

Unidad lógico aritmético de un bit realizada con circuitos reconfigurables

J. González-Contreras¹, R. E. Lozoya-Ponce¹ y L. J. Ontañón-García²

¹ Tecnológico Nacional de México campus Chihuahua, División de Estudios de Posgrado e Investigación, Av. Tecnológico No. 2909 Col. 10 de Mayo, C.P. 31310 Chihuahua, Chih., México, D24061345@chihuahua.tecnm.mx, ricardo.lp@chihuahua.tecnm.mx

² Universidad Autónoma de San Luis Potosí, Coordinación Académica Región Altiplano Oeste, Salinas, San Luis Potosí 78600, México, luis.ontanon@uaslp.mx

Resumen

Este trabajo presenta el diseño y desarrollo de una Unidad Lógico Aritmética (ALU) reconfigurable de 1 bit basada en circuitos analógicos, utilizando componentes accesibles como amplificadores operacionales y transistores. A través del concepto de celdas lógicas reconfigurables (CLR), inspiradas en redes neuronales artificiales, se implementa un sistema capaz de realizar múltiples funciones lógicas y aritméticas simplemente ajustando ciertos parámetros del circuito. La propuesta mejora versiones anteriores al optimizar el diseño, permitir el uso de señales invertidas, y reducir la necesidad de componentes adicionales. Se realizaron simulaciones y pruebas experimentales que confirman su correcto funcionamiento, destacando su utilidad como herramienta didáctica para explorar conceptos de reconfiguración analógica en sistemas híbridos.

Palabras clave— ALU, Celda lógica reconfigurable, Amplificadores operacionales, Operaciones lógicas.

One-bit Arithmetic Logic Unit Implemented with Reconfigurable Circuits

Abstract

This work presents the design and development of a 1-bit reconfigurable Arithmetic Logic Unit (ALU) based on analog circuits, using accessible components such as operational amplifiers and transistors. Through the concept of Reconfigurable Logic Cells (RLCs), inspired by artificial neural networks, a system is implemented that can perform multiple logic and arithmetic functions by simply adjusting certain circuit parameters. The proposed design improves upon previous versions by optimizing the structure, enabling the use of inverted signals, and reducing the need for additional components. Simulations and experimental tests confirm its correct operation, highlighting its value as an educational tool for exploring analog reconfiguration concepts in hybrid systems.

Keywords— ALU, Reconfigurable logic cell, Operational amplifiers, Logical operations.

I. INTRODUCCIÓN

La evolución de los circuitos reconfigurables ha estado fuertemente influenciada por los avances en los sistemas digitales; sin embargo, la electrónica analógica también ha desempeñado un papel importante, particularmente mediante el uso de amplificadores operacionales (op-amps, por sus siglas en inglés *operational amplifier*). Según Sedra y sus colegas, los op-amps han posibilitado incrementar la capacidad de adaptación de los circuitos analógicos, permitiendo implementar funciones digitales en el dominio analógico y ampliando así las capacidades de los sistemas híbridos [1].

En la actualidad, el diseño de sistemas electrónicos exige un equilibrio entre flexibilidad, eficiencia y adaptabilidad, especialmente en contextos donde coexisten señales analógicas y digitales. Aunque las soluciones digitales, como los ASICs y las plataformas FPGA, dominan la industria por su alto rendimiento, suelen ser menos accesibles en aplicaciones educativas, experimentales o de bajo costo, donde también es crucial la simplicidad de implementación y la posibilidad de manipulación directa.

En estos entornos, las opciones que permiten explorar operaciones lógicas y aritméticas fuera del dominio puramente digital son limitadas, lo que restringe el alcance de la enseñanza y la experimentación en sistemas híbridos. Por ello, resulta necesaria una alternativa que permita implementar funciones digitales empleando componentes analógicos accesibles y versátiles.

Una posible solución la ofrecen los sistemas reconfigurables analógicos, que buscan equilibrar los beneficios de la lógica digital con la manipulación directa propia del dominio analógico. Este enfoque intermedio se ilustra en la Fig. 1, donde se posicionan tecnologías que no son ni completamente digitales ni exclusivamente analógicas, pero que permiten una exploración funcional y educativa de ambos mundos [2].

