

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

**A CMOS-Memristor based Logic-in-Memory circuit
implementation using a ReRAM Dynamic Model calibrated with
experimental data**

Martín Santiago Gavilánez Aguilar

Ingeniería Electrónica

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

Quito, 17 de mayo de 2024

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

**A CMOS-Memristor based Logic-in-Memory circuit implementation using
a ReRAM Dynamic Model calibrated with experimental data**

Martín Santiago Gavilánez Aguilar

Nombre del profesor, Título académico

**Luis Miguel Procel, PhD
Silvana Guitarra, PhD
Ramiro Taco, PhD**

Quito, 17 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: Martín Santiago Gavilánez Aguilar

Código: 00211960

Cédula de identidad: 1721035960

Lugar y fecha: Quito, 17 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UNPUBLISHED DOCUMENT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

Un circuito de Lógica en Memoria (LiM) basado en tecnología CMOS-memristor ha sido implementado en Synopsys Custom Compiler, utilizando memristores con un modelo dinámico calibrado con datos experimentales. Inicialmente, analizamos el comportamiento eléctrico de una muestra de ReRAM basado en HfO₂ para calibrar el modelo del memristor. Este proceso de calibración genera la variabilidad inherente reportada para estos dispositivos. Después, se implementó una arquitectura 2T2R para una operación LiM de 2 palabras de 1 bit con cuatro memristores calibrados que, según el análisis, funcionan bajo los principios de una memoria. Luego, se implementó un Amplificador de Señal para realizar operaciones lógicas AND, NAND, NOR y OR LiM entre las dos palabras almacenadas en la memoria. Finalmente, se implementó un módulo Sumador LiM para calcular una adición entre las palabras. Las salidas demuestran que las operaciones lógicas LiM y el Sumador mostraron una alta precisión, lo que implica que la variabilidad del memristor no afecta la operación deseada. El circuito LiM fue caracterizado en términos de rendimiento y potencia considerando sus variabilidades.

Palabras clave: ReRAMs, Memristor, LiM, Lógica en Memoria, Amplificador de Señal, operaciones lógicas, Full-Adder, 2T2R.

ABSTRACT

A CMOS-memristor technology-based Logic In Memory (LiM) circuit has been implemented in Synopsys Custom Compiler, employing memristors using a dynamic model calibrated with experimental data. Initially, we analyzed the electrical behavior of a sample of a HfO₂-based ReRAM for calibrating the memristor model. This calibration process generates the inherent variability reported for these devices. After, a 2T2R architecture was implemented for 2 words of 1-bit LiM operation with four calibrated memristors that, according to the analysis, work under the principles of a memory. Then, a Sense Amplifier was implemented to perform logic AND, NAND, NOR and OR LiM operations between the two words stored in memory. Finally, a LiM Full Adder module was implemented to compute a one-bit addition between the words. The outputs demonstrate that the LiM logic operations and Full Adder exhibited a high accuracy, which implies that the memristor's variability does not affect the desired operation. The LiM circuit was characterized in terms of performance and power, considering its variabilities.

Key words: ReRAMs, Memristor, LiM, Logic in Memory, Sense Amplifier, logic operations, Full-Adder, 2T2R.

TABLA DE CONTENIDO

1	Introducción	10
2	Desarrollo del Tema.....	13
2.1	Caracterización de ReRAM, modelo de simulación y calibración	13
2.1.1	Datos experimentales de ReRAM.....	13
2.1.2	Modelo dinámico de memristor	14
2.1.3	Calibración de modelo dinámico de memristor	17
2.2	Arquitectura LiM.....	18
2.2.1	Celda de bits 2T2R.....	19
2.3	Operaciones LiM.....	20
2.3.1	Amplificador de Señal	23
2.3.2	Operaciones lógicas LiM	24
2.3.3	Sumador LiM	26
2.3.4	Optimizaciones	27
2.4	Resultados.....	28
2.4.1	Robustez.....	32
2.4.2	Rendimiento.....	33
2.4.3	Consumo de potencia.....	38
3	Conclusiones	40
4	Referencias bibliográficas	42

ÍNDICE DE TABLAS

Tabla 1. Voltajes usados para simulación de circuito LiM.....	29
---	----

ÍNDICE DE FIGURAS

Figura 1. (a) Formación de filamentos conductores y (b) representación de los estados de resistencia en la curva IV de la ReRAM.....	11
Figura 2. Curvas experimentales de una muestra de 85x85 nm ² basadas en HfO ₂	13
Figura 3. Modelo dinámico de memristor en HSPICE (Aguirre et al., 2021).	15
Figura 4. Modelo dinámico de memristor que incluye variabilidad en HSPICE.	16
Figura 5. Curvas IV simuladas con modelo dinámico de memristor calibrado con datos experimentales.....	17
Figura 6. Verificación de calibración del modelo dinámico de memristor.....	18
Figura 7. Celda de bits 2T2R.	19
Figura 8. Símbolo Memristor.....	20
Figura 9. Arreglo de memoria de 2 palabras con tamaño de 1 bit.	21
Figura 10. Arreglo de memoria de 2 palabras para operaciones LiM.	22
Figura 11. Topología de Amplificador de Señal.....	23
Figura 12. Pasos evaluar operaciones LiM.	25
Figura 13. Topología e implementación de LiM-FA.....	26
Figura 14. Análisis de tamaño del transistor que habilita escritura.	28
Figura 15. Resultados de simulación para operaciones LiM AND y NAND.....	30
Figura 16. Resultados de simulación para operaciones LiM NOR y OR.	30
Figura 17. Resultados de simulación para sumador LiM.	31
Figura 18. Distribución del tiempo de retardo para el peor caso de la salida S.....	34
Figura 19. Distribución del tiempo de retardo para el peor caso de la salida C _{OUT}	34
Figura 20. Distribución del tiempo de subida para el peor caso de la salida S.....	35
Figura 21. Distribución del tiempo de subida para el peor caso de la salida C _{OUT}	36
Figura 22. Distribución del tiempo de bajada para el peor caso de la salida S.....	37
Figura 23. Distribución del tiempo de bajada para el peor caso de la salida C _{OUT}	37
Figura 24. Distribución de potencia estática del circuito LiM.....	38
Figura 25. Resultados de variabilidad de Potencia Dinámica del circuito LiM.	39

1 INTRODUCCIÓN

La rápida reducción de escala de la tecnología de semiconductor complementario de óxido metálico (CMOS) ha llevado a la búsqueda de nuevos materiales para la reducción del consumo energético, controlar límites de tamaño y la integración de dispositivos en nuevas arquitecturas para circuitos y sistemas (Zhu et al., 2019; Vourkas & Sirakoulis, 2016). En este contexto, el diseño lógico basado en memristores es un concepto emergente dentro del área de sistemas de cómputo inteligentes que podrían aumentar la densidad de dispositivos y mejorar la velocidad de funcionamiento de los circuitos (Vourkas & Sirakoulis, 2016; Ielmini, 2016).

Las memorias resistivas de acceso aleatorio, ReRAMs, también conocidas como memristores (Strukov et al., 2008), son una excelente opción para ser introducidas dentro del marco computacional no convencional gracias a sus cualidades favorables, especialmente relacionadas al tamaño de escala nanométrica, las propiedades de almacenamiento y la compatibilidad con las estructuras CMOS (Ielmini, 2016; Liu et al., 2019). Este dispositivo, propuesto por primera vez por Chua (1971), es una memoria no volátil (NVM) que tiene una estructura Metal-Aislante-Metal (MIM). Esta funciona gracias al cambio de estado resistivo que tiene un comportamiento de conmutación entre dos estados estables, el estado de alta resistencia (HRS) y el estado de baja resistencia (LRS). La conmutación se produce aplicando un potencial de polarización que provoca el movimiento de vacancias de oxígeno dentro de la capa aislante, lo cual, crea y destruye filamentos conductores (ver Fig. 1a). Estos son procesos estocásticos que se conocen como SET y RESET, y ocurren cuando el estado de resistencia pasa de HRS a LRS y de LRS a HRS, respectivamente (ver Fig. 1b). Las transiciones de estados generan fluctuaciones en la respuesta de corriente-voltaje (Ielmini, 2016).

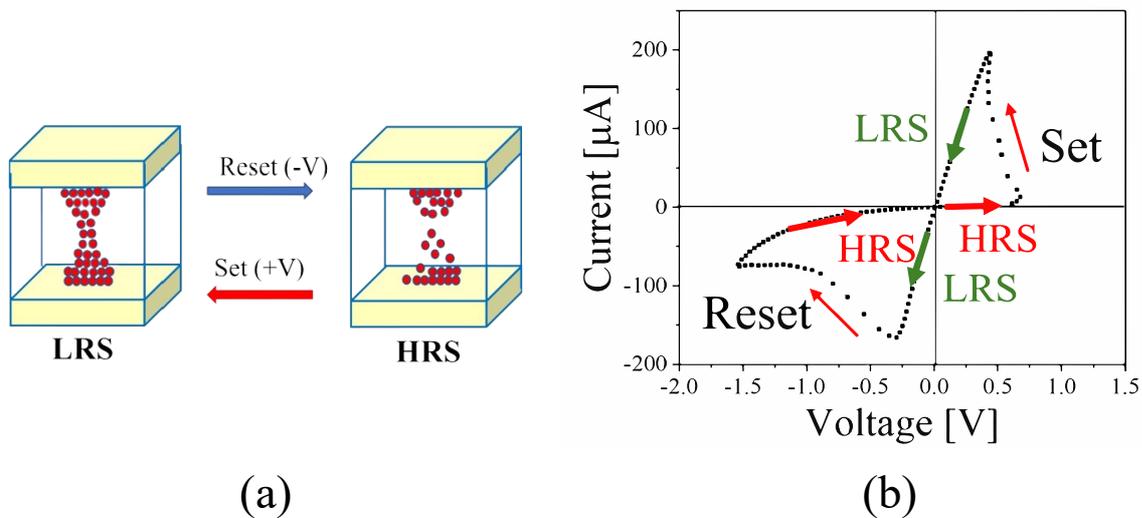


Figura 1. (a) Formación de filamentos conductores y (b) representación de los estados de resistencia en la curva IV de la ReRAM.

Debido a las atractivas características que presentan las ReRAMs, los modelos para reproducir su comportamiento son múltiples en la literatura (Panda et al., 2018). Sin embargo, un requisito crítico para modelar este dispositivo es la predictividad de su comportamiento eléctrico. En este trabajo se implementa un modelo dinámico propuesto por Aguirre et al. (2021), que como su nombre lo indica, responde a la frecuencia y es capaz de trabajar a altas velocidades. Es importante considerar la velocidad con la que trabajan los dispositivos ReRAM, ya que es superior comparada con lo usualmente usado en industria para NVM (NAND Flash) (Guitarra et al., 2021).

Como las ReRAM son compatibles con la estructura CMOS, pueden ser un elemento alternativo para el diseño de circuitos de Lógica en Memoria (LiM). Esta combinación utilizaría el uso lógico y las capacidades de memoria para evitar la clásica limitación de ancho de banda de datos entre las unidades de procesamiento y la memoria (L. O. Chua, 1971). Se han propuesto algunos diseños de LiM basados en memristores en la literatura como los propuestos por Kvatinsky et al. (2014) o (Chen et al., 2020). Por ejemplo, en

Memristor-Aided Logic (MAGIC) de Kvatinsky et al.(2014), se construyen diferentes compuertas lógicas utilizando una arquitectura de arreglos de memoria hechos de memristores. El cálculo se realiza aplicando pulsos de tensión a toda la red. Otro ejemplo es la opción es de arquitectura 2T2R de (Chen et al., 2020), en donde se usan 2 memristores y 2 transistores por celda unitaria de bit de memoria y se realizan los cálculos mediante descargas de líneas a través de memristores.

