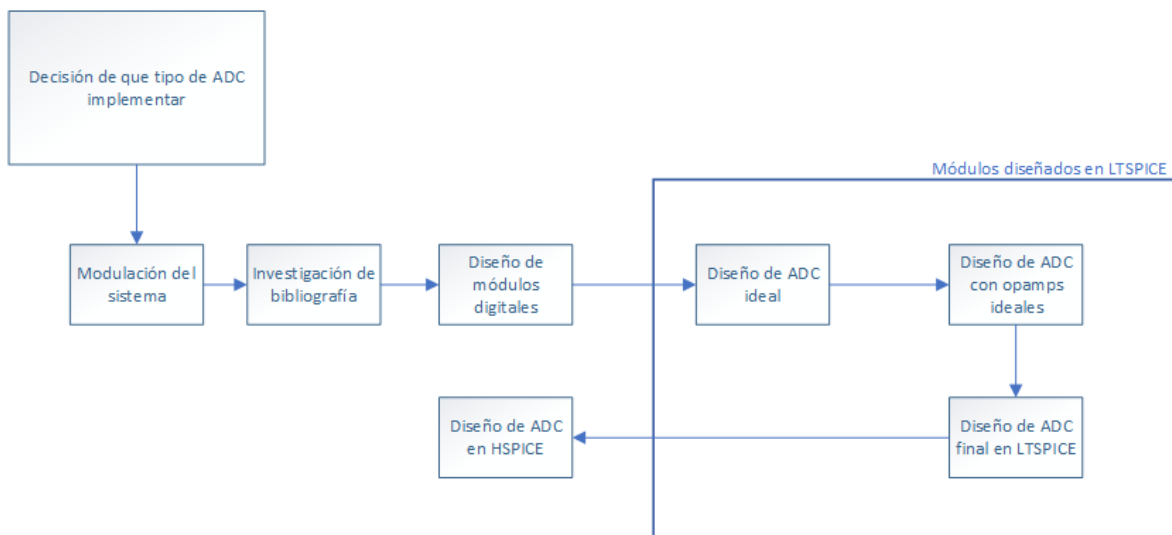
En este capítulo se tratará el tema de los alcances que se esperan obtener de este proyecto. Antes del diseño de cualquier sistema es necesario conocer los alcances que se espera conseguir. A continuación, se destacarán una serie de características que se espera que posea el presente proyecto.

Se espera comprobar la efectividad de una metodología de diseño a partir de un prototipado rápido, implementando circuitos ideales analógicos que faciliten la comprobación de la arquitectura. Para luego des abstraer módulos individuales, teniendo un sistema completo funcionando en todo momento. Diseñar en la herramienta Hspice, un ADC cíclico con tecnología de 50 nm. Otro alcance destacado consiste en diseñar un primer prototipo escalable para siguientes versiones con el objetivo de poder llegar a fabricar el circuito ADC en algún momento.

## VI. Metodología

En el presente capítulo se presentará la metodología empleada en los diferentes módulos de diseño del sistema. Para realizar el proyecto fue necesario realizar cuatro prototipos del sistema, los primeros tres prototipos fueron desarrollados en la herramienta LTspice y el cuarto prototipo fue desarrollado en la herramienta Hspice.

**Figura No.20.** Diagrama de flujo de elaboración de proyecto



Como se muestra en la Figura No.20, la primera etapa, del desarrollo de este proyecto, es la decisión de qué tipo de ADC implementar. La decisión del desarrollo de un ADC cíclico fue lo primero a tomar en cuenta en este proyecto. La elección de esta arquitectura se debe a que ya se contaba con experiencia, en proyectos anteriores, implementándola.

La siguiente etapa del desarrollo es la modulación del sistema. Con esto se refiere a que se divide el sistema submódulos, las cuales se representan en diagramas de bloques, como se puede observar la Figura No.1, está dividido en siete módulos principales, de los cuales cuatro son totalmente digitales y tres son analógicos. Es destacable mencionar que se escogieron estos módulos por su funcionalidad, abstrayendo en sí su respectivo diseño.

La investigación de la bibliografía fue de mucha ayuda para la realización de este proyecto. Con ella se pudo llegar a aclarar la forma en la cual se iba a diseñar el sistema. Del mismo modo se acaró la fiabilidad del sistema.

Posteriormente se realizó una serie de etapas de diseño de circuitos en el programa LTspice. Se buscó este programa por su escalabilidad, su precisión, fácil implementación. En la etapa inicial de diseño se desarrollaron los módulos que fueran totalmente digitales, como es el caso de los comparadores, el contador, el shift register. Esto se realizó de esa forma para luego poder tener un mayor enfoque en los módulos analógicos del sistema. Luego de completar los módulos digitales se realizó una serie de etapas para poder llegar a completar el sistema analógico, para posteriormente terminar el diseño de todo el sistema mixto. Estas etapas son las siguientes: (1) diseño de ADC ideal. (2) Diseño de ADC con opamps ideales. (3) Diseño de ADC final en LTspice, con transistores de 50 nm. (4) Diseño de ADC en Hspice, con tecnología de 50 nm.

El diseño de ADC ideal se refiere a que se implementaron componentes ideales para una elaboración más rápida precisa del sistema. También se buscaba corroborar la arquitectura de forma ideal. En esta etapa se utilizaron fuentes dependientes e independientes, switches y multiplicadores analógicos ideales.

En el momento en que ya existía un sistema completo funcionando, se buscó modificar de forma individual cada módulo analógico, de tal forma que al terminar el diseño de cada módulo el sistema mantendría las mismas entradas y salidas que en la etapa anterior. Se implementaron opamps ideales con los que contaba ya el programa LTspice. Fue necesario rediseñar completamente cada sistema con el fin de acercarse a un diseño final.

Ya con un diseño anterior terminado se cambió el enfoque a la implementación y calibración de opamps utilizando tecnología de 50 nm para reemplazar los transistores ideales utilizados anteriormente. En este proceso fue necesario, en ocasiones, cambiar el diseño anterior o recalibrar el sistema.

Después de la correcta elaboración en el software LTspice, se hace el desarrollo en el software Hspice. Este implementando los modelos respectivos de transistores a usar en el diseño final. En este proceso fue necesario volver a hacer recalibraciones del sistema para poder llegar a tener resultados similares, en exactitud, al anterior proceso.

## **VII. Diseño del sistema**

En este capítulo se presentará el proceso del diseño de cada módulo, comenzando por los módulos digitales y siguiendo por los analógicos. Es de tomar a consideración que la metodología de diseño es diferente para los módulos digitales y los módulos analógicos. Por lo que el detalle y el procedimiento de diseño que se explicará varía de la misma forma.

### **A. Contador digital**

Una de las características más importantes del ADC cíclico es que trabaja con un número determinado de etapas, por lo que es necesario tener un circuito que se encargue del conteo de dichas etapas; esta función la lleva a cabo el contador digital.

#### **1. Circuitos que componen este módulo**

Este circuito esta realizado por subcircuitos, tales como flip-flops, que a su vez han sido realizados por compuertas not y puertas de transmisión (transmission gates). A continuación, se dará una explicación más detallada de los circuitos que se implementan en este módulo.

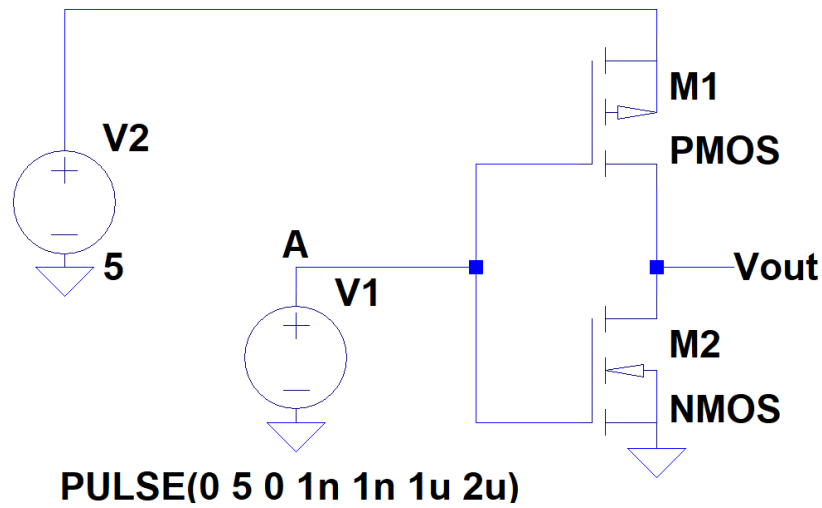
##### **a. Compuerta NOT**

Este circuito de una entrada y una salida tiene la función de negar una señal digital. De tal modo que, si en su entrada se encuentra un uno lógico, en su salida será un cero lógico o viceversa; esto se puede apreciar mejor en la Tabla No.1.

**Tabla No.1.** Tabla de verdad de compuerta NOT

INPUT	OUTPUT
A	NOT A
0	1
1	0

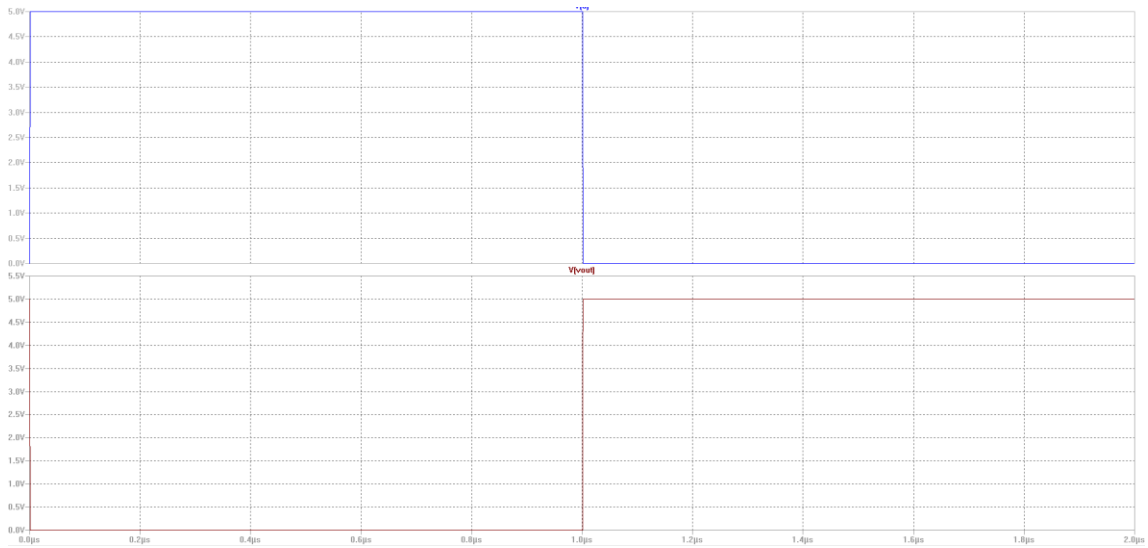
**Figura No.21.** Esquemática compuerta not



Este circuito posee dos transistores, uno tipo PMOS y otro tipo NMOS. En la Figura No.21 se puede apreciar el esquemático de la compuerta NOT. Cabe destacar que en circuitos digitales tipo CMOS es recomendable hacer el transistor pmos más grande que el nmos. Los resultados de este circuito se pueden observar en la Figura No.22.

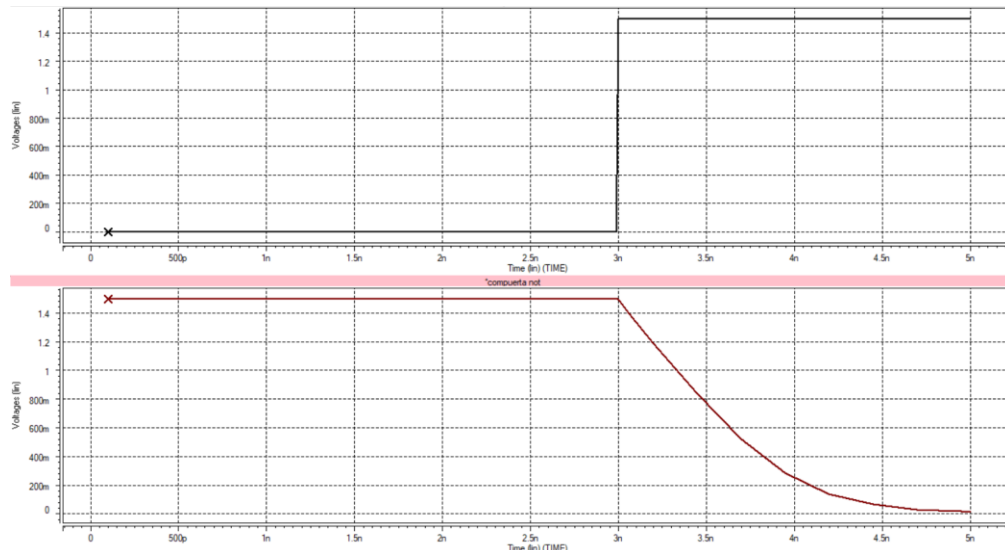


**Figura No.22.** Resultados compuerta NOT en LTspice



Estos resultados son los conseguidos con el programa LTspice. A continuación, se presenta la respectiva simulación en Hspice.

**Figura No.23.** Resultados de compuerta not en Hspice. En negro entra del sistema, en rojo salida del sistema.



En figura anterior se puede observar los resultados de la simulación de la compuerta NOT. Se destaca que los tamaños de los transistores son 200nm/50nm, para el transistor PMOS y 100n/50n para el transistor NMOS. Se puede observar en las gráficas que la entrada cambia de un cero lógico a un uno lógico en el tiempo igual a 3 nano segundos, posteriormente la señal de salida cambia, esto retardándose 2 nano segundos, siendo este retardo, para esta aplicación, lo suficientemente corto como para ser considerado despreciable.

Los demás subcircuitos que conlleven compuertas tipo not, implementarán esta configuración. El código de Hspice de la compuerta se presenta en el apéndice B1.

**b. Puerta de transmisión (transmission gate)**

Es un circuito con tres señales de entrada y una señal de salida. Sus entradas son un voltaje de entrada  $V_i$ , la señal de control y el negado de esta. En su salida puede tener dos configuraciones, la primera es una alta impedancia y la segunda es copiar el voltaje de entrada  $V_i$ . Esto se puede observar en la siguiente tabla.

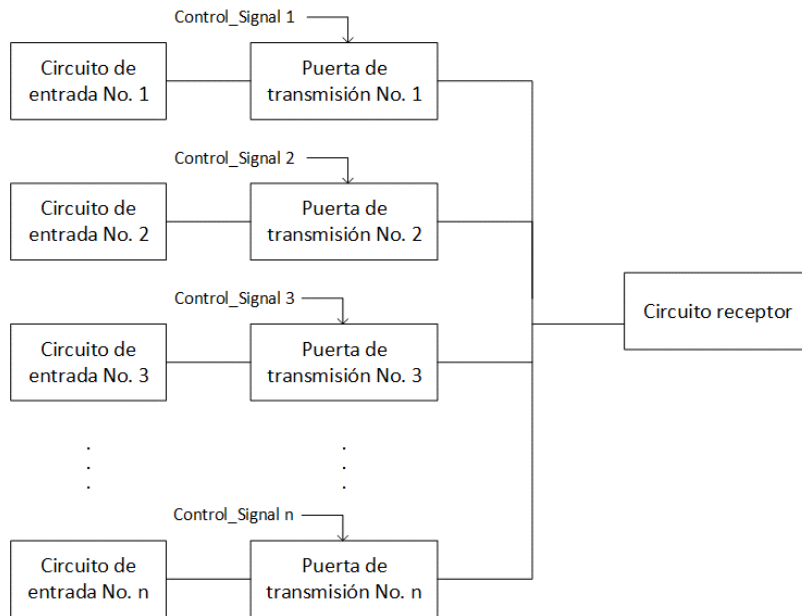
**Tabla No.2.** Tabla de verdad circuito de puerta de transmisión

CONTROL		INPUT	OUTPUT
pMOS	nMOS	IN	OUT
0	1	0	0
		1	1
1	0	0	Z
		1	

Se hace la mención que este circuito es especialmente útil para el diseño de flip-flop tipo D y para el diseño de circuitos analógicos, pues el hecho que pueda tener en ocasiones una salida con alta impedancia permite poder tener más de una entrada

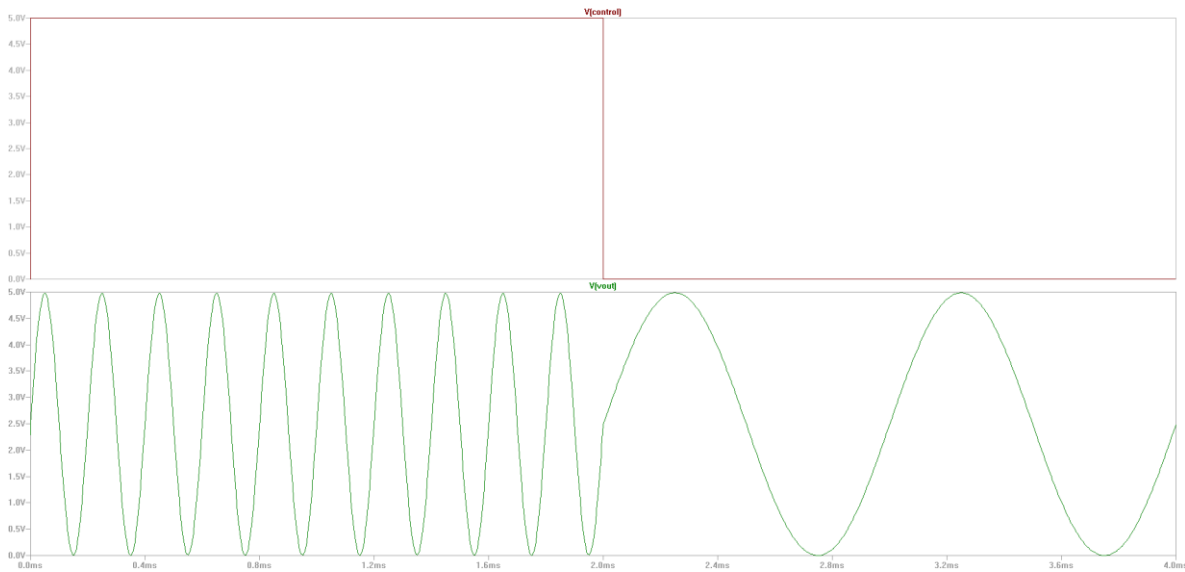
conectada en paralelo, lo que permite a un circuito recibir estados de varios circuitos, en diferentes tiempos. Esto se puede observar en la siguiente figura.

**Figura No.24.** Conexión de varios circuitos de entrada a través de puertas de transmisión, diagrama



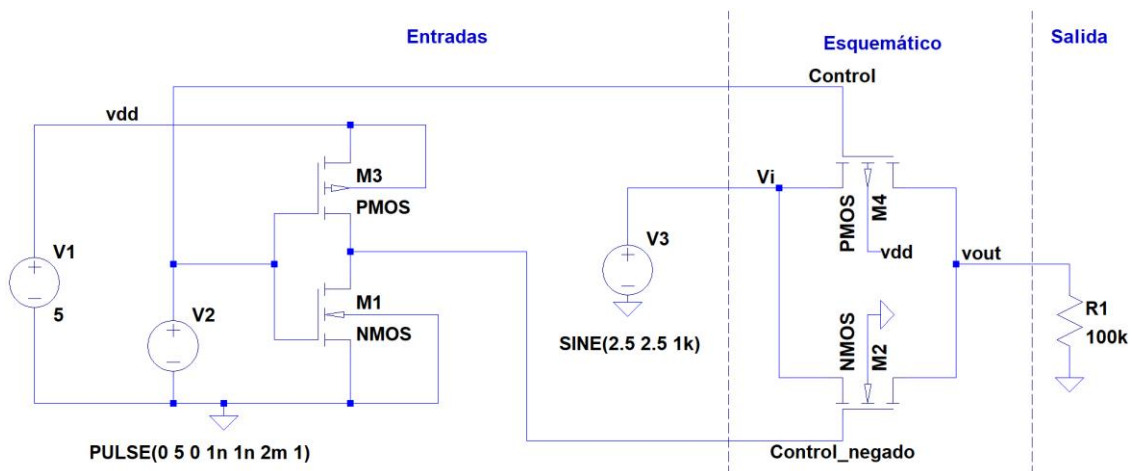
El funcionamiento de un transmission gate se puede observar en la Figura No.25, donde se conectan dos de estos, con diferentes señales de entrada, a una misma salida. En este caso fue posible usar la misma señal de control luego de negarla, pero en el caso de haber más sistemas interconectados esto no sería posible.