En este contexto, el presente trabajo propone una solución basada en componentes estándar y una estructura modular, orientada principalmente a fines académicos y experimentales. Esta propuesta se alinea con la tendencia actual en el diseño de unidades lógico-aritméticas (ALUs), la cual está en constante

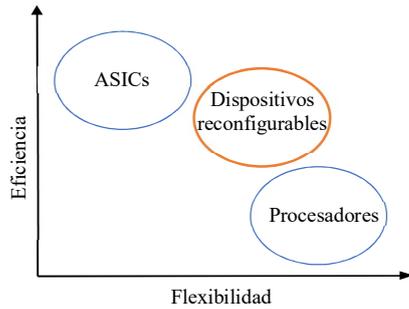


Fig. 1 Comparativa entre las técnicas de cómputo.

evolución para mejorar el rendimiento y la eficiencia energética [3], [4]. El objetivo es demostrar un sistema que, sin alcanzar la eficiencia ni la flexibilidad total de las plataformas digitales programables, constituye una alternativa viable para explorar operaciones lógicas y aritméticas mediante celdas lógicas reconfigurables (CLR) basadas en amplificadores operacionales.

En la literatura, diversos autores han explorado la implementación de compuertas lógicas mediante circuitos analógicos reconfigurables. Peng y sus colaboradores [5] introdujeron el concepto de celdas lógicas basadas en sistemas lineales por tramos, logrando compuertas básicas como AND, OR y XOR mediante ajustes de parámetros. Sinha y Ditto [6] propusieron sistemas dinámicos caóticos para emular funciones lógicas, lo cual abrió una nueva línea de exploración, pero con alta complejidad experimental. Posteriormente, en [7], se desarrolló una ALU de 1 bit y 2 bits utilizando múltiples CLR, aunque la implementación requería un mayor número de niveles de voltaje (hasta 11) y utilizaba encapsulados y configuraciones más complejas.

La presente propuesta se diferencia y aporta respecto a estos antecedentes en varios aspectos clave: (1) se ha optimizado el circuito de la CLR utilizando únicamente amplificadores operacionales LM324 y transistores 2N3904, simplificando la implementación práctica; (2) se ha logrado reducir el número de niveles de voltaje requeridos a solo 6, mejorando la eficiencia sin sacrificar la funcionalidad; (3) se implementaron 16 funciones lógicas y aritméticas en un diseño compacto de 1 bit, duplicando la capacidad reportada en [7]; y (4) se enfatiza la aplicabilidad en entornos didácticos y experimentales mediante componentes accesibles y estructura modular. Estas mejoras permiten avanzar hacia circuitos reconfigurables más eficientes, accesibles y adaptables, fortaleciendo la exploración de arquitecturas híbridas en la electrónica moderna. Cabe destacar que esta propuesta no busca sustituir soluciones digitales consolidadas, sino ofrecer una alternativa complementaria para la exploración académica y el desarrollo de sistemas híbridos.

II. DESARROLLO

En este trabajo, se abordan las CLR como redes neuronales artificiales (ANNs, por sus siglas en inglés Artificial Neural Networks) al igual que en los trabajos [8] y [9], específicamente se aborda como redes de un sólo nodo. En este contexto, cada

compuerta lógica se conceptualiza como una "neurona" artificial, diseñada para realizar una operación lógica en su salida, denotada como Y .

La salida Y de la CLR se obtiene a partir de una combinación ponderada de las entradas, seguida de una función de activación. Sea x_i cada una de las entradas binarias (con $i = 1, 2, \dots, n$) y w_i los pesos asignados a cada entrada, los cuales determinan su influencia en la salida final. Además, se introduce un término de *bias* u *offset*, representado como una entrada constante x_0 con un peso asociado w_0 .

$$g(x) = w_0x_0 + \sum_{i=1}^n w_ix_i \tag{1}$$

Donde:

- “ $g(x)$: valor intermedio antes de aplicar la función umbral,”
- “ x_i : entradas binarias,”
- “ w_i : pesos correspondientes a cada entrada,”
- “ x_0 : entrada de *bias* (normalmente una señal constante),”
- “ w_0 : peso del *bias*.”