En este trabajo se implementa, optimiza y caracteriza la arquitectura 2T2R de (Chen et al., 2020) con tecnología basada en memristores y CMOS para hacer operaciones de LiM y un Sumador de 1 bit de LiM. Los memristores son simulados mediante el modelo dinámico de Aguirre et al. (2021), al cual se le incluyó la variabilidad ciclo a ciclo reportado para estos dispositivos (Guitarra et al., 2021), y que fue previamente calibrado con datos experimentales, mientras que, para los transistores se utilizó el Process Design Kit (PDK) de 65nm de TSMC. Los criterios que se tomaron en cuenta son los más importantes en este tipo de circuitos: la tasa de error, el consumo de energía y el análisis temporal. En este sentido, un circuito LiM solo puede ser confiable cuando se comprenden los efectos de las variaciones de resistencia en la tecnología de celda elegida (Fieback et al., 2022).

2 DESARROLLO DEL TEMA

2.1 Caracterización de ReRAM, modelo de simulación y calibración

2.1.1 Datos experimentales de ReRAM

Se analizó la respuesta eléctrica de ReRAMs basados en óxido de hafnio (HfO_2) con una arquitectura de un transistor-un resistor (1T1R) para calibrar el memristor utilizado en este trabajo. La estructura de conmutación resistiva, compuesta por TiN (30 nm)/ HfO_2 (5 nm)/Hf (10 nm)/TiN (30 nm), ocupa un área de $85 \times 85 \text{ nm}^2$ (Guitarra et al., 2021). IMEC proporcionó estas muestras y la caracterización eléctrica se realizó utilizando una estación de cuatro puntas, la Cascade Alessi REL 4800 en la Universidad San Francisco de Quito. Los dispositivos siguieron un proceso de electroformación, para su activación, con una corriente de cumplimiento de 5mA. El estudio involucró 120 curvas de corriente-voltaje (IV) experimentales que revelaron la variabilidad de ciclo a ciclo previamente reportada para este tipo de dispositivos (Guitarra et al., 2021). Este comportamiento se debe a la naturaleza estocástica de la conmutación resistiva.

En la Fig. 2 se puede apreciar 105 de las 120 curvas de la muestra y cómo estas varían ciclo a ciclo.

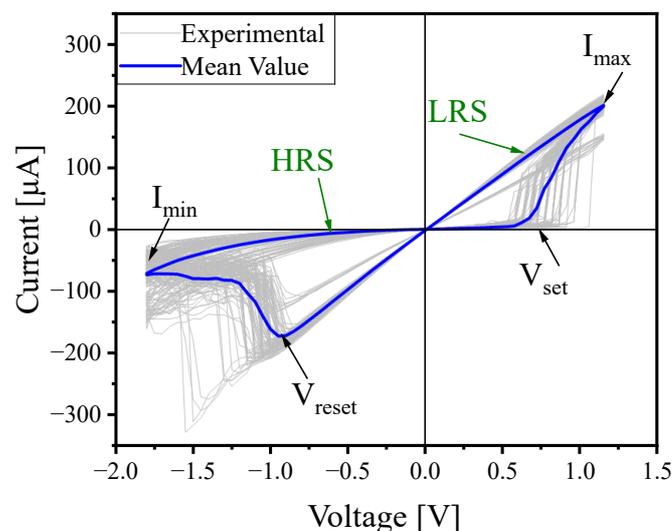


Figura 2. Curvas experimentales de una muestra de $85 \times 85 \text{ nm}^2$ basadas en HfO_2 .

Se han excluido 15 curvas que representan valores atípicos para mejorar la referencia de datos experimentales que permitirán la calibración del modelo.

Además, para la calibración se realizó un análisis estadístico de los datos experimentales para determinar los valores medios (μ) y la desviación estándar (σ). Los parámetros esenciales para la calibración están marcados en la Figura 2. Se encontró que $V_{\text{set}} = (0.74V, 0.10V)$, $V_{\text{reset}} = (-1.09V, 0.16V)$, $I_{\text{min}} = (-73.69\mu A, 33.41\mu A)$, $I_{\text{max}} = (200.30\mu A, 15.95\mu A)$, donde los valores corresponden a (μ, σ).

2.1.2 Modelo dinámico de memristor

Han surgido numerosos modelos, cada uno variando en precisión, para comprender y predecir el comportamiento de la ReRAM. Una de las aproximaciones más aceptadas es el modelo de memristor, un modelo de circuito especializado que captura las características eléctricas esenciales de los dispositivos ReRAM. En este estudio, se empleó el modelo dinámico introducido por Aguirre et al. que requiere dos ecuaciones interrelacionadas. Una ecuación para el transporte del electrón (ecuación de transporte, ET) y una segunda que tiene en cuenta el efecto de memoria (2021).

La ecuación para la característica IV se da por (1).

$$I = I_0(\lambda) [e^{\beta\alpha(\lambda)(V-IR_s(\lambda))} - e^{-(1-\beta)\alpha(\lambda)(V-IR_s(\lambda))}] \quad (1)$$

donde $I_0(\lambda) = I_{\text{min}}(1 - \lambda) + \lambda I_{\text{max}}$ es el factor de amplitud de corriente, α es un parámetro relacionado con las características particulares del mecanismo de conducción, y R_s es una resistencia en serie. La ecuación (1) es el resultado de considerar una barrera de potencial parabólica invertida con altura Φ como el dispersor de túnel (Aguirre et al., 2021). Tanto α como R_s tienen la misma dependencia en λ que I_0 . I_{min} e I_{max} son los valores mínimo y máximo de la amplitud de corriente, respectivamente, y $0 < \beta < 1$ expresa la asimetría en el potencial en los dos extremos del CF. (Aguirre et al., 2021)

En cuanto a la ecuación de memoria (EM), normalmente es una ecuación diferencial impulsada por un voltaje o una corriente (L. Chua, 2011). Esta ecuación tiene algunas dependencias no lineales, así como de la llamada función de ventana. Además cumple con una serie de requisitos como la homogeneidad dimensional, el equilibrio dinámico y el comportamiento de nivelación para grandes sesgos opuestos, con valores finales de 0 y 1 para HRS y LRS, respectivamente (Aguirre et al., 2021). Físicamente, la EM expresa el movimiento de iones/vacantes causado por el potencial aplicado externamente. (Aguirre et al., 2021)

La ecuación diferencial de primer orden más simple está dada por la ecuación (2):

$$\frac{d\lambda}{dt} = \frac{1 - \lambda}{\tau_S(V)} - \frac{\lambda}{\tau_R(V)} \quad (2)$$

“donde $\tau_{S,R}$ son tiempos característicos vinculados a los transitorios de SET y RESET (para $V > 0$ y $V < 0$, respectivamente).” (Aguirre et al., 2021).

Las expresiones de $\tau_{S,R}$ están dadas por las ecuaciones (3):

$$\tau_{S,R}(V) = \tau_{0S,0R} e^{-\frac{V}{V_{0S,0R}}} \quad (3)$$

donde $\tau_{0S,0R}$ son parámetros de calibración al igual que $V_{0S,0R}$.

```
.subckt memdiode p n
.param H0 = 0 beta = 0.5
* Parámetros de transición
.param T0s = 8.5e+3 V0s = 6.8e-2 T0r = 1e+4 V0r = 1e-1
* Parámetros I-V
.param imin = 5e-7 imax = 9.5e-5 alphamin = 1e+0 alphamax = 1
.rsmmin = 3.8e+1
+ rsmmax = 3.8e+1
* Funciones auxiliares
.param I0(x) = 'imax*x + imin*(1-x)'
.param A(x) = 'alphamax*x + alphamin*(1-x)'
.param Rss(x) = 'rsmmax*x + rsmmin*(1-x)'
*****H-V*****
EV A gnd! vol = 1
RH H A R = 'T0s*exp(-V(p,n)/V0s)'
RD H gnd! R = 'T0r*exp(V(p,n)/V0r)'
CH H gnd! C = 1 IC = 'H0'
*****I-V*****
RS p D R = 'Rss(V(H))'
GD D n cur = 'I0(V(H)) * (exp(beta*A(V(H))*V(D,n)) - exp(-(1-beta)*A(V(H))*V(D,n)))'
.ends memdiode
```

Figura 3. Modelo dinámico de memristor en HSPICE (Aguirre et al., 2021).

```

.subckt memdiode p n
.param H0=0 beta=0.3
*Variability parameters
.param
+ sigT0s=300*0.0027
+ sigV0s=0.08*3.7
+ sigT0r=150*0.004
+ sigV0r=0.108*1.5
+ sigimin=32u*16000
+ sigimax=230u
+ sigalpha=0.925*0.01
+ sigrs=700*0.001
*Variability functions parameters
.param
+ varT0s=agauss(0,sigT0s,3)
+ varV0s=agauss(0,sigV0s,3)
+ varT0r=agauss(0,sigT0r,3)
+ varV0r=agauss(0,sigV0r,3)
+ varimin=agauss(0,sigimin,3)
+ varimax=agauss(0,sigimax,3)
+ varalpha=agauss(0,sigalpha,3)
+ varrs=agauss(0,sigrs,3)
*Transition parameters
.param
+ T0s=300*(1+varT0s)
+ V0s=0.08*(1+varV0s)
+ T0r=150*(1+varT0r)
+ V0r=0.108*(1+varV0r)
*I-V parameters
.param
+ imin=32u*(1+varimin)
+ imax=230u*(1+varimax)
+ alphamin=0.925*(1+varalpha)
+ alphamax=0.925*(1+varalpha)
+ rsmin=700*(1+varrs)
+ rsmax=700*(1+varrs)
*****H-V*****
EV A gnd! vol=1
RH H A R='T0s*exp(-V(p,n)/V0s) '
RD H gnd! R='T0r*exp(V(p,n)/V0r) '
CH H gnd! C=1 IC='H0'
*****I-V*****
RS p D R='rsmax*V(H) + rsmin*(1-V(H)) '
GD D n cur='(imax*V(H) + imin*(1-V(H))) * (exp(beta*(alphamax*V(H)
+ alphamin*(1-V(H)))*V(D,n)) -exp(-(1-beta)*(alphamax*V(H) +
alphamin*(1-V(H)))*V(D,n))) '
.ends memdiode

```

Figura 4. Modelo dinámico de memristor que incluye variabilidad en HSPICE.

Las ecuaciones de este modelo fueron implementadas en HSPICE (ver Fig. 3), lenguaje que permite la simulación circuital del comportamiento del memristor y además permite la calibración con los datos experimentales.

Para añadir la variabilidad de ciclo a ciclo, se incluyó en el modelo la desviación estándar de los parámetros de calibración haciendo uso de las funciones de distribución de parámetros de HSPICE. En este caso debido al comportamiento gaussiano de los datos experimentales se

utilizó la función AGAUSS, que añade una distribución gaussiana a un parámetro a partir de un valor nominal y su variación absoluta (Synopsys Inc., 2008). El modelo resultante se puede apreciar en la Fig. 4.

2.1.3 Calibración de modelo dinámico de memristor

Para la calibración del modelo se realizaron pruebas y simulaciones paramétricas para encontrar el mejor ajuste del modelo a la forma de las curvas experimentales. Los estudios paramétricos se realizaron con un análisis en dc, ya que los datos de las curvas experimentales fueron tomados con una frecuencia muy baja, por lo que es apropiado calibrar el modelo en unas condiciones similares.

Para calibrar el modelo, en primer lugar, se tomó una curva experimental arbitraria como referencia para ajustar los parámetros del modelo y luego se incluyeron las variabilidades al modelo. De esta manera, se logró variar los estados estables de resistividad, LRS y HRS, y también los puntos de transición de estados, SET y RESET.

