

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

**Diseño de Memoria RERAM con Verilog-A para aplicación en
circuitos ULG y Full Adder**

Ulices Andrés Espinoza Terán

Ing. Electrónica y Automatización

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero Electrónico

Quito, 20 de mayo de 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

**Diseño de Memoria RERAM con Verilog-A para aplicación en circuitos
ULG y Full Adder**

Ulices Andrés Espinoza Terán

Nombre del profesor Tutor: Luis Miguel Procel, PhD

Quito, 20 de mayo de 2024

© DERECHOS DE AUTOR

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en la Ley Orgánica de Educación Superior del Ecuador.

Nombres y apellidos: Ulices Andrés Espinoza Terán

Código: 00213956

Cédula de identidad: 1205361486

Lugar y fecha: Quito, 20 de mayo de 2024

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en <http://bit.ly/COPETHeses>.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on <http://bit.ly/COPETHeses>.

RESUMEN

Este proyecto se centra en el diseño de una memoria ReRAM basada en HfO₂, utilizando Verilog-A para modelar las propiedades del dispositivo, con un enfoque particular en su capacidad para almacenar datos y realizar operaciones lógicas. Además, se explora cómo la incorporación de funciones analógicas en los parámetros de diseño puede potenciar la eficiencia de una memoria ReRAM en un modelo compacto, aportando nuevas posibilidades para la mejora del rendimiento. Se analiza y optimiza un modelo compacto estático, destacando su contribución al estudio de circuitos lógicos digitales basados en memorias ReRAM. Se considera su implementación en circuitos de puertas lógicas digitales (ULG) y en circuitos de sumador completo, utilizando el PDK de 90[nm] de Synopsys. Los resultados demuestran que la conmutación entre el estado de alta resistencia (HRS) y el estado de baja resistencia (LRS) en los memristores, afecta el voltaje de salida y el rendimiento, a pesar de la optimización del factor de aspecto y el balance de transistores en los buffers para restaurar las señales de los circuitos lógicos.

Palabras clave: ReRAM, Funciones Analógicas, Modelo Compacto, Circuitos Lógicos, Puertas Lógicas, ULG, Sumador Completo, PDK, HRS, LRS.

ABSTRACT

This project focuses on the design of an HfO₂-based ReRAM memory, using Verilog-A to model the properties of the device, with particular emphasis on its ability to store data and perform logic operations. In addition, we explore how incorporating analog functions into the design parameters can increase the efficiency of a ReRAM memory in a compact model, providing new possibilities for performance improvement. A static compact model is analyzed and optimized, highlighting its contribution to the study of digital logic circuits based on ReRAM memories. Its implementation in digital logic gate (ULG) and full adder circuits is considered using the Synopsys 90[nm] PDK. The results show that the switching between the high resistive state (HRS) and low resistive state (LRS) in the memristors affects the output voltage and the performance, despite the optimization of the aspect ratio and the transistor balance in the buffers to restore the signals in the logic circuits.

Key words: ReRAM, Analog Functions, Compact Model, Logic Circuits, Logic Gates, ULG, Full Adder, PDK, HRS, LRS.

TABLA DE CONTENIDO

1. Introducción.....	10
2. ANTECEDENTES.....	12
3. DISEÑO DE MODELO RERAM.....	15
4. LÓGICA RACIONAL DE MEMRISTORES	20
5. SIMULACIÓN Y RESULTADOS.....	25
5.1.1. Simulación: Lógica de Operación	25
5.1.2. Resultados: Retardo y Consumo de energía	26
5.2.1. Simulación: Optimización y Lógica de Operación	28
5.2.2. Resultados: Retardo y Consumo de energía	30
6. Conclusiones Y TRABAJO FUTURO	31
7. Referencias bibliográficas.....	33
8. Anexo A: Código verilog-a de una reram basada en hfo2	34

ÍNDICE DE TABLAS

Tabla #4.1.3. Tabla de Verdad del Circuito de ULG.	22
Tabla #4.2.3. Tabla de Verdad del Circuito Full Adder.	24
Tabla #5.1.2. a. Parámetros de Tiempo para el Circuito ULG.	27
Tabla #5.1.2. b. Parámetro de Consumo de Energía Promedio el Circuito ULG.	27
Tabla #5.2.2. a. Parámetros de Tiempo para el Circuito Full Adder.	30
Tabla #5.2.2. b. Parámetro de Consumo de Energía Promedio del Circuito Full Adder.	30

ÍNDICE DE FIGURAS

Figura #2.1. IV Curvas intrínsecas de RERAMS basados en HFO ₂ con arquitectura 1T1R y un área de 85x85 nm.	13
Figura #3.1. Estructura de definición de funciones analógicas en Verilog-A.	17
Figura #3.2. Simulación de una ReRAM basada en HfO ₂ de 85x85nm ² en Custom Compiler con el PDK de 90[nm] – Configuración 1T1R.	19
Figura #4.1.1. (a) Puertas lógicas AND/OR basadas en Lógica Racionalizada de Memristores (MRL). (b) Implementación del Circuito de Puerta Lógica Universal (ULG).	20
Figura #4.1.2. Simulación del Circuito ULG basado en Memristores – Etapa #1.	22
Figura #4.2.1. Circuito Full Adder basado en Memristores mediante conexión de 3 circuitos ULG.	23
Figura #4.2.2. Simulación del Circuito Full Adder basado en Memristores – Etapa #1.	24
Figura #5.1.1. Simulación del Circuito ULG basado en Memristores – 4 Etapas.	26
Figura #5.2.1. a. Optimización del Circuito Full Adder basado en Memristores.	28
Figura #5.2.1. b. Simulación del Circuito Full Adder basado en Memristores – 4 Etapas. ...	29

1. INTRODUCCIÓN

En el ámbito de la electrónica digital, la evolución constante de la tecnología ha llevado a la necesidad de explorar nuevas formas de mejorar la eficiencia y el rendimiento de los circuitos. En este contexto, las memorias ReRAM, también conocidas como memristores, han surgido como una solución prometedora (Zhu, 2019). Estos dispositivos son capaces de combinar la funcionalidad de almacenamiento de datos y operaciones lógicas, lo que los convierte en una opción atractiva para el diseño de circuitos lógicos digitales (Ioannis & Georgios Ch., 2016).

Las memorias de acceso aleatorio resistiva (ReRAM), son dispositivos que ofrecen una serie de ventajas que las hacen ideales para su introducción en el marco de la computación no convencional (Strukov, Snider, Stewart, & Williams, 2008). Entre estas ventajas se encuentran su tamaño a nano escala, sus propiedades de almacenamiento y su compatibilidad con las estructuras CMOS. Esta compatibilidad es crucial ya que permite evitar la limitación de ancho de banda de datos entre las unidades de procesamiento y la memoria, aportando una mejora en la estructura lógica en los circuitos digitales y a la restauración de la señal mediante inversores y buffers (Ielmini, 2016). Además, estos dispositivos presentan un comportamiento de conmutación entre dos estados estables, el Estado de Alta Resistividad (HRS) y el Estado de Baja Resistividad (LRS), lo que les permite realizar operaciones lógicas y de almacenamiento (Liu, Zheng, Wang, Shen, & Liang, 2019).

En este proyecto, se utiliza Verilog-A para el diseño de modelos compactos de memristores. Verilog-A permite describir el comportamiento de un dispositivo en términos de ecuaciones matemáticas que representan las características del terminal del dispositivo. Estas ecuaciones se implementan en el bloque analógico del modelo, lo que mejora la fluidez y el funcionamiento del modelo compacto (Open Verilog International, 1996). Además, este

proyecto hace un aporte al manual de Verilog-A, con la inclusión de funciones de tipo analógico que contienen las ecuaciones que describen el funcionamiento del memristor y se conectan con el bloque analógico del modelo.