Posteriormente, la salida Y se calcula aplicando la función de activación f sobre $g(x)$. En este caso, se emplea una función tipo comparador de ventana, definida mediante los umbrales K_- (umbral inferior) y K_+ (umbral superior), de la siguiente forma:

$$Y = f(g(x)) = \begin{cases} 1, & \text{si } K_+ \geq g(x) \geq K_- \\ 0, & \text{en otro caso} \end{cases} \tag{2}$$

Donde:

- “ K_- : umbral inferior que marca el límite mínimo para la activación,”
- “ K_+ : umbral superior que marca el límite máximo para la activación,”
- “ Y : salida binaria de la celda (1 o 0).”

Esta formación permite que la CLR implemente funciones lógicas tanto simples como complejas, ajustando dinámicamente la ventana de activación mediante la configuración adecuada de los pesos, el bias y los valores de umbral. Este proceso se ilustra en la Fig. 2.

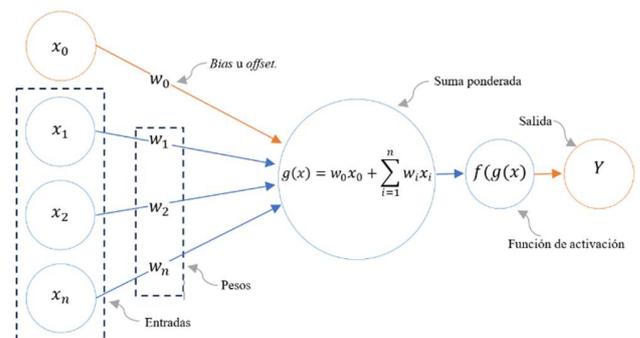


Fig. 2 Red de la celda lógica reconfigurable.

Es importante resaltar que los valores de K_+ y K_- definen la ventana o umbral de acción de la compuerta lógica reconfigurable y debe de cumplir la condición de la ecuación 3.

$$K_+ > K_- \tag{3}$$

El diseño de las CLR's en este trabajo guarda similitud conceptual con el modelo clásico del perceptrón, ya que ambos manejan señales analógicas y generan salidas binarias mediante la aplicación de un umbral. Sin embargo, es importante destacar que este diseño no implementa redes neuronales artificiales formales en el sentido moderno: no incorpora múltiples capas ni algoritmos de aprendizaje automático o adaptación dinámica de los pesos.

Una característica destacada de estas compuertas es su capacidad para realizar múltiples operaciones lógicas simplemente ajustando los pesos w_i o modificando los umbrales de activación. Para ello, el dispositivo ideal es el amplificador operacional, ya que permite implementar ganancias ponderadas y realizar la suma de múltiples señales de entrada.

III. CELDA LÓGICA RECONFIGURABLE

Una CLR permite, con sólo dos entradas, la obtención de 16 funciones lógicas distintas a su salida, como se detalla en la Tabla I. Las compuertas no comerciales se presentan con sus expresiones booleanas.

TABLA I
COMPUERTAS POSIBLES CON DOS ENTRADAS.

Función	x_2x_1				Expresión lógica
	00	01	10	11	
f_0	0	0	0	0	GND
f_1	0	0	0	1	AND
f_2	0	0	1	0	$x_2\bar{x}_1$
f_3	0	0	1	1	x_2
f_4	0	1	0	0	\bar{x}_2x_1
f_5	0	1	0	1	x_1
f_6	0	1	1	0	XOR
f_7	0	1	1	1	OR
f_8	1	0	0	0	NOR
f_9	1	0	0	1	XNOR
f_{10}	1	0	1	0	\bar{x}
f_{11}	1	0	1	1	$\overline{\bar{x}x}$
f_{12}	1	1	0	0	\bar{x}
f_{13}	1	1	0	1	$\bar{x}\bar{x}$
f_{14}	1	1	1	0	NAND
f_{15}	1	1	1	1	Vcc

Vcc: Voltaje de corriente continua
GND: Ground/Tierra

En trabajos anteriores [7], [10], [11], [12], suelen utilizar el circuito mostrado en la Fig. 3 para la realización de estas 16 funciones, este circuito se simuló utilizando el software de NI Multisim 14.2, el cual involucra dos configuraciones clave: el sumador inversor de dos entradas y el comparador de ventana con dos amplificadores LM311.