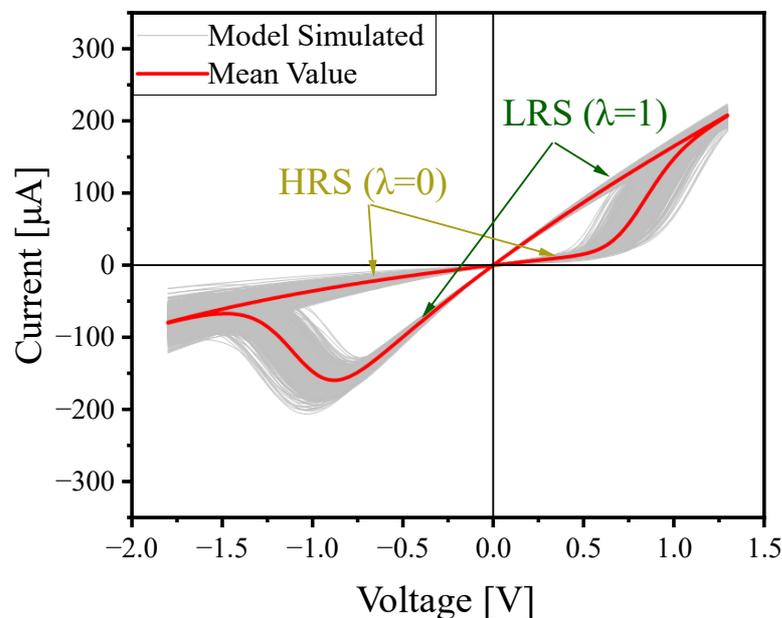


Figura 5. Curvas IV simuladas con modelo dinámico de memristor calibrado con datos experimentales.

El resultado de la calibración se puede observar en la Fig. 5, la cual muestra 1000 curvas simuladas que varían ciclo a ciclo junto con su curva promedio. Las curvas promedio de los datos experimentales y las curvas simuladas a partir del modelo dinámico calibrado, se muestran en la Fig. 6. Se puede apreciar un buen ajuste conseguido en la calibración.

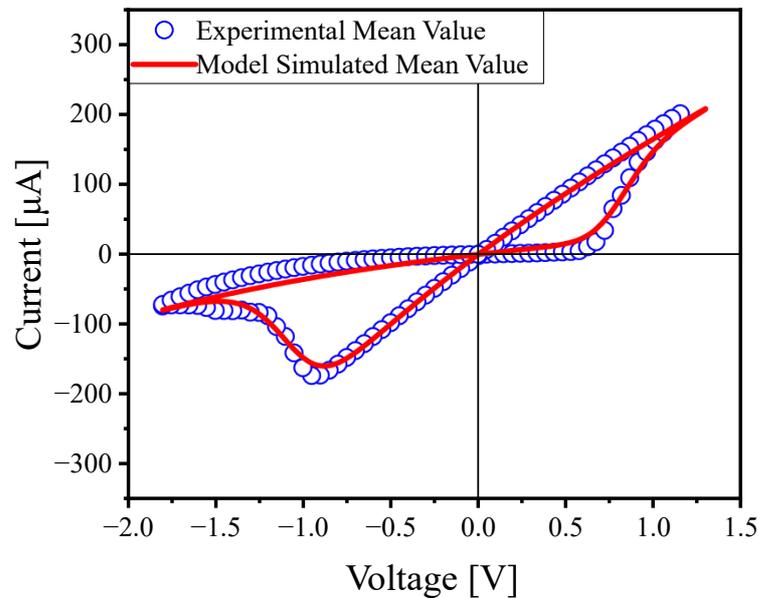


Figura 6. Verificación de calibración del modelo dinámico de memristor.

2.2 Arquitectura LiM

Las arquitecturas de Computación en Memoria (CiM) realizan cálculos directamente en el chip de memoria, lo que reduce la necesidad de mover datos hacia y desde el procesador. Estas se clasifican según el lugar donde se produce el resultado del cálculo, es decir, en la matriz de memoria (CiM-A) o en los circuitos periféricos (CiM-P) (Lebdeh et al., 2019). La lógica de Snider (2005) es un ejemplo de arquitecturas CiM-A, mientras que, lógica de Scouting (Xie et al., 2017) y la Multiplicación de Matrices Booleanas en barras cruzadas memristivas (Velasquez & Jha, 2016) son ejemplos de CiM-P. CiM-A almacena sus resultados en la matriz misma, lo que introduce una alta carga de escritura en las celdas que

limita la vida útil del circuito. En contraste, CiM-P utiliza solo operaciones de lectura para realizar cálculos y, por lo tanto, no sufre de este problema (Fieback et al., 2022).

En este trabajo se implementó una arquitectura CiM de tipo CiM-P. La arquitectura elegida es una propuesta por Chen et al., la cual es una CiM reconfigurable 2T2R que puede soportar operaciones de Memoria Ternaria de Contenido Direccional (TCAM), Producto Punto en Memoria (IM-DP) y LiM (2020). Esta última es la operación en la que se enfocará este trabajo. Además, es importante mencionar que la arquitectura 2T2R también puede funcionar como un ReRAM 1T1R convencional (arquitectura más usada en la literatura) en situaciones donde no se requieren operaciones CiM (Chen et al., 2020).

2.2.1 Celda de bits 2T2R

La Fig. 7 muestra la celda de bits 2T2R para la arquitectura propuesta Chen et al. (2020), esta consiste en dos celdas de bits 1T1R que comparten una línea de bits (BL). Cada fila en una matriz de ReRAMs comparte un par de líneas de palabra, línea de palabra izquierda (WLL) y línea de palabras de derecha (WLR), mientras que cada columna comparte una línea de bits (BL) y un par de líneas de origen, línea de origen (SL) y línea de origen complementaria (SLB).

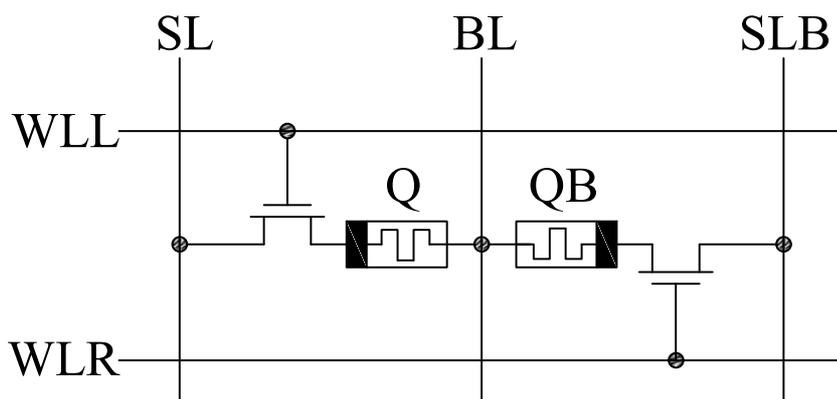


Figura 7. Celda de bits 2T2R.

Vale la pena recalcar que un dispositivo ReRAM o memristor está representado por el símbolo de la Fig. 8, en donde se muestra su doble polaridad con sus terminales p y n. Las

flechas indican la dirección en la que tiene que ir la corriente para realizar SET o RESET del dispositivo.

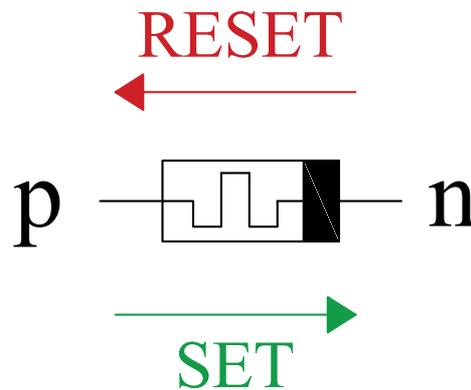


Figura 8. Símbolo Memristor.

El par de dispositivos ReRAM (Q, QB) en la celda de bits representa el dato 1 lógico con $(Q, QB) = (HRS, LRS)$ y el dato 0 lógico con $(Q, QB) = (LRS, HRS)$ (Chen et al., 2020).

Cuando la celda de bits se utiliza para almacenamiento normal como NVM, la representación complementaria proporciona beneficios significativos, como una menor tasa de errores de bits y una detección más rápida en comparación con la convencional ReRAM 1T1R (Chen et al., 2020).

2.3 Operaciones LiM

En la arquitectura 2T2R, se pueden ejecutar operaciones lógicas en memoria cuando se realiza la lectura de las palabras o bits de la palabra que contiene la memoria (Chen et al., 2020). Para el propósito de este trabajo, se usará la memoria más pequeña a considerar (un arreglo de 2 palabras de 1 bit) para realizar las operaciones lógicas. De esta manera se usarán solamente dos celdas de bits 2T2R, donde cada una representa a 1 palabra de 1 bit. Por conveniencia estas palabras se llamarán X e Y. (Ver Fig. 9)

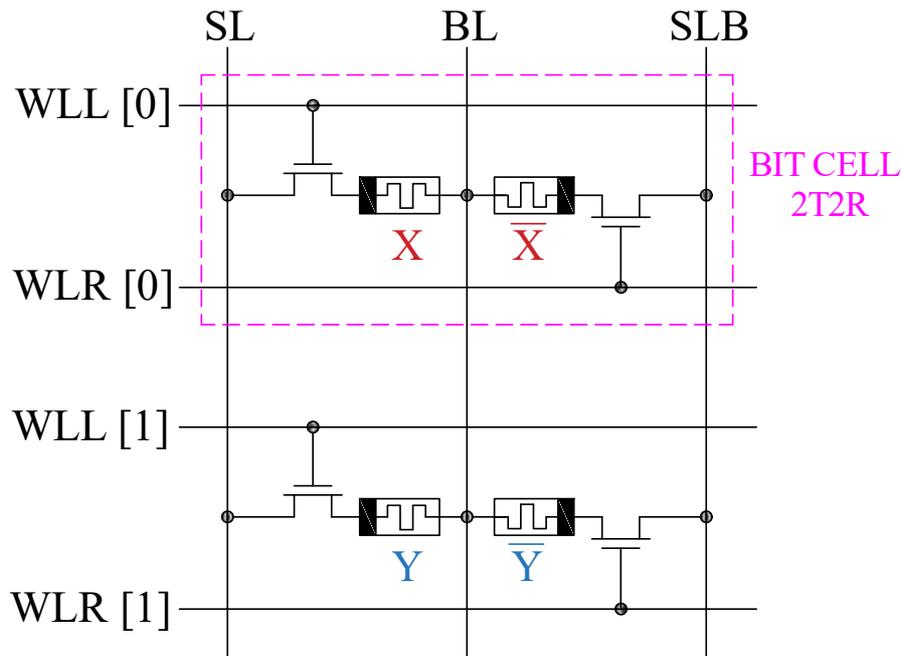


Figura 9. Arreglo de memoria de 2 palabras con tamaño de 1 bit.

Para realizar leer la memoria, se debe descargar el voltaje de lectura (V_{READ}) a través de un memristor. Esto se lo hace en 2 pasos: primero se debe precargar SL o SLB a V_{READ} , y segundo, se deja flotando SL o SLB (según sea el caso) mientras BL se conecta a la referencia y se activa WL (usando un voltaje V_{WRITE}) del memristor a leer. De esta manera, el voltaje precargado en SL o SLB se puede descargar a través del memristor que se intenta leer. Se puede conocer el estado resistivo del memristor a través de este proceso ya que si se descarga SL o SLB de manera rápida, esto significa que el estado resistivo del memristor es de LRS, y si se descarga lentamente, estará en HRS. Dependiendo de si se está midiendo el valor o su complemento se puede interpretar si es un 0 lógico o 1 lógico.

Para hacer posible estos 2 pasos del proceso de lectura se sumaron 2 circuitos al arreglo de memoria de 2-bits. Uno es el circuito de precarga y el segundo el amplificador de señal. (ver Fig. 10).