**Figura No.25.** Gráfica de ejemplo de conexión de varios circuitos de entrada a través puertas de transmisión.

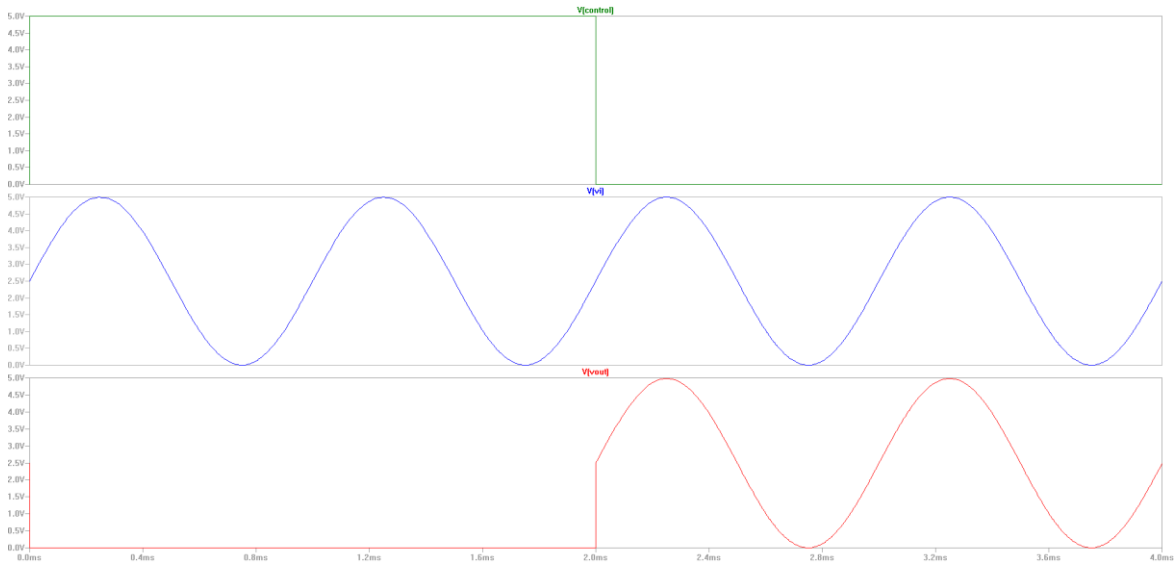


La siguiente figura se muestra el esquemático de una puerta de transmisión, junto con las respectivas entradas y su salida para hacerlo funcionar correctamente y así poder mostrar su funcionamiento como se observa en la Figura No.27.

**Figura No.26.** Esquemático de circuito puerta de transmisión

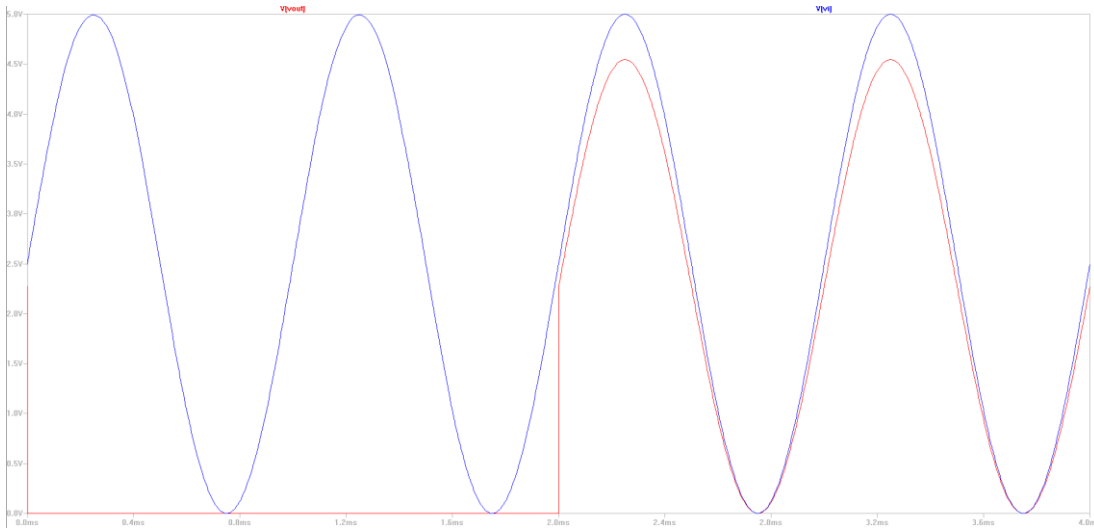


**Figura No.27.** Resultados, en verde la señal de control, en azul la señal de entrada y en rojo la señal de salida

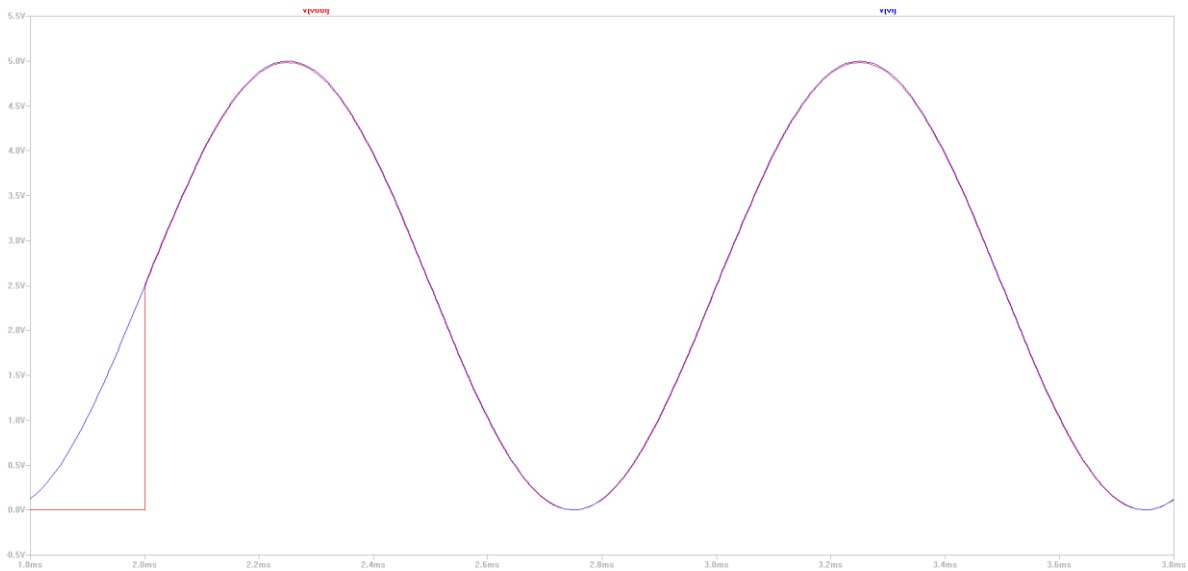


En este circuito se puede destacar la importancia del tamaño de los transistores. Pues si estos no están bien dimensionados, la salida varía de la entrada, generando un error en el sistema. La Figura No.28 muestra en una misma gráfica la señal de salida y la señal de entrada cuando los transistores tienen la misma dimensión, 32nm/32nm para todos los dos transistores utilizados. Se puede observar como los voltajes mayores no pasan correctamente, son atenuados de forma no lineal. Sin embargo, en la Figura No.29 se puede observar cómo, luego de un correcto dimensionamiento, se puede obtener una salida que se parezca a la entrada respectivamente. En este caso se incrementó el tamaño del ancho ( $w$ ) del transistor pmos en 1024nm.

**Figura No.28.** Error por dimensionamiento de los transistores en puertas de transmisión, en azul señal de entrada, en rojo señal de salida.



**Figura No.29.** Correcto dimensionamiento de transistores en puertas de transmisión. En azul señal de entrada, en rojo señal de salida.

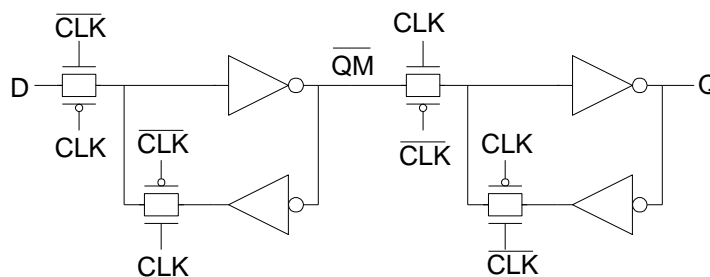


### c. Flip-flop tipo D

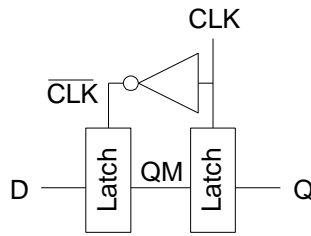
Para poder generar un conteo es necesario poder almacenar un estado actual, esto es posible con circuitos tipo flip-flops. Estos guardan un dato hasta que un parámetro de entrada, un flanco positivo, indica que vuelvan a cargar un nuevo valor. Este circuito cuenta con dos señales de entrada y dos señales de salida. Una señal de entrada, llamada D, recibe los datos a guardar en un futuro, otra entrada es la señal de reloj respectivamente, que indica cuando almacenar los datos que recibe la entrada D. Posee dos señales de salida, una es llamada Q, la otra su negado, llamada Qn. Como funciona un flip-flop tipo D es que cuando se da un flanco positivo en la señal de reloj, la salida Q copia lo que había en ese momento en la entrada D.

Este circuito se realizó implementando los circuitos mencionados anteriormente, compuertas NOT y puertas de transmisión. Específicamente 4 puertas de transmisión y 4 compuertas NOT. Existen varios diseños para hacer un flip flop tipo D, pero se optó por esta por su versatilidad y por qué cumple con los requisitos de diseño.

**Figura No.30.** Esquemático flip-flop tipo D

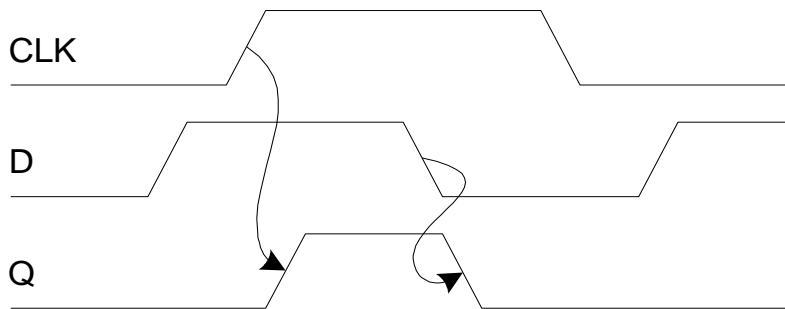


**Figura No.31.** Esquemático de un flip flop tipo D a partir de latches



Existen varios tipos de circuitos que almacenan estados, entre ellos se encuentran unos llamados latches, su funcionamiento consiste en almacenar el estado cuando está el CLK en cero lógico, cuando el CLK está en uno lógico, la salida Q copia el estado de la entrada D. Esto se puede observar en la Figura No.32. En la Figura No.31 se puede observar que se implementaron dos de estos circuitos para formar el circuito flip flop tipo D.

**Figura No.32.** Gráfica de funcionamiento de un latch

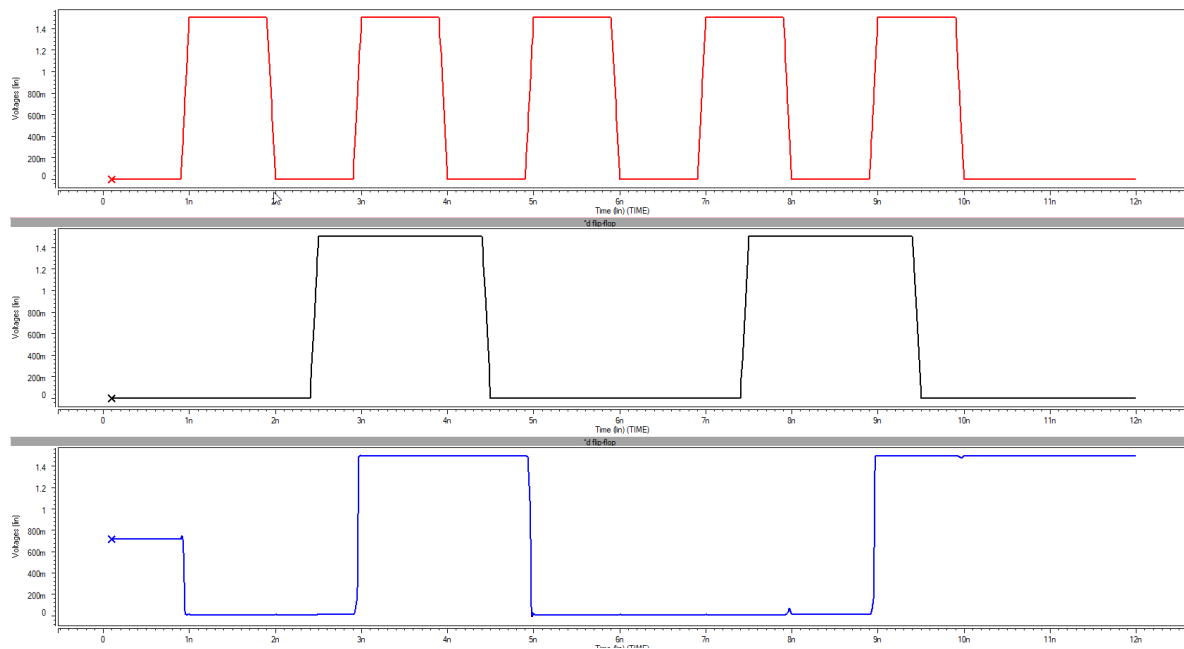


En el momento de escribir el código en Hspice para el flip flop se cometió el error de poner como vdd un voltaje de 5 voltios, esto produjo que la señal de Q se comportara de forma no debida, como se observa en la Figura No.34. Se puede observar que esta no posee cambios bien definidos, sino que tiene cambios parciales y estos se producen en el momento en que CLK y D cambian de flanco. Esto se debe a que el voltaje de  $V_{GS}$  de los transistores, que seguía siendo de 1.5 V, no llegó a saturar el transistor y entonces la salida de estos no llegó a ser el voltaje de vdd en ningún momento.

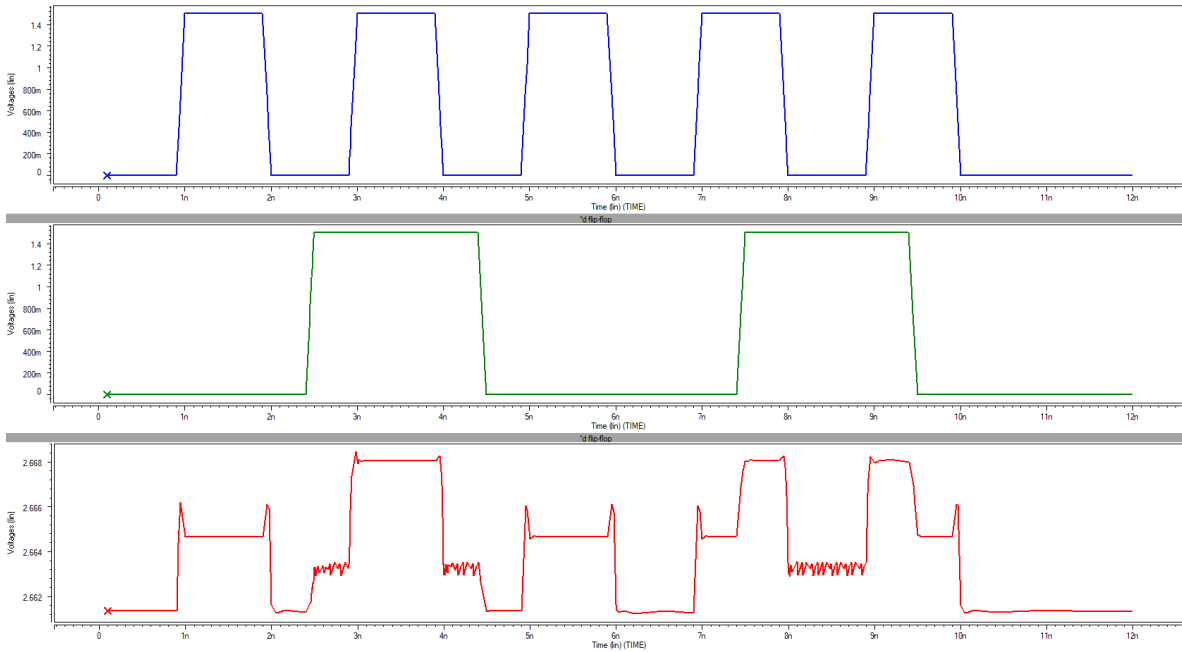


Se puede apreciar el correcto funcionamiento de circuito flip flop tipo D en la Figura No.33. Se hablarán de ciertos puntos específicos de la Figura No.33 para resaltar su correcto funcionamiento. En el tiempo de 3ns se puede observar que la señal de CLK tiene un flanco positivo, en ese momento la señal D, de entrada, estaba en un estado positivo, lo que produce que la señal de salida, Q, tenga un cambio de estado positivo. Luego de eso se puede observar en el tiempo de 5ns como la señal de CLK tiene otro flanco positivo, mientras que en ese momento la señal de entrada D se contaba con un cero lógico, entonces la señal de Q tiene un cambio de estado en ese momento y baja su señal a cero voltios

**Figura No.33.** Funcionamiento correcto de flip-flop tipo D. En rojo señal de CLK, en negro señal D y en azul señal Q.



**Figura No.34.** Funcionamiento incorrecto de flip-flop tipo D. En azul señal de CLK, en verde señal D y en rojo señal de Q.

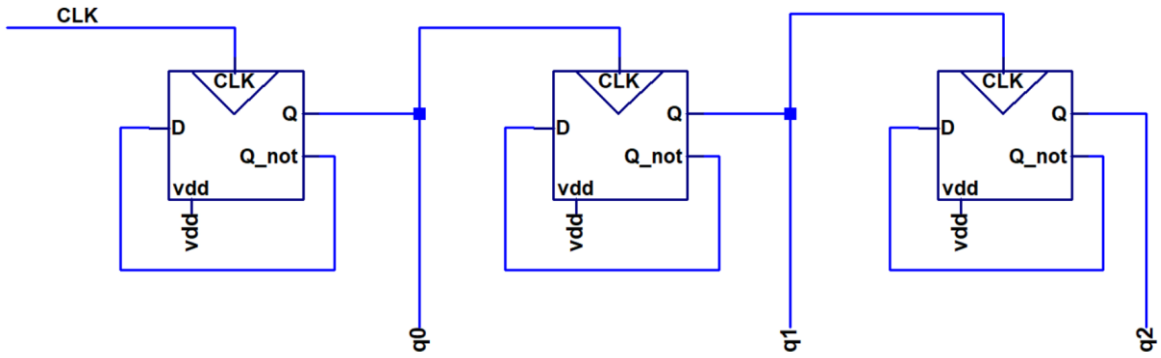


## 2. Contador digital de tres bits

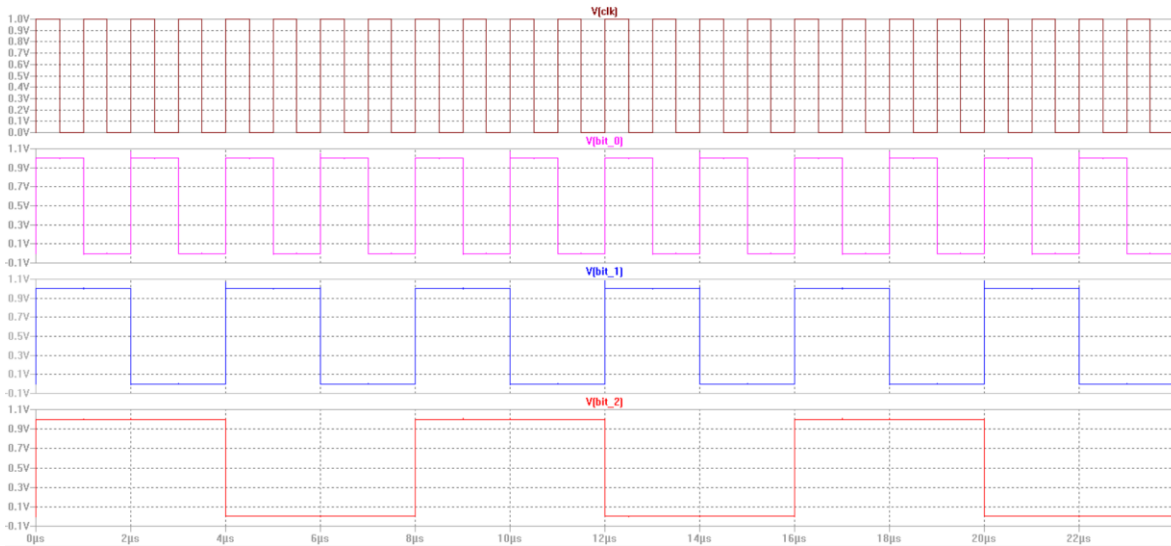
El ADC cíclico realiza la conversión a digital luego de una serie de ciclos de conversión. Estos ciclos son proporcionales a la resolución del ADC. Este ADC maneja una resolución constante a través del tiempo, por lo que es necesario tener un número de ciclos constante por conversión. Para que el circuito conozca el número de ciclos se implementó un contador digital de tres bits.

El diseño de un circuito contador de  $n$  bits, requiere de la implementación a su vez de  $n$  flip-flops. En este caso se realizó de 3 bits, por lo cual se usaron 3 flip-flops. Como se puede apreciar en la Figura No.35, este circuito consiste en dos entradas y tres salidas. Sus entradas son vdd (la alimentación, en este caso es 1V) y clk (el reloj del sistema). Sus salidas son q0, q1, q2 siendo estos los bits de conteo. Y en la Figura No.36 se puede observar las gráficas de las señales de clk y de los bits que llevan el conteo de ciclos.