Además, se utiliza la lógica racional de memristores (MRL), en relación con el modelo compacto de la memoria ReRAM para combinar lógica y memoria en un solo dispositivo, superando las limitaciones de ancho de banda entre unidades de procesamiento y memoria. Los MRL utilizan la resistencia programable de los dispositivos memristivos para computar funciones booleanas de AND/OR, integrándose eficientemente con la lógica CMOS estándar (Liu, Zheng, Wang, Shen, & Liang, 2019). Dicha lógica sirve como complemento para la implementación circuitos de puertas lógicas digitales (ULG) y circuitos de sumador completo mediante el kit de diseño de procesos (PDK) de 90[nm] de Synopsys, una herramienta esencial para la fase de simulación y optimización que permite evaluar el retardo y el consumo energético de los circuitos (Synopsys, 2021).

Este trabajo está organizado de la siguiente forma: En la Sección 2, se presentan los antecedentes experimentales de la memoria ReRAM, el uso de Verilog-A para el diseño de modelo compacto y la MRL para analizar circuitos lógicos mediante el PDK. En la Sección 3, se explica el proceso de adaptación, caracterización y calibración del modelo del memristor de LT-spice a Verilog-A junto al funcionamiento del código. En la Sección 4, se explica la lógica racional del memristor junto a la influencia de la transición entre LRS y HRS en la operación de circuitos de puerta de lógica universal (ULG) y circuitos de sumador completo. En la Sección 5, se presenta el proceso de optimización de los circuitos lógicos empleados y se discuten los resultados de simulación en términos de retardo y consumo de energía. Finalmente, en la Sección 6 se resume el documento y se plantea el trabajo futuro del proyecto.

2. ANTECEDENTES

La evolución de la tecnología electrónica ha llevado a la exploración y desarrollo de nuevas formas de mejorar la eficiencia y el rendimiento de los circuitos. En este sentido, los antecedentes de este trabajo se centran en recopilar el historial de cuatro áreas clave que han demostrado ser prometedoras en la búsqueda de soluciones innovadoras. Estas áreas son: el modelo ReRAM basado en HfO₂, el modelo compacto mediante Verilog-A, la lógica racional de memristores y el kit de diseño de procesos.

2.1. Modelo ReRAM basado en HfO₂

Teniendo en cuenta que el modelo se basa en la tecnología de memoria resistiva, el desarrollo y la implementación de modelos compactos para dispositivos memristivos han sido un área de investigación activa en los últimos años. Miranda (2015) presentó un modelo compacto para las curvas I-V en dispositivos memristivos no lineales de conmutación resistiva bipolar. Este modelo, que se basa en la ecuación del diodo generalizada y la ecuación diferencial de Duhem, ha sido fundamental para entender el comportamiento de los memristores y su aplicación en el diseño de circuitos.

Posteriormente, Miranda y Fröhlich (2019) propusieron un modelo compacto para dispositivos de conmutación resistiva complementaria (CRS) utilizando memdiodos. Este trabajo introdujo una modificación en el modelo original de memdiode para tener en cuenta el efecto snapback (SB) en la conmutación resistiva, lo cual ha demostrado ser crucial para una representación precisa de la conmutación en dispositivos CRS.

En el trabajo reciente de Guitarra, Taco, Gavilánez, Yépez, & Espinoza (2023), se utilizó un modelo de memristor calibrado con datos experimentales de ReRAMs basadas en HfO₂. Estos dispositivos, que presentan una arquitectura de un transistor-one-resistor (1T1R), han demostrado ser particularmente útiles para la implementación de circuitos lógicos y

aritméticos. La pila de conmutación resistiva de estos dispositivos basados en HfO₂, consiste en TiN (30 nm)/HfO₂ (5 nm)/Hf (10 nm)/TiN (30 nm), tiene un área de 85x85nm², lo que permite su integración en diseños de circuitos a nanoescala.

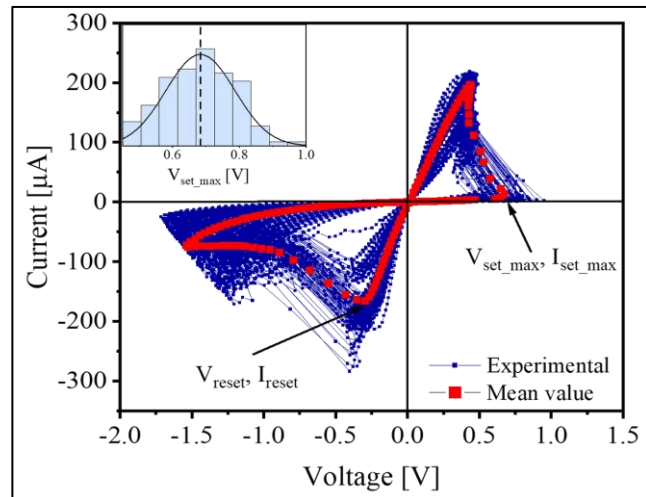


Figura #2.1. IV Curvas intrínsecas de RERAMS basados en HFO₂ con arquitectura 1T1R y un área de 85x85 nm. (Guitarra, Taco, Gavilánez, Yépez, & Espinoza, 2023)

Este trabajo proporciona un marco sólido para la implementación de memristores en circuitos lógicos y aritméticos, y destaca la importancia de la calibración precisa del modelo de memristor para lograr un rendimiento óptimo. En este sentido, el proyecto actual busca explorar aún más las posibilidades de los memristores basados en HfO₂ en modelo compacto para el estudio de circuitos digitales, tales como: ULG y Full Adder y así aprovechar los avances realizados en trabajos anteriores.

2.2. Modelo Compacto mediante Verilog-A

La propuesta de diseño de un memristor en modelo compacto se vincula a la investigación y estudio de un lenguaje de descripción de hardware (HDL) conocido como Verilog A. Sin embargo, Miranda y Fröhlich (2019) describen el modelo compacto como la representación simplificada de un circuito electrónico que describe su comportamiento de

manera precisa y eficiente, lo cual resulta útil en la predicción del comportamiento del dispositivo tanto a nivel de circuito como simulación. Por tal motivo, el diseño de la estructura se basa en manual de Open Verilog Internacional (1996) tanto para el manejo de palabras clave en la recreación del módulo principal del memristor como para la inclusión de funciones externas al bloque analógico. Cabe mencionar que, el manual no recomienda el uso de funciones externas al bloque analógico debido a la cantidad de errores que presentan en la sintaxis del código de Verilog-A.

2.3. Lógica Racional de Memristores

La lógica racional de memristores (MRL) ha surgido como un método innovador para el diseño de circuitos lógicos. Esta técnica, que utiliza memristores y buffers CMOS tradicionales, permite lograr cualquier función lógica con una mayor eficiencia y estabilidad en comparación con otros métodos basados en memristores. Así mismo, permite el diseño de compuertas lógicas AND/OR con una mayor compatibilidad con la tecnología CMOS donde se evidencia una reducción del consumo de energía y del área de diseño de circuitos integrados (Liu, Zheng, Wang, Shen, & Liang, 2019). Así mismo, la investigación presenta una propuesta de circuito de puerta lógica universal (ULG) y sumador de acarreo anticipado (CLA) basados en MRL para la reducción de complejidad del circuito y recuperación de señal.

2.4. Kit de Diseño de Procesos

El kit de diseño de procesos (PDK) se aplica como una herramienta esencial para la fase de simulación y optimización que permite evaluar el retardo y el consumo energético de los circuitos debido a su variedad de librerías para el diseño de circuitos integrados (Synopsys, 2021). Cabe mencionar que, para el proyecto actual se emplea el PDK de 90 [nm] con el modelo TT25 que permite un voltaje umbral de hasta 2.5V al momento de realizar la simulación para la obtención de resultados de medición.

3. DISEÑO DE MODELO RERAM

En esta sección se detalla el proceso de adaptación de una ReRAM basada en HfO₂ procedente de un trabajo reciente de Guitarra, Taco, Gavilánez, Yépez, & Espinoza (2023) enfocado a la descripción del modelo variable de memristor en el entorno de LT-spice. Es importante recordar el planteamiento de diseñar un código de Verilog-A que permita la optimización del comportamiento del dispositivo, por lo que, se hace un énfasis en el uso de funciones analógicas como aporte principal para el diseño de modelos compactos. Por otro lado, este modelo no incluye variabilidad y lo clasifica como modelo estático para la descripción del comportamiento del dispositivo de acuerdo con la calibración de la investigación mencionada anteriormente que resulta útil para estimar el rendimiento del modelo compacto.