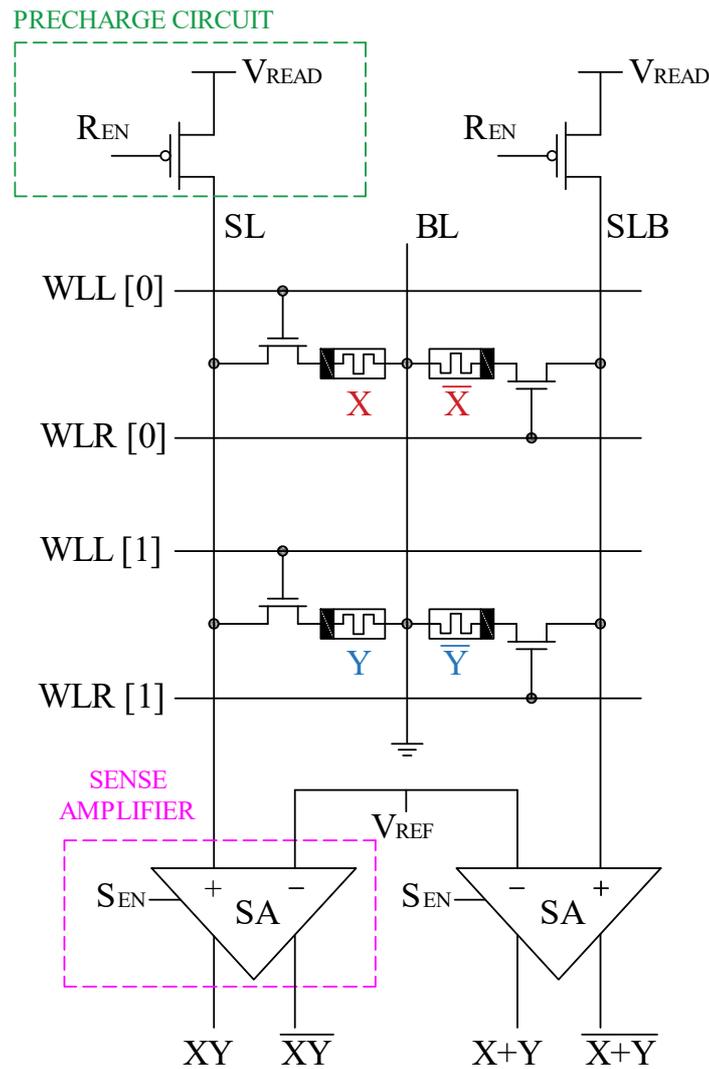


Figura 10. Arreglo de memoria de 2 palabras para operaciones LiM.

Por un lado, el circuito de precarga es un transistor PMOS, que permite precargar SL y SLB al valor del voltaje de lectura (V_{READ}) a través de una señal de habilitación de lectura R_{EN} , cuando esta se encuentra en 0L. También permite dejar flotando estos nodos una vez que se requiera realizar la lectura activando la señal R_{EN} .

Por el otro lado, el amplificador de señal (SA) detectará si la descarga fue rápida o lenta. La topología de este circuito y la manera en que esto se realiza será abordado con más profundidad más adelante en el texto.

2.3.1 Amplificador de Señal

El amplificador de señal implementado para este trabajo fue propuesto por Rahman et al., este es un comparador dinámico con enganche convencional, que es el más utilizado debido a su alta impedancia de entrada, alta velocidad y salida de oscilación completa (2014). Estas características fueron claves para la selección del SA, ya que son necesarias para la implementación de LiM de este trabajo. La topología del amplificador de señal se puede observar en la Fig. 11.

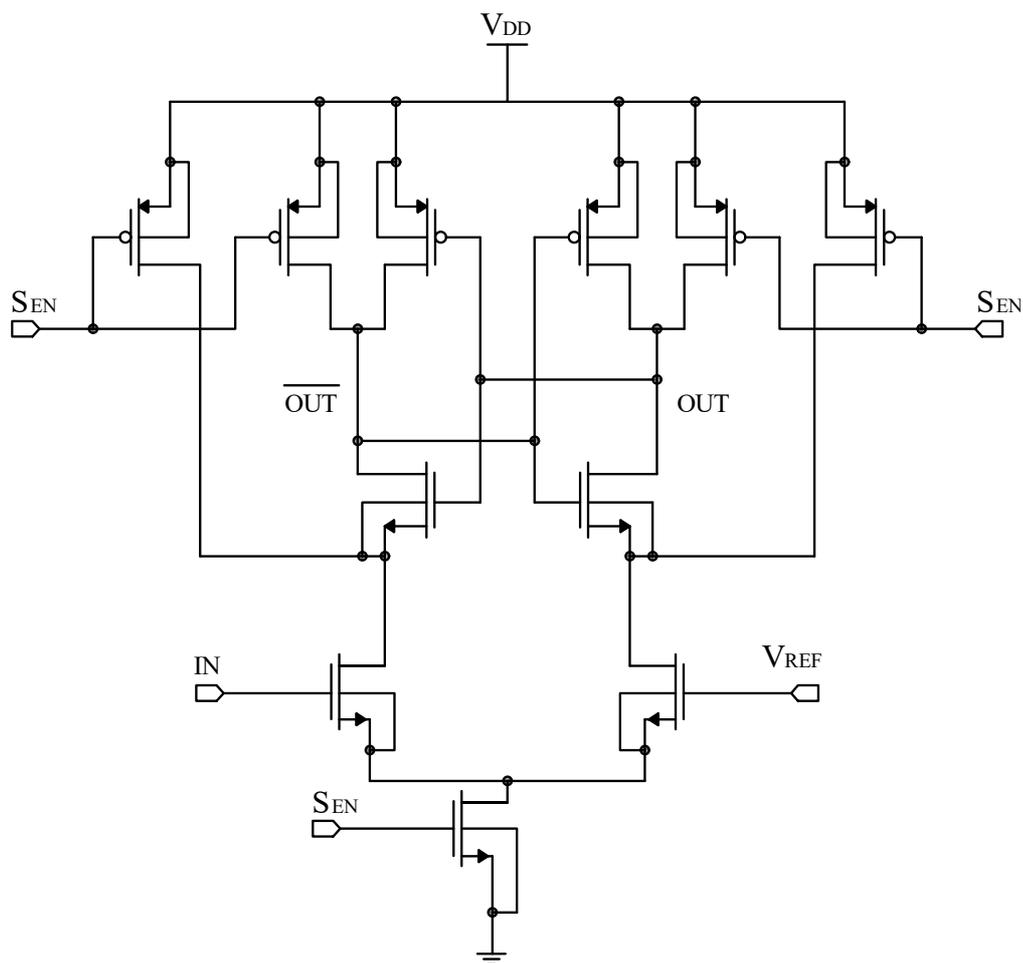


Figura 11. Topología de Amplificador de Señal.

El funcionamiento del amplificador de señal viene dado en dos etapas: la primera es de precarga y la segunda de evaluación. Para la etapa de precarga, la señal de S_{EN} debe mantenerse en 0L, con ello precarga las salidas OUT y \overline{OUT} al voltaje nominal V_{DD} .

Posteriormente en la etapa de evaluación, S_{EN} debe cambiar a 1 lógico (1L) permitiendo la descarga de los nodos precargados. Estos se descargarán dependiendo de si el voltaje en la entrada IN es mayor al voltaje de referencia (V_{REF}). En el caso de ser mayor, el nodo de \overline{OUT} se descargará más rápido que OUT y por tanto, se habilita el PMOS que carga el nodo OUT a 1L y como tenemos inversores cruzados se mantendrá descargado el nodo de \overline{OUT} , teniendo un \overline{OUT} de 0 lógico (0L). Por el lado contrario si el voltaje en IN es menor a V_{REF} , el nodo que se descargaría más rápido será OUT , con lo que las salidas serían las contrarias al caso anterior.

2.3.2 Operaciones lógicas LiM

Las operaciones lógicas que se realizaron son las cuatro más básicas: AND, NAND, NOR y OR. La manera en que se ejecutan es en tres etapas: Las dos primeras son las etapas de lectura, donde se deben leer todos los memristores al mismo tiempo. Para esto, primero se precargan SL y SLB, y segundo, se activan todas las WLLs y WLRs. Finalmente, en la tercera etapa, se activa la señal de S_{EN} , que debió estar desactivada durante las dos primeras etapas para precargar el amplificador de señal, y con ello evaluamos el voltaje de SL y SLB teniendo como referencia V_{REF} (ver Fig. 12).

El responsable de obtener las operaciones lógicas AND y NAND, es el amplificador de señal de la izquierda, es decir, el que está conectado a SL y el responsable de las operaciones OR y NOR, será el de la derecha conectado a SLB.

La manera en la que funcionan estas operaciones lógicas es similar en todos los casos, pero con el objetivo realizar una explicación, se tomará como ejemplo la operación AND. En el caso de que se tenga almacenados 1L en X e Y, los memristores de la izquierda estarán en HRS, y su R_{eq} , será la de tener dos resistencias del valor de HRS en paralelo, es decir la mitad de HRS, lo que es mucho mayor a LRS. Debido a que se tiene una resistencia muy alta, la

descarga de SL será muy lenta, lo que provoca mantener un valor de voltaje cercano a 1L durante más tiempo. Este fenómeno permite que el amplificador de señal detecte esta descarga lenta como un 1L en su salida y 0L en su salida complemento, mediante la comparación del voltaje de SL con V_{REF} .

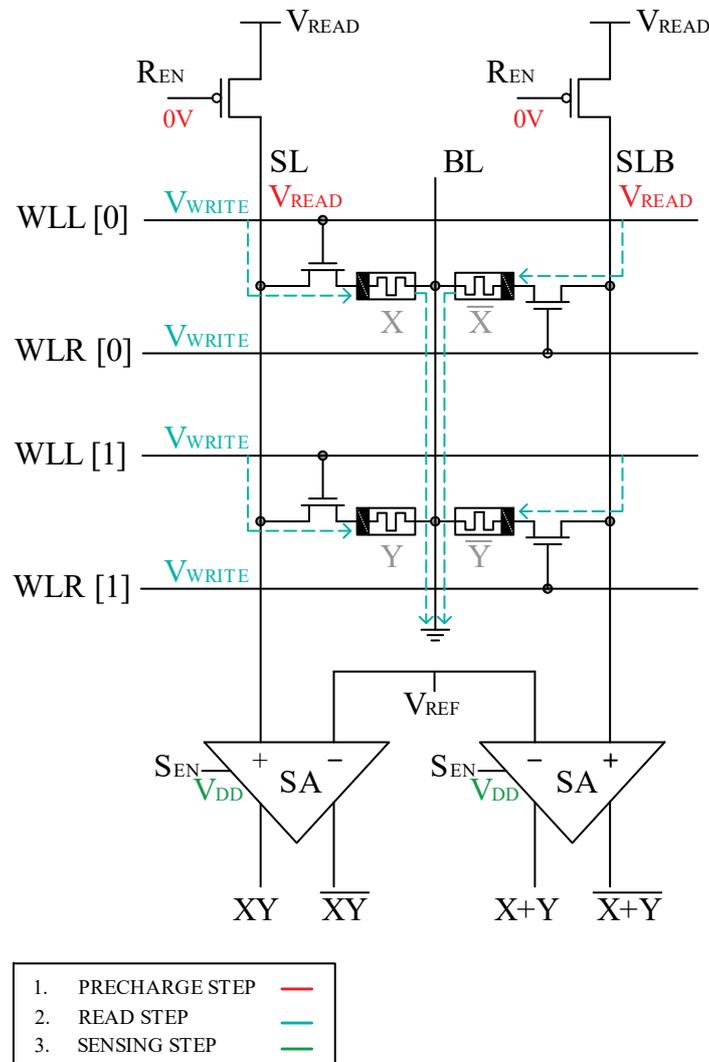


Figura 12. Pasos evaluar operaciones LiM.

Sin embargo, en otro caso, cuando se tenga almacenados diferentes valores lógicos en X e Y o se almacene en ambos un 0L, la R_{eq} del lado izquierdo será de LRS, teniendo LRS en paralelo con HRS o HRS en paralelo con LRS o LRS en paralelo con LRS, respectivamente. En estos casos como la resistencia es baja la descarga de SL será muy rápida, con lo cual al evaluar con el amplificador de señal, la descarga rápida se detectará como un 0L en la salida, y

un 1L en la salida complemento. La operación NOR es exactamente igual a la AND, solamente que se usan los valores complementos almacenados y la descarga de SLB, y debido a ello, el resultado deja de ser una operación AND y pasa a ser una operación NOR. Vale la pena recalcar que, como los amplificadores de señal tienen salidas complemento, se pueden obtener las operaciones lógicas complemento, en este caso la NAND y la OR, obteniendo así, las cuatro operaciones lógicas en un solo ciclo. Esto último es importante ya que nos va a permitir implementar el sumador de un bit con la información almacenada en memoria.