Con este circuito permite obtener de manera sencilla las funciones $f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_{10}$ y f_{14} variando los voltajes negativos de los umbrales de acción. Sin embargo, algunas

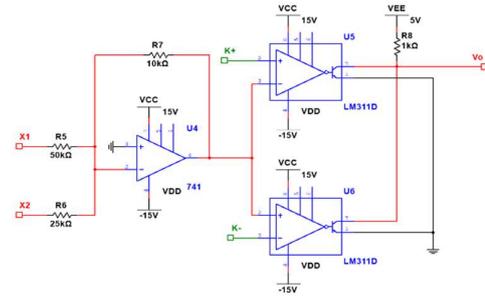


Fig. 3 Circuito de CLR encontrada en la literatura.

funciones, como $f_8, f_9, f_{10}, f_{11}, f_{12}, f_{13}$ y f_{15} requieren voltajes positivos para su correcta operación. Además, para funciones que dependen de una ganancia negativa, es decir, f_9, f_{11} y f_{13} , es necesario utilizar un circuito externo, como un amplificador inversor adicional con ganancia unitaria.

Con estas áreas de mejora, se propone una ampliación del diseño considerando los siguientes aspectos:

1. Aumento de entradas: Se incrementa el número de entradas de dos a cuatro, permitiendo integrar las tres entradas necesarias para operaciones completas de suma/resta y una adicional para un posible offset, específicamente con ganancias 1, 2, 3, 4.
2. Nueva configuración de suma ponderada: Se reemplaza el sumador inversor convencional por una configuración no inversora. Esto permite trabajar con voltajes positivos en los umbrales de acción, mejorando la versatilidad del diseño.
3. Uso de amplificadores y transistores de propósito general: Se sustituye el LM311 por un amplificador operacional cuádruple LM324 combinado con dos transistores 2N3904. Esta configuración ofrece mayor flexibilidad, ya que permite obtener salidas invertidas y no invertidas directamente en el circuito, eliminando la necesidad de ajustes externos.
4. Inversión de señales de entrada: Se añade la capacidad de manejar señales invertidas directamente desde el circuito mediante entradas con pesos negativos (-1, -2, -3, -4). Esto permite realizar operaciones tanto con señales originales como invertidas en el mismo diseño.
5. Obtención de salidas complementarias: El diseño permite generar tanto V_0 como $-V_0$ a partir del mismo circuito, facilitando la implementación de funciones lógicas complementarias como AND/NAND, OR/NOR, y XOR/XNOR.

Finalmente, el circuito propuesto, que se muestra en la Fig. 4, permite integrar estas mejoras con un diseño más compacto y flexible, logrando mayor adaptabilidad para diversas aplicaciones.

Posteriormente este circuito es reemplazado por el subcircuito de la Fig. 5 para las siguientes simulaciones.

Cabe destacar que el sistema ha sido diseñado con una separación de aproximadamente 1 V entre los niveles de umbral, lo que mejora la robustez frente a ruido o perturbaciones externas. Además, dado el carácter reconfigurable del circuito, los valores de voltaje y las

resistencias pueden reajustarse para adaptarse a distintas necesidades específicas del usuario, ofreciendo una alta flexibilidad en el diseño.

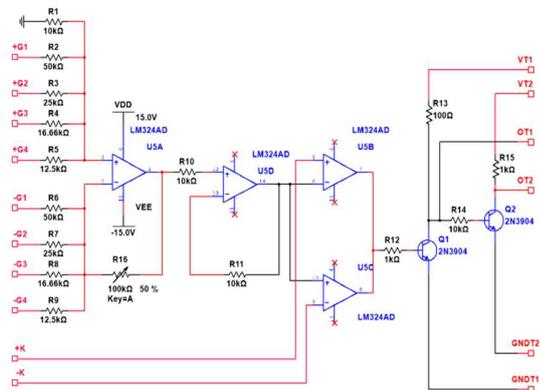


Fig. 4 Circuito propuesto para la CLR.

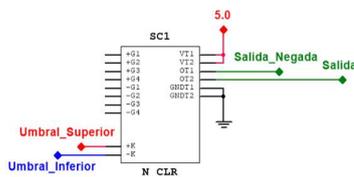


Fig. 5 Subcircuito para la CLR.

Antes de realizar el diseño a una placa de circuito impreso (PCB del inglés *printed circuit board*), se realizaron simulaciones que comprobaran su correcto funcionamiento. Como el sumador de dos bits en la Fig. 6 y un comparador de dos bits en la Fig. 7.