3.1. Código de Verilog-A y su Funcionamiento

El diseño del código de Verilog-A requiere de la introducción de operaciones matemáticas que describan el comportamiento del dispositivo. En este caso, nos vamos a guiar de dos ecuaciones clave propuestas por Miranda (2019) de las cuales destaca la ecuación característica de Corriente-Voltaje (I-V) de un solo memdiodo dada por:

$$I = \text{sgn}(V_D) * \{(\alpha R_S)^{-1} W\{\alpha R_S I_0(\lambda) \exp[\alpha(\text{abs}(V_D) + R_S I_0(\lambda))]\} - I_0(\lambda)\} \quad (1)$$

En esta ecuación, $V_D = V - I * R_{SS}$ representa la caída de tensión a través del memdiódo. La amplitud de corriente del diodo se denota como $I_0 = I_{min}(1 - \lambda) + I_{max}(\lambda)$, donde I_{min} e I_{max} son los valores mínimo y máximo de la amplitud de corriente, respectivamente. La ecuación también incluye una constante α y una resistencia en serie R_S , ambas pueden tener una dependencia similar de λ si es necesario. La ecuación utiliza el valor absoluto de la tensión $\text{abs}(V_D)$ y la función signo sgn . Esta ecuación es la solución de una

estructura dieléctrica simétrica con resistencia en serie R_s , y utiliza la función de Lambert W . A medida que I_0 cambia de I_{min} a I_{max} y viceversa, la curva I-V cambia su forma de exponencial (HRS) a lineal (LRS), como se observa experimentalmente en muchos dispositivos RS. Las corrientes inversas del diodo se desprecian por completo. La ecuación expresa que sólo un diodo dentro del memdiodo M se activa a la vez dependiendo de la polaridad de polarización aplicada (Miranda & Fröhlich, 2019).

Por otro lado, la segunda ecuación es una parte integral del modelo del memdiodo que incluye una segunda resistencia en serie, R_{ss} , además de la resistencia en serie R_s . Esta resistencia adicional puede sustituir a la resistencia en serie externa, R . En esta ecuación, λ es un parámetro de control que varía entre 0 y 1 y determina el estado de memoria del dispositivo o histerón, la cual está dada por:

$$\lambda(V_D) = \min\{\Gamma^-(V_D), \max[\lambda(\overline{V_D}), \Gamma^+(V_D)]\} \quad (2)$$

Las funciones $\Gamma^+(V_D)$ y $\Gamma^-(V_D)$, que son funciones de cresta positiva y negativa respectivamente, definen los límites exteriores del histerón. Estas funciones dependen de las velocidades de transición η^+ y η^- , y las tensiones de set y reset, V^+ y V^- , respectivamente. El valor de V_D es la tensión un paso de tiempo antes, y este paso de tiempo es dictado por el motor de simulación LTspice. Cualquier punto dentro de esta estructura matemática representa un posible estado del dispositivo. El valor inicial utilizado en la ecuación 2, λ_0 , determina si el dispositivo comienza en el estado de baja resistencia (LRS) cuando $\lambda_0 = 1$ o en el estado de alta resistencia (HRS) cuando $\lambda_0 = 0$. Nótese que el valor de λ calculado a partir de la ecuación 2 proporciona el estado del sistema, que se introduce en la ecuación 1 para calcular la corriente (Miranda & Fröhlich, 2019).

Una vez entendido el modelo matemático que describen el comportamiento del memristor que se quiere adaptar en Verilog-A se aplica el manual de Open Verilog

Internacional (1996) debido a la estructura del código debido a su carácter sistemático donde el orden afecta los resultados de diseño. Es así como, partiendo de las instrucciones de diseño proporcionadas por el manual de Verilog-A para un dispositivo en modelo compacto y análisis de errores posibles de sintaxis se estableció un código de Verilog-A funcional de una ReRAM basada en HfO₂, el cual se encuentra en el Anexo A.

El funcionamiento de este código se basa en la definición del módulo memristor, el cual tiene cuatro terminales: *p*, *n*, *H*, *out*. Donde *p* y *n* son las entradas y salidas, mientras que *H* y *out* son solo salidas. Todas estas señales son de tipo eléctrico. Luego, se definen los parámetros del modelo de la ReRAM, como las corrientes de encendido y apagado (*ion* y *ioff*), los voltajes de ajuste y reinicio (*vset* y *vres*), las tasas de transición de ajuste y reinicio (*nset* y *nres*), entre otros. Además, se declaran varias variables internas que se utilizan en el cálculo de las ecuaciones del modelo.

Cómo mencioné antes el manual no recomienda el uso de funciones debido a sus numerosos problemas de sintaxis que pueden resultar un desafío sin una búsqueda previa y siguiendo la estructura de definición que propone el manual. Sin embargo, esta investigación comprueba que es posible definir funciones fuera del bloque analógico siguiendo la siguiente estructura:

```

/* Auxiliary Function Block -----
// ExtraFunction GGD(x), Recreation for Verilog-A
analog function real GGD;
  input x, y, z, k;
  real x, y, z, k;
  begin
    GGD = 1/(1+exp(1.702*ln(1-k*z*(x-y))/k));
  end
endfunction

```

Figura #3.1. Estructura de definición de funciones analógicas en Verilog-A.

La Figura #3.1 muestra la estructura empleada bloque de funciones auxiliares, donde se definen varias funciones que se utilizan en el bloque analógico. Por ejemplo, la función *GGD* es una función de densidad de probabilidad generalizada Gaussiana, y *vsetOrVT* determina si se debe usar *vset* o *VT* en la función $S(x)$. Entonces, para lograr la correcta sintaxis de código, es necesario permitir la conexión de las funciones externas con el bloque con la palabra clave *analog* y el tipo de dato que va a trabajar, en este caso *real*. Acto seguido, es necesario definir las variables de entrada con *input* para luego definir el tipo de dato que van a manejar estas variables de estado, en este caso *real*.

Sin embargo, el manual de Verilog-A establece que el tipo de dato que deben manejar las variables de entrada debe ser de tipo *electrical*, pero esto genera un error de sintaxis debido a la definición previa de los nodos eléctricos del dispositivo ya que redefinirlos no está permitido. Posteriormente, para colocar las ecuaciones matemáticas que representen el comportamiento del dispositivo es necesario inicializar/terminar dicha sección con *begin/end* y de igual forma para el final de la función con *endfunction*.

Finalmente, en el bloque analógico, que es el corazón del modelo, se calculan las ecuaciones del modelo. Aquí se calculan las variables *R*, *S*, *BH*, *CH*, *a_VH*, *RS_VH*, *I0_VH*, *lambda_V*, *sgn*, *w_x*, *BF* y *BFxRSS*. Estas variables se utilizan para calcular las corrientes y voltajes en el dispositivo como se evidenció en las ecuaciones (1) y (2). En resumen, este código modela el comportamiento de una ReRAM basada en HfO2 utilizando Verilog-A.