2.3.3 Sumador LiM

Para construir el sumador LiM, se ha implementado un nuevo módulo digital llamado LiM-FA. Este módulo tiene cinco entradas, de las cuales, cuatro son las salidas de ambos amplificadores de señal, es decir, las operaciones lógicas AND, NAND, NOR y OR, y la entrada de Carry In (C_{IN}), característica de un sumador. Este módulo fue propuesto y optimizado en su camino crítico por Chen et al. (2020). La topología e implementación en el arreglo de memoria de 2 bits se muestran en la Fig. 13.

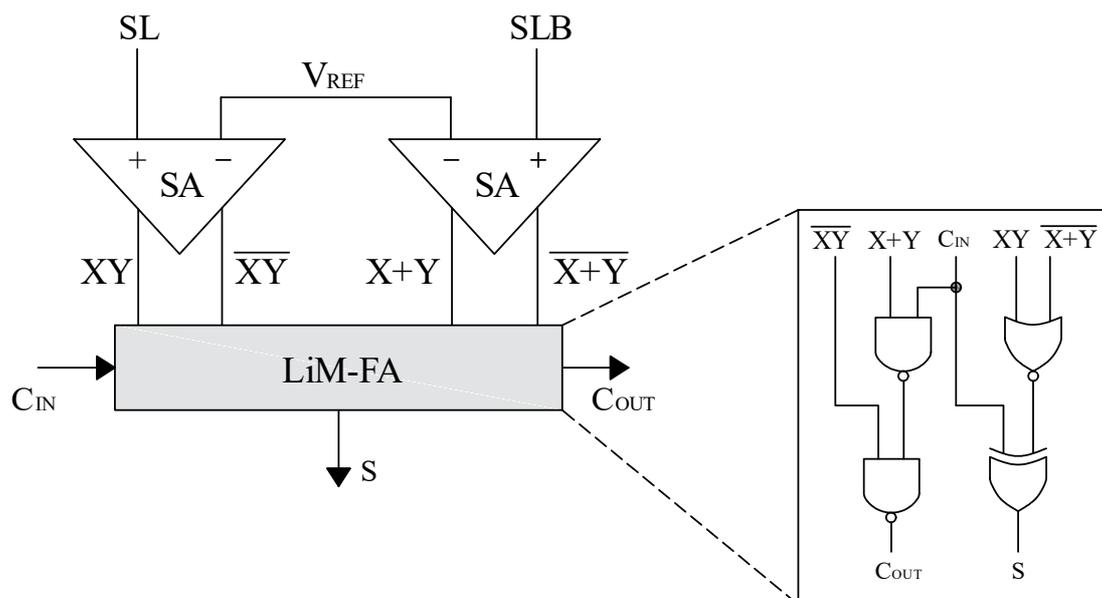


Figura 13. Topología e implementación de LiM-FA.

Para ejecutar la suma, simplemente se siguen los mismos pasos usados para evaluar las operaciones lógicas. Esto debido a que el módulo LiM-FA es puramente combinacional y no necesita de ninguna señal para empezar su funcionamiento.

2.3.4 Optimizaciones

2.3.4.1 Operaciones lógicas LiM

Con la implementación del modelo calibrado del memristor, el funcionamiento de las operaciones LiM no tenían el comportamiento esperado. Esto se debía principalmente porque la relación entre HRS y LRS (HRL) en esta calibración es de alrededor de 10, y esta arquitectura se había probado que funcionaba de manera adecuada con hasta 50 de HRL (Chen et al., 2020). Esta relación mientras más alta asegura el correcto funcionamiento del circuito ya que incrementará la diferencia entre una descarga lenta o rápida de SL o SLB y en consecuencia facilitará al SA la detección.

La solución encontrada para a este problema, fue incrementar el tamaño del transistor conectado a WL. Al agrandar el transistor, también aumenta su capacitancia, lo que permite una descarga de SL y SLB más lenta. La contraparte de realizar esto es que aumenta el tiempo de precarga de estas líneas, por lo que se hizo un estudio de cuál es el tamaño más adecuado del transistor para conseguir un correcto funcionamiento de las operaciones LiM sin perjudicar demasiado el tiempo de precarga de SL y SLB.

La Fig. 14 muestra un estudio paramétrico de los múltiplos del tamaño mínimo para el nodo tecnológico de 65nm.

Las operaciones LiM empezaron a dar un correcto funcionamiento a partir de 5 veces el tamaño mínimo del transistor, sin embargo se eligió un tamaño de 6 veces más

grande que el mínimo debido a que es más robusto y no existe una diferencia significativa en el tiempo de precarga con respecto a menores tamaños.

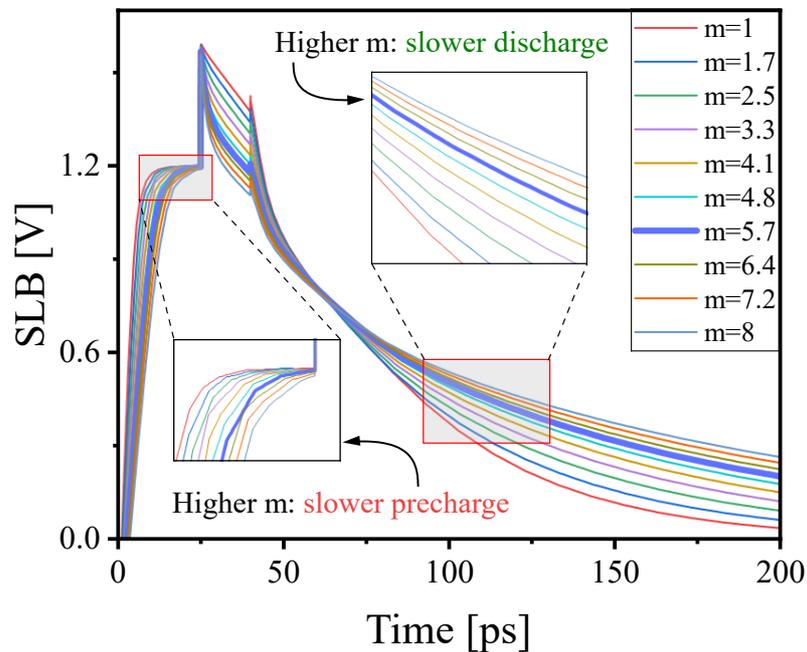


Figura 14. Análisis de tamaño del transistor que habilita escritura.

2.3.4.2 Sumador LiM

En cuanto al sumador LiM, las compuertas lógicas que contiene el módulo LiM-FA (ver Fig. 13) se implementaron usando tecnología CMOS y se optimizaron con el dimensionamiento teórico para que se tenga el mismo tiempo de retardo que un inversor. Esto implica una relación 2 a 1 del ancho del transistor PMOS frente al NMOS para compensar la movilidad inferior de los huecos en el PMOS (H.E.Weste & Harris, 2010). Cabe recalcar que se mantuvo la longitud mínima del nodo tecnológico para todos los transistores.

2.4 Resultados

Se simularon mil curvas en análisis Montecarlo para observar la variabilidad ciclo a ciclo de los memristores implementados y así evaluar el circuito LiM. Para ello, se inicializó la

memoria con cada una de las cuatro posibles combinaciones de palabras de 1 bit que se podrían almacenar. De esta manera, se puede verificar el correcto funcionamiento para todos los casos. Los voltajes usados para la simulación están detallados en la Tabla 1.

	Voltaje [V]
<i>Voltaje de polarización (V_{DD})</i>	1.2
<i>Voltaje de referencia (V_{REF})</i>	0.8
<i>1 lógico</i>	1.2
<i>0 lógico</i>	0

Tabla 1. Voltajes usados para simulación de circuito LiM.

El voltaje de polarización (V_{DD}), es el voltaje nominal para el nodo tecnológico usado (65nm). De igual manera, para V_{READ} , V_{WRITE} , R_{EN} y S_{EN} cuando están en 1L, se usó el voltaje nominal, lo que también implica que para las salidas, 1.2V se interpretará como 1L y 0V, como 0L. En cuanto a V_{REF} , se usó 0.8V debido a que asegura un funcionamiento correcto y de alto rendimiento del amplificador de señal.

Una idea que vale la pena recalcar es que V_{READ} es aplicado durante un tiempo muy corto para que de esta manera no corrompa el dato guardado en el memristor. Teóricamente el voltaje de lectura tiene que ser menor a los voltajes de SET y RESET del memristor (Chen et al., 2020), que en este caso tienen un valor nominal 0.8V y 1.2V, respectivamente. Sin embargo la velocidad reportada de escritura de estos dispositivos es de 5ns (Guitarra et al., 2021), por lo que si se aplica este voltaje por un tiempo muy inferior (se aplica por alrededor de 50ps), no existirá corrupción del dato almacenado.

Como se puede observar en la Fig. 15 y Fig. 16, las operaciones lógicas funcionan correctamente a pesar de la variabilidad incluida en el modelo dinámico calibrado del memristor.

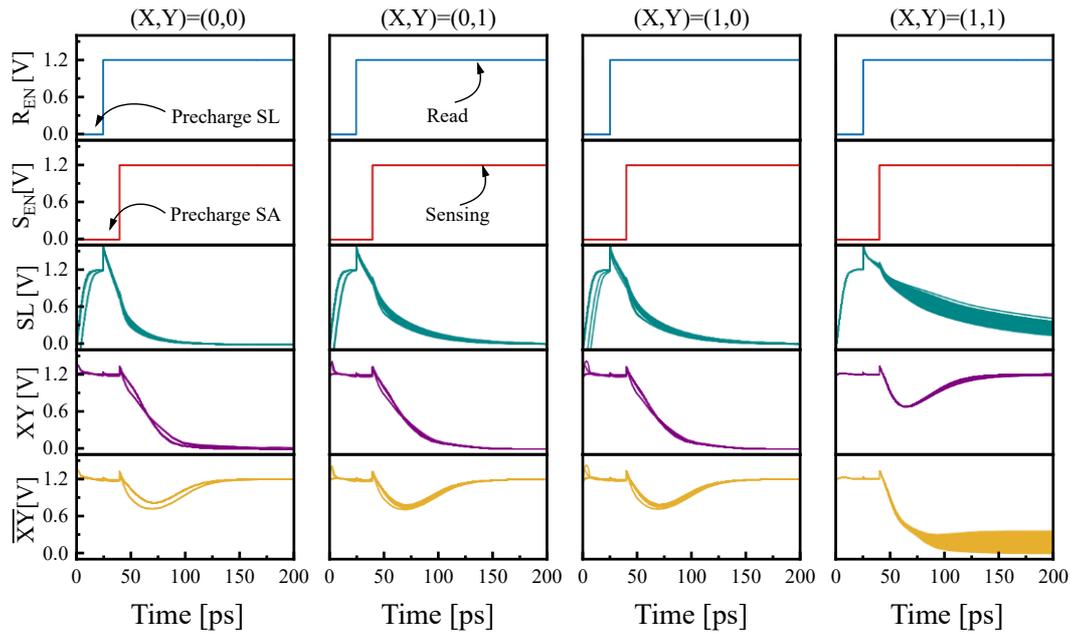


Figura 15. Resultados de simulación para operaciones LiM AND y NAND.