IV. DISEÑO DE LA ALU DE 1 BIT CON 16 FUNCIONES

En el diseño de la ALU, se utilizaron las entradas x_1 , x_2 y para las operaciones de suma y resta también se toma en cuenta la entrada C_{in} , se necesitaron de 4 selectores S3, S2, S1 y S0. Para aminorar el uso de fuentes, se procura utilizar los niveles de voltaje de alimentación en los umbrales de acción. A la salida se obtiene las 16 funciones detalladas en la Tabla II.

La ALU se completó con 11 CLR's, distribuidos en tres secciones mostradas en la Fig. 8:

- Sección 1 de operaciones lógicas no negadas: 3 CLR's se encargan de estas funciones y de la operación $\overline{x_2}$.
- Sección 2 de operaciones aritméticas: 4 CLR's realizan las operaciones de suma y resta, incluyendo el cálculo de los acarreo.
- Sección 3 del multiplexor y operaciones negadas: 4 CLR's procesan las operaciones lógicas negadas y combinan las salidas de las demás secciones en sólo dos señales de salida (Bit 0 y Bit 1).

SUMADOR

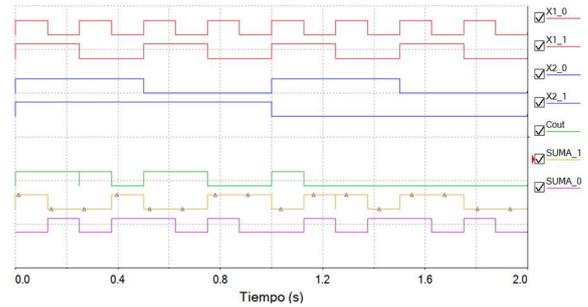


Fig. 6 Simulación para sumador de dos bits completo con 4 CLR's.

COMPARADOR

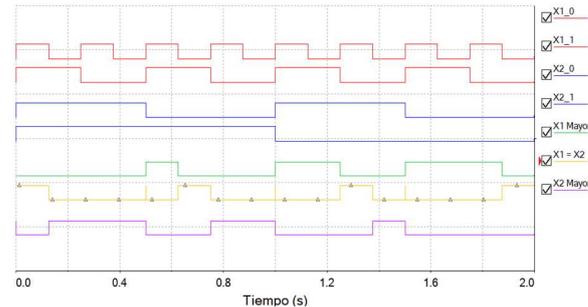


Fig. 7 Simulación para comparador de dos bits con 3 CLR's.

TABLA II
FUNCIONES DE LA ALU.

Selector				Salida	
S3	S2	S1	S0	Bit 1	Bit 0
0	0	0	0	Vcc	$x_1 < x_2$
0	0	0	1	Vcc	$x_1 > x_2$
0	0	1	0	Vcc	x_1
0	0	1	1	Vcc	AND
0	1	0	0	Cout +	$x_1 + x_2 + C_{in}$
0	1	0	1	Cout -	$x_1 - x_2 - C_{in}$
0	1	1	0	Vcc	XOR
0	1	1	1	Vcc	OR
1	0	0	0	Vcc	$\overline{x_1 < x_2}$
1	0	0	1	Vcc	$\overline{x_1 > x_2}$
1	0	1	0	Vcc	$\overline{x_1}$
1	0	1	1	Vcc	NAND
1	1	0	0	GND	$\overline{x_2}$
1	1	0	1	GND	x_2
1	1	1	0	Vcc	XNOR
1	1	1	1	Vcc	NOR

Las primeras 4 funciones se obtienen de una misma CLR, su conexión se muestra en la Fig. 9. Para x_1 , se asigna un peso de $w_1 = 7$, por lo tanto, se conecta x_1 a las ganancias 3 y 4 identificadas como +G3 y +G4; para x_2 , un peso de $w_2 = 2$ (+G2). Además, se utilizan las señales de los selectores S0 y S1 en la suma ponderada con los pesos de -3 y -5 respectivamente (-G3 y -G1 junto con -G4). La señal del selector S2 habilita la salida, que debe estar en un nivel bajo.