3.2. Simulación del Modelo ReRAM

Una vez adaptado el modelo compacto del memristor se procede con la recreación del modelo ReRAM con arquitectura de un transistor y una resistencia (1T1R). Cabe mencionar que, conectar el memristor a una resistencia es un circuito equivalente, pues la resistencia externa representa la influencia del transistor en la configuración 1T1R. (Guitarra, Taco,

Gavilánez, Yépez, & Espinoza, 2023). Además, mediante el uso de la herramienta de diseño de circuitos integrados PDK de 90[nm] se obtuvieron los siguientes resultados:

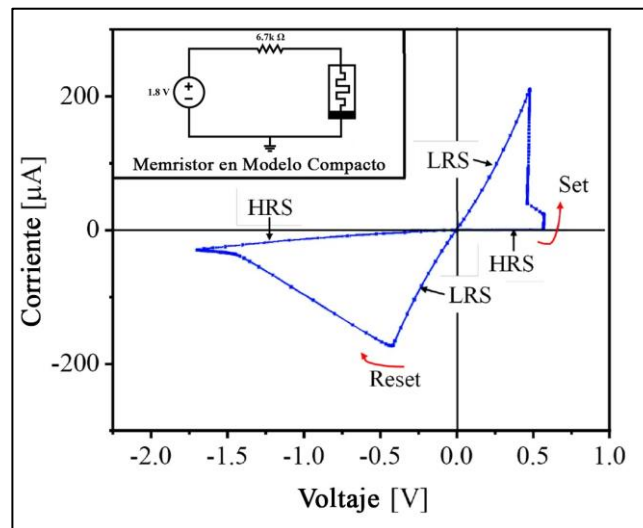


Figura #3.2. Simulación de una ReRAM basada en HfO₂ de 85x85nm² en Custom Compiler con el PDK de 90[nm] – Configuración 1T1R.

La Figura #3.2 evidencia el correcto diseño del modelo compacto de un memristor, pues si la comparamos con la Figura #2.1 perteneciente al modelo previo de LT-spice que se usó como base para su adaptación en Verilog-A, se comprueba la característica de conmutación resistiva complementaria entre dos estados estables del memristor. En este caso, se observa que el voltaje mínimo para efectuar estado de reset, conmutación de estado de baja resistencia (LRS) a estado de alta resistencia (HRS), es de 1.8V. Mientras que, el voltaje mínimo para efectuar estado de set, conmutación de estado de alta resistencia (HRS) a estado de baja resistencia (LRS), es de 0.5V. Finalmente, se evidencia que la adaptación del modelo de LT-spice a modelo compacto junto a la herramienta de simulación PDK de 90[nm] redujo el voltaje necesario de activación del memristor, pues en una investigación previa realizada por Guitarra, Taco, Gavilánez, Yépez, & Espinoza (2023), se necesitó un voltaje de 3.5V, mientras que, en el proyecto actual se puede trabajar desde 1.8V con resultados satisfactorios.

4. LÓGICA RACIONAL DE MEMRISTORES

En esta sección se detalla el funcionamiento lógico de operación del circuito de Puertas Lógicas Universales (ULG) y el circuito Sumador Completo (Full Adder). Así mismo, se incluye las señales de salida de estos circuitos lógicos en la primera etapa de simulación para luego comprobar su desempeño y restauración de señal para el cumplimiento de la lógica operacional.

4.1. Operación del Circuito de Puertas Lógicas Universales (ULG)

En base al diseño del modelo compacto del memristor basado en modelo compacto propuesto en la Sección 3, se procede con su aplicación en circuitos digitales. En este caso, la investigación de Liu, Zheng, Wang, Shen, y Liang (2019) referente a la aplicación de la lógica racionalizada de memristores (MRL) para el diseño de puertas lógicas AND/OR con memristores debido a su compatibilidad con la tecnología CMOS convencional y su polaridad de conexión. Además, ofrece la estructura base para el diseño de un circuito de Puertas Lógicas Universales (ULG) que se muestra a continuación.

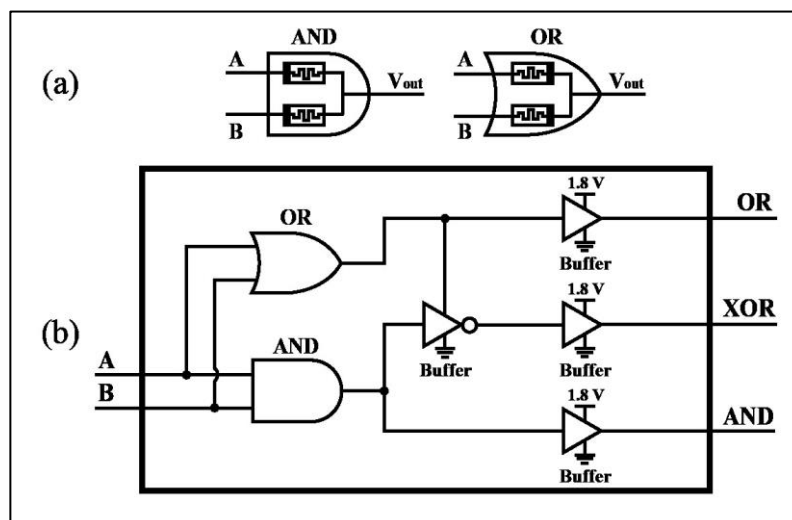


Figura #4.1.1. (a) Puertas lógicas AND/OR basadas en Lógica Racionalizada de Memristores (MRL). (b) Implementación del Circuito de Puerta Lógica Universal (ULG).

De acuerdo con la Figura #4.1.1. (a), se diseñaron puertas lógicas AND y OR mediante la conexión de dos memristores en la polarización apropiada. En la puerta AND, la salida será “1” solo si ambas entradas son “1”, lo que significa que ambas entradas deben ser verdaderas para que la salida sea verdadera. Mientras que, en la puerta OR, la salida será “1” si al menos una de las entradas es “1”, lo que significa que la salida será verdadera si al menos una de las entradas es verdadera (Liu, Zheng, Wang, Shen, & Liang, 2019).

Por otro lado, cuando las entradas son diferentes, se produce un flujo de corriente desde el terminal de alto voltaje (“1” lógico) al terminal de bajo voltaje (“0” lógico), lo que provoca un cambio en el estado del memristor. Esto significa que, si el voltaje aplicado supera el umbral positivo, el memristor cambia de un estado de alta resistencia (HRS) a un estado de baja resistencia (LRS). Por otro lado, si el voltaje supera el umbral negativo, el sistema cambia de LRS a HRS (Liu, Zheng, Wang, Shen, & Liang, 2019).

Al integrar las estructuras de las puertas lógicas AND y OR del diseño lógico MRL con transistores PMOS y NMOS, se puede implementar un circuito de Puerta Lógica Universal (ULG) basada en memristores de manera eficiente, pues los memristores aportan conmutación resistiva al circuito, lo que resulta en un multiplexor con la salida de tres puertas lógicas (AND, OR y XOR) (Teimoori, Ahmadi, Alirezaee, & Ahmadi, 2016). Sin embargo, al momento de realizar la simulación del circuito se evidenció un problema en la restauración de las señales AND y OR del circuito que provocaban una salida de voltaje inferior al voltaje de entrada de 1.8V, lo cual anula el funcionamiento de la señal XOR.

Por lo que, se propone la inclusión de tres buffers ubicados en la salida de las compuestas lógicas para solventar los problemas salida de voltaje de estas, lo cual se evidencia en la Figura #4.1.1. (a). Esta corrección mejora significativamente el voltaje de

salida y permite la correcta restauración de señal para comprobar su funcionamiento lógico como se ve a continuación.

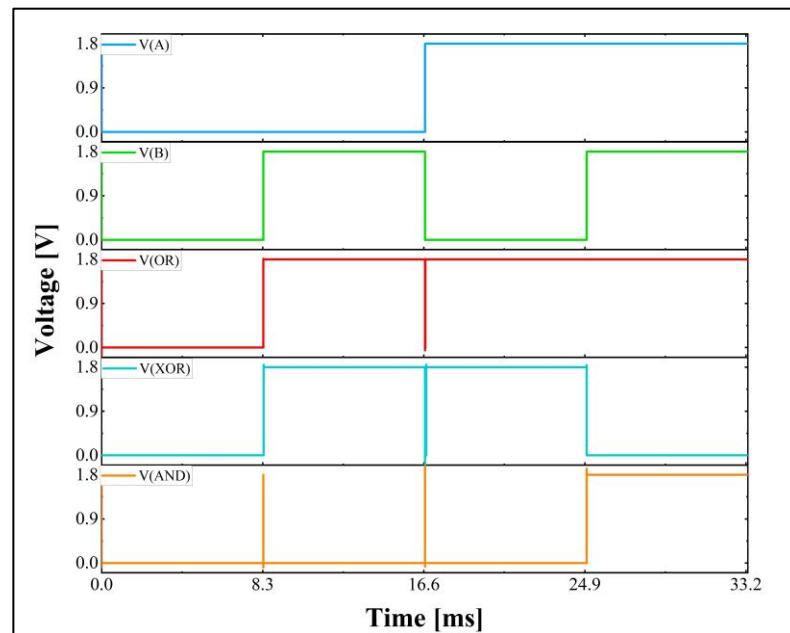


Figura #4.1.2. Simulación del Circuito ULG basado en Memristores – Etapa #1.