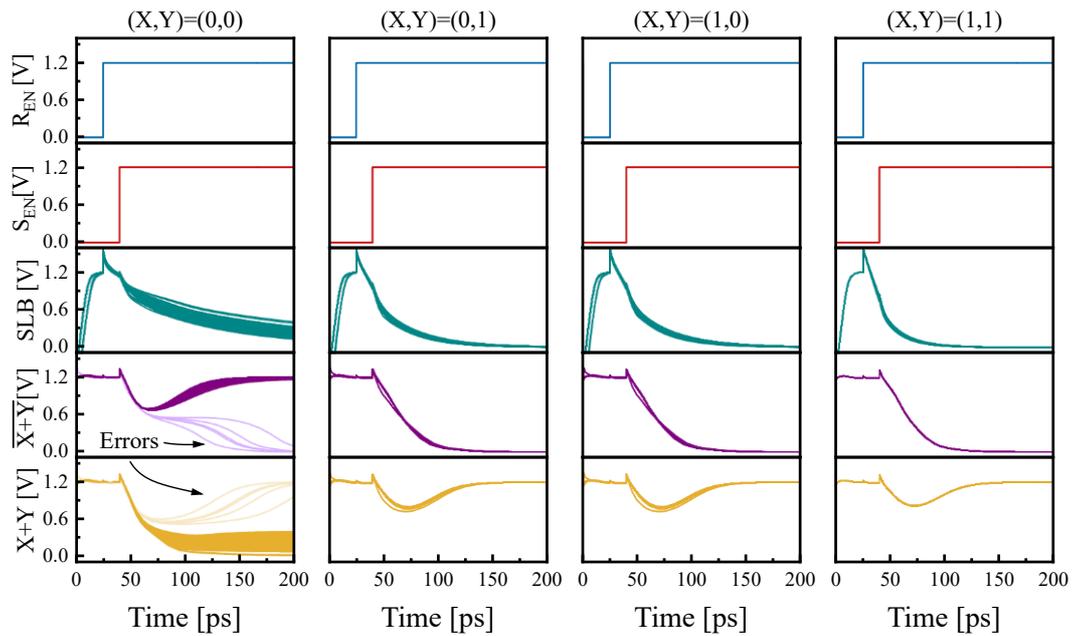


Figura 16. Resultados de simulación para operaciones LiM NOR y OR.

Se puede determinar que existe más variabilidad en los casos donde ambos memristores leídos están en HRS. Esto se debe a que el estado de alta resistencia presenta una variabilidad más alta ciclo a ciclo respecto a LRS (Guitarra et al., 2021).

El nivel lógico de la operación NAND en el caso de que ambas palabras almacenadas sean 1L, presenta la mayor variabilidad de las salidas de las operaciones lógicas. Esto podría ser corregido con un buffer apropiadamente dimensionado que esté conectado a esta salida.

Similarmente, en la Fig. 17, se muestra la simulación del Sumador LiM. Aquí se puede apreciar que para los 8 casos posibles de las combinaciones de las entradas X, Y y C_{IN} en un sumador, el LiM-FA funciona de manera correcta.

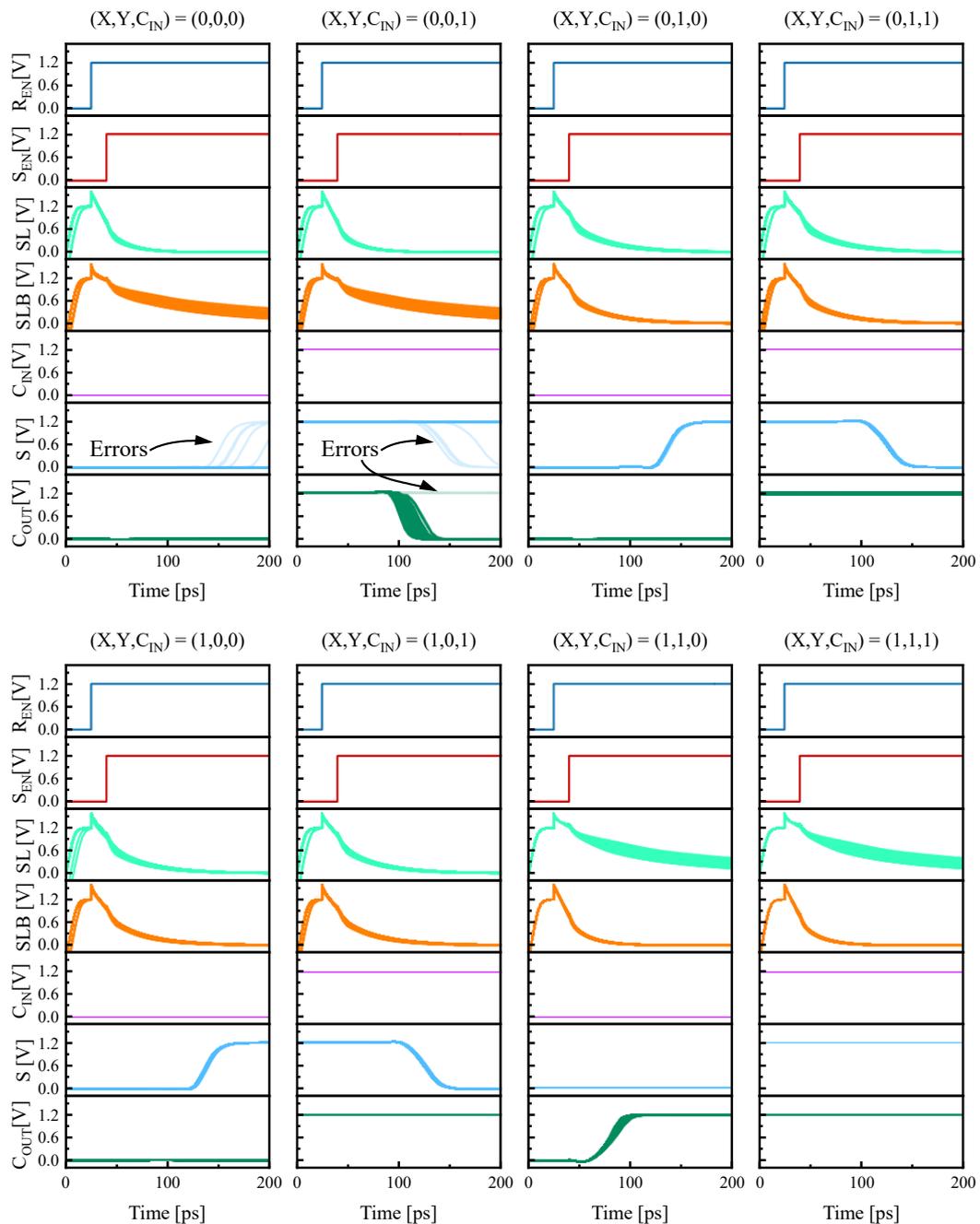


Figura 17. Resultados de simulación para sumador LiM.

2.4.1 Robustez

Un análisis de robustez es esencial para comprender si el circuito será fiable. Los circuitos de LiM de memristores necesitan de un entendimiento amplio de su variabilidad ya que todos los parámetros a caracterizar del circuito dependen de la variabilidad de la celda resistiva (Fieback et al., 2022).

2.4.1.1 Operaciones lógicas LiM

En cuanto a las operaciones LiM AND, NAND, NOR y OR, existe un caso en las operaciones NOR y OR en donde aparecen errores debido a la variabilidad. En este caso los valores almacenados en la memoria son $(X,Y) = (0, 0)$. Los errores son solamente de 5 de las 1000 curvas, teniendo una tasa de error del 0,5% (ver Fig. 16). El circuito podría ser más robusto y reducir esta tasa de error a cero por ciento, con dos mejoras: La primera, mejorando la calibración del modelo dinámico del memristor para incrementar HLR y así conseguir una mayor diferencia entre las descargas rápidas y lentas de SL o SLB; La segunda es encontrar una topología de amplificador de señal que sea más robusta y que su precisión de detección sea mayor.

2.4.1.2 Sumador LiM

El sumador de 1 bit LiM, es robusto por sí mismo por el hecho de ser un módulo CMOS. Sin embargo, como se ve en la Fig. 17, cuenta con errores debido a las fallas de sus entradas. En este caso, como es dependiente de las operaciones AND, NAND, NOR y OR, entonces las salidas del sumador, S y Cout, conservarán los mismos errores de estas operaciones. En otras palabras las salidas del sumador también cuentan con un 0.5% de tasa de error para los casos donde $(X,Y) = (0, 0)$ (ver Fig. 17).

2.4.2 Rendimiento

El rendimiento de un circuito se puede demostrar con sus características de tiempo, en este caso fueron medidas tres características de las salidas del LiM-FA. El tiempo de retardo de propagación (t_{pd}), que permite comprobar cuanto tarda el circuito en obtener la salida una vez iniciada su ejecución, el tiempo de subida (t_{rise}), que permite verificar el tiempo que demora la señal de salida en pasar de 0L a 1L, y el tiempo de bajada (t_{fall}), que permite determinar el tiempo que toma la señal de salida en pasar de 1L a 0L.

2.4.2.1 Tiempo de retardo de propagación

Para medir el tiempo de retardo de propagación se escogió el peor caso de retardo para la salida S y para la salida C_{OUT}. Esto ocurrió cuando la memoria tenía almacenado $(X,Y) = (0,1)$ y la señal C_{IN} era 0L, y cuando la memoria tenía almacenado $(X,Y) = (0,0)$ y la señal C_{IN} era 1L, respectivamente.

La técnica usada para medir t_{pd} fue tomar el tiempo entre cuando la señal de R_{EN} llega al 50% de V_{DD}, al ser activada y dejar a SL y SLB flotando, y cuando la señal de la salida también llegue al 50% de V_{DD}, en camino al valor correcto de la operación.

Esta técnica es una de las más usadas para medir t_{pd} (H.E.Weste & Harris, 2010) y por ello se empleada en este trabajo.

En la Fig. 18 y Fig. 19 se observa la distribución gaussiana de la variabilidad del retardo de propagación, junto con su media y desviación estándar, para las salidas de S y Cout del LiM-FA respectivamente.

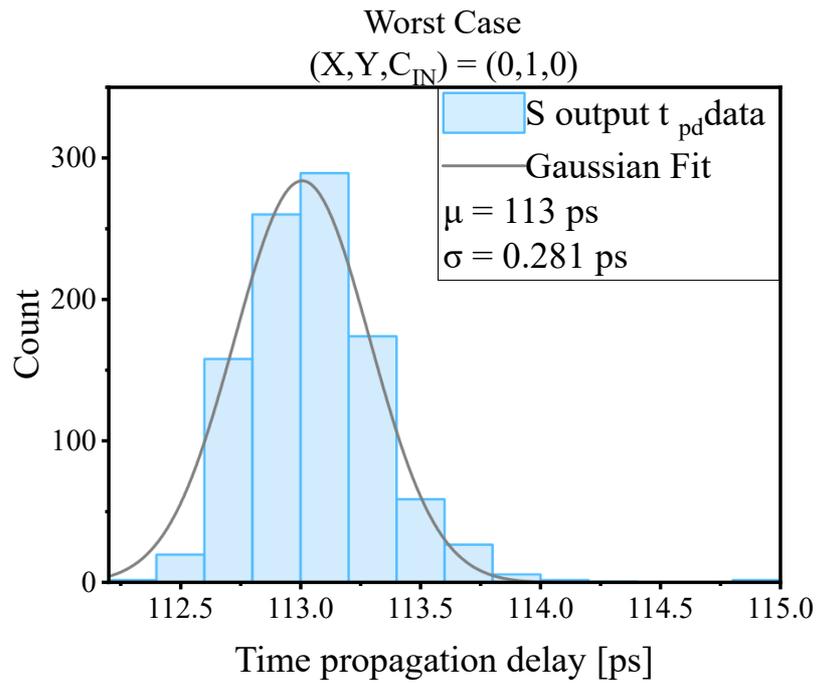


Figura 18. Distribución del tiempo de retardo para el peor caso de la salida S.