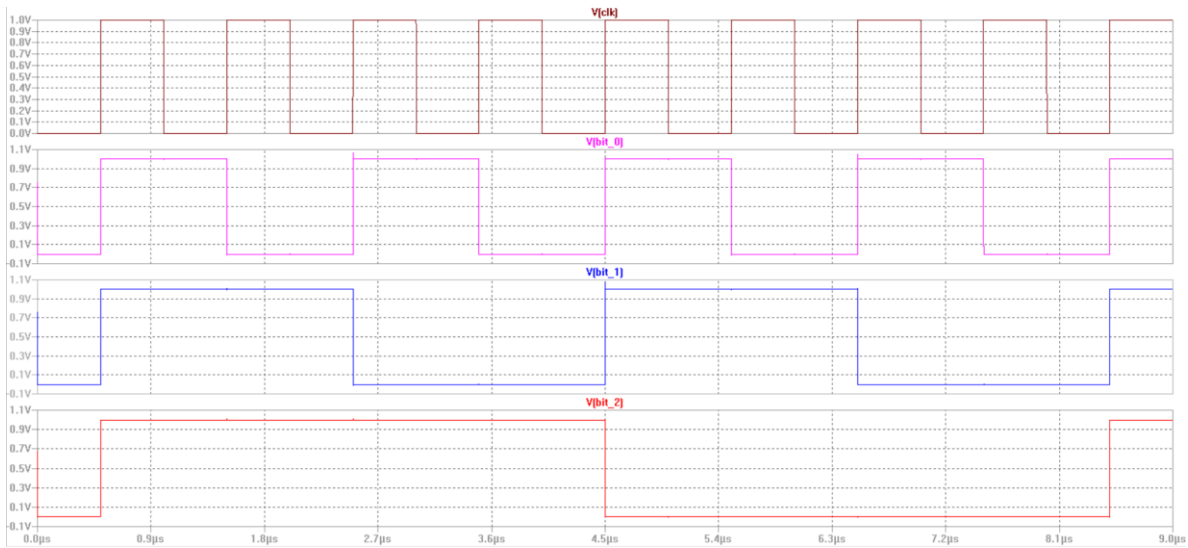
**Figura No.35.** Circuito contador de 3 bits



**Figura No.36.** Resultados circuito contador. En corinto señal clk, en rosado bit 0, en azul bit 1 y en rojo bit 2



**Figura No.37.** Resultados de circuito contador con señal de reloj desfasada



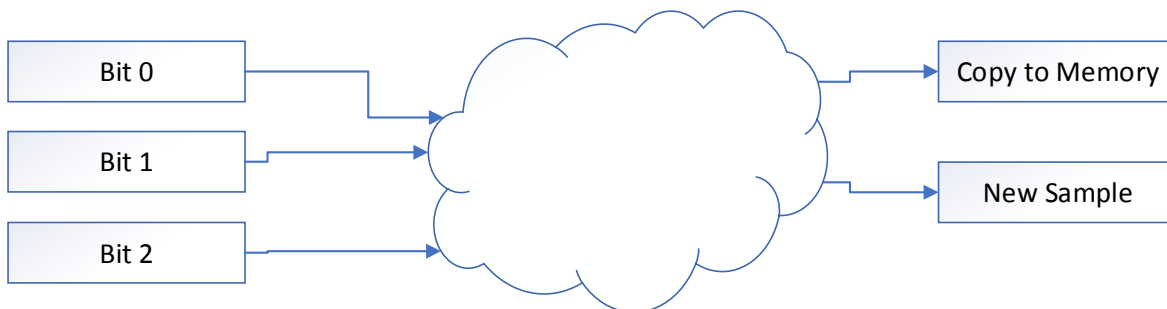
Se puede observar en la Figura No.36 que el conteo inicia en el número más alto, en este caso el número siete en binario, y que el conteo es descendente, esto debido a que al inicio todos los valores del contador se encuentran en cero (Figura No.37, con reloj desfasado para hacer notar el conteo). La entrada D del flip-flop está conectada a la salida Q\_not, como se observa en la Figura No.35, por lo que todos los bits del contador al recibir el primer flanco positivo de reloj se cambian a uno. La razón por la cual el contador va en descenso tiene que ver con la forma en que están conectados los flip-flops, el hecho que siguen el mismo patrón de estar conectados para que en flanco positivo de la señal, bit, anterior genere un cambio. Posibles soluciones para generar un conteo positivo serían negar todos los bits, o sacarlos del Q\_not del flip-flop, puesto este es el negado de la señal de salida. Otra solución sería que cambiar la lógica del segundo y tercer bit, que los flip-flops cambien en flanco bajo, esto se podría conseguir conectando la salida Q\_not al clk del siguiente flip-flop.

## B. Comparador digital

Sabiendo en qué ciclo se encuentra la conversión del ADC, se generan banderas para indicar el momento en el que ocurren ciertos eventos durante el proceso, pues el circuito no es capaz de entender qué etapa de la conversión está sucediendo. Un comparador digital corrobora el estado de las entradas y a través de lógica booleana procesa una señal que representa la realización de cierto acontecimiento esperado. Las entradas de este comparador son estados del sistema, en este caso, los bits de salida del contador, la señal de reloj y las señales de energía, vdd y gnd.

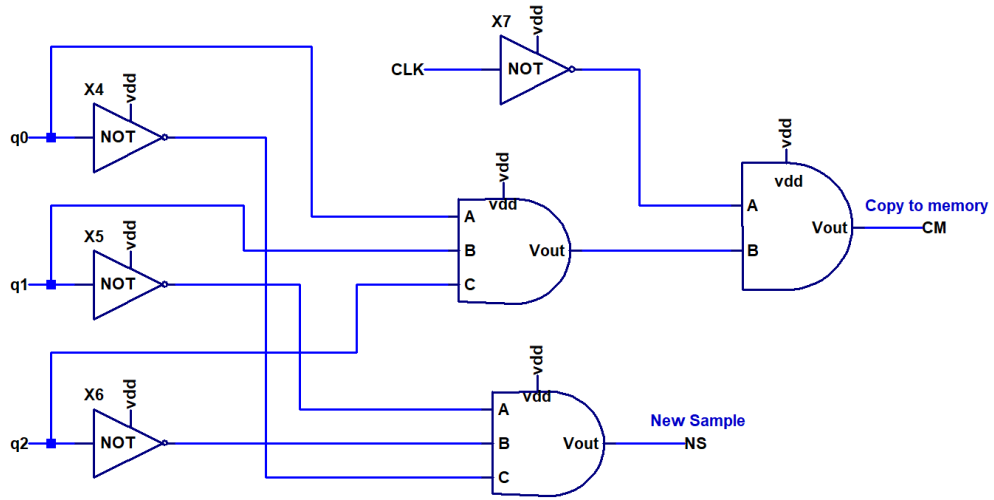
En este caso particular es necesario obtener dos estados, banderas, del comparador. La primera de ellas es la bandera que indica al shift register en qué momento ya terminó el proceso de conversión y es preciso copiar los bits de este a la memoria, llamada en este proyecto como "Copy to Memory". La segunda bandera indica cuando es momento de iniciar una nueva conversión y obtener una muestra a convertir del exterior del circuito, llamada "New sample". En la Figura No.38 se representa conceptualmente este comparador.

**Figura No.38.** Concepto comparador implementado

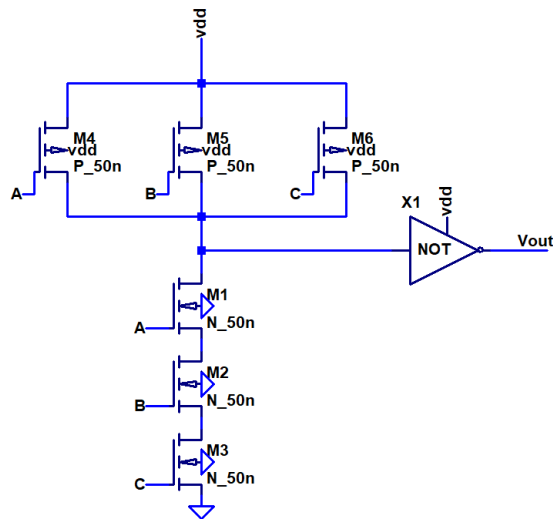


Es posible apreciar el esquemático del circuito comparador en la Figura No.39. Se puede ver que este circuito está compuesto de compuertas not, compuertas AND de tres entradas y compuertas AND de dos entradas.

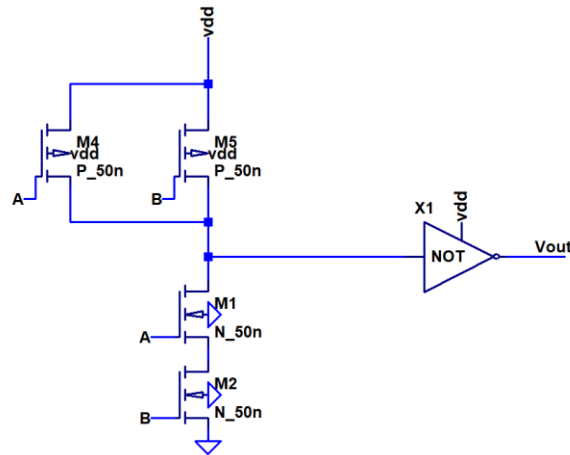
**Figura No.39.** Esquemático comparador digital



**Figura No.40.** Esquemático AND de tres entradas



**Figura No.41.** Esquemático AND de dos entradas



En la Figura No.40 y Figura No.41 se puede observar los esquemáticos de los circuitos AND de tres y dos entradas utilizados para el diseño del comparador digital.

Una de las dificultades presente en este módulo, fue definir en qué momento y duración del ciclo desplegar las banderas para el correcto funcionamiento del circuito. En el caso de la señal llamada "New sample" se definió cuando el contador tenía el valor de siete en binario y contaba con la duración de 1 ciclo. Puesto es una señal que tiene un conteo regresivo se dice que el valor siete en el contador corresponde al primer valor de esa conversión y la bandera funciona alimentando a un transmission gate, mientras esta esté levantada el transmission gate deja pasar la señal deseada al muestreador de señales y tomando en cuenta el funcionamiento de este muestreador de señales, que se explicara posteriormente, la primera señal se muestra en el valor siete del contador, para que sea procesada en el siguiente ciclo.

En el caso de la señal llamada "Copy to Memory" era necesario copiar del shift register a la memoria la señal después de terminar toda la conversión, pero antes de iniciar una nueva conversión, antes de la señal "New sample", esto corresponde al valor cero del contador. El problema que surgió entonces fue que coincidía con la última conversión del

ciclo. Por lo tanto, fue necesario atrasar la señal medio ciclo. Esta señal alimenta a un flip-flop, por lo tanto, el flanco de subida es lo importante en esta señal. Se atrasó la señal al pasar la señal de salida con el reloj del sistema a través de un and de dos entradas.

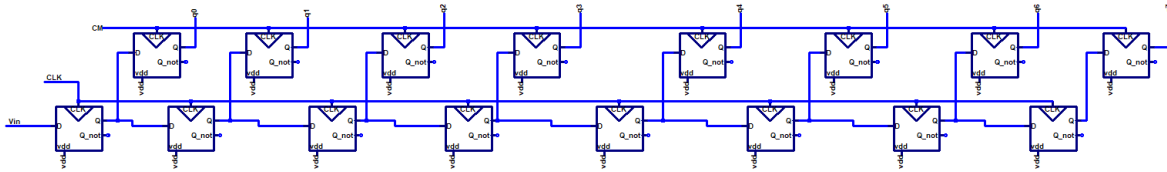
### **C. Circuito Shift Register y memoria**

El ADC de este trabajo tiene la característica de que proporciona los bits de uno en uno por cada ciclo. Luego de esto es necesario un sistema que tome esos bits de cada ciclo y luego los pase a una memoria donde se mantengan por un periodo de conversión sin cambiar. Este subcircuito es el que proporciona la salida del sistema general, hacia el exterior del circuito, brindando el resultado de la conversión luego de su correcto procesamiento. Sus entradas consisten en diferentes señales: la señal de reloj, la bandera "Copy to memory" (explicada anteriormente), la señal de salida del comparador analógico, la alimentación del sistema (vdd y gnd).

La salida del comparador analógico proporciona la información sobre la conversión de ese ciclo, correspondiente a un bit. Ese bit se guarda por un ciclo en el primer flip-flop, luego en el siguiente ciclo es movido, ese bit de información, al siguiente flip-flop y así por el número de ciclos que constituyen la conversión, en este caso ocho ciclos. En el momento que ya se ha dado la conversión una bandera, llamada "Copy to Memory" indica que toda la información que se ha mantenido en los flip-flops del shift register va a ser movida a un banco de flip-flops que componen una memoria donde se va a guardar la información de conversión por un ciclo de conversión completo. Para luego ser reemplazado por otro valor de conversión de un ciclo posterior. El circuito diseñado enteramente por flip-flops tipo D se observa en la Figura No.42.



**Figura No.42.** Esquemático de circuito shift register y memoria



En los siguientes tres incisos (D, E, F) se postularán los circuitos analógicos implementados en el sistema.

#### **D. Comparador analógico**

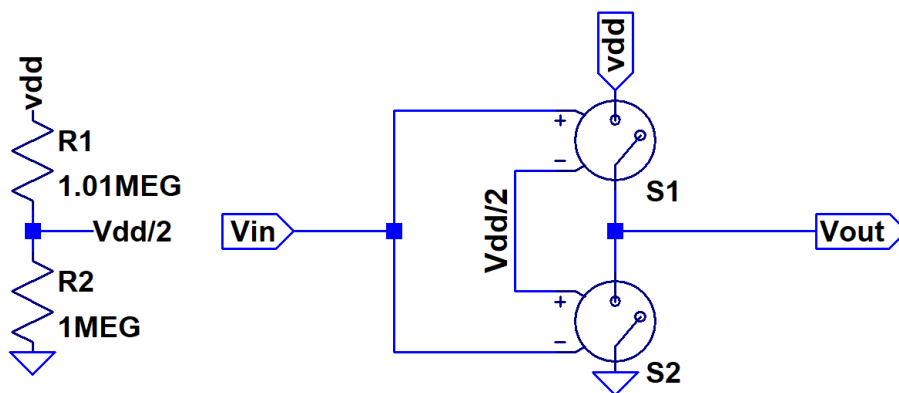
Existen varios tipos de ADC, entre los cuales destaca el tipo flash. Este ADC funciona implementando un comparador por cada rango (o estados, explicado en el capítulo anterior) de conversión. En algunos ADCs, la forma de funcionar es identificando entre todos los sub-rangos posibles el valor digital que coincida con el valor analógico de entrada, en este ejemplo se implementan varios comparadores. En otras ocasiones se implementa una forma de conversión iterativa, en la cual se reutiliza el mismo comparador varias veces. En conclusión, el circuito comparador analógico es fundamental en el diseño de convertidores analógicos a digital, independientemente de la arquitectura a utilizar.

El diseño con comparadores analógicos consiste en que se seleccione entre dos o más posibles sub-rangos donde asignar la señal de entrada. Para este trabajo, se introduce una señal que tiene un rango entre 0V y 1V. Se compara si es mayor al valor máximo de

comparación dividido dos, 0.5V. Si es mayor que este valor el sistema devuelve un valor positivo en binario, correspondiente a 1V. en caso contrario, el sistema devuelve un valor de negativo en binario, que corresponde a 0V. Por lo tanto, la entrada de este sistema es el valor por comparar, el cual viene de la salida del circuito muestreador. La salida del sistema corresponde al valor en binario que representa la iteración correspondiente, funcionan como entrada para los circuitos shift register y el circuito ajustador. En el shift register sirve para procesar el valor de conversión que representa la señal analógica a convertir. En el circuito ajustador sirve para identificar como operar la señal que será procesada en el siguiente ciclo del proceso de conversión.

La primera etapa de diseño de este circuito consistió en realizar un circuito que cumpliera con su funcionamiento para probar su implementación, por lo tanto, se realizó un circuito a base de switches ideales en el software LTspice, como se observa en la Figura No.43.

**Figura No.43.** Esquemático de circuito ideal de comparador analógico



**If  $V_{in} > V_{dd}/2$  ...  $V_{out} = v_{dd}$ ... else...  $V_{out} = gnd$**

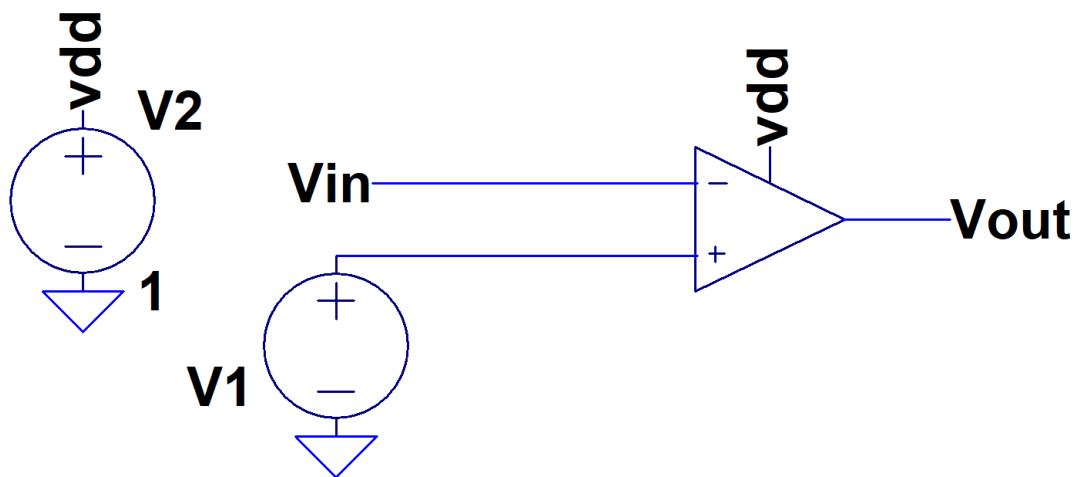
El funcionamiento del circuito de la Figura No.43 consiste en que, si  $V_{in}$  es mayor que  $V_{dd}/2$  entonces la salida  $V_{out}$  es igual a  $v_{dd}$ , si  $V_{in}$  es menor que  $V_{dd}/2$  entonces la

salida  $V_{out}$  es igual a  $gnd$ . Este diseño contribuyó para corroborar que este subcircuito junto con los demás circuitos si producían los resultados deseados del sistema. Otro aspecto importante de este circuito fue el poder identificar algunos parámetros necesarios a tomar en cuenta a la hora de modificar este circuito. Ya funcionando todos los sistemas conectados se pasó a la modificación de este circuito.

Se prosiguió a modificar este circuito para realizar uno que tuviera la misma (o lo más parecido posible) función de transferencia, pero implementando opamps diseñados con transistores.

La primera prueba de circuito es el presentado en la Figura No.44. Este circuito no contaba con las características necesarias para ser implementado en el diseño final, era muy lento al momento de hacer un cambio de un 1 lógico a un 0 lógico y viceversa y también otra limitante era que exigía demasiada corriente de entrada por lo que los circuitos que producían la señal de entrada se veían afectados por esto, degradando entonces la señal de entrada del circuito.

**Figura No.44.** Comparador implementando opamp con transistores de 50 nm

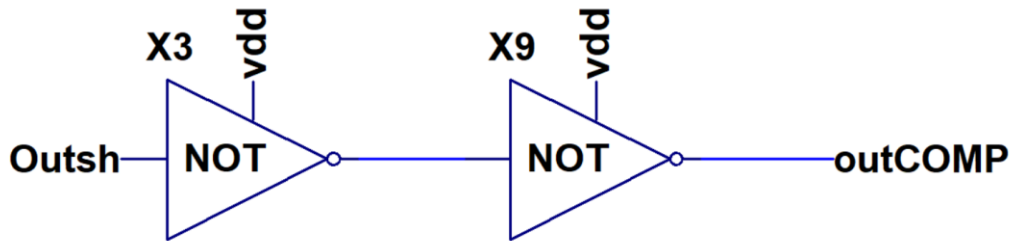


Se hizo otra prueba implementando otro circuito amplificador que saturara la señal de entrada, pero del mismo modo que el anterior circuito, este contaba con características

que hacían no factible su implementación, este comparador poseía un gran margen de error en su salida al momento de comparar la señal; otro factor negativo para no tomar en cuenta este diseño fue la potencia y la complejidad que requería éste, no se consideraban óptimas para las características de diseño que este circuito contaba con relación a las que se pretendían obtener.

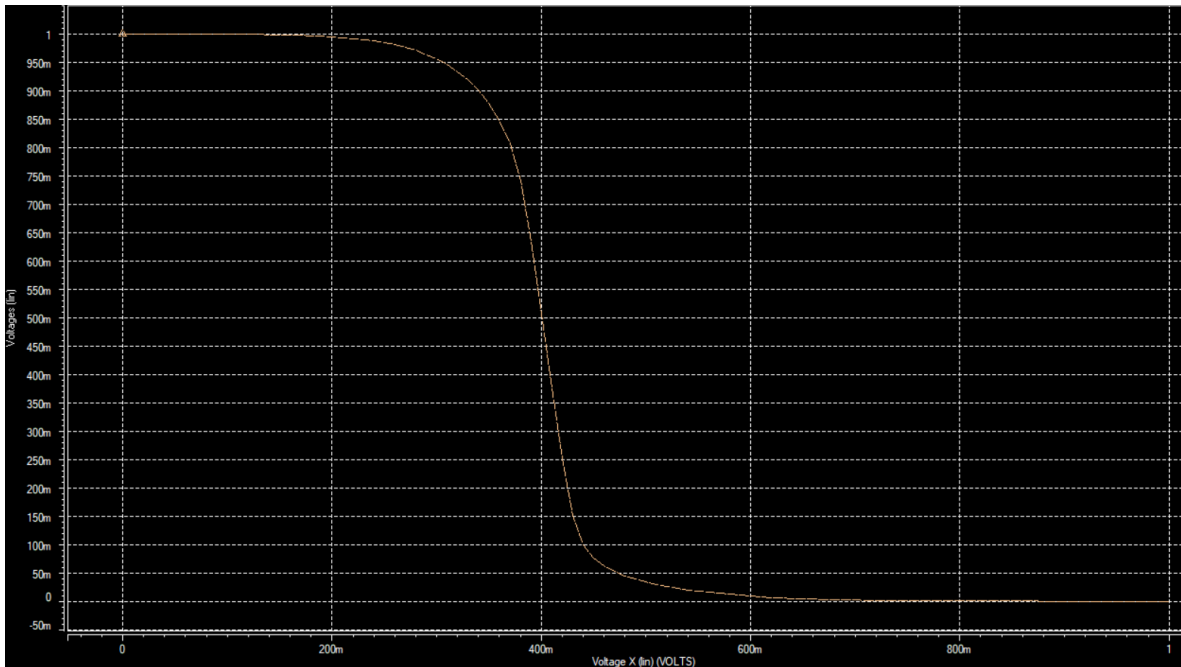
Luego de investigar otros métodos, se llegó a considerar la opción de implementar inversores. Este circuito contaba con mejores características que los anteriores, como la sencillez del diseño. En su implementación en LTspice se diseñaron dos inversores iguales con una relación l/w de  $\frac{1}{2}$  para los nmos y  $\frac{1}{4}$  para los pmos.