A partir del análisis del comportamiento de las puertas lógicas AND, OR y XOR de la Figura #4.1.2 se puede evidencia una salida de voltaje estable, que varía entre un voltaje 0V y 1.8V (“0” lógico y “1” lógico respectivamente), el cual se mantiene estable con el paso del tiempo. Adicionalmente, se realizó la siguiente tabla de verdad que comprueba el correcto funcionamiento lógico del circuito de Puerta Lógica Universal (ULG) (Teimoori, Ahmadi, Alirezaee, & Ahmadi, 2016).

A	B	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Tabla #4.1.3. Tabla de Verdad del Circuito de ULG.

4.2. Operación del Circuito Sumador Completo (Full Adder)

En base al diseño del circuito de Puerta Lógica Universal (ULG) es posible el desarrollo de circuitos integrados más complejos, tal es el caso de un circuito Sumador Completo (Full Adder), el cual se basa en la conexión de tres circuitos ULG. Este circuito Full Adder tiene tres entradas (A, B, Cin), que representan dos números binarios que se sumarán y el acarreo de la suma anterior respectivamente. Además, este circuito cuenta con dos salidas (S y Cout), que representan el bit de suma de los números de entrada (S) y el acarreo de salida que se llevará a la siguiente etapa de suma si es necesario (Cout) (Teimoori, Ahmadi, Alirezaee, & Ahmadi, 2016). La implementación de este circuito se muestra a continuación:

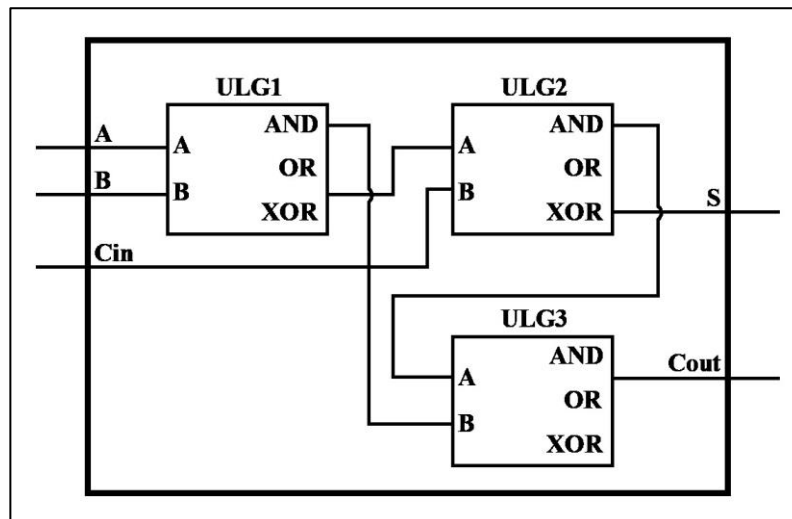


Figura #4.2.1. Circuito Full Adder basado en Memristores mediante conexión de 3 circuitos ULG.

La Figura #4.2.1 muestra la implementación del circuito de Sumador Completo (Full Adder) cuyo funcionamiento se basa en la interconexión de tres circuitos ULG donde se evidencia que para realizar la operación de suma binaria de los números de entrada, cada ULG realiza una parte de la operación de suma y los resultados se combinan para obtener la señal de salida con la suma completa (Teimoori, Ahmadi, Alirezaee, & Ahmadi, 2016).

Dicha lógica de operación se evidencia en la correcta restauración de señal del circuito vista a continuación.

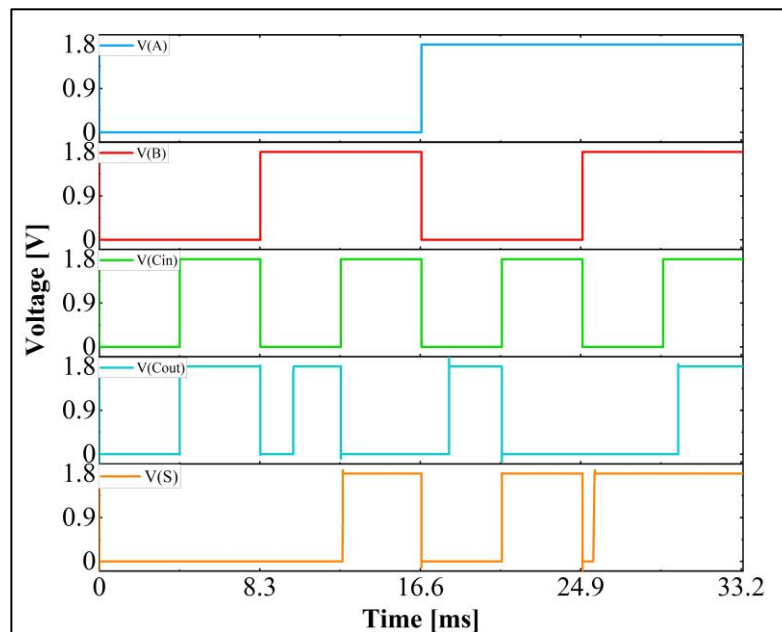


Figura #4.2.2. Simulación del Circuito Full Adder basado en Memristores – Etapa #1.

A partir del análisis del comportamiento de las señales de salida S y C_{out} de la Figura #4.2.2 se puede evidencia una salida de voltaje estable, que varía entre un voltaje 0V y 1.8V (“0” lógico y “1” lógico respectivamente), el cual se mantiene estable con el paso del tiempo. Adicionalmente, se realizó la siguiente tabla de verdad que comprueba el correcto funcionamiento lógico del circuito de Sumador Completo (Full Adder) (Guitarra, Taco, Gavilánez, Yépez, & Espinoza, 2023).

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Tabla #4.2.3. Tabla de Verdad del Circuito Full Adder.

5. SIMULACIÓN Y RESULTADOS

En esta sección se muestran los resultados de simulación de acuerdo funcionamiento lógico de operación del circuito de Puertas Lógicas Universales (ULG) y el circuito Sumador Completo (Full Adder) en cuatro etapas para evidenciar cambios en el rendimiento de los circuitos, siendo la etapa cuatro el punto de estabilización del circuito con el pasar del tiempo. Así mismo, se detalla el proceso de optimización del circuito Sumador Completo (Full Adder) y los resultados obtenidos de retardo y consumo de energía de ambos circuitos.

5.1. Circuito de Puertas Lógicas Universales (ULG)

La obtención de resultados de medición se obtuvo con la herramienta de kit de proceso de diseños PDK de 90[nm] con el modelo TT25 que permite simulaciones con un máximo de 2.5V. Sin embargo, para la recolección de datos se va a ser lo más fiel posible al modelo compacto diseñado previamente que trabaja con 1.8V.