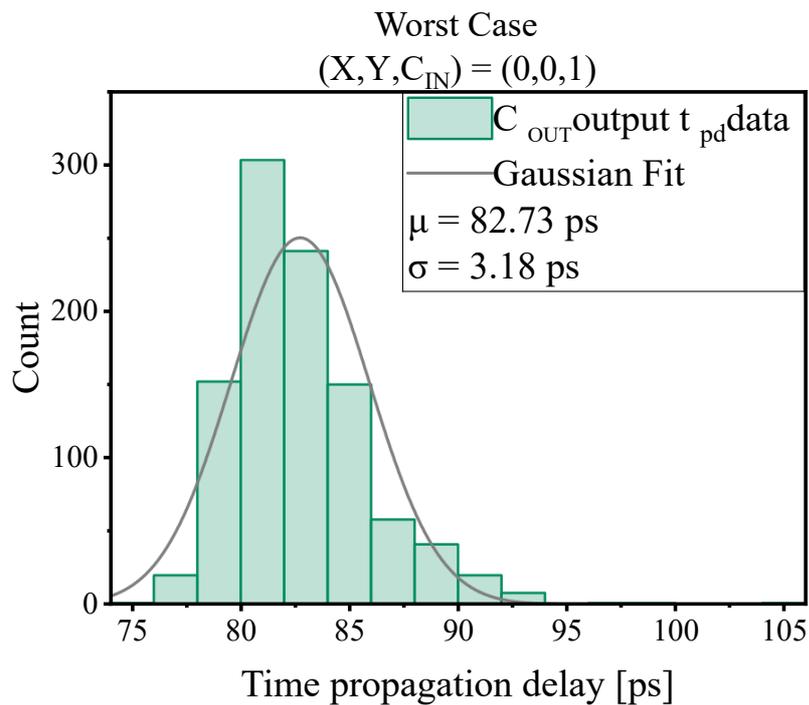


Figura 19. Distribución del tiempo de retardo para el peor caso de la salida C_{OUT}.

El t_{pd} más grande lo tiene la salida S con 113ps. Este valor es un excelente tiempo de retardo ya que permite trabajar la lógica en memoria con una frecuencia de hasta 8.8 GHz.

2.4.2.2 Tiempo de subida

Para medir el tiempo de subida, también se escogió el peor caso para ambas salidas, S y Cout. La técnica usada para medir t_{rise} , nuevamente es la más usada en la literatura (H.E.Weste & Harris, 2010), y consiste en tomar el tiempo desde que la señal es un 10% de V_{DD} hasta que llega al 90% de V_{DD} . La Fig. 20 y Fig. 21 muestran las distribuciones de los tiempos de subida, sus medias y desviaciones estándar de ambas salidas, S y Cout, respectivamente.

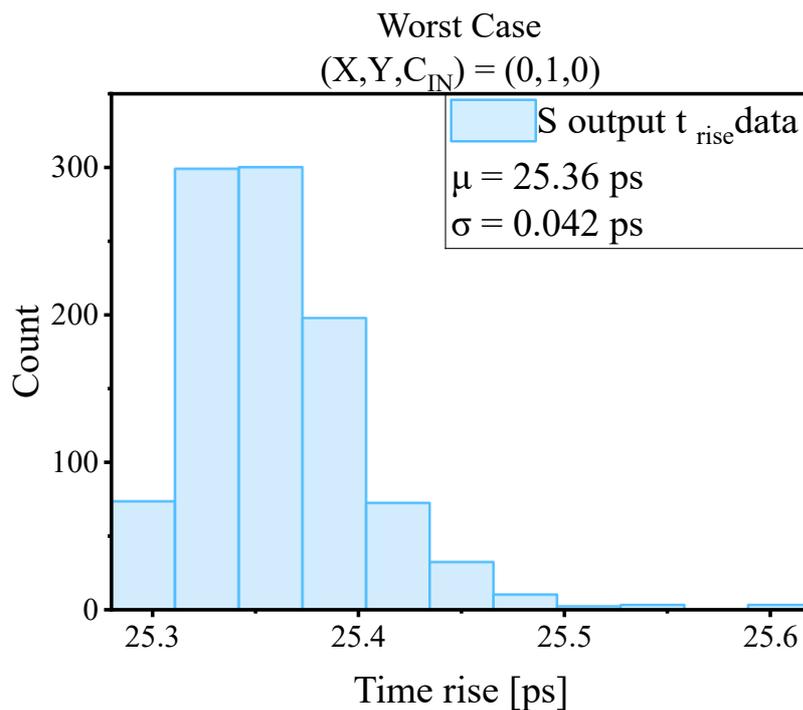


Figura 20. Distribución del tiempo de subida para el peor caso de la salida S.

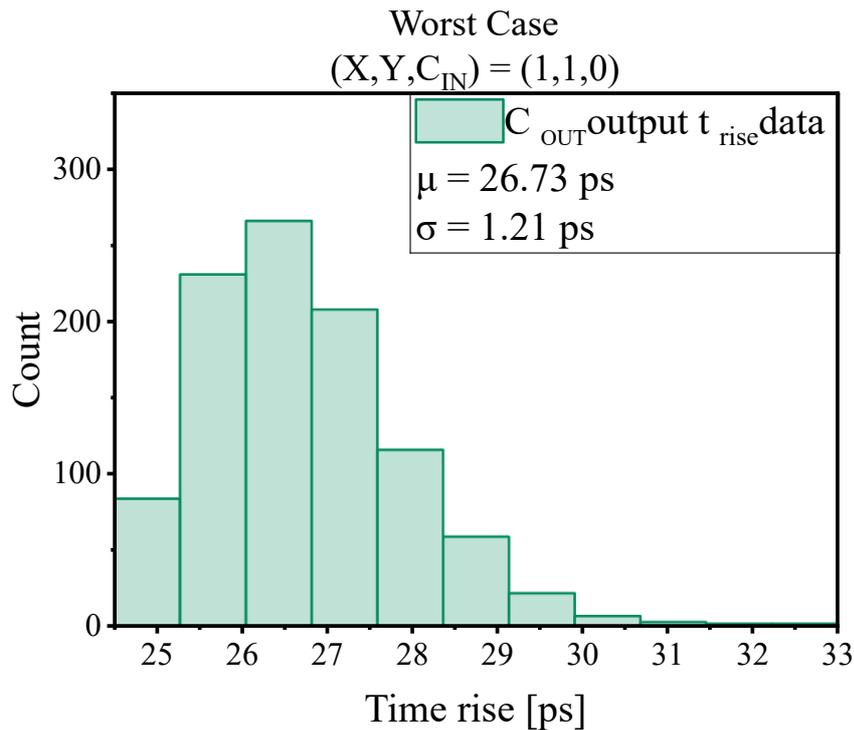


Figura 21. Distribución del tiempo de subida para el peor caso de la salida C_{OUT}.

Los valores de tiempo de subida para ambas salidas del LiM-FA están balanceadas, es decir, cuentan con un valor medio muy similar. Esta es un excelente resultado ya que idealmente las diferentes salidas de un sistema deberían tener las mismas características (Harris & Harris, 2015).

2.4.2.3 Tiempo de bajada

Para medir el t_{fall} , se escogió nuevamente el peor caso para ambas salidas, S y Cout. La técnica usada es la misma que para el t_{rise} , solamente que ahora se toma el tiempo desde que la señal es un 90% de V_{DD} hasta que llega al 10% de V_{DD} . La Fig. 22 y Fig. 23 muestran las distribuciones de los tiempos de subida, sus medias y desviaciones estándar para ambas salidas, S y Cout, respectivamente.

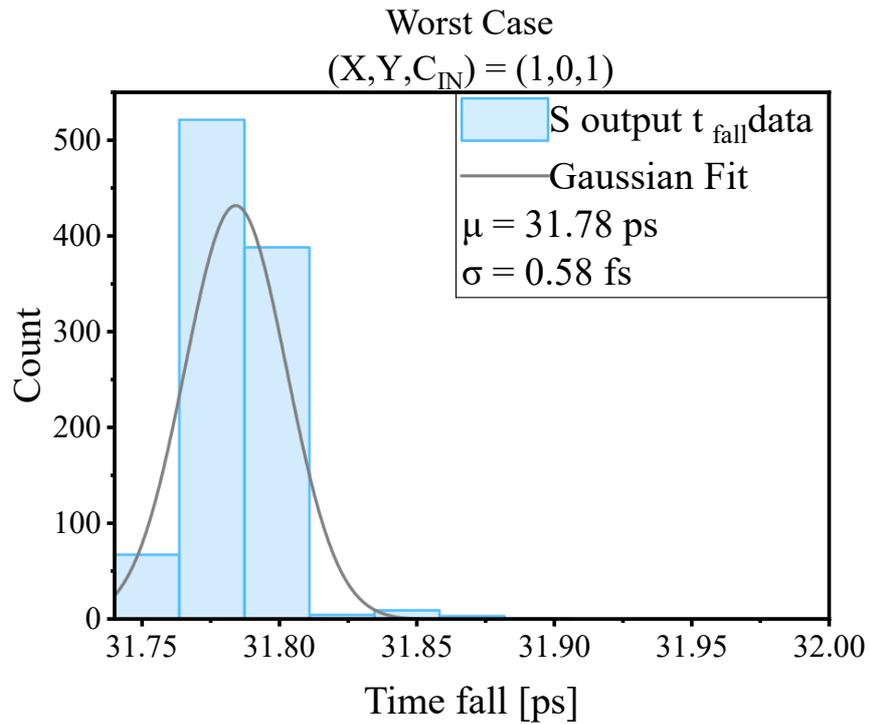


Figura 22. Distribución del tiempo de bajada para el peor caso de la salida S.

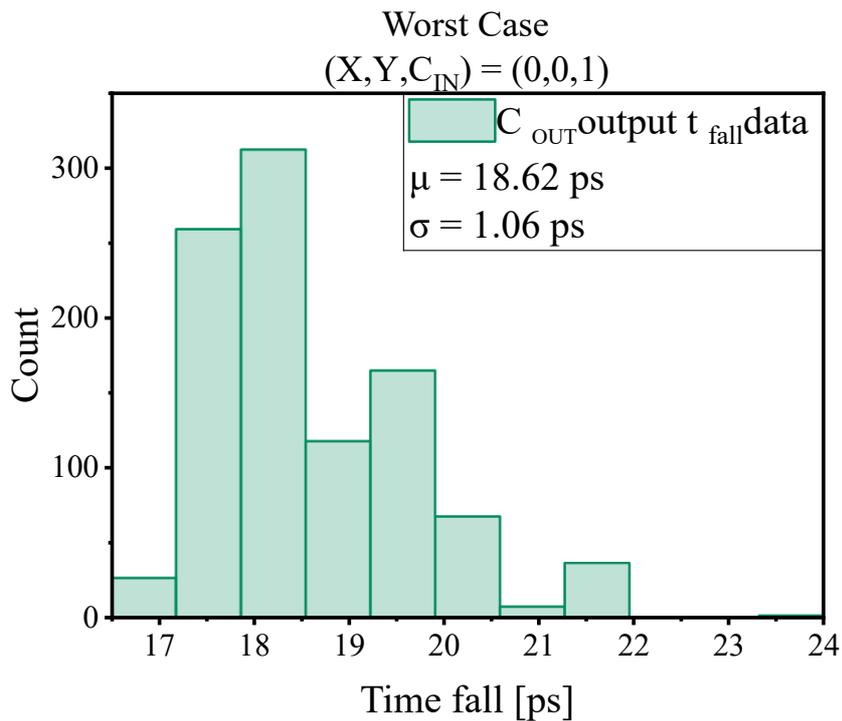


Figura 23. Distribución del tiempo de bajada para el peor caso de la salida C_{OUT}.

Los valores de tiempo de bajada para ambas salidas del LiM-FA, contrario a lo que sucede con t_{rise} , están desbalanceadas y cuentan con una diferencia de 13.16 ps.

2.4.3 Consumo de potencia

El bajo consumo de potencia es de alta importancia en un circuito hoy en día. Esto permite a los diseñadores de productos optimizar la duración de la batería en aplicaciones portátiles o reducir la huella de carbono en dispositivos conectados permanentemente a la corriente (H.E.Weste & Harris, 2010).

En esta sección se realizan mediciones de potencia estática, la cual es la potencia consumida por el circuito cuando se encuentra apagado debido a efectos de corrientes de fuga en transistores, y la potencia dinámica, que es la que se consume cuando se cargan o descargan las capacitancias de carga de la compuerta de un transistor cuando se activa o desactiva (H.E.Weste & Harris, 2010).

2.4.3.1 Potencia estática

La Fig. 24 muestra la distribución gaussiana, la media y la desviación estándar de la potencia estática del circuito LiM.