**Figura No.45.** Circuito comparador usando doble inversor



Posteriormente en el momento de realizar este diseño en Hspice se contó con el problema que la curva, en el análisis AC, del inversor estaba corrida por lo que su punto de comparación estaba desfasado. Otro error que se producía era que en la salida aparecían voltajes no deseados, esto se debía a que la curva debía tener una mayor pendiente. Esto se puede observar en la Figura No.46, donde se puede apreciar como la curva está centrada en 400 mV y la pendiente debe ser mayor para poder tener los resultados requeridos en este diseño.

**Figura No.46.** Análisis AC inversor con error de corrimiento y pendiente en Hspice



Este circuito debe contar con una gran precisión puesto errores en la comparación pueden provocar un resultado totalmente incorrecto en la conversión. Para la correcta calibración de este circuito es necesario ajustar el desfase que posee e incrementar la pendiente de forma significativa. De tal modo que si la señal de entrada de este circuito pasara de los 500mV entonces el comparador lo indicara, de forma precisa, cambiando el valor de su salida.

## **E. Circuito sujetador de señales**

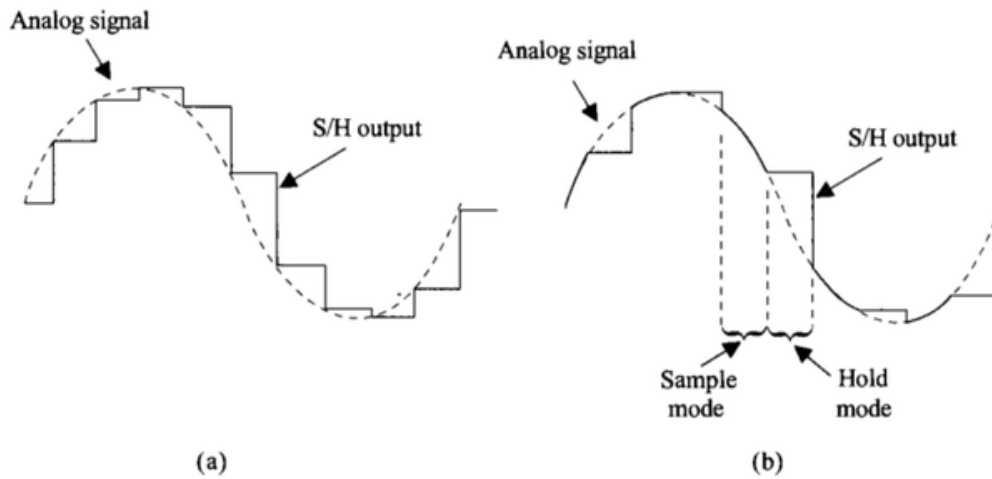
El circuito sujetador de señales o S/H "Sample and hold" es estrictamente necesario para cualquier tipo de ADC. Para el correcto análisis de datos es necesario poder retener un dato en un intervalo de tiempo, es necesario que el resultado de una conversión permanezca cierto tiempo retenido en su salida.

Entre las entradas que este circuito posee destacan la señal de entrada del exterior y la señal que retorna del circuito ajustador de señales. A través de dos circuitos

transmission gate se elige que señal servirá como entrada, de las dos señales mencionadas anteriormente, en cada ciclo. Al inicio de la conversión, en el primer ciclo, es necesario obtener una muestra del exterior para luego procesarla y obtener de forma recursiva una nueva muestra que servirá para el segundo ciclo, del mismo modo la muestra de este segundo ciclo determina la siguiente muestra. Por lo que se utiliza la entrada del exterior en un ciclo y luego se utiliza la entrada de la señal que retorna del circuito ajustador las siguientes veces por cada conversión que se realice. El que señal dejar pasar esta determinado por una bandera llamada NS "New sample", que indica a los transmission gate que estado tomar en qué momento.

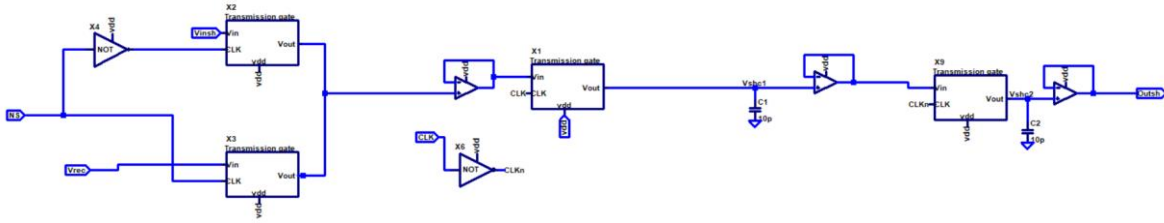
La forma como se realizó el circuito S/H fue implementando dos circuitos T/H. Como se puede observar en la Figura No.47 un circuito Track and hold tiene una etapa llamada "Sample mode", la cual consiste en seguir la señal de entrada, la siguiente etapa llamada "hold mode", consiste en retener la señal cierta cantidad de tiempo. Así mismo se puede observar como un circuito S/H solo retiene la señal por el intervalo de tiempo que dura el ciclo, para luego retener otro intervalo de tiempo. Si luego de contar con un circuito T/H se coloca otro circuito T/H en serie al primero y desfasado en tiempo por medio ciclo, lo que ocurre es que el segundo T/H tiene su etapa de Sample mode mientras que el primer T/H tiene su etapa de Hold mode, para luego el segundo T/H pasara a su etapa de Hold mode, por lo que mantendría la señal retenida, como lo hace un circuito S/H, todo un ciclo completo.

**Figura No.47.** Salida de (a) circuito S/H ideal y (b) circuito track and hold (T/H)



Como se puede observar en la Figura No.48, este circuito está formado por tres amplificadores operacionales, que han sido usados como seguidores de voltaje para poder compensar la corriente que otros circuitos demandarían de los capacitores y así poder hacer que los capacitores cuenten con un voltaje constante cuando es requerido. Se cuenta con cuatro circuitos transmission gate, de los cuales dos de ellos son para determinar cuál va a ser la señal de entrada a ser muestreada. Los otros dos son parte de los dos circuitos T/H utilizados, sirven para aislar los al circuito y así poder mantener el Hold mode que corresponde a cada uno de estos. Dos inversores son utilizados para negar la señal NS y la señal reloj, en el caso de la señal NS se busca que con una bandera controlar dos circuitos que funcionan sincronizados, mientras uno funciona el otro deja de funcionar; en el caso de la señal de reloj invertida sirve para controlar los dos circuitos T/H desfasados medio ciclo. Por último, se cuenta con dos capacitores que sirven para mantener el voltaje en el hold mode.

**Figura No.48.** Esquemático circuito muestreador



## F. Circuito ajustador de señales

Previamente se ha mencionado que la señal de entrada del ADC se modifica para poder ser reutilizada en futuros ciclos en la conversión. Este circuito modifica la señal analógica muestreada en el circuito S/H de tal forma que puede ser utilizada nuevamente. Este circuito permite poder reutilizar los otros circuitos un número N de ciclos. Hacer un circuito que trate una señal de forma recursiva hace necesario tener que ajustar una futura señal en el momento que se cuenta con la presente señal.

La función de este circuito consiste en operar la señal de tal forma que cumpla con el algoritmo de un ADC cíclico, con esto se refiere a que, dependiendo del tipo de señal en este momento, así será la futura señal por operar. Este circuito cumple con la siguiente ecuación:  $V_{out} = 2V_{in}$ , no obstante dependiendo de una señal proveniente del comparador ( llamada "COMP" en la Figura No.50) la salida del sistema se le restaría un voltaje constante llamado  $V_{CM}$ , de tal forma que  $V_{out} = 2V_{in} - V_{cm}$ , donde  $V_{CM} = V_{dd} = 1$  V. Por lo tanto:

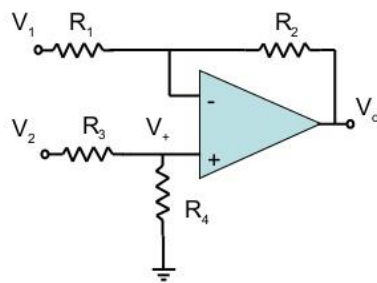
$$V_{out} = \begin{cases} 2V_{in}, & \text{Si } COMP = 0V \\ 2V_{in} - V_{cm}, & \text{Si } COMP = 5V \end{cases}$$

Las entradas de este circuito son la señal proveniente del comparador COMP, los voltajes de alimentación y la señal proveniente del S/H. Este circuito solo cuenta con una salida, esta llega al circuito S/H.



Los subcircuitos que lo conforman son dos transmission gate, dos amplificadores operacionales, un inversor y seis resistencias. Se utilizó un amplificador operacional para formar un circuito restador, se puede observar en la Figura No.49, previo al ajuste del circuito se buscó que  $R_1 = R_3$  y  $R_2 = R_4$ , de tal modo que este circuito no recibiera  $V_{cm}$  sino  $V_{cm}/2$  para así hacer que  $R_2 = 2R_1$ , de este modo se cumpliría la función de transferencia requerida por el sistema. Posteriormente se utilizaron dos transmission gate para determinar que si resta  $V_{cm}/2$  o  $0V$  a la señal, de tal modo que cumpliera con la ecuación antes mencionada. Otro amplificador operacional fue utilizado como seguidor para con este brindar la corriente que necesitara el circuito restador. El inversor se utilizó para controlar los dos transmission gate con una misma señal.

**Figura No.49.** Circuito restador



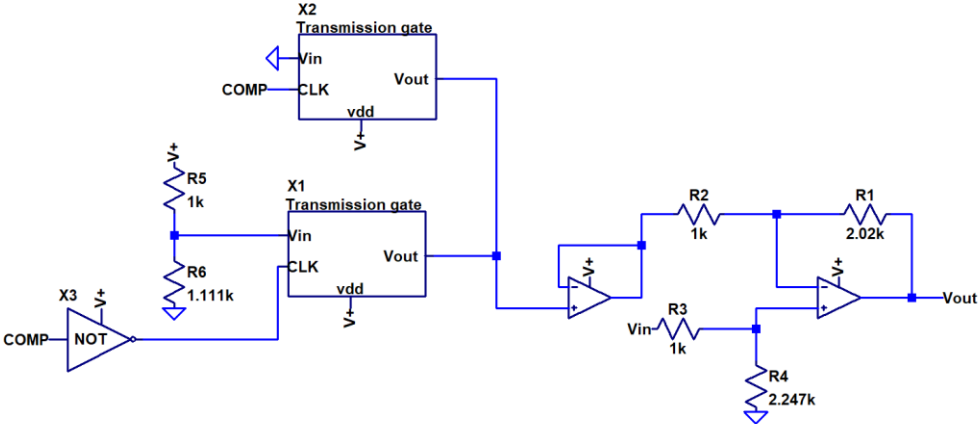
Aplicando superposición:

$$V_o = V_2 \cdot \left( \frac{R_4}{R_3 + R_4} \right) \cdot \left( 1 + \frac{R_2}{R_1} \right) - V_1 \cdot \frac{R_2}{R_1}$$

Si hacemos  $R_1 = R_3$  y  $R_2 = R_4$

$$V_o = \frac{R_2}{R_1} \cdot (V_2 - V_1)$$

**Figura No.50.** Esquemático de circuito ajustador de señales



## VIII. Resultados

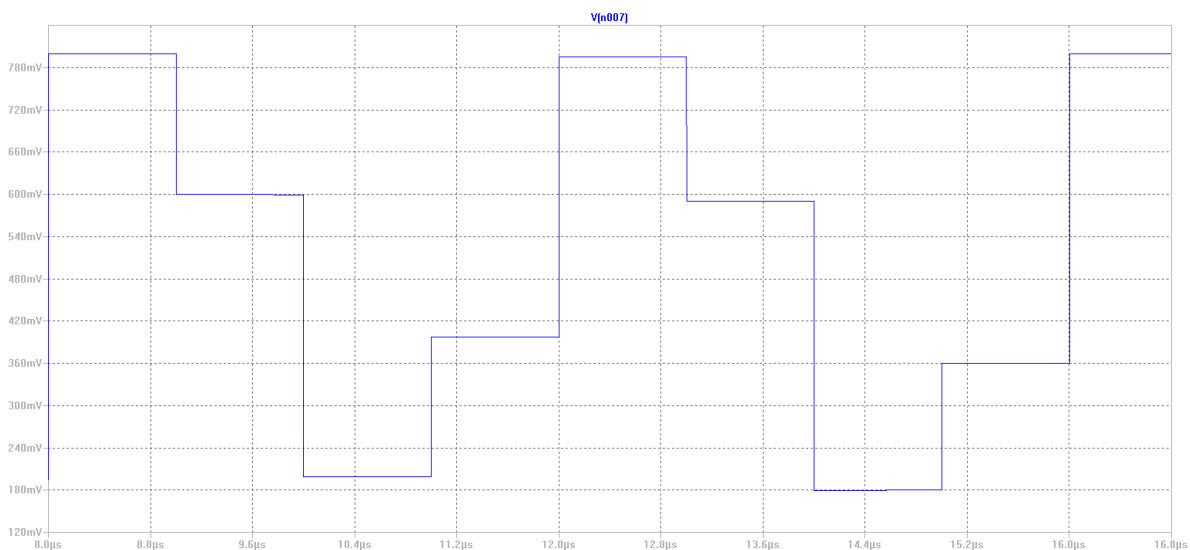
En este capítulo se muestran varias simulaciones del ADC diseñado y de sus subcircuitos. Así mismo se mencionarán otros resultados destacados.

Se considera que su tasa máxima de conversión es de 125 KHz por palabra. Se realizaron pruebas a esta tasa de conversión, se puede observar los resultados en las siguientes gráficas a continuación, donde el voltaje de entrada a convertir es de 0.8V.

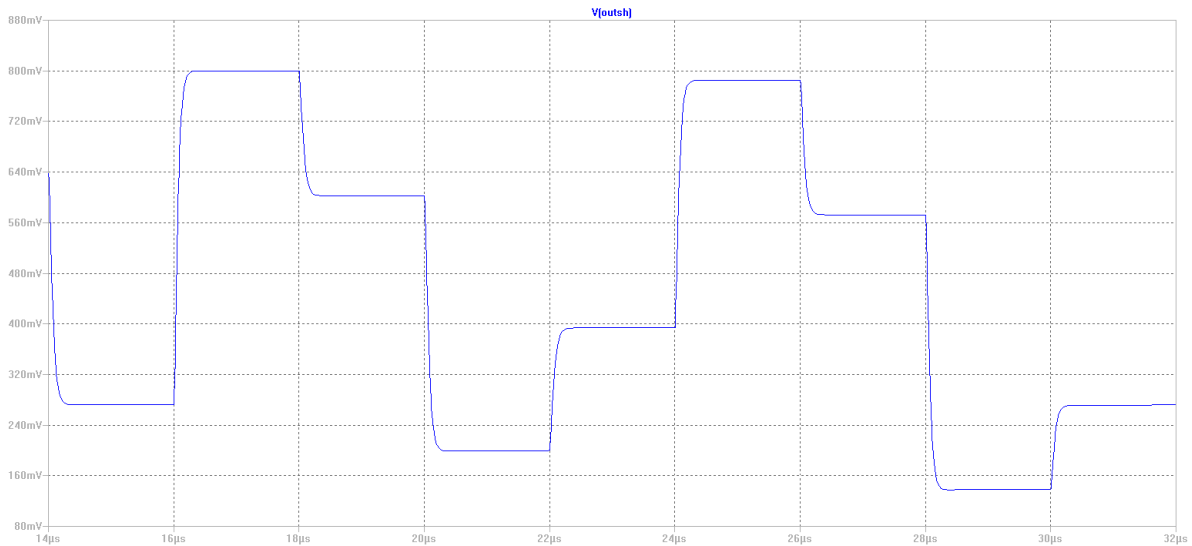
La Figura No.51 y la Figura No.52 muestran resultados de dos circuitos ADC cíclico donde se implementaron componentes ideales con el fin de abstraer y simplificar el diseño del sistema.

En la siguiente gráfica se muestran resultados de la salida del S/H de un ADC cíclico que fue diseñado a partir de componentes ideales, tales como switches ideales y fuentes dependientes.

**Figura No.51.** Simulación en LTspice de ADC cíclico con circuitos analógicos ideales, conversión de 800 mV @ 125 kHz

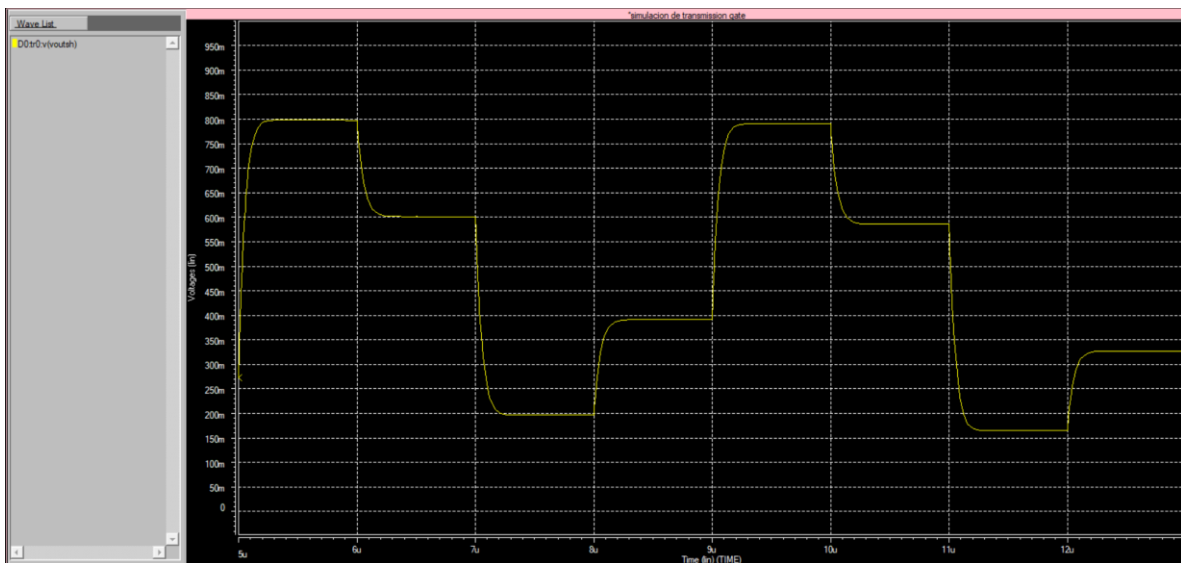


**Figura No.52.** Simulación en LTspice de ADC cíclico con amplificadores operacionales ideales, conversión de 800 mV @ 125 kHz



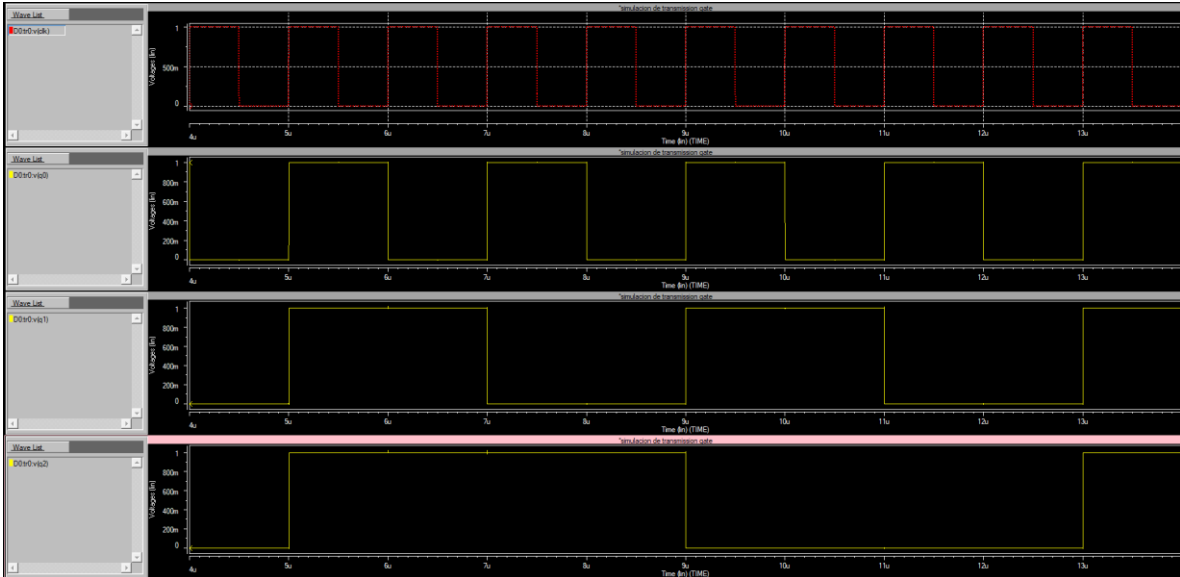
A continuación, se presentan resultados del circuito ADC cíclico el cual está diseñado a partir de transistores con tecnología de 50 nm, simulado en el software Hspice.