5.1.1. Simulación: Lógica de Operación

Los resultados de simulación evidencian el cumplimiento de la lógica de operación del circuito de Puertas Lógicas Universales propuesto en la Tabla #4.1.3. Sin embargo, es importante destacar un leve aumento de retardo en el circuito. Este comportamiento va según lo esperado, pues el modelo de la memoria ReRAM basada en HfO₂ presenta conmutación resistiva complementaria (CRS), la cual evidencia cómo estas estructuras basadas en memristores alternan entre un estado de alta resistencia (HRS) y un estado de baja resistencia (LRS) para generar una ventana conductora entre dos voltajes umbral. Además, la conmutación entre los estados HRS y LRS se logra mediante procesos de Set (transición de HRS a LRS) y procesos de reset (transición de LRS a HRS) indispensables para la durabilidad y resistencia a varios ciclos lectura y escritura del memristor. (Miranda & Fröhlich, 2019)

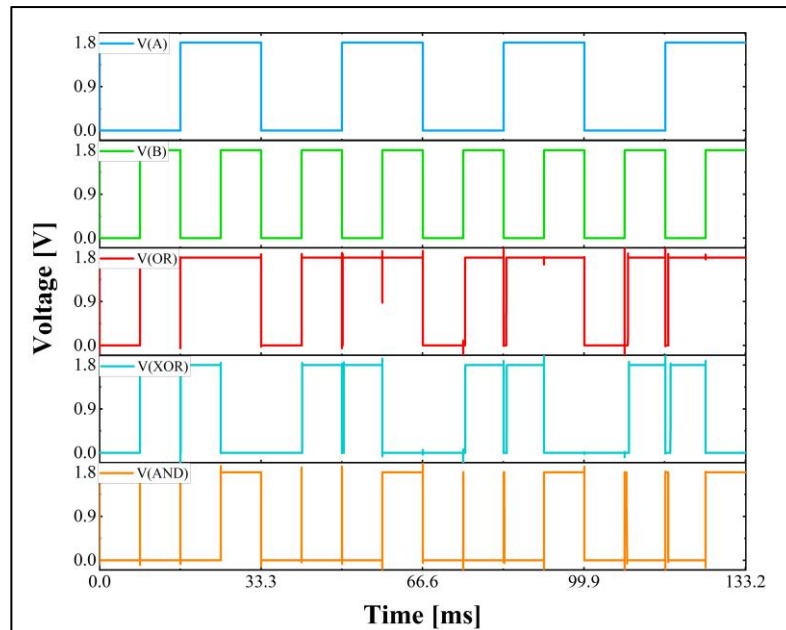


Figura #5.1.1. Simulación del Circuito ULG basado en Memristores – 4 Etapas.

La Figura #5.1.1 evidencia el correcto funcionamiento del circuito de Puertas Lógicas Universales (ULG) y su correcta restauración de señal debido a los buffers incluidos en la salida de las compuertas AND, OR y XOR. Cabe mencionar que, se evidencia un aumento de retardo en las señales de salida hasta alcanzar la etapa de optimización del sistema que se evidencia en la cuarta etapa de simulación que es el punto óptimo donde el circuito evidencia durabilidad y resistencia a varios ciclos lectura y escritura del memristor.

5.1.2. Resultados: Retardo y Consumo de energía

La medición de retardo se realizó en la cuarta etapa de simulación donde se evidencia la presencia notoria de retardo del circuito. Los resultados obtenidos se basan en tres aspectos importantes donde t_r detalla el tiempo de subida (de “0” a “1” lógico) desde el 10% hasta el 90% del voltaje de entrada. Mientras que, t_f detalla el tiempo de bajada (de “1” a “0” lógico) desde el 90% hasta el 10% del voltaje de entrada. Finalmente, el tiempo de retardo de propagación t_{pd} se mide desde el 50% del voltaje de entrada hasta el 50% del voltaje de salida, en este caso 0.9V.

	t_r [μs]	t_f [μs]	t_{pd} [μs]
AND	8.75×10^{-5}	1.77×10^{-4}	636
OR	31.8	2.25×10^{-4}	653
XOR	110.1	5.98×10^{-5}	1100

Tabla #5.1.2. a. Parámetros de Tiempo para el Circuito ULG.

La Tabla #5.1.2. a. evidencia que la compuerta AND tiene el tiempo de subida más corto de 8.75×10^{-5} [μs], lo que indica que es la más rápida en cambiar de un estado lógico “0” a “1”. Por otro lado, la compuerta XOR tiene el tiempo de bajada más corto de 5.98×10^{-5} [μs], lo que significa que es la más rápida en cambiar de un estado lógico “1” a “0”. En cuanto al tiempo de retardo de propagación, que es el tiempo que tarda la señal en pasar del 50% del voltaje de entrada al 50% del voltaje de salida, la compuerta AND también es la más rápida con un tiempo de 636 [μs].

	Power [μA]
ULG	0.813

Tabla #5.1.2. b. Parámetro de Consumo de Energía Promedio el Circuito ULG.

Finalmente, la Tabla #5.1.2. b. evidencia que el consumo de energía promedio es de 0.813 [μA], lo cual es un resultado satisfactorio obtenido como resultado de la optimización del modelo del memristor durante su diseño en modelo compacto y debido a la inclusión de buffers para la correcta restauración del voltaje de salida de las compuertas lógicas.

5.2. Circuito Sumador Completo (Full Adder)

La obtención de resultados de medición se obtuvo con la herramienta de kit de proceso de diseños PDK de 90[nm] con el modelo TT25 que permite simulaciones con un máximo de 2.5V. Sin embargo, para la recolección de datos se va a ser lo más fiel posible al modelo compacto diseñado previamente que trabaja con 1.8V.

5.2.1. Simulación: Optimización y Lógica de Operación

De acuerdo, al estudio del comportamiento en simulación del circuito Sumador Completo (Full Adder), fue necesaria la aplicación de una optimización en el tamaño de los transistores para alcanzar el correcto balance entre señal restaurada del voltaje de salida, retardo y consumo de energía de este circuito. Cabe mencionar que el valor de los transistores PMOS es el doble del valor de los transistores NMOS.

Por lo que, se realizó un análisis transiente del parámetro de width (ancho) de los transistores NMOS donde se evidencia que entre más grande sean los transistores del circuito, el circuito será más rápido. Sin embargo, debido al modelo de memristores que utiliza este proyecto existe un rango umbral con los extremos de 10 [μm] y 60 [μm] evidencian un aumento de retardo, resultando en un promedio donde el valor que mantiene el balance entre señal restaurada del voltaje de salida, retardo y consumo de energía de este circuito es de 29 [μm].

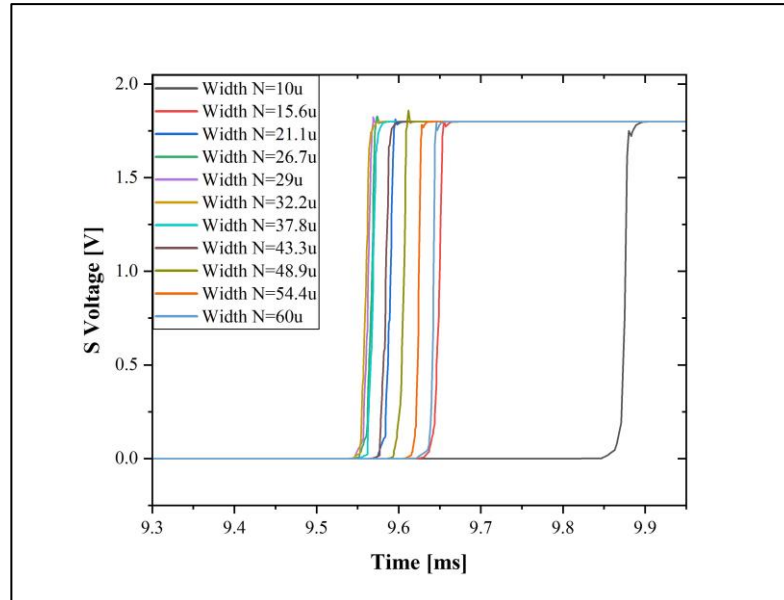


Figura #5.2.1. a. Optimización del Circuito Full Adder basado en Memristores.

Una vez optimizado el circuito Sumador Completo se evidencia el cumplimiento de la lógica de operación del circuito de Sumador Completo propuesto en la Tabla #4.2.3. Sin

embargo, es importante mencionar que el circuito acarrea el retardo visto previamente en el circuito de Puerta Lógica Universal (ULG). En este caso, se utilizaron tres ULG para el circuito Full Adder, lo cual justifica el aumento notorio del retardo incluso. Tener en cuenta que sin optimización el retardo es mucho más notorio y en algunos casos es posible perder la señal de salida debido al balance incorrecto de los transistores CMOS. El responsable de estos resultados es el modelo de la memoria ReRAM basada en HfO₂ con el que se realizó este proyecto debido a su comportamiento de conmutación resistiva complementaria (CRS) como se mencionó previamente en el circuito ULG (Miranda & Fröhlich, 2019).