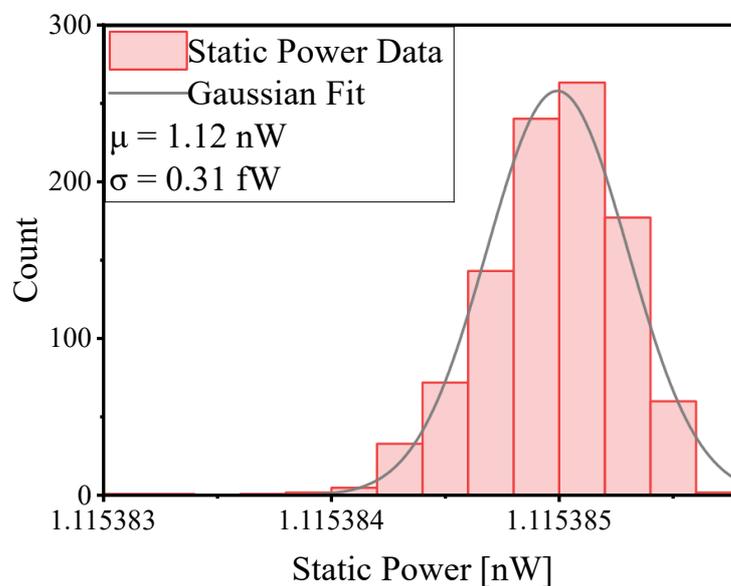


Figura 24. Distribución de potencia estática del circuito LiM.

Como se puede observar, la potencia estática tiene una variabilidad extremadamente baja debido a que existe muy poca corriente circulando a través de los memristores. Lo que indica esta potencia es que el circuito aún estando apagado, estará consumiendo 1.12nW.

2.4.3.2 Potencia dinámica

La Fig. 25 muestra las medias y desviaciones estándar para los consumos de potencia dinámica en los diferentes casos de operación del circuito LiM.

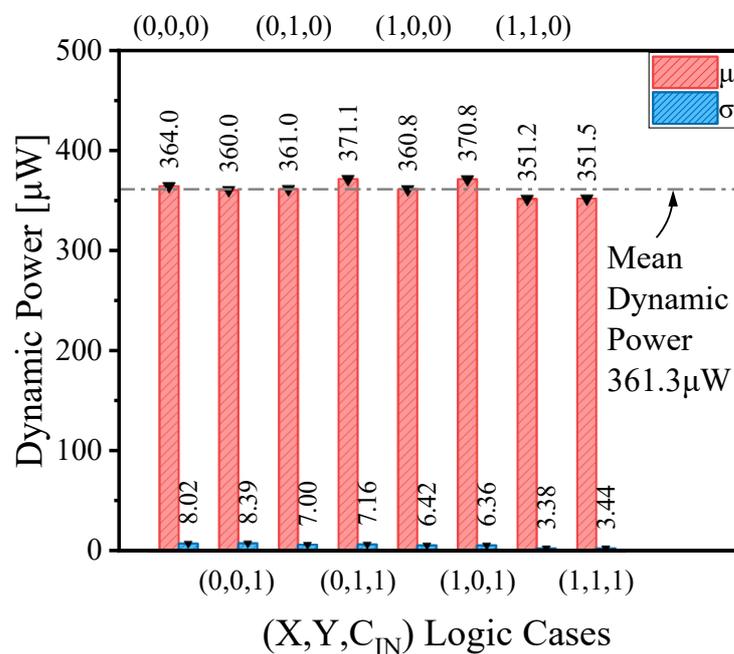


Figura 25. Resultados de variabilidad de Potencia Dinámica del circuito LiM.

El caso para el que existe mayor consumo de potencia dinámica es cuando están almacenadas dos palabras distintas en la memoria y la señal de C_{IN} se encuentra en 1L. Vale la pena destacar que la potencia dinámica consumida es muy similar para todos los casos y se encontró que el valor medio es de 361.3 μW.

Si tomamos en cuenta la potencia dinámica y estática, la potencia total consumida en el circuito en su peor caso de operación será de 371.1012 μW, siendo la potencia estática prácticamente despreciada.

3 CONCLUSIONES

- Se consiguió calibrar un modelo dinámico de memristor con datos experimentales para que el comportamiento de corriente-voltaje del dispositivo simule un dispositivo ReRAM de HfO_2 con un área de $85 \times 85 \text{ nm}^2$ fabricado. Además, se incluyó la variabilidad ciclo a ciclo característica de los memristores. Esto se logró mediante el estudio de curvas experimentales tomadas de la muestra, donde se obtuvieron estadísticas de media y desviación estándar de los puntos estratégicos de la curva que formaban parte, como parámetros, del modelo. El resultado fue un buen ajuste del modelo dinámico, con lo que se pudo simular el circuito LiM en un entorno más cercano a lo que sería si se fabricara.
- Se implementó una arquitectura para LiM usando memristores calibrados y transistores del PDK de 65nm de TSMC. Esta arquitectura se conforma por celdas de bits 2T2R, en la que se almacena un dato y su complemento de manera no volátil haciendo uso de los dos estados de resistividad que tiene un memristor, y las cuales fueron optimizadas conseguir realizar LiM. El arreglo usado fue uno de 2 palabras de 1 bit para realizar operaciones lógicas AND, NAND, NOR y OR en memoria, y también computar una suma LiM. Además se implementaron los circuitos de precarga de las líneas de origen, un amplificador de señal para detectar y evaluar la operación lógica que requerida, y un módulo CMOS LiM-FA que permite el cómputo en memoria de la suma de las dos palabras almacenadas.
- Las salidas de las operaciones lógicas AND y NAND, por un lado, mostraron una excelente robustez en todos los casos de datos almacenados en memoria frente a la variabilidad ciclo a ciclo del memristor, pero por otro lado las operaciones lógicas NOR y OR presentaron un 0.5% de error para el caso en que en la memoria se tiene

almacenado 0L en las dos palabras. En cuanto al sumador, debido a que sus salidas dependen del funcionamiento correcto de todas las operaciones lógicas mencionadas previamente, este conservó el 0.5% de error para este mismo caso. Además, el rendimiento de las salidas del sumador fue alto debido a que el tiempo de retardo de propagación máximo fue de 113ps lo que le permite al circuito LiM trabajar hasta 8.8 GHz de velocidad. En cuanto a tiempos subida y de bajada, se encontró que son de 25.36ps y 31.78ps para la salida S y de 26.73ps y 18.62ps para la salida C_{OUT}, respectivamente. Finalmente, se obtuvo un consumo de potencia máximo del circuito LiM de 371.1012 μ W.

- Como trabajo a futuro, este circuito de LiM se podría extender para trabajar con más de 2 palabras y que cada palabra contenga más bits, de esta manera se puede hacer un análisis de cómo funcionarían las operaciones lógicas entre más de 2 palabras y un estudio del camino crítico de los Carrys en el sumador cuando se tiene palabras de más de 1 bit.

4 REFERENCIAS BIBLIOGRÁFICAS

- Aguirre, F. L., Pazos, S. M., Palumbo, F., Suñé, J., & Miranda, E. (2021). SPICE Simulation of RRAM-Based Cross-Point Arrays Using the Dynamic Memdiode Model. *Frontiers in Physics*, 9. <https://doi.org/10.3389/fphy.2021.735021>
- Chen, Y., Lu, L., Kim, B., & Kim, T. T. H. (2020). Reconfigurable 2T2R ReRAM Architecture for Versatile Data Storage and Computing In-Memory. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 28(12). <https://doi.org/10.1109/TVLSI.2020.3028848>
- Chua, L. (2011). Resistance switching memories are memristors. *Applied Physics A: Materials Science and Processing*, 102(4). <https://doi.org/10.1007/s00339-011-6264-9>
- Chua, L. O. (1971). Memristor—The Missing Circuit Element. *IEEE Transactions on Circuit Theory*, 18(5). <https://doi.org/10.1109/TCT.1971.1083337>
- Fieback, M., Munch, C., Gebregiorgis, A., Medeiros, G. C., Taouil, M., Hamdioui, S., & Tahoori, M. (2022). PVT Analysis for RRAM and STT-MRAM-based Logic Computation-in-Memory. *Proceedings of the European Test Workshop, 2022-May*. <https://doi.org/10.1109/ETS54262.2022.9810436>
- Guitarra, S., Raymond, L., & Trojman, L. (2021). Stochastic multiscale model for HfO₂-based resistive random access memories with 1T1R configuration. *Solid-State Electronics*, 176. <https://doi.org/10.1016/j.sse.2020.107947>
- Harris, S. L., & Harris, D. M. (2015). Digital Design and Computer Architecture: ARM Edition. In *Digital Design and Computer Architecture: ARM Edition*. <https://doi.org/10.1016/C2013-0-14352-8>

- H.E.Weste, N., & Harris, D. M. (2010). CMOS VLSI Design: A Circuits and Systems Perspective (4th Edition). In *CMOS VLSI Design: A Circuits and Systems Perspective (4th Edition)*.
- Ielmini, D. (2016). Resistive switching memories based on metal oxides: Mechanisms, reliability and scaling. In *Semiconductor Science and Technology* (Vol. 31, Issue 6). <https://doi.org/10.1088/0268-1242/31/6/063002>
- Kvatinsky, S., Belousov, D., Liman, S., Satat, G., Wald, N., Friedman, E. G., Kolodny, A., & Weiser, U. C. (2014). MAGIC - Memristor-aided logic. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 61(11). <https://doi.org/10.1109/TCSII.2014.2357292>
- Lebdeh, M. A., Reinsalu, U., Du Nguyen, H. A., Wong, S., & Hamdioui, S. (2019). Memristive device based circuits for computation-in-memory architectures. *Proceedings - IEEE International Symposium on Circuits and Systems, 2019-May*. <https://doi.org/10.1109/ISCAS.2019.8702542>
- 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. <https://doi.org/10.1109/ACCESS.2019.2907976>
- Panda, D., Sahu, P. P., & Tseng, T. Y. (2018). A Collective Study on Modeling and Simulation of Resistive Random Access Memory. In *Nanoscale Research Letters* (Vol. 13). <https://doi.org/10.1186/s11671-017-2419-8>
- Rahman, L. F., Reaz, M. B. I., Yin, C. C., Ali, M. A. M., & Marufuzzaman, M. (2014). Design of high speed and low offset dynamic latch comparator in 0.18 μm CMOS process. *PLoS ONE*, 9(10). <https://doi.org/10.1371/journal.pone.0108634>
- Snider, G. (2005). Computing with hysteretic resistor crossbars. *Applied Physics A: Materials Science and Processing*, 80(6). <https://doi.org/10.1007/s00339-004-3149-1>

- Strukov, D. B., Snider, G. S., Stewart, D. R., & Williams, R. S. (2008). The missing memristor found. *Nature*, *453*(7191). <https://doi.org/10.1038/nature06932>
- Synopsys Inc. (2008). *HSPICE® User Guide: Simulation and Analysis*.
- Velasquez, A., & Jha, S. K. (2016). Parallel boolean matrix multiplication in linear time using rectifying memristors. *Proceedings - IEEE International Symposium on Circuits and Systems, 2016-July*. <https://doi.org/10.1109/ISCAS.2016.7538937>
- Vourkas, I., & Sirakoulis, G. C. (2016). Emerging memristor-based logic circuit design approaches: A review. In *IEEE Circuits and Systems Magazine* (Vol. 16, Issue 3). <https://doi.org/10.1109/MCAS.2016.2583673>
- Xie, L., Nguyen, H. A. D., Yu, J., Kaichouhi, A., Taouil, M., Alfailakawi, M., & Hamdioui, S. (2017). Scouting Logic: A Novel Memristor-Based Logic Design for Resistive Computing. *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI, 2017-July*. <https://doi.org/10.1109/ISVLSI.2017.39>
- Zhu, Z., Ma, M., Liu, J., Xu, L., Chen, X., Yang, Y., Wang, Y., & Yang, H. (2019). A general logic synthesis framework for memristor-based logic design. *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD, 2019-November*. <https://doi.org/10.1109/ICCAD45719.2019.8942111>