**Figura No.53.** Resultados de máxima tasa de conversión, señal de salida del circuito S/H

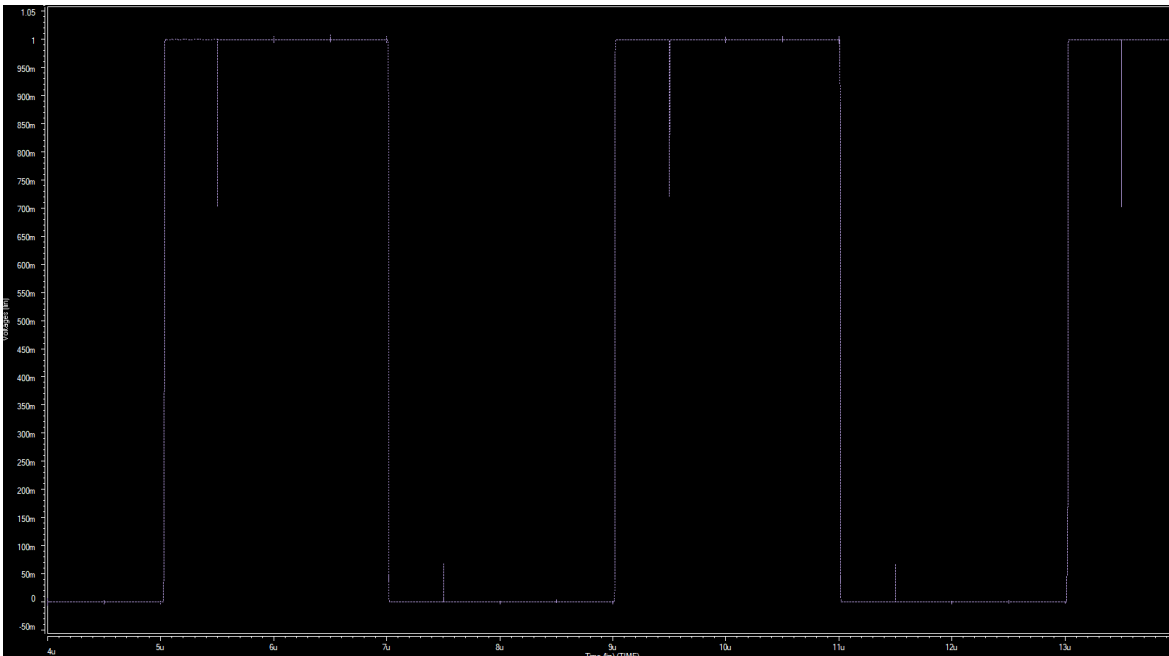


A continuación, se muestran diferentes resultados de subcircuitos a una tasa de 125 kHz por conversión.

**Figura No.54.** Contador digital de tres bits, en rojo señal de reloj, en amarillo señales de bit 0, bit 1 y bit 2 del contador

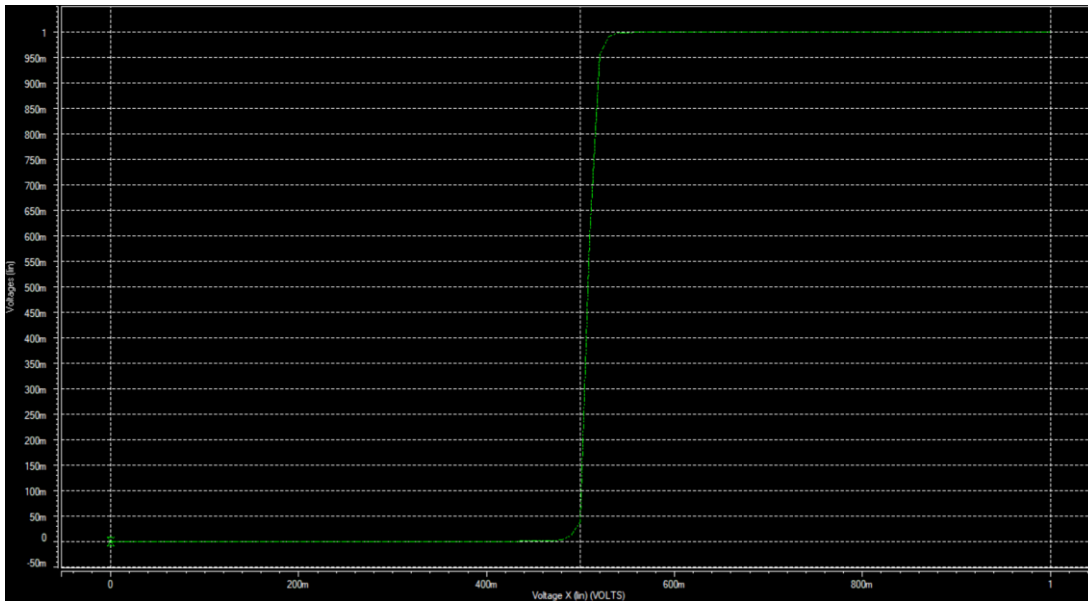


**Figura No.55.** Salida de comparador analógico

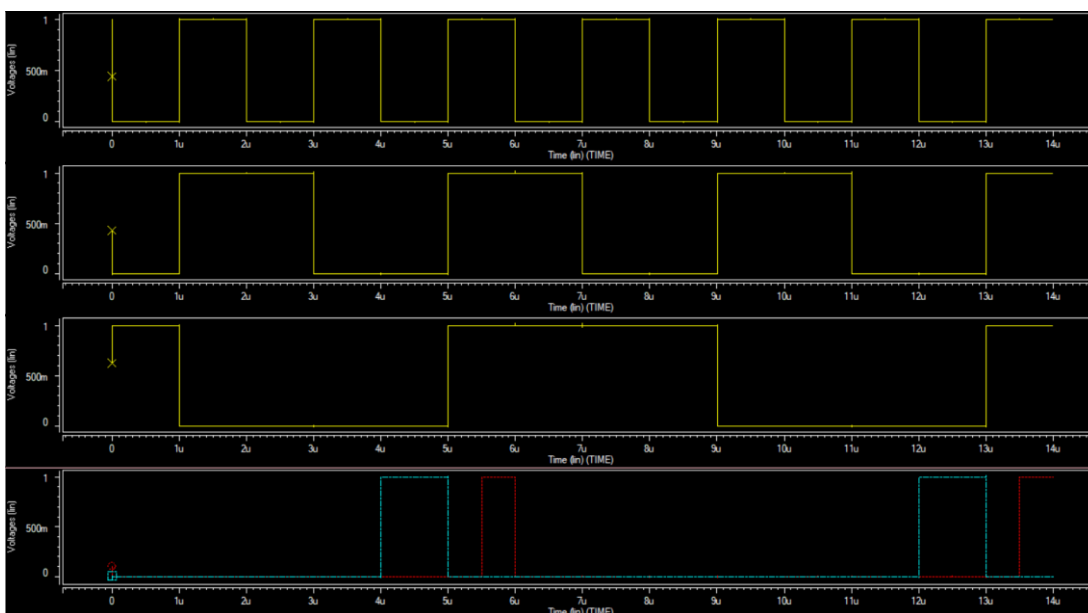


En el capítulo de diseño se señaló que era necesario modificar los inversores que sirven como comparadores. El resultado de la curva DC de estos fue la siguiente.

**Figura No.56.** Curva DC de inversores utilizados como comparador analógico

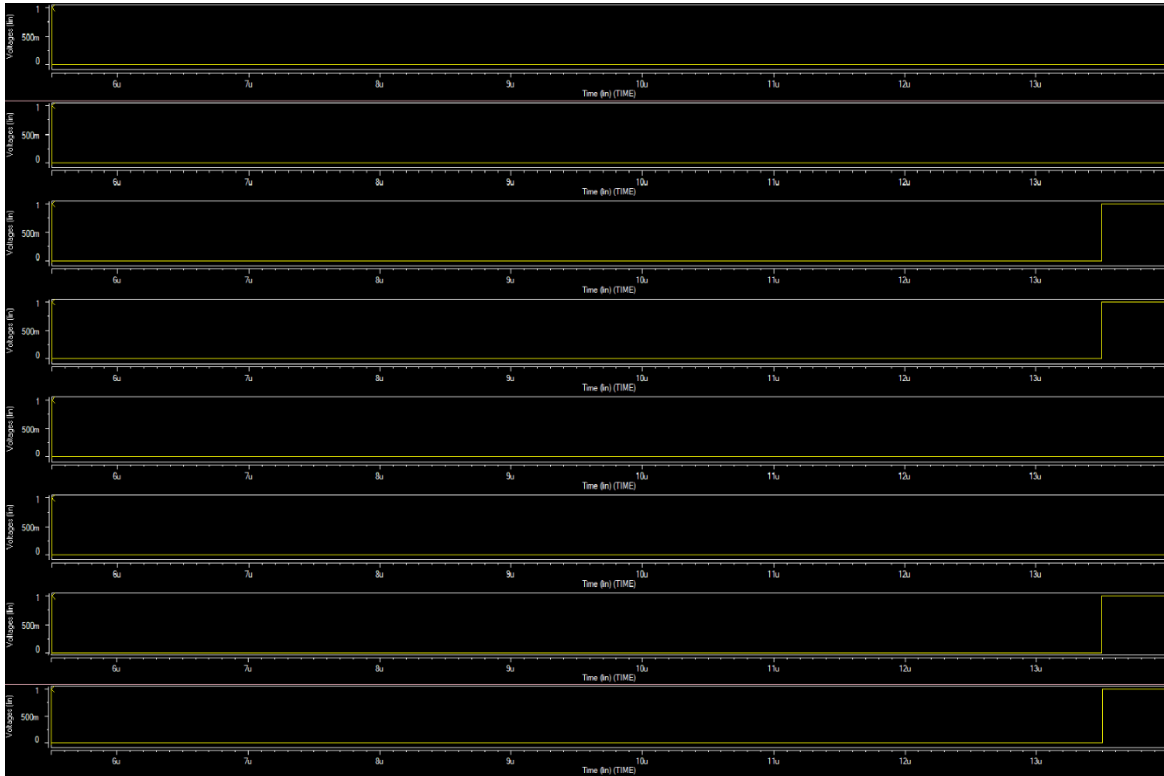


**Figura No.57.** Salida de comparador digital; en amarillo señales de contador; en azul señal para indicar nuevo muestreo; en rojo señal para copiar del shift register a la memoria

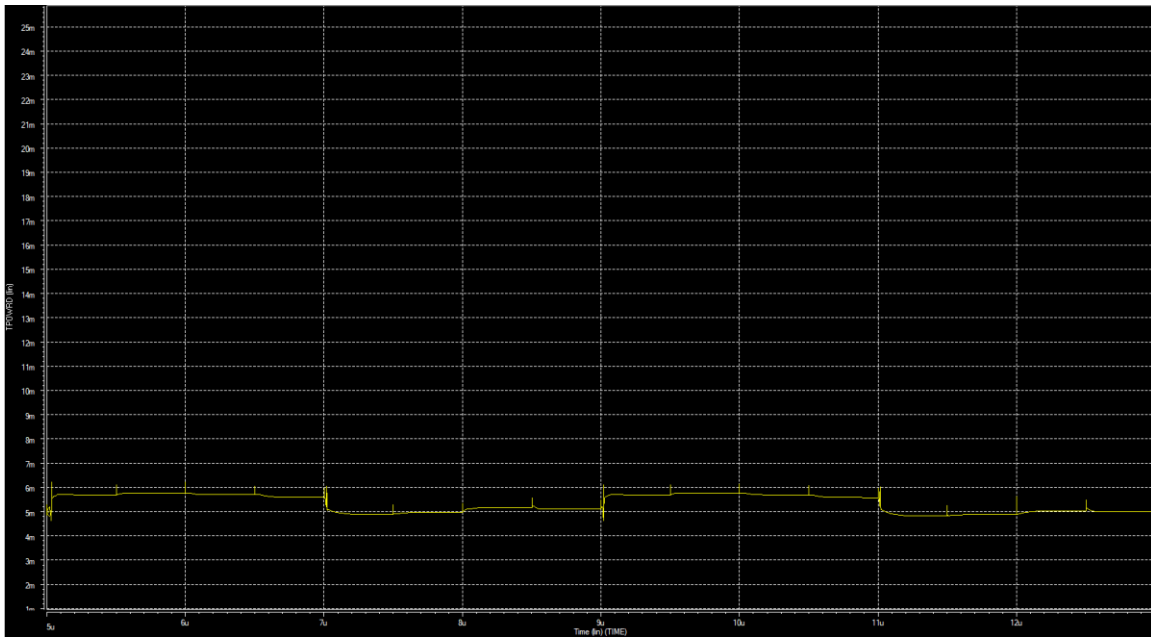


En la siguiente figura se puede observar los resultados que se guardan en la memoria de la conversión de 0.8 V, los resultados se producen a partir del tiempo 13.5 u segundos.

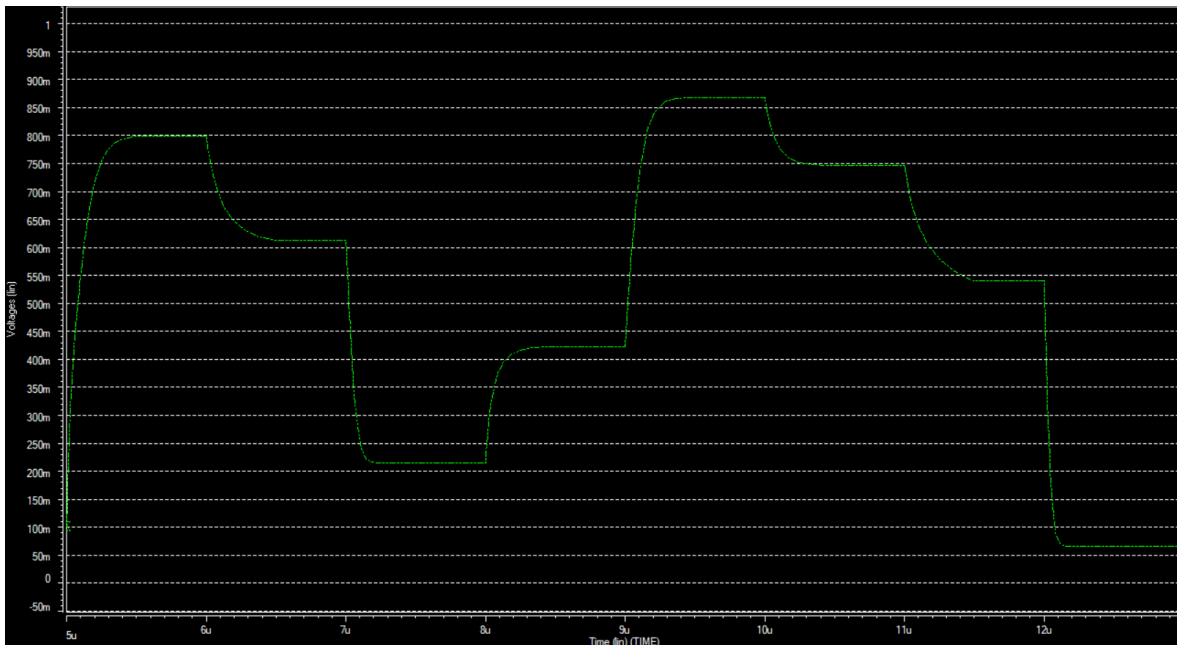
**Figura No.58.** Resultados de conversión en subcircuito memoria, siendo el LSB la gráfica de hasta arriba y el MSB la gráfica de hasta abajo.



**Figura No.59.** Consumo de potencia de circuito ADC cíclico



**Figura No.60.** Simulación de ADC cíclico implementando tecnología de 32 nm al diseño actual, señal de salida de circuito S/H



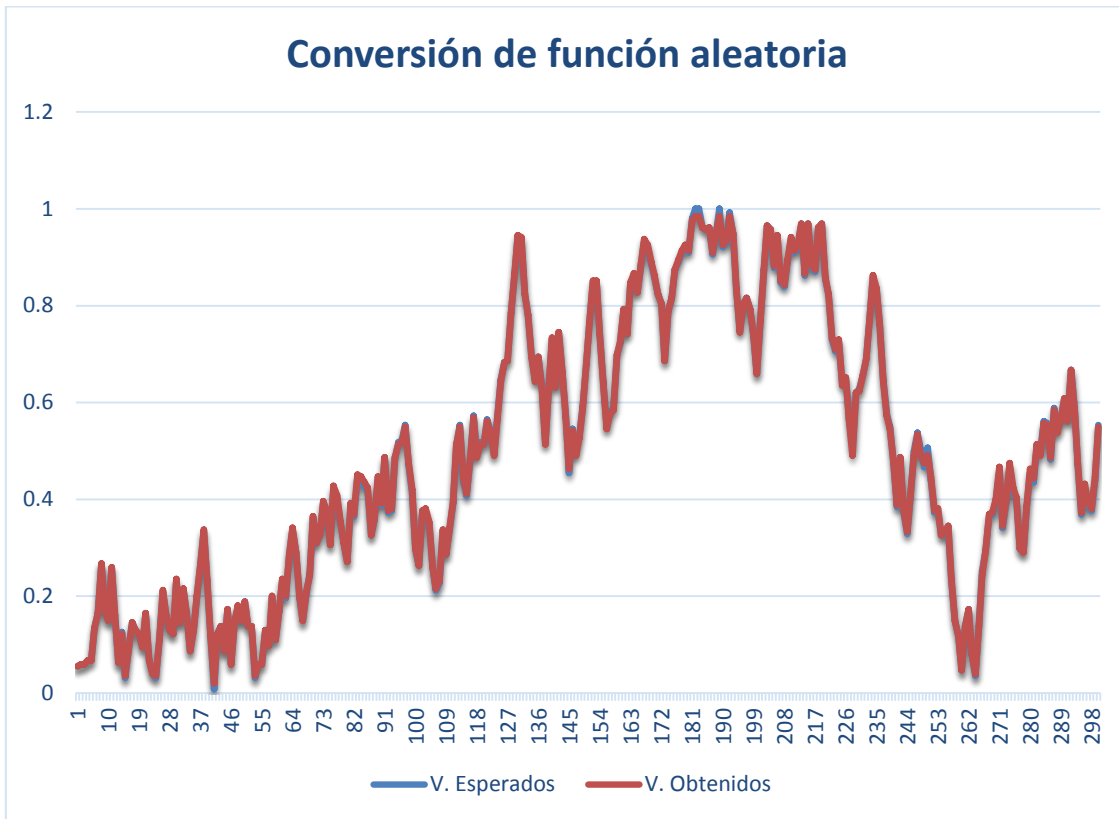


**Tabla No.3.** Resultados de simulación de ADC cíclico con una entrada de 800mV @ 125 Ksps

Tecnología	50 nm
Resolución	8 bits
Radio de conversión	Hasta 125 Ksps
Frecuencia de reloj	Hasta 1 MHz
Voltaje de alimentación	1 V
Voltaje de referencia	1V
Consumo de potencia promedio	5.3425mW @ 125 Ksps
Número de transistores implementados	878 mosfets
% de error máximo con respecto al rango completo	1.57%
% de conversión sin error	41.41%
% de error relativo máximo con respecto al valor específico de conversión	300%
Desviación estándar con respecto al error en prueba de conversión de todos los posibles valores	0.7497 LSB
Media aritmética del error	0.7254 LSB

Se realizó una simulación con una señal aleatoria de entrada, para ver la salida del ADC y comparar esta con la señal que se esperaba obtener.

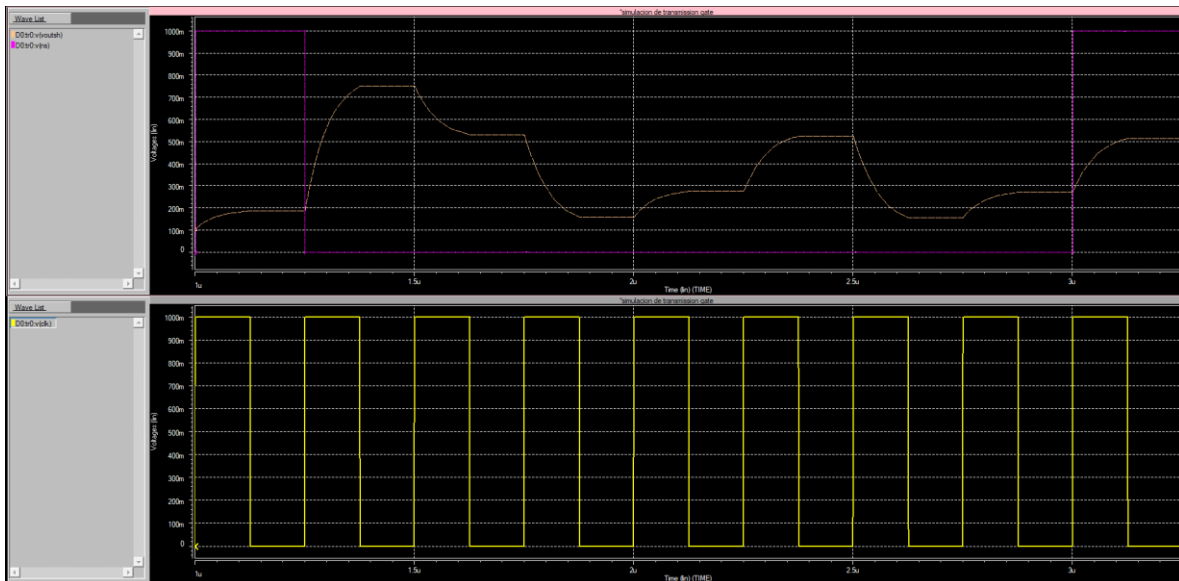
**Figura No.61.** Gráfica conversión de función aleatoria, resultados obtenidos vs resultados esperados



## IX. Discusión de resultados

En el momento de determinar la tasa máxima de conversión, se pudo observar que cuando se iba incrementando la frecuencia de reloj, lo primero que limitó la tasa de conversión el sistema fueron los capacitores del circuito S/H. Puesto que no se cargan los capacitores en el tiempo, la señal que se muestrea es errónea desde un principio y el resultado es incorrecto desde el principio. En la Figura No.62 se trató de convertir un valor de voltaje analógico de 0.8V, pero se puede observar que en el primer ciclo no alcanza el tiempo para que los capacitores lleguen a cargarse al valor de 0.8V.