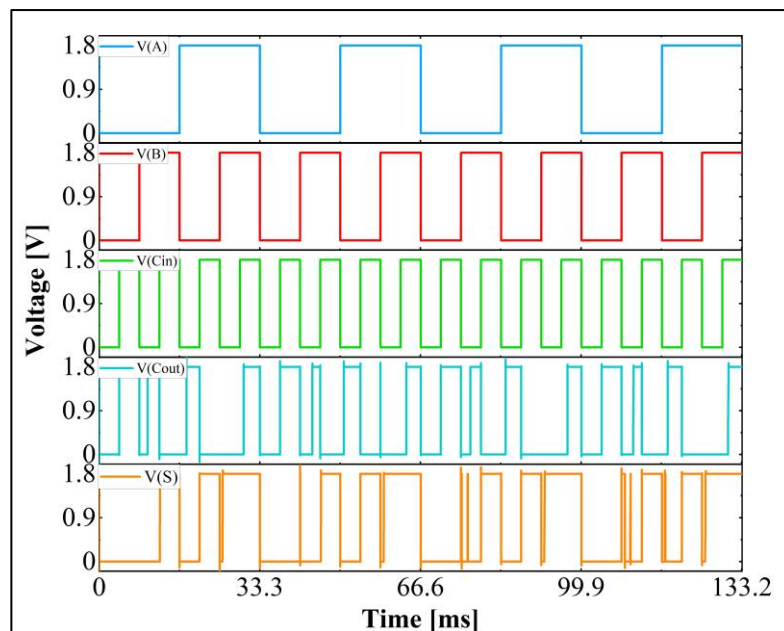


Figura #5.2.1. b. Simulación del Circuito Full Adder basado en Memristores – 4 Etapas.

La Figura #5.2.1 evidencia el correcto funcionamiento del circuito Sumador Completo (Full Adder) y su correcta restauración de señal debido a la optimización del tamaño de los transistores. Cabe mencionar que, se evidencia un aumento de retardo en las señales de salida hasta alcanzar la etapa de optimización del sistema que se evidencia en la cuarta etapa de simulación que es el punto óptimo donde el circuito evidencia durabilidad y resistencia a varios ciclos lectura y escritura del memristor.

5.2.2. Resultados: Retardo y Consumo de energía

La medición de retardo se realizó en la cuarta etapa de simulación donde se evidencia la presencia notoria de retardo del circuito. Los resultados obtenidos se basan en tres aspectos importantes donde t_r detalla el tiempo de subida (de “0” a “1” lógico) desde el 10% hasta el 90% del voltaje de entrada. Mientras que, t_f detalla el tiempo de bajada (de “1” a “0” lógico) desde el 90% hasta el 10% del voltaje de entrada. Finalmente, el tiempo de retardo de propagación t_{pd} se mide desde el 50% del voltaje de entrada hasta el 50% del voltaje de salida, en este caso 0.9V.

	t_r [μs]	t_f [μs]	t_{pd} [μs]
S	16.6	3.56×10^{-4}	1230
C_{out}	1.76×10^{-3}	0.0754	721

Tabla #5.2.2. a. Parámetros de Tiempo para el Circuito Full Adder.

La Tabla #5.2.2. a. evidencia que la salida S tiene un tiempo de subida más largo de 16.6 [μs] y un tiempo de bajada más corto de 3.56×10^{-4} [μs]. Mientras que la salida Cout tiene un tiempo de subida más corto de 1.76×10^{-3} [μs] y un tiempo de bajada más largo de 0.0754 [μs]. Sin embargo, la salida Cout tiene un tiempo de retardo de propagación más corto de 721 [μs], lo que indica que puede responder más rápidamente a los cambios en el voltaje de entrada.

	Power [μA]
Full Adder	405

Tabla #5.2.2. b. Parámetro de Consumo de Energía Promedio del Circuito Full Adder.

Finalmente, la Tabla #5.2.2. b. evidencia que el consumo de energía promedio es de 405 [μA], lo cual es un resultado satisfactorio obtenido como resultado de la optimización del tamaño de los transistores CMOS evidenciado en la Figura #5.2.1 (a) y se evidencia cómo el consumo de potencia se multiplica debido al uso de tres circuitos ULG para el funcionamiento del circuito Full Adder.

6. CONCLUSIONES Y TRABAJO FUTURO

El diseño de memorias resistivas de acceso aleatorio en un modelo compacto ha resultado ser un gran aporte para los métodos de inclusión de funciones. Este aporte radica en la aclaración sobre la definición y la invocación de funciones en Verilog-A, lo que permite realizar cálculos en el bloque analógico del circuito de manera más fluida y con un menor tiempo de ejecución. Esto resulta en una optimización del modelo LT-spice, que sirvió como base para esta investigación. Las curvas de simulación del Modelo de ReRAM basado en HfO₂ requieren de un voltaje umbral de 1.8 V para pasar de LRS a HRS (Reset) y 0.5V para pasar de HRS a LRS (Set).

El uso del método de diseño lógica racionalizada de memristores (MRL) para la implementación de puertas lógicas en el circuito ULG permite aplicar la lógica de divisor de voltaje y conocer el comportamiento de la corriente en función del voltaje umbral del Memristor. Se obtuvieron resultados satisfactorios en la optimización del circuito Full Adder, ya que se evidencia la teoría de transistores donde a mayor tamaño, mayor velocidad. Sin embargo, debido a la conmutación entre HRS y LRS de los transistores, se evidencia un umbral de optimización, siendo el valor de 29u el balance perfecto entre el tamaño del transistor, la restauración lógica de la señal y la reducción de los tiempos de retardo.

Finalmente, como trabajo futuro, se puede agregar variabilidad al diseño, dado que el proyecto actual presenta un modelo compacto estático. Este plus puede ofrecer resultados de variabilidad de la tensión de salida en el peor de los casos del circuito de Puertas Lógicas Universales (ULG) y del circuito Sumador Completo (Full Adder) para las señales de salida de cada uno. Según su comportamiento, se analiza “0” lógico o “1” lógico mediante gráficas de distribución Gaussiana (Guitarra, Taco, Gavilánez, Yépez, & Espinoza, 2023). Por otro lado, recordemos que este proyecto se basa en el uso de un PDK de 90[nm] para uso estudiantil, lo

cual abre la posibilidad de trasladar este proyecto a una tecnología diferente, más completa y con una configuración más precisa en la fabricación de dispositivos, como la tecnología del PDK de 65[nm].

Adicionalmente, este proyecto puede contribuir al diseño de futuros proyectos enfocados en el diseño de modelos compactos basados en lenguajes de descripción de hardware (HDL) como Verilog-A. Los aportes en la estructura y manejo de funciones analógicas externas que mantienen su conexión con el bloque analógico son fundamentales, ya que preservan la lógica y orden del código para programar el comportamiento de dispositivos. Este fue precisamente el caso de este proyecto con el diseño de una memoria ReRAM basada en HfO₂. Además, es importante mencionar que se pueden buscar mejoras directas en el rendimiento del modelo mediante una calibración de los parámetros de diseño. Esta calibración permitiría ajustar el comportamiento del dispositivo a las necesidades específicas de cada aplicación, optimizando así el rendimiento general del sistema. En este sentido, el trabajo realizado en este proyecto no solo proporciona una base sólida para futuras investigaciones, sino que también abre la puerta a nuevas posibilidades de exploración y mejora en el campo del diseño de circuitos basados en HDL.

7. REFERENCIAS BIBLIOGRÁFICAS

- Guitarra, S., Taco, R., Gavilánez, M., Yépez, J., & Espinoza, U. (2023). Assessment of a universal logic gate and a full adder circuit based on CMOS-memristor technology. *Solid-State Electronics*, 108704.
- Ielmini, D. (2016). Resistive switching memories based on metal oxides: mechanisms, reliability and scaling. *Semiconductor Science and Technology*, 31(6), 063002.
- Ioannis, V., & Georgios Ch., S. (2016). Emerging Memristor-Based Logic Circuit Design Approaches: A Review. *IEEE Circuits and Systems Magazine*, 16(3), 15-30.
- Kvatinsky, S., Wald, N., Satat, G., Kolodny, A., Weiser, U. C., & Friedman, E. G. (2012). MRL — Memristor Ratioed Logic. 13th International Workshop on Cellular Nanoscale Networks and their Applications, 1-6.
- Liu, G., Zheng, L., Wang, G., Shen, Y., & Liang, Y. (2019). A Carry Lookahead Adder Based on Hybrid CMOS-Memristor Logic Circuit. *IEEE Access*, 7, 43691-6.
- Miranda, E. (2015). Compact Model for the Major and Minor Hysteretic I-V Loops in Nonlinear Memristive Devices. *IEEE Transactions on Nanotechnology*, 787-789.
- Miranda, E., & Fröhlich, K. (2019). Compact Modeling of Complementary Resistive Switching Devices Using Memdiodes. *IEEE Transactions on Electron Devices*, 2831-2836.
- Open Verilog International. (1996). *Verilog-A Language Reference Manual: Analog Extensions to Verilog HDL (Versión 1.0)*. California 95032: Los Gatos.
- Puglisi, F. M., Pacchioni, L., Zagni, N., & Pavan, P. (2018). Energy-Efficient Logic-in-Memory I-bit Full Adder Enabled by a Physics-Based RRAM Compact Model. 48th European Solid-State Device Research Conference (ESSDERC), 50-53.
- SIMetrix Technologies Ltd. (2010). *SIMETRIX SPICE AND MIXED MODE SIMULATION VERILOG-A MANUAL*. Thatcham, RG18 4QN, United Kingdom: SIMetrix Technologies Ltd.
- Strukov, D. B., Snider, G. S., Stewart, D. R., & Williams, R. S. (2008). The Missing Memristor Found. *Nature*, 453(7191), 80-3.
- Synopsys. (12 de Diciembre de 2021). How DTCO Advances the Chip Design Process. Obtenido de Synopsys Blog - Chip Design: <https://www.synopsys.com/blogs/chip-design/how-dtco-advances-chip-design-process.html>
- Teimoori, M., Ahmadi, A., Alirezaee, S., & Ahmadi, M. (2016). A novel hybrid CMOS-memristor logic circuit using Memristor Ratioed Logic. 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 1-4.
- Zhu, Z. a. (2019). *A General Logic Synthesis Framework for Memristor-based logic Design*. Westminster: ICCAD, 1-8.

8. ANEXO A: CÓDIGO VERILOG-A DE UNA RERAM BASADA EN HFO2

```
// Created Wed Apr 10 01:44:44 2024

/* Libraries Definition -----*/
#include "constants.vams"
#include "disciplines.vams"

/* Module Definition -----*/
module memristor (p, n, H, out);
    inout p, n;                // Declare inputs/outputs: p, n
    output H, out;            // Declare outputs: H, out
    electrical p, n, H, out;  // Declare electrical signals: p, n,
H, out

    /* Model Parameters -----*/
    parameter real ion = 6e-4, aon = 2, ron = 1;        // Parameters Diode ON
    parameter real ioff = 0.5e-6, aoff = 0.56, roff = 1; // Parameters Diode OFF
    parameter real vset = 0.62, vres = -0.35;          // Set and Reset Voltage
    parameter real nset = 18, nres = 3.3;              // Set and Reset Transition Rates
    parameter real ISB = 2.31e-5, VT = 0.45;          // Trigger Snapback and Transition
Voltage
    parameter real kset = -1, kres = 1;                // Set and Reset Skewness
    parameter real H0 = 0, RSS = 1, RPP = 1e10;        // Initial State and Resistances
    parameter real CH0 = 1e-3;                        // Memory Capacitor

    /* Internal Variables -----*/
    real R = 0;
    real vset_VT = 0;
    real S = 0;
    real BF = 0;
    real BH = 0;
    real CH = 0;
    real BH_CH = 0;
    real a_VH = 0;
    real RS_VH = 0;
    real IO_VH = 0;
    real lambda_V = 0;
    real sgn = 0;
    real w_x = 0;
    real BFxRSS = 0;

    /* Auxiliary Function Block -----*/
    // ExtraFunction GGD(x), Recreation for Verilog-A
    analog function real GGD;
        input x, y, z, k;
        real x, y, z, k;
        begin
            GGD = 1/(1+exp(1.702*ln(1-k*z*(x-y))/k));
        end
    endfunction

    // Condition for vset or VT value for S(x) function
    analog function real vsetOrVT;
```

```

input BF, ISB, vset, VT;
real BF, ISB, vset, VT;
begin
  if(BF <= ISB) begin
    vsetOrVT = vset; // If BF <= ISB, vsetOrVT is equal to
vset
  end
  else begin
    vsetOrVT = VT; // If BF > ISB, vsetOrVT is equal to
VT
  end
end
endfunction

// AFunction: a(V(H))
analog function real calc_a_VH;
input V_H, aon, aoff;
real V_H, aon, aoff;
begin
  calc_a_VH = aon * V_H + aoff * (1-V_H);
end
endfunction

// AFunction: RS(V(H))
analog function real calc_RS_VH;
input V_H, ron, roff;
real V_H, ron, roff;
begin
  calc_RS_VH = ron * V_H + roff * (1-V_H);
end
endfunction

// AFunction: IO(V(H))
analog function real calc_IO_VH;
input V_H, ion, ioff;
real V_H, ion, ioff;
begin
  calc_IO_VH = ion * V_H + ioff * (1-V_H);
end
endfunction

// ExtraFunction: Lambda(V)
analog function real calc_lambda_V;
input V_pout;
real V_pout;
begin
  calc_lambda_V = (1/(1 + exp(-100*V_pout))); // Origin of State Equation
end
endfunction

// ExtraFunction: sgn(), Recreation for Verilog-A
analog function real calc_sgn;
input lambda_V;
real lambda_V;
begin

```

```

    calc_sgn = 2 * lambda_V - 1;
end
endfunction

// AFunction: w(x), Lambert Function
analog function real calc_w_x;
    input x;
    real x;
begin
    calc_w_x = abs(ln(1+x) * (1-(ln(1+ln(1+x)))) / (2+ln(1+x)));
end
endfunction

// Analog Block -----*/
analog begin
    R = GGD(V(p, out), vres, nres, kres);           // R(x)
    vset_VT = vsetOrVT(BF, ISB, vset, VT);
    S = GGD(V(p, out), vset_VT, nset, kset);       // S(x)

    BH = min(R, max(S, V(H)));                    // Device History
    CH = CH0 * ddt(V(H));
    BH_CH = BH - CH;
    V(H) <+ BH_CH;                               // Update V(H)

    a_VH = calc_a_VH(V(H), aon, aoff);           // a(V(H))
    RS_VH = calc_RS_VH(V(H), ron, roff);         // RS(V(H))
    IO_VH = calc_IO_VH(V(H), ion, ioff);         // IO(V(H))

    lambda_V = calc_lambda_V(V(p, out));         // Lambda(V)
    sgn = calc_sgn(lambda_V);                    // sgn(Lambda)
    w_x = calc_w_x(a_VH * RS_VH * IO_VH * limexp(a_VH * (abs(V(p, out)) + RS_VH * IO_VH)));
// w(x)

    BF = sgn * (1/(a_VH * RS_VH) * w_x - IO_VH); // BF - Current Potencial
    I(p, out) <+ BF;                             // I flowing from p to out, updated by
BF value
    BFxRSS = BF * RSS;
    V(out, n) <+ BFxRSS;                         // V between out and n, updated by BF
value
end
endmodule

```