**Figura No.62.** Etapas de conversión en salida del S/H (café) con error por tasa de conversión inadecuada, señal de nueva conversión en morado y señal de reloj en amarillo



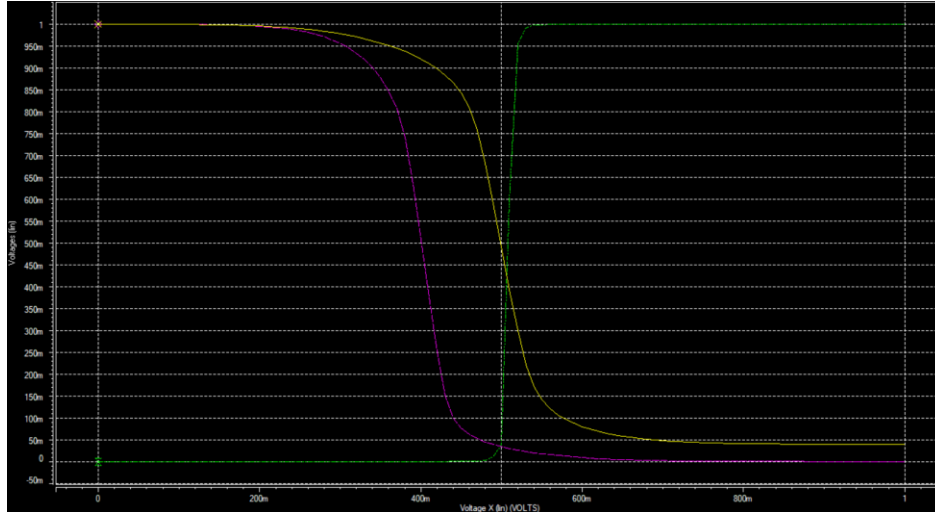
La tasa de conversión es posible mejorarla disminuyendo la capacitancia de los dos capacitores del circuito S/H, no obstante, el problema radica en que por la baja capacitancia los capacitores no serían capaces de mantener un voltaje cierta cantidad de tiempo y entonces el voltaje se vería alterado en frecuencias bajas.

Puesto el circuito muestreador se divide en dos circuitos T/H se ve como la señal se queda en estado hold por un tiempo. El circuito solo cuenta con medio ciclo de reloj para cargar los capacitores con el valor que posteriormente deben mantener en estado hold.

En el capítulo de diseño se menciona porque el contador digital es descendente y no ascendente, del mismo modo se describe cómo se puede modificar para hacer un contador ascendente. Se destaca que se dejó el diseño con este contador descendente porque era más sencillo y para fines de la aplicación no había ninguna diferencia significativa como para elegir diseñar el otro contador.

En el comparador analógico se utilizaron dos inversores, esto ayudo en gran medida a rectificar la señal, esto se puede apreciar en la Figura No.55. Fue necesario modificar estos inversores, puesto en la comparación se generaban errores muy grandes que hacían imposible una comparación efectiva. Esto se puede observar en la Figura No.63, la señal original en un principio era igual a la del segundo inversor (señal morada). Se puede observar que existe un gran desfase que provoca que en voltajes abajo del punto de comparación, 500 mV, existan salidas no correctas. Otro problema consiste en la pendiente de curva que esa señal posee, puesto de su poca pendiente, en los resultados era posible observar voltajes no saturados en la salida de este comparador, lo cual era otra fuente de error en la salida. Al modificar el sistema se puede observar como la señal se mejoró en estos dos aspectos en gran medida, pero por la gran precisión que es necesaria para la comparación se recomienda mejorar este sistema para futuros trabajos.

**Figura No.63.** Análisis DC de comparador e inversores que lo componen, en amarillo análisis DC de primer inversor; en morado análisis DC de segundo inversor; en verde análisis DC de los dos inversores en serie.



El propósito de generar un contador era el de poder tener una idea de en qué parte de la conversión se estaba y así generar ciertas banderas necesarias para que el sistema actuara de una forma especial en cierto momento, para esto era necesario tener un comparador digital que identificara dichos momentos a partir de los datos que generaba el contador. Fueron necesarias dos señales, la primera indicaba cuando obtener datos del exterior para empezar una nueva comparación. La segunda indicaba en qué momento pasar la información del shift register a la memoria, al final de la conversión.

Por motivos de escalabilidad, en el momento de describir el ADC en Hspice, el tamaño de los transistores se desarrolló de forma paramétrica. De esta forma se hizo factible el poder incorporar y probar otras tecnologías. Se hicieron pruebas con tecnología de 32 nm. En la Figura No.60 se puede observar los resultados obtenidos de esto. Lo primero a mencionar es que existe un error de precisión que se considera como des calibración del sistema. En el circuito ajustador hay un opamp que se implementa como parte de un circuito restador, se considera que este fue el que se vio más afectado al momento que se cambió de tecnología, por lo que se considera que este error de

precisión se arreglaría ajustando las resistencias y capacitores que forman este circuito. El error de precisión se puede apreciar en la Figura No.60 desde el segundo muestreo, por lo que el circuito S/H o muestreador se dice que sigue funcionando correctamente y el error se encuentra en la operación que se realiza en el circuito ajustador.

Entre los problemas que conlleva este diseño es la alta precisión necesaria, puesto que con un leve error en la calibración, este se multiplica en cada una de las iteraciones de tal forma que se vuelve muy significativo. Esto ocurre en su mayor parte en el circuito llamado ajustador, pero también ocurren en el circuito muestreador o S/H.

En la Figura No.51 se puede observar los resultados de la simulación de la conversión de 800 mV @ 125 kHz implementando circuitos ideales para los subcircuitos analógicos, tales como fuentes dependientes y switches ideales. Esto para poder hacer prueba de la arquitectura a implementar. En los resultados se puede observar que en los cambios de voltaje se cuenta con una respuesta instantánea. El circuito muestreador retiene el voltaje de forma muy precisa, sin variaciones puesto el sistema ya no cuenta con problemas de corrientes de fuga o desacoples.

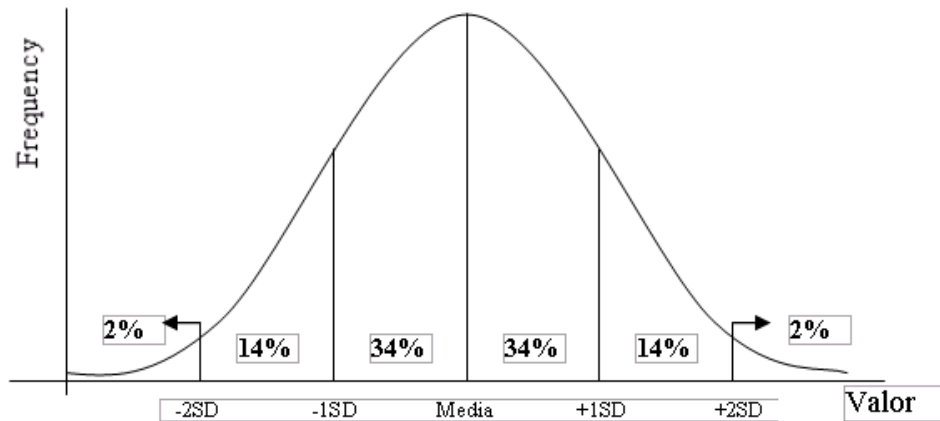
En la Figura No.52 se puede observar los resultados de la simulación de la conversión de 800 mV @ 125 kHz implementando amplificadores operacionales ideales. El software LTspice ya contaba con estos componentes y fueron de mucha ayuda para pasar a una etapa menos abstracta en el diseño de forma rápida. Se puede observar como ya se cuenta con problemas de ajuste desde este momento, esto porque el ajuste debe ser muy preciso. Si se compara con la Figura No.53 se puede observar que tiene un slew rate mayor y el circuito muestreador retiene la señal de forma más precisa.

En la Figura No.59 se puede ver la gráfica de potencia contra tiempo, el tiempo que se muestra en la gráfica es el que el ADC requiere para terminar una conversión. En la Tabla No.3 se puede observar el valor promedio de potencia.

Se realizó un programa en lenguaje de programación Python que ejecutara el Hspice y así simulara el ADC con todos los posibles valores de conversión que este permite. Los datos de estas simulaciones se pueden encontrar en el Apéndice C. En la Tabla No.3 se puede observar varios resultados importantes obtenidos de las simulaciones antes mencionadas, el porcentaje de error con respecto al valor se refiere a la relación entre el error de cierta medición con respecto al valor que debería de obtener en dicha simulación, en un principio es muy grande puesto los valores que se deberían obtener son pequeños. Por ejemplo, un error de 1 LSB es muy significativo cuando el valor de conversión es el equivalente a 00000001, sería un voltaje analógico de 0.0039 V, entonces el ADC con error de 1 LSB tendría una salida de valor 2 (en decimal) y no 1 (en decimal) como debería de ser, pero si fuera un error de 1 LSB en una conversión de voltaje que representara 255 (en decimal), sería un voltaje analógico de 1 V, este sería de 254 (en decimal) lo cual sería considerablemente menos significativo que el ejemplo anterior. El porcentaje de error máximo con respecto al valor medido fue de un 300 %, este error se produjo en el momento de convertir el valor más pequeño que le es posible convertir a este ADC, este error va disminuyendo muy rápidamente conforme los voltajes de entrada son mayores.

De las 256 posibles conversiones, 41.41% de los valores no cuentan con error de conversión. Del restante 58.59% se puede decir que el mayor error es de 1.57% con relación al rango total, lo cual lo hace ser un error aceptable para aplicaciones que no requieran de una alta precisión. La media aritmética es 0.7254 LSB y la desviación estándar del error es de 0.7497 LSB, esto dice que la mayor cantidad de errores poseen un valor que se encuentran a una desviación estándar positiva y una negativa de la media del error, como se puede observar en la Figura No.64, la cual dice que generalmente el 68% de los datos se concentran a 1 desviación estándar de la media aritmética y generalmente el 96% de los datos se concentra a un valor de dos desviaciones estándar.

**Figura No.64.** Interpretación de desviación estándar



Se realizó un programa que introdujera una señal aleatoria al circuito ADC y se guardaron los resultados de las dos señales, la original y la que se obtendría con el ADC, de este modo se puede observar como se distorsiona la señal al pasar por el ADC. En la Figura No.61 se pueden observar resultados de la simulación mencionada, en azul la señal original y en anaranjado la señal convertida, pasada por el ADC, se puede observar a simple vista que si son muy parecidas las curvas y solo se puede observar unas pequeñas variaciones no del todo significativas.



## X. Conclusiones

- Se logró desarrollar un comparador analógico a través de dos inversores modificando el tamaño y el voltaje que de bulk de los transistores que componían los inversores.
- El diseño no proporciona 100% de exactitud en todas las conversiones debido a no linealidades y error que se genera en el mismo sistema.
- El circuito comparador diseñado con dos inversores da resultados aceptables para los alcances de este proyecto
- El diseño del circuito en LTspice implementando cajas negras con componentes ideales facilitó pasar a un diseño final en Hspice implementando transistores de 32 y 50 nm.
- Los resultados en la herramienta Hspice y en la herramienta LTspice fueron muy similares sin tomar en cuenta el diseño del comparador analógico
- La tasa de conversión máxima fue de 125 KHz por conversión
- Para implementar otra tecnología en el sistema es necesario calibrar el sistema nuevamente.
- El sistema con tecnología de 32nm consume menos de la mitad de potencia que el sistema a 50 nm para una conversión.
- El mayor error es de 1.57% con respecto al rango total.
- La media aritmética del error es de 0.7254 LSB y la desviación estándar del error es de 0.7497.

## **XI. Recomendaciones**

- Se recomienda modificar el módulo de ajustador, para que tenga una mejor precisión y así introducir menor error al sistema. De este modo el sistema podría tener mayor resolución.
- Se recomienda investigar la posibilidad de implementar opamps full diferencial, puesto se considera que con su implementación es posible llegar tener menor error en el sistema.
- Se recomienda continuar con la metodología de diseño, utilizando cajas negras al principio para facilitar el rápido diseño, en la elaboración de circuitos similares.
- Se recomienda tener presente la aplicación de este circuito para la hora de diseñar el S/H saber que tiempos de conversión y muestreo deben ser realizados.
- Se recomienda recalibrar el sistema si se pensara continuar con el proyecto a una tecnología de 32 nm.

## XII. Bibliografía

A. D. I. Engineering, Data Conversion Handbook, 1 edition. Amsterdam; Boston: Newness, 2004.

Analog Devices, "Sample-and-Hold Amplifiers," p. 21.

Atmel Corporation, "AVR127: Understanding ADC Parameters," p. 24, 2016.

Baker R. J., CMOS Circuit Design, Layout, and Simulation, 3rd Edition, 3 edition. Piscataway, NJ: Hoboken, NJ: Wiley-IEEE Press, 2010.

Carsten F. J., "Advanced Techniques for Fast Offset Cancellation Limit Amplifiers," Massachusetts Institute of Technology, 2008.

da Branca Martins - Cyclic ADC with Programmable Resolution and Bias C.pdf." .

da Branca Martins, M. F. "Cyclic ADC with Programmable Resolution and Bias Current Adaptation," p. 4.

Dimitrov D. L., "A Cyclic Analog to Digital Converter for CMOS image sensors," Linköpings universitet, Sweden, 2013.

Douglas H., "CMOS Operational Amplifiers," in CMOS Analog Circuit Design, vol. Chapter 6, p. 36.

Fayomi, C. J. B. , G. W. Roberts, and M. Sawan, "Low-voltage CMOS analog bootstrapped switch for sample-and-hold circuit: design and chip characterization," in 2005 IEEE International Symposium on Circuits and Systems, 2005, pp. 2200-2203 Vol. 3.

Feddeler J. and B. Lucas, "Freescale Semiconductor, Inc...," p. 22.

Fuller, L. "SPICE Examples," 02-Jan-2014.

GmbH, I., A. kumar Puppala, and J. J. Wikner, "Design of a Low Power Cyclic / Algorithmic Analog-to-Digital Converter in a 130 nm CMOS Process," 2012. [Online]. Available: **[/paper/Design-of-a-Low-Power-Cyclic-%2F-Algorithmic-in-a-130-GmbH-Puppala/c21a15adc5221ec9e9602ef9555b3e7cc6625c27](http://paper/Design-of-a-Low-Power-Cyclic-%2F-Algorithmic-in-a-130-GmbH-Puppala/c21a15adc5221ec9e9602ef9555b3e7cc6625c27)**.

HSPICE [Online]. Available: **[http://utdallas.edu/~akshay.sridharan/index\\_files/Page9295.htm](http://utdallas.edu/~akshay.sridharan/index_files/Page9295.htm)**.

Hsu I. K. C., "A 70 MHz CMOS band-pass sigma-delta analog-to-digital converter for wireless receivers," 1999.

- Insoo, K. and C. Kyusun, "Sample & Hold Circuits," The Pennsylvania State University, Spring-2011.
- Josephsen S., "An Ultra-Low Power SAR-ADC in 65 nm CMOS Technology," Norwegian University of Science and Technology, Norway, 2013.
- Kakoty P. and K. Das, "DESIGN OF A HIGH FREQUENCY AND LOW POWER SUCCESSIVE APPROXIMATION ADC," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Energy, vol. 2, no. 8, pp. 3680–3692, Jan. 1970.
- Le B., T. W. Rondeau, J. H. Reed, and C. W. Bostian, "Analog-to-digital converters," IEEE Signal Processing Magazine, vol. 22, no. 6, pp. 69–77, Nov. 2005.
- Maruyama, T., K. Aihara, and M. Hotta, "Robust Cyclic ADC Architecture Based on  $\beta$ -Expansion | Request PDF." [Online]. Available: [https://www.researchgate.net/publication/270441911\\_Robust\\_Cyclic\\_ADC\\_Architecture\\_Based\\_on\\_beta-Expansion](https://www.researchgate.net/publication/270441911_Robust_Cyclic_ADC_Architecture_Based_on_beta-Expansion).
- Neha, Y. Amana, and C. Ila, "Design and Analysis of Comparators using 180nm CMOS Technology," vol. 7, p. 6, Jun. 2016.
- Nimish, K. , "HSpice - Device Level Circuit Simulation." [Online]. Available: <https://class.ee.washington.edu/cadta/hspice/>. [Accessed: 04-Aug-2018].
- Orchanian, S., G. Rudho, and A. Soares Jr., "Fundamental Blocks for a Cyclic Analog-to-Digital Converter," 2009.
- Rapuano S. et al., "ADC parameters and characteristics," IEEE Instrumentation Measurement Magazine, vol. 8, no. 5, pp. 44–54, Dec. 2005.
- Saranovac T., "Cyclic ADC based on novel concept of Zero-Crossing Detection," Swiss Federal Institute of Technology, 2012.
- Satish, Kashyap, Lec 28 Comparator Design, vol. 28. IIT Professors.
- Sivakumar - 2012 - Triple Sampling an Application to a 14b 10MSs Cyc.pdf." .
- Sivakumar, B. "Triple Sampling an Application to a 14b 10MS/s Cyclic ADC," ARIZONA STATE UNIVERSITY, 2012.
- Tangel A. and K. Choi, "'The CMOS Inverter' as a Comparator in ADC Designs," Analog Integrated Circuits and Signal Processing, vol. 39, no. 2, pp. 147–155, May 2004.

Texas Instruments, "ADC Accuracy Part 2: Total unadjusted error explained - Precision Hub - Archives - TI E2E Community." [Online]. Available: [https://e2e.ti.com/blogs\\_archives/b/precisionhub/archive/2014/10/14/adc-accuracy-part-2-total-unadjusted-error-explained](https://e2e.ti.com/blogs_archives/b/precisionhub/archive/2014/10/14/adc-accuracy-part-2-total-unadjusted-error-explained). [Accessed: 03-Aug-2018].

Texas Instruments, "ADC Performance Parameters- Convert the Units Correctly!," p. 8, May 2013.

## XIII. Apéndice

### A. Especificaciones de transistores MOSFET de tecnología de 50 nm utilizados en este proyecto

```
.model NMOS_50n nmos level = 54
+binunit = 1      paramchk= 1      mobmod = 0
+capmod = 2      igcmod = 1      igbmod = 1      geomod = 0
+diomod = 1      rdsmod = 0      rbodymod= 1      rgatemod= 1
+permod = 1      acnqsmod= 0      trnqsmod= 0
+tnom = 27      toxo = 1.4e-009  toxp = 7e-010  toxm = 1.4e-009
+epsrox = 3.9   wint = 5e-009    lint = 1.2e-008
+ll = 0         wl = 0         lln = 1         wln = 1
+lw = 0         ww = 0         lwn = 1         wwn = 1
+lw1 = 0        ww1 = 0        xpart = 0        toxref = 1.4e-009
+vth0 = 0.22    k1 = 0.35    k2 = 0.05    k3 = 0
+k3b = 0        w0 = 2.5e-006  dvt0 = 2.8    dvt1 = 0.52
+dvt2 = -0.032  dvt0w = 0    dvt1w = 0    dvt2w = 0
+dsb = 2        minv = 0.05   voffl = 0     dvtp0 = 1e-007
+dvtp1 = 0.05  lpe0 = 5.75e-008  lpeb = 2.3e-010  xj = 2e-008
+ngate = 5e+020  ndep = 2.8e+018  nsd = 1e+020   phin = 0
+cdsc = 0.0002  cdsbc = 0     cdsdc = 0     cit = 0
+voff = -0.15   nfactor = 1.2  eta0 = 0.15   etab = 0
+vfb = -0.55    u0 = 0.032    ua = 1.6e-010  ub = 1.1e-017
+uc = -3e-011   vsat = 1.1e+005  a0 = 2        ags = 1e-020
+a1 = 0         a2 = 1        b0 = -1e-020  b1 = 0
+keta = 0.04    dwg = 0       dwb = 0       pclm = 0.18
+pdiblc1 = 0.028  pdiblc2 = 0.022  pdiblc3 = -0.005  drout = 0.45
+pvag = 1e-020   delta = 0.01   pscbe1 = 8.14e+008  pscbe2 = 1e-007
+fprout = 0.2    pdits = 0.2   pditsd = 0.23  pditsl = 2.3e+006
+rsh = 3         rdsw = 150    rsw = 150     rdw = 150
+rdsmin = 0      rdswmin = 0   rswmin = 0     prwg = 0
+prwb = 6.8e-011  wr = 1        alpha0 = 0.074  alpha1 = 0.005
+beta0 = 30      agidl = 0.0002  bgidl = 2.1e+009  cgidl = 0.0002
+egidl = 0.8
+aigbacc = 0.012  bigbacc = 0.0028  cigbacc = 0.002
+nigbacc = 1     aigbinv = 0.014  bigbinv = 0.004  cigbinv = 0.004
+eigbinv = 1.1  nigbinv = 3     aigc = 0.017    bigc = 0.0028
+cigc = 0.002   aigsd = 0.017   bigsd = 0.0028  cigsd = 0.002
+nigc = 1       poxedge = 1     pigcd = 1       ntox = 1
+xrcrg1 = 12     xrcrg2 = 5
+cgso = 6.238e-010  cgdo = 6.238e-010  cgbo = 2.56e-011  cgdl = 2.495e-10
+cgs1 = 2.495e-10  ckappas = 0.02   ckappad = 0.02   acde = 1
+moin = 15       noff = 0.9      voffcv = 0.02
+kt1 = -0.21     kt1l = 0.0      kt2 = -0.042     ute = -1.5
+ua1 = 1e-009    ub1 = -3.5e-019  uc1 = 0          prt = 0
+at = 53000
+fnoimod = 1     tnoimod = 0
+jss = 0.0001    jsws = 1e-011   jswgs = 1e-010   njs = 1
+ijthsfwd = 0.01  ijthsrev = 0.001  bvs = 10         xjbvs = 1
+jsd = 0.0001    jswd = 1e-011   jswgd = 1e-010   njd = 1
+ijthdfwd = 0.01  ijthdrev = 0.001  bvd = 10         xjbvd = 1
+pbs = 1         cjs = 0.0005    mjs = 0.5        pbsws = 1
+cjsws = 5e-010  mjsws = 0.33    pbswgs = 1       cjswgs = 5e-010
+mjswgs = 0.33   pbd = 1         cjd = 0.0005     mjd = 0.5
+pbswd = 1       cjswd = 5e-010  mjswd = 0.33     pbswgd = 1
+cjswgd = 5e-010  mjswgd = 0.33   tpb = 0.005      tcj = 0.001
+tpbsw = 0.005  tcjsw = 0.001   tpbswg = 0.005   tcjswg = 0.001
+xtis = 3        xtid = 3
+dmcg = 0e-006   dmci = 0e-006   dmdg = 0e-006   dmcgt = 0e-007
+dwj = 0e-008    xgw = 0e-007    xgl = 0e-008
+rshg = 0.4      gbmin = 1e-010  rbpb = 5         rbpd = 15
+rbps = 15       rbdb = 15       rbsb = 15        ngcon = 1
```

\*

```
.model PMOS_50n pmos level = 54
+binunit = 1      paramchk= 1      mobmod = 0
+capmod = 2      igcmod = 1      igbmod = 1      geomod = 0
+diomod = 1      rdsmmod = 0      rbodymod= 1      rgatemod= 1
+permod = 1      acnqsmmod= 0      trnqsmmod= 0
+tnom = 27      toxex = 1.4e-009  toxp = 7e-010  toxm = 1.4e-009
+epsrox = 3.9    wint = 5e-009      lint = 1.2e-008
+ll = 0          wl = 0          lln = 1          wln = 1
+lw = 0          ww = 0          lwn = 1          wwn = 1
+lwl = 0         wwl = 0         xpart = 0         toxref = 1.4e-009
+vth0 = -0.22    k1 = 0.39      k2 = 0.05      k3 = 0
+k3b = 0         w0 = 2.5e-006      dvt0 = 3.9     dvt1 = 0.635
+dvt2 = -0.032  dvt0w = 0      dvt1w = 0      dvt2w = 0
+dsusb = 0.7     minv = 0.05    voffl = 0      dvtp0 = 0.5e-008
+dvtp1 = 0.05   lpe0 = 5.75e-008  lpeb = 2.3e-010  xj = 2e-008
+ngate = 5e+020  ndep = 2.8e+018  nsd = 1e+020    phin = 0
+cdsc = 0.000258  cdsdb = 0      cdsd = 6.1e-008  cit = 0
+voff = -0.15    nfactor = 2     eta0 = 0.15     etab = 0
+vfb = 0.55      u0 = 0.0095     ua = 1.6e-009    ub = 8e-018
+uc = 4.6e-013   vsat = 90000    a0 = 1.2         ags = 1e-020
+a1 = 0          a2 = 1          b0 = -1e-020     b1 = 0
+keta = -0.047   dwg = 0         dwb = 0          pclm = 0.55
+pdiblc1 = 0.03  pdiblc2 = 0.0055  pdiblc3 = 3.4e-008  drout = 0.56
+pvag = 1e-020   delta = 0.014    pscbe1 = 8.14e+008  pscbe2 = 9.58e-007
+fprout = 0.2    pdits = 0.2      pditsd = 0.23     pditsl = 2.3e+006
+rsh = 3         rdsb = 250      rsw = 160         rdw = 160
+rdsbmin = 0     rdwmin = 0       rswmin = 0        prwg = 3.22e-008
+prwb = 6.8e-011  wr = 1          alpha0 = 0.074    alpha1 = 0.005
+beta0 = 30      agidl = 0.0002   bgidl = 2.1e+009  cgidl = 0.0002
+egidl = 0.8
+aigbacc = 0.012  bigbacc = 0.0028  cigbacc = 0.002
+nigbacc = 1     aigbinv = 0.014  bigbinv = 0.004  cigbinv = 0.004
+eigbinv = 1.1   nigbinv = 3      aigc = 0.69      bigc = 0.0012
+cigc = 0.0008   aigsd = 0.0087  bigsd = 0.0012  cigsd = 0.0008
+nigc = 1        pokedge = 1      pigcd = 1         ntox = 1
+xrcrg1 = 12     xrcrg2 = 5
+cgso = 7.43e-010  cgdo = 7.43e-010  cgbo = 2.56e-011  cgdl = 1e-014
+cgs1 = 1e-014   ckappas = 0.5    ckappad = 0.5     acde = 1
+moin = 15       noff = 0.9       voffcv = 0.02
+kt1 = -0.19     kt1l = 0         kt2 = -0.052     ute = -1.5
+ua1 = -1e-009   ub1 = 2e-018     uc1 = 0           prt = 0
+at = 33000
+fnoimod = 1     tnoimod = 0
+jss = 0.0001    jsws = 1e-011    jswgs = 1e-010    njs = 1
+ijthsfwd = 0.01  ijthsrev = 0.001  bvs = 10          xjbvs = 1
+jsd = 0.0001    jswd = 1e-011    jswgd = 1e-010    njd = 1
+ijthdfwd = 0.01  ijthdrev = 0.001  bvd = 10          xjbvd = 1
+pbs = 1         cjs = 0.0005     mjs = 0.5         pbsws = 1
+cjsws = 5e-010  mjsws = 0.33     pbswgs = 1        cjswgs = 5e-010
+mjswgs = 0.33   pbd = 1          cjd = 0.0005     mjd = 0.5
+pbswd = 1       cjswd = 5e-010   mjswd = 0.33     pbswgd = 1
+cjswgd = 5e-010  mjswgd = 0.33    tpb = 0.005      tcj = 0.001
+tpbsw = 0.005  tcjsw = 0.001    tpbswg = 0.005   tcjswg = 0.001
+xtis = 3        xtld = 3
+dmcg = 0e-006   dmci = 0e-006    dmdg = 0e-006    dmcgt = 0e-007
+dwj = 0e-008    xgw = 0e-007     xgl = 0e-008
+rshg = 0.4      gbmin = 1e-010   rbpb = 5          rbpd = 15
+rbps = 15       rbdb = 15        rbsb = 15         ngcon = 1
```

## B. Especificaciones de transistores MOSFET de tecnología de 32 nm utilizados en este proyecto

\* PTM 32nm NMOS

.model NMOS\_32n nmos level = 54

```

+version = 4.0      binunit = 1      paramchk= 1      mobmod = 0
+capmod = 2        igcmod = 1      igbmod = 1      geomod = 1
+diomod = 1        rdsmod = 0      rbodymod= 1     rgatemod= 1
+permod = 1        acnqsmod= 0      trnqsmod= 0

+tnom = 27         toxo = 1.65e-9   toxp = 1.0e-9   toxm = 1.65e-9
+dtox = 0.65e-9   epsrox = 3.9      wint = 5e-009   lint = 2.7e-009
+ll = 0           wl = 0           lln = 1         wln = 1
+lw = 0           ww = 0           lwn = 1         wwn = 1
+lwl = 0          wwl = 0          xpart = 0       toxref = 1.65e-9
+xl = -14e-9

+vth0 = 0.5088    k1 = 0.4         k2 = 0.0        k3 = 0
+k3b = 0          w0 = 2.5e-006   dvt0 = 1        dvt1 = 2
+dvt2 = 0         dvt0w = 0       dvt1w = 0       dvt2w = 0
+dsusb = 0.1      minv = 0.05     voffl = 0       dvtp0 = 1.0e-011
+dvtp1 = 0.1      lpe0 = 0        lpeb = 0        xj = 1.0e-008
+ngate = 2e+020   ndep = 4.12e+018 nsd = 2e+020    phin = 0
+cdsc = 0.000    cdsbc = 0       cdsdc = 0       cit = 0
+voff = -0.13     nfactor = 2.3   eta0 = 0.0042   etab = 0
+vfb = -0.55      u0 = 0.0389     ua = 6e-010     ub = 1.2e-018
+uc = 0           vsat = 178470   a0 = 1.0        ags = 0
+a1 = 0           a2 = 1.0        b0 = 0          b1 = 0
+keta = 0.04      dwg = 0         dwb = 0         pclm = 0.02
+pdiblc1 = 0.001 pdiblc2 = 0.001 pdiblc3 = -0.005 dROUT = 0.5
+pvag = 1e-020    delta = 0.01    pscbe1 = 8.14e+008 pscbe2 = 1e-007
+fprout = 0.2     pdits = 0.01    pditsd = 0.23   pditsl = 2.3e+006
+rsh = 5          rdsw = 150     rsw = 75        rdw = 75
+rdswmin = 0     rdwmin = 0     rswmin = 0      prwg = 0
+prwb = 0         wr = 1          alpha0 = 0.074   alpha1 = 0.005
+beta0 = 30       agidl = 0.0002  bgidl = 2.1e+009 cgidl = 0.0002
+egidl = 0.8

+aignbacc = 0.012  bigbacc = 0.0028  cigbacc = 0.002
+nignbacc = 1      aignbinv = 0.014  bigbinv = 0.004  cigbinv = 0.004
+eignbinv = 1.1    nignbinv = 3      aigc = 0.012    bigc = 0.0028
+cigc = 0.002      aigsd = 0.012    bigsd = 0.0028  cigsd = 0.002
+nigc = 1          poxedge = 1      pigcd = 1        ntox = 1

+xrcrg1 = 12       xrcrg2 = 5
+cgso = 0.85e-010 cgdo = 0.85e-010 cgbo = 2.56e-011 cgdl = 2.653e-10
+cgs1 = 2.653e-10 ckappas = 0.03   ckappad = 0.03  acde = 1
+moin = 15         noff = 0.9       voffcv = 0.02

+kt1 = -0.11      kt1l = 0         kt2 = 0.022     ute = -1.5
+ua1 = 4.31e-009 ub1 = 7.61e-018 uc1 = -5.6e-011 prt = 0
+at = 33000

+fnoimod = 1      tnoimod = 0

+jss = 0.0001     jsws = 1e-011    jswgs = 1e-010  njs = 1
+ijthsfwd = 0.01  ijthsrev = 0.001 bvs = 10         xjbvs = 1
+jsd = 0.0001     jswd = 1e-011    jswgd = 1e-010  njd = 1
+ijthdfwd = 0.01  ijthdrev = 0.001 bvd = 10         xjbvd = 1
+pbs = 1          cjs = 0.0005     mjs = 0.5       pbsws = 1
+cjsws = 5e-010   mjsws = 0.33     pbswgs = 1      cjswgs = 3e-010
+mjswgs = 0.33    pbd = 1          cjd = 0.0005    mjgd = 0.5
+pbswd = 1        cjswd = 5e-010   mjswd = 0.33    pbswgd = 1
+cjswgd = 5e-010 mjswgd = 0.33    tpb = 0.005     tcj = 0.001
+tpbsw = 0.005   tcjsw = 0.001    tpbswg = 0.005  tcjswg = 0.001

```



```

+xtis = 3      xtid = 3

+dmcg = 0e-006  dmci = 0e-006  dmdg = 0e-006  dmcgt = 0e-007
+dwj  = 0.0e-008  xgw  = 0e-007   xgl  = 0e-008

+rshg = 0.4     gbmin = 1e-010  rbpb = 5      rbpd = 15
+rbps = 15     rbdb = 15     rbsb = 15     ngcon = 1

* PTM 32nm PMOS

.model PMOS_32n pmos level = 54

+version = 4.0   binunit = 1     paramchk= 1     mobmod = 0
+capmod = 2     igcmod = 1     igbmod = 1     geomod = 1
+diomod = 1     rdsmod = 0     rbodymod= 1     rgatemod= 1
+permod = 1     acnqsmod= 0     trnqsmod= 0

+tnom = 27     toxo = 1.75e-009  toxp = 1.0e-009  toxm = 1.75e-009
+dtox = 0.75e-9  epsrox = 3.9     wint = 5e-009   lint = 2.7e-009
+ll = 0        wl = 0        lln = 1        wln = 1
+lw = 0        ww = 0        lwn = 1        wwn = 1
+lwl = 0       wwl = 0       xpart = 0      toxref = 1.75e-009
+xl = -14e-9

+vth0 = -0.450  k1 = 0.4        k2 = -0.01     k3 = 0
+k3b = 0        w0 = 2.5e-006  dvt0 = 1      dvt1 = 2
+dvt2 = -0.032  dvt0w = 0      dvt1w = 0     dvt2w = 0
+dsub = 0.1     minv = 0.05     voffl = 0     dvtp0 = 1e-011
+dvtp1 = 0.05   lpe0 = 0        lpeb = 0     xj = 1.0e-008
+ngate = 2e+020  ndep = 3.07e+018  nsd = 2e+020  phin = 0
+cdsc = 0.000   cdscb = 0       cdscd = 0     cit = 0
+voff = -0.126  nfactor = 2.3   eta0 = 0.0042  etab = 0
+vfb = 0.55     u0 = 0.00355   ua = 2.0e-009  ub = 0.5e-018
+uc = 0         vsat = 70000   a0 = 1.0       ags = 1e-020
+a1 = 0         a2 = 1         b0 = 0         b1 = 0
+keta = -0.047  dwg = 0        dwb = 0        pclm = 0.1
+pdiblc1 = 0.001  pdiblc2 = 0.001  pdiblc3 = 3.4e-008  drout = 0.6
+pvag = 1e-020  delta = 0.01    pscbe1 = 8.14e+008  pscbe2 = 9.58e-007
+fprout = 0.2    pdits = 0.08   pditsd = 0.23  pditsl = 2.3e+006
+rsh = 5        rdsw = 150     rsw = 75       rdw = 75
+rdswmin = 0    rdwmin = 0     rswmin = 0     prwg = 0
+prwb = 0       wr = 1         alpha0 = 0.074  alpha1 = 0.005
+beta0 = 30     agidl = 0.0002  bgidl = 2.1e+009  cgidl = 0.0002
+egidl = 0.8

+aigbacc = 0.012  bigbacc = 0.0028  cigbacc = 0.002
+nigbacc = 1     aigbinv = 0.014  bigbinv = 0.004  cigbinv = 0.004
+eigbinv = 1.1   nigbinv = 3      aigc = 0.69     bigc = 0.0012
+cigc = 0.0008  aigsd = 0.0087  bigsd = 0.0012  cigsd = 0.0008
+nigc = 1        poxedge = 1     pigcd = 1       ntox = 1

+xrcrg1 = 12     xrcrg2 = 5
+cgso = 0.85e-010  cgdo = 0.85e-010  cgbo = 2.56e-011  cgdl = 2.653e-10
+cgs1 = 2.653e-10  ckappas = 0.03   ckappad = 0.03  acde = 1
+moin = 15       noff = 0.9       voffcv = 0.02

+kt1 = -0.11     kt1l = 0        kt2 = 0.022     ute = -1.5
+ua1 = 4.31e-009  ub1 = 7.61e-018  uc1 = -5.6e-011  prt = 0
+at = 33000

+fnoimod = 1     tnoimod = 0

+jss = 0.0001    jsws = 1e-011   jswgs = 1e-010  njs = 1
+ijthsfwd = 0.01  ijthsrev = 0.001  bvs = 10       xjbvs = 1
+jsd = 0.0001   jswd = 1e-011   jswgd = 1e-010  njd = 1
+ijthdfwd = 0.01  ijthdrev = 0.001  bvd = 10       xjbvd = 1

```

```
+pbs = 1      cjs = 0.0005  mjs = 0.5    pbsws = 1
+cjsws = 5e-010  mjsws = 0.33  pbswgs = 1    cjswgs = 3e-010
+mjswgs = 0.33  pbd = 1      cjd = 0.0005  mjd = 0.5
+pbswd = 1      cjswd = 5e-010  mjswd = 0.33  pbswgd = 1
+cjswgd = 5e-010  mjswgd = 0.33  tpb = 0.005  tcj = 0.001
+tpbsw = 0.005  tcjsw = 0.001  tpbswg = 0.005  tcjswg = 0.001
+xtis = 3      xtid = 3

+dmcg = 0e-006  dmci = 0e-006  dmdg = 0e-006  dmcgt = 0e-007
+dwj = 0.0e-008  xgw = 0e-007  xgl = 0e-008

+rshg = 0.4     gbmin = 1e-010  rbpb = 5      rbpd = 15
+rbps = 15     rbdb = 15      rbsb = 15     ngcon = 1
```

## C. Resultados de corrida de los 256 posibles valores de conversión

Se realizó un programa en el lenguaje de programación python que comprobara los 256 valores de conversión. Este programa introduce los valores de voltaje, cambiándolos en el archivo .sp, ejecuta el Hspice con el correspondiente archivo editado, lee el archivo .lis, extrae los resultados de la conversión y finalmente los guarda en un archivo de texto. En la Tabla No.4 se puede observar los resultados.

**Tabla No.4.** Resultados de corrida de los 256 valores posibles de conversión

Voltaje de Entrada	Valor digital obtenido	Conversión obtenida (valor decimal)	Conversión esperada	Conversión esperada (valor decimal)	Error individual en decimal	% de error con respecto al rango completo	% Error relativo con respecto al valor específico
0.0039	00000100	4	00000001	1	3	1.18%	300.00%
0.0078	00000101	5	00000010	2	3	1.18%	150.00%
0.0118	00000101	5	00000011	3	2	0.78%	66.67%
0.0157	00000110	6	00000100	4	2	0.78%	50.00%
0.0196	00000110	6	00000101	5	1	0.39%	20.00%
0.0235	00000111	7	00000110	6	1	0.39%	16.67%
0.0275	00001000	8	00000111	7	1	0.39%	14.29%
0.0314	00001001	9	00001000	8	1	0.39%	12.50%
0.0353	00001010	10	00001001	9	1	0.39%	11.11%
0.0392	00001010	10	00001010	10	0	0.00%	0.00%
0.0431	00001011	11	00001011	11	0	0.00%	0.00%
0.0471	00001100	12	00001100	12	0	0.00%	0.00%
0.051	00001101	13	00001101	13	0	0.00%	0.00%
0.0549	00001110	14	00001110	14	0	0.00%	0.00%
0.0588	00001111	15	00001111	15	0	0.00%	0.00%
0.0627	00010000	16	00010000	16	0	0.00%	0.00%
0.0667	00010001	17	00010001	17	0	0.00%	0.00%
0.0706	00010010	18	00010010	18	0	0.00%	0.00%
0.0745	00010011	19	00010011	19	0	0.00%	0.00%
0.0784	00010100	20	00010100	20	0	0.00%	0.00%
0.0824	00010101	21	00010101	21	0	0.00%	0.00%
0.0863	00010110	22	00010110	22	0	0.00%	0.00%
0.0902	00010111	23	00010111	23	0	0.00%	0.00%
0.0941	00011000	24	00011000	24	0	0.00%	0.00%
0.098	00011001	25	00011001	25	0	0.00%	0.00%
0.102	00011010	26	00011010	26	0	0.00%	0.00%
0.1059	00011011	27	00011011	27	0	0.00%	0.00%

Voltaje de Entrada	Valor digital obtenido	Conversión obtenida (valor decimal)	Conversión esperada	Conversión esperada (valor decimal)	Error individual en decimal	% de error con respecto al rango completo	% Error relativo con respecto al valor específico
0.1098	00011100	28	00011100	28	0	0.00%	0.00%
0.1137	00011101	29	00011101	29	0	0.00%	0.00%
0.1176	00011110	30	00011110	30	0	0.00%	0.00%
0.1216	00011111	31	00011111	31	0	0.00%	0.00%
0.1255	00011111	31	00100000	32	1	0.39%	3.13%
0.1294	00100001	33	00100001	33	0	0.00%	0.00%
0.1333	00100010	34	00100010	34	0	0.00%	0.00%
0.1373	00100011	35	00100011	35	0	0.00%	0.00%
0.1412	00100100	36	00100100	36	0	0.00%	0.00%
0.1451	00100101	37	00100101	37	0	0.00%	0.00%
0.149	00100110	38	00100110	38	0	0.00%	0.00%
0.1529	00100111	39	00100111	39	0	0.00%	0.00%
0.1569	00101000	40	00101000	40	0	0.00%	0.00%
0.1608	00101001	41	00101001	41	0	0.00%	0.00%
0.1647	00101010	42	00101010	42	0	0.00%	0.00%
0.1686	00101011	43	00101011	43	0	0.00%	0.00%
0.1725	00101100	44	00101100	44	0	0.00%	0.00%
0.1765	00101110	46	00101101	45	1	0.39%	2.22%
0.1804	00101111	47	00101110	46	1	0.39%	2.17%
0.1843	00101111	47	00101111	47	0	0.00%	0.00%
0.1882	00110000	48	00110000	48	0	0.00%	0.00%
0.1922	00110001	49	00110001	49	0	0.00%	0.00%
0.1961	00110011	51	00110010	50	1	0.39%	2.00%
0.2	00110100	52	00110011	51	1	0.39%	1.96%
0.2039	00110101	53	00110100	52	1	0.39%	1.92%
0.2078	00110110	54	00110101	53	1	0.39%	1.89%
0.2118	00110111	55	00110110	54	1	0.39%	1.85%
0.2157	00111000	56	00110111	55	1	0.39%	1.82%
0.2196	00111001	57	00111000	56	1	0.39%	1.79%
0.2235	00111010	58	00111001	57	1	0.39%	1.75%
0.2275	00111011	59	00111010	58	1	0.39%	1.72%
0.2314	00111100	60	00111011	59	1	0.39%	1.69%
0.2353	00111101	61	00111100	60	1	0.39%	1.67%
0.2392	00111101	61	00111101	61	0	0.00%	0.00%
0.2431	00111110	62	00111110	62	0	0.00%	0.00%
0.2471	00111110	62	00111111	63	1	0.39%	1.59%
0.251	00111110	62	01000000	64	2	0.78%	3.13%
0.2549	01000001	65	01000001	65	0	0.00%	0.00%
0.2588	01000010	66	01000010	66	0	0.00%	0.00%
0.2627	01000011	67	01000011	67	0	0.00%	0.00%
0.2667	01000100	68	01000100	68	0	0.00%	0.00%

Voltaje de Entrada	Valor digital obtenido	Conversión obtenida (valor decimal)	Conversión esperada	Conversión esperada (valor decimal)	Error individual en decimal	% de error con respecto al rango completo	% Error relativo con respecto al valor específico
0.2706	01000101	69	01000101	69	0	0.00%	0.00%
0.2745	01000110	70	01000110	70	0	0.00%	0.00%
0.2784	01000111	71	01000111	71	0	0.00%	0.00%
0.2824	01001000	72	01001000	72	0	0.00%	0.00%
0.2863	01001001	73	01001001	73	0	0.00%	0.00%
0.2902	01001010	74	01001010	74	0	0.00%	0.00%
0.2941	01001011	75	01001011	75	0	0.00%	0.00%
0.298	01001100	76	01001100	76	0	0.00%	0.00%
0.302	01001101	77	01001101	77	0	0.00%	0.00%
0.3059	01001110	78	01001110	78	0	0.00%	0.00%
0.3098	01001111	79	01001111	79	0	0.00%	0.00%
0.3137	01010000	80	01010000	80	0	0.00%	0.00%
0.3176	01010001	81	01010001	81	0	0.00%	0.00%
0.3216	01010010	82	01010010	82	0	0.00%	0.00%
0.3255	01010011	83	01010011	83	0	0.00%	0.00%
0.3294	01010101	85	01010100	84	1	0.39%	1.19%
0.3333	01010110	86	01010101	85	1	0.39%	1.18%
0.3373	01010111	87	01010110	86	1	0.39%	1.16%
0.3412	01011000	88	01010111	87	1	0.39%	1.15%
0.3451	01011001	89	01011000	88	1	0.39%	1.14%
0.349	01011010	90	01011001	89	1	0.39%	1.12%
0.3529	01011011	91	01011010	90	1	0.39%	1.11%
0.3569	01011100	92	01011011	91	1	0.39%	1.10%
0.3608	01011101	93	01011100	92	1	0.39%	1.09%
0.3647	01011110	94	01011101	93	1	0.39%	1.08%
0.3686	01011111	95	01011110	94	1	0.39%	1.06%
0.3725	01100000	96	01011111	95	1	0.39%	1.05%
0.3765	01100001	97	01100000	96	1	0.39%	1.04%
0.3804	01100010	98	01100001	97	1	0.39%	1.03%
0.3843	01100011	99	01100010	98	1	0.39%	1.02%
0.3882	01100100	100	01100011	99	1	0.39%	1.01%
0.3922	01100101	101	01100100	100	1	0.39%	1.00%
0.3961	01100110	102	01100101	101	1	0.39%	0.99%
0.4	01100111	103	01100110	102	1	0.39%	0.98%
0.4039	01101000	104	01100111	103	1	0.39%	0.97%
0.4078	01101001	105	01101000	104	1	0.39%	0.96%
0.4118	01101011	107	01101001	105	2	0.78%	1.90%
0.4157	01101100	108	01101010	106	2	0.78%	1.89%
0.4196	01101101	109	01101011	107	2	0.78%	1.87%
0.4235	01101110	110	01101100	108	2	0.78%	1.85%
0.4275	01101111	111	01101101	109	2	0.78%	1.83%

Voltaje de Entrada	Valor digital obtenido	Conversión obtenida (valor decimal)	Conversión esperada	Conversión esperada (valor decimal)	Error individual en decimal	% de error con respecto al rango completo	% Error relativo con respecto al valor específico
0.4314	01101111	111	01101110	110	1	0.39%	0.91%
0.4353	01110001	113	01101111	111	2	0.78%	1.80%
0.4392	01110010	114	01110000	112	2	0.78%	1.79%
0.4431	01110011	115	01110001	113	2	0.78%	1.77%
0.4471	01110100	116	01110010	114	2	0.78%	1.75%
0.451	01110101	117	01110011	115	2	0.78%	1.74%
0.4549	01110110	118	01110100	116	2	0.78%	1.72%
0.4588	01110111	119	01110101	117	2	0.78%	1.71%
0.4627	01111000	120	01110110	118	2	0.78%	1.69%
0.4667	01111001	121	01110111	119	2	0.78%	1.68%
0.4706	01111001	121	01111000	120	1	0.39%	0.83%
0.4745	01111010	122	01111001	121	1	0.39%	0.83%
0.4784	01111011	123	01111010	122	1	0.39%	0.82%
0.4824	01111100	124	01111011	123	1	0.39%	0.81%
0.4863	01111100	124	01111100	124	0	0.00%	0.00%
0.4902	01111100	124	01111101	125	1	0.39%	0.80%
0.4941	01111101	125	01111110	126	1	0.39%	0.79%
0.498	01111101	125	01111111	127	2	0.78%	1.57%
0.502	01111101	125	10000000	128	3	1.18%	2.34%
0.5059	01111101	125	10000001	129	4	1.57%	3.10%
0.5098	10000011	131	10000010	130	1	0.39%	0.77%
0.5137	10000011	131	10000011	131	0	0.00%	0.00%
0.5176	10000011	131	10000100	132	1	0.39%	0.76%
0.5216	10000100	132	10000101	133	1	0.39%	0.75%
0.5255	10000101	133	10000110	134	1	0.39%	0.75%
0.5294	10000110	134	10000111	135	1	0.39%	0.74%
0.5333	10000111	135	10001000	136	1	0.39%	0.74%
0.5373	10001000	136	10001001	137	1	0.39%	0.73%
0.5412	10001001	137	10001010	138	1	0.39%	0.72%
0.5451	10001010	138	10001011	139	1	0.39%	0.72%
0.549	10001011	139	10001100	140	1	0.39%	0.71%
0.5529	10001100	140	10001101	141	1	0.39%	0.71%
0.5569	10001101	141	10001110	142	1	0.39%	0.70%
0.5608	10001110	142	10001111	143	1	0.39%	0.70%
0.5647	10001111	143	10010000	144	1	0.39%	0.69%
0.5686	10010000	144	10010001	145	1	0.39%	0.69%
0.5725	10010001	145	10010010	146	1	0.39%	0.68%
0.5765	10010010	146	10010011	147	1	0.39%	0.68%
0.5804	10010011	147	10010100	148	1	0.39%	0.68%
0.5843	10010100	148	10010101	149	1	0.39%	0.67%
0.5882	10010101	149	10010110	150	1	0.39%	0.67%

Voltaje de Entrada	Valor digital obtenido	Conversión obtenida (valor decimal)	Conversión esperada	Conversión esperada (valor decimal)	Error individual en decimal	% de error con respecto al rango completo	% Error relativo con respecto al valor específico
0.5922	10010110	150	10010111	151	1	0.39%	0.66%
0.5961	10010111	151	10011000	152	1	0.39%	0.66%
0.6	10011000	152	10011001	153	1	0.39%	0.65%
0.6039	10011001	153	10011010	154	1	0.39%	0.65%
0.6078	10011011	155	10011011	155	0	0.00%	0.00%
0.6118	10011100	156	10011100	156	0	0.00%	0.00%
0.6157	10011101	157	10011101	157	0	0.00%	0.00%
0.6196	10011110	158	10011110	158	0	0.00%	0.00%
0.6235	10011111	159	10011111	159	0	0.00%	0.00%
0.6275	10011111	159	10100000	160	1	0.39%	0.63%
0.6314	10100000	160	10100001	161	1	0.39%	0.62%
0.6353	10100001	161	10100010	162	1	0.39%	0.62%
0.6392	10100010	162	10100011	163	1	0.39%	0.61%
0.6431	10100011	163	10100100	164	1	0.39%	0.61%
0.6471	10100100	164	10100101	165	1	0.39%	0.61%
0.651	10100110	166	10100110	166	0	0.00%	0.00%
0.6549	10100111	167	10100111	167	0	0.00%	0.00%
0.6588	10101000	168	10101000	168	0	0.00%	0.00%
0.6627	10101001	169	10101001	169	0	0.00%	0.00%
0.6667	10101010	170	10101010	170	0	0.00%	0.00%
0.6706	10101011	171	10101011	171	0	0.00%	0.00%
0.6745	10101100	172	10101100	172	0	0.00%	0.00%
0.6784	10101101	173	10101101	173	0	0.00%	0.00%
0.6824	10101110	174	10101110	174	0	0.00%	0.00%
0.6863	10101111	175	10101111	175	0	0.00%	0.00%
0.6902	10110000	176	10110000	176	0	0.00%	0.00%
0.6941	10110001	177	10110001	177	0	0.00%	0.00%
0.698	10110010	178	10110010	178	0	0.00%	0.00%
0.702	10110011	179	10110011	179	0	0.00%	0.00%
0.7059	10110101	181	10110100	180	1	0.39%	0.56%
0.7098	10110110	182	10110101	181	1	0.39%	0.55%
0.7137	10110111	183	10110110	182	1	0.39%	0.55%
0.7176	10111000	184	10110111	183	1	0.39%	0.55%
0.7216	10111001	185	10111000	184	1	0.39%	0.54%
0.7255	10111010	186	10111001	185	1	0.39%	0.54%
0.7294	10111011	187	10111010	186	1	0.39%	0.54%
0.7333	10111100	188	10111011	187	1	0.39%	0.53%
0.7373	10111101	189	10111100	188	1	0.39%	0.53%
0.7412	10111101	189	10111101	189	0	0.00%	0.00%
0.7451	10111110	190	10111110	190	0	0.00%	0.00%
0.749	10111110	190	10111111	191	1	0.39%	0.52%

Voltaje de Entrada	Valor digital obtenido	Conversión obtenida (valor decimal)	Conversión esperada	Conversión esperada (valor decimal)	Error individual en decimal	% de error con respecto al rango completo	% Error relativo con respecto al valor específico
0.7529	10111110	190	11000000	192	2	0.78%	1.04%
0.7569	11000001	193	11000001	193	0	0.00%	0.00%
0.7608	11000010	194	11000010	194	0	0.00%	0.00%
0.7647	11000011	195	11000011	195	0	0.00%	0.00%
0.7686	11000100	196	11000100	196	0	0.00%	0.00%
0.7725	11000101	197	11000101	197	0	0.00%	0.00%
0.7765	11000110	198	11000110	198	0	0.00%	0.00%
0.7804	11000111	199	11000111	199	0	0.00%	0.00%
0.7843	11001000	200	11001000	200	0	0.00%	0.00%
0.7882	11001001	201	11001001	201	0	0.00%	0.00%
0.7922	11001010	202	11001010	202	0	0.00%	0.00%
0.7961	11001011	203	11001011	203	0	0.00%	0.00%
0.8	11001100	204	11001100	204	0	0.00%	0.00%
0.8039	11001101	205	11001101	205	0	0.00%	0.00%
0.8078	11001110	206	11001110	206	0	0.00%	0.00%
0.8118	11001111	207	11001111	207	0	0.00%	0.00%
0.8157	11010000	208	11010000	208	0	0.00%	0.00%
0.8196	11010001	209	11010001	209	0	0.00%	0.00%
0.8235	11010010	210	11010010	210	0	0.00%	0.00%
0.8275	11010011	211	11010011	211	0	0.00%	0.00%
0.8314	11010101	213	11010100	212	1	0.39%	0.47%
0.8353	11010110	214	11010101	213	1	0.39%	0.47%
0.8392	11010111	215	11010110	214	1	0.39%	0.47%
0.8431	11011000	216	11010111	215	1	0.39%	0.47%
0.8471	11011001	217	11011000	216	1	0.39%	0.46%
0.851	11011010	218	11011001	217	1	0.39%	0.46%
0.8549	11011011	219	11011010	218	1	0.39%	0.46%
0.8588	11011100	220	11011011	219	1	0.39%	0.46%
0.8627	11011101	221	11011100	220	1	0.39%	0.45%
0.8667	11011110	222	11011101	221	1	0.39%	0.45%
0.8706	11011111	223	11011110	222	1	0.39%	0.45%
0.8745	11011111	223	11011111	223	0	0.00%	0.00%
0.8784	11100001	225	11100000	224	1	0.39%	0.45%
0.8824	11100010	226	11100001	225	1	0.39%	0.44%
0.8863	11100011	227	11100010	226	1	0.39%	0.44%
0.8902	11100100	228	11100011	227	1	0.39%	0.44%
0.8941	11100101	229	11100100	228	1	0.39%	0.44%
0.898	11100110	230	11100101	229	1	0.39%	0.44%
0.902	11100111	231	11100110	230	1	0.39%	0.43%
0.9059	11101000	232	11100111	231	1	0.39%	0.43%
0.9098	11101001	233	11101000	232	1	0.39%	0.43%



Voltaje de Entrada	Valor digital obtenido	Conversión obtenida (valor decimal)	Conversión esperada	Conversión esperada (valor decimal)	Error individual en decimal	% de error con respecto al rango completo	% Error relativo con respecto al valor específico
0.9137	11101010	234	11101001	233	1	0.39%	0.43%
0.9176	11101011	235	11101010	234	1	0.39%	0.43%
0.9216	11101100	236	11101011	235	1	0.39%	0.43%
0.9255	11101101	237	11101100	236	1	0.39%	0.42%
0.9294	11101110	238	11101101	237	1	0.39%	0.42%
0.9333	11101111	239	11101110	238	1	0.39%	0.42%
0.9373	11110000	240	11101111	239	1	0.39%	0.42%
0.9412	11110001	241	11110000	240	1	0.39%	0.42%
0.9451	11110010	242	11110001	241	1	0.39%	0.41%
0.949	11110011	243	11110010	242	1	0.39%	0.41%
0.9529	11110011	243	11110011	243	0	0.00%	0.00%
0.9569	11110100	244	11110100	244	0	0.00%	0.00%
0.9608	11110101	245	11110101	245	0	0.00%	0.00%
0.9647	11110110	246	11110110	246	0	0.00%	0.00%
0.9686	11110111	247	11110111	247	0	0.00%	0.00%
0.9725	11110111	247	11111000	248	1	0.39%	0.40%
0.9765	11111000	248	11111001	249	1	0.39%	0.40%
0.9804	11111001	249	11111010	250	1	0.39%	0.40%
0.9843	11111001	249	11111011	251	2	0.78%	0.80%
0.9882	11111010	250	11111100	252	2	0.78%	0.79%
0.9922	11111011	251	11111101	253	2	0.78%	0.79%
0.9961	11111011	251	11111110	254	3	1.18%	1.18%
1	11111011	251	11111111	255	4	1.57%	1.57%

## D. Programa en python para revisión resultados de ADC

```
import os
```

```
"""
```

Esta función busca el nombre de la variable en el archivo .lis y devuelve su valor expresado en el mismo archivo

```
"""
```

```
def search_text(buscado):
```

```
    datafile = file('measure.lis', 'r')
```

```
    found = False
```

```
    valor_voltaje = 0.0
```

```
    for line in datafile:
```

```
        if buscado in line:
```

```
            #print line
```

```
            datafile.close()
```

```
            valor_voltaje = float(line.split(" ")[len(line.split(" "))-1])
```

```
            #print "valor de voltaje es = " + str(valor_voltaje)
```

```
            break
```

```
    return valor_voltaje
```

```
"""
```

Esta función crea un archivo txt con la información de las corridas

entradas:

palabra de conversion

valor de voltaje de entrada

salidas:

```

'''
def insertar_dato(adc_word_value, adc_ideal_value, analog_value):
    a = format(int(str(bin(adc_ideal_value))[2:], 2), '08b') # pasar de un valor decimal a
binario, con numero de bits
    b = int('101',2) # pasar de binario a decimal

    print a
    print adc_word_value

    datafile = file('datos2.txt', 'a+')
    datafile.write(str(adc_word_value) + " " + str(adc_ideal_value) + " " + str(analog_value) + "
" + str(a) + "\n")
    datafile.close()

```

#-----

```

'''
Esta funcion edita el archivo .sp para generar
entrada:
    el valor a ingresar a la fuente de voltaje
salida:
    debe ingresar algo asi en el archivo sp - V6 Vin gnd 0.8
'''

```

```

'''
def copiar_sp():
    datafile = open('plantilla_Setup.txt', 'r')
    temp_data = datafile.read()
    datafile.close()
'''

```

```

return temp_data

def modificar_sp(vin, temp_data):

    new_data = temp_data.replace("<<voltage_value>>", "V6 Vin gnd " + str(vin))

    datafile = open('Setup.sp', 'w+')
    datafile.write(new_data)
    datafile.close()

'''
Crear palabra de simulacion actual
entradas:
    la resolucion del ADC
salida:
    resultado de los 8 bits ya concatenados como 1 o 0
'''

def crear_palabra(resolucion):

    actual_word = ""
    # resolucion = resolucion - 1
    # obtener crear palabra nueva
    for i in range(resolucion):
        buscado = "res" + str(i) + "=" # se crea la palabra a buscar en el .lis
        #print "la variable buscado = " + buscado
        found = search_text(buscado) #se obtiene el n valor del .lis
        #print "la variable found = " + str(found)
        if found > 0.5:

```

```

        digTemp = 1
    else:
        digTemp = 0
    actual_word = actual_word + str(digTemp) # se concatena para formar la palabra

return actual_word

# empeiza codigo ejecutable

plantilla = copiar_sp() # se guarda la plantilla para solo modificarla en un futuro y no
abrirla varias veces
adc_word = []
rango_inicial_corrida = 1
rango_final_corrida = 256

for i in range(rango_inicial_corrida,rango_final_corrida):
    vin_value = round((i)/255.0, 4)
    print i
    print "vin =" + str(vin_value)
    modificar_sp(vin_value, plantilla) # se crea o modifica el Setup.sp
    print "Se ha creado el archivo Setup.sp satisfactoriamente!"

    os.system('hspice -i Setup.sp -o measure')
    print "HSPICE Compilado satisfactoriamente!"

temp_adc_word = crear_palabra(8)
adc_word.append(temp_adc_word)
print temp_adc_word

```

```
print(str(len(adc_word)) + " simulaciones procesadas")
```

```
for i in range(len(adc_word)):
```

```
    insertar_dato(adc_word[i], i + rango_inicial_corrida, round((i+  
rango_inicial_corrida)/255.0, 4) )
```

```
    print "dato "+ str(i) + " almacenado!"
```

## XIV. Glosario

- **ADC ideal:** tipo de ADC hipotético, donde la resolución de este es infinita
- **ADC perfecto:** ADC que realiza el proceso de cuantificación de forma exacta
- **Convertor analógico a digital:** sistema (circuito electrónico) que convierte la señal de voltaje analógico a una señal digital.
- **Distorsión armónica total:** relación de la suma de las potencias de las frecuencias armónicas y la potencia de la frecuencia fundamental.
- **Distorsión señal a ruido:** relación entre el valor RMS de la amplitud de la señal y el valor RMS de todos los demás componentes espectrales.
- **EDA:** automatización de diseño electrónico (Electronic Design Automation); software dedicado a la concepción, diseño y producción de sistemas electrónicos.
- **Electric VLSI Design System:** herramienta de software de automatización de diseño electrónico; esta herramienta fue implementada en este proyecto.
- **Error de offset:** la variación que existe entre la función de transferencia del actual ADC y el ADC perfecto en el punto cero, medido en el bit LSB.
- **Error de Ganancia:** desviación del punto medio del último paso del actual ADC con respecto al punto medio del último paso del ADC ideal.
- **Error de escala total:** representa la desviación de la última transición del actual ADC con respecto del ADC perfecto. Se mide en término de LSB o voltios.
- **Error total de desajuste:** representa la máxima desviación entre la función de transferencia del ADC actual y la ideal.
- **Número efectivo de bits:** número con el cual el ADC se comporta como un ADC perfecto.
- **No linealidad diferencial:** máxima y mínima diferencia de anchura de un paso entre la actual función de transferencia y la función de transferencia del ADC perfecto.
- **No linealidad integral:** diferencia máxima vertical entre la curva actual y la ideal.

- **Rango dinámico libre de espurios:** rango dinámico entre el armónico y el mayor espurio.
- **Relación señal a ruido:** la relación entre niveles de potencia entre la señal de salida con respecto del ruido que posee esta, medido en dB.
- **Sobre muestreo:** proceso que consiste en muestrear a mayor frecuencia que la frecuencia de Nyquist.
- **Static Free Software:** empresa que desarrollo software Electric VLSI Design System.
- **Verilog:** lenguaje de descripción de hardware, HDL, se usa en el diseño de sistemas electrónicos.
- **VHDL:** lenguaje de descripción de hardware utilizado para el diseño de circuitos digitales. Definido por la IEEE. Su nombre proviene de VHSIC (Very High Speed Integrated Circuit) y HDL (Hardware Description Language).