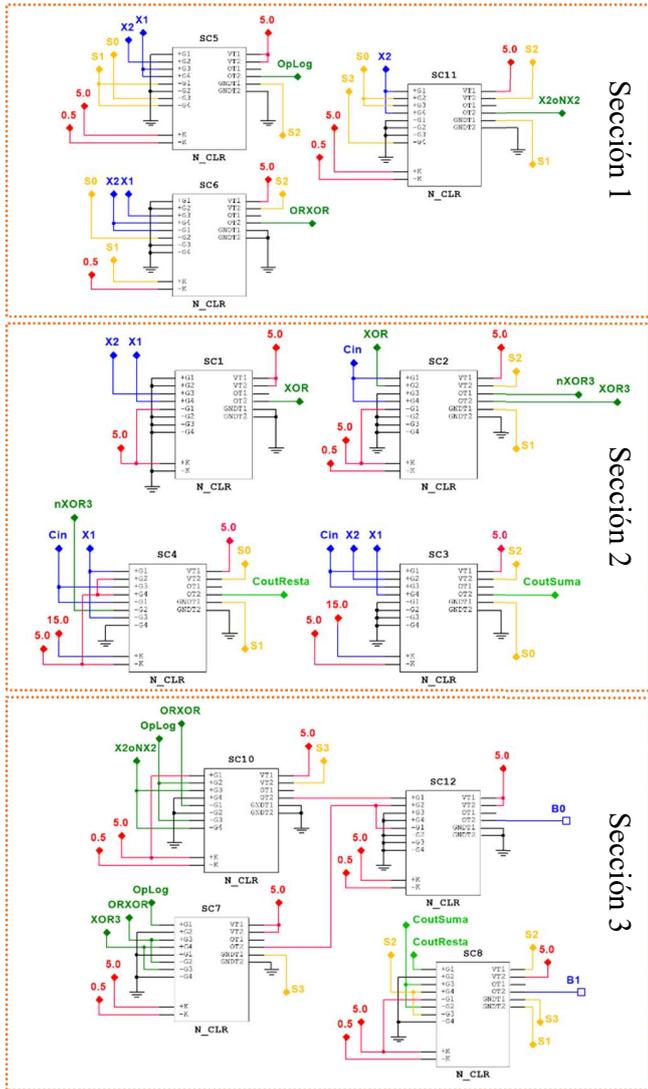


Fig. 8 ALU de 1 bit utilizando 11 CLR.

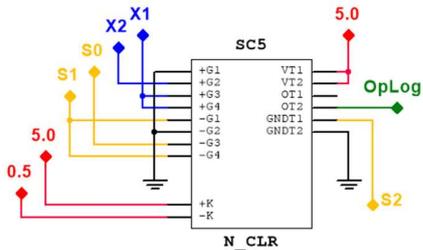


Fig. 9 CLR que obtiene 4 operaciones lógicas.

La Tabla III muestra los valores de la suma ponderada de esta primera CLR, como el umbral está definido en $K_+ = 5 V$ y $K_- = 0.5 V$. Según la permutación de los selectores:

- En 0000, la salida corresponde a $x_2 > x_1$.
- En 0001, la salida cambiara a $x_1 > x_2$.
- En 0010, la salida será igual a x_1 .
- En 0011, la salida corresponde a la compuerta AND.

TABLA III
PESOS PARA LA CLR DE 4 OPERACIONES.

S2	S1	S0	x_2	x_1	$g(x) = 7x_1 + 2x_2 - 3S_0 - 5S_1$	Función
0	0	0	0	0	0	
0	0	0	0	1	7	$x_1 < x_2$
0	0	0	1	0	2	
0	0	0	1	1	9	
0	0	1	0	0	-3	
0	0	1	0	1	4	$x_1 > x_2$
0	0	1	1	0	-1	
0	0	1	1	1	6	
0	1	0	0	0	-5	
0	1	0	0	1	2	x_1
0	1	0	1	0	-3	
0	1	0	1	1	4	
1	1	0	0	0	-8	
0	1	1	0	1	-1	AND
0	1	1	1	0	-6	
0	1	1	1	1	1	

Para el resto de permutaciones con $S2 = 1$ la salida no estará habilitada. Finalmente, la señal generada por esta CLR se dirige a la salida Bit 0 cuando se habilita otra CLR de la sección 3 con $S3 = 0$.

Las siguientes funciones de la Tabla IV son las operaciones aritméticas. Para realizar la suma y la resta, que corresponden ambas a una XOR de tres entradas, se emplean dos CLR de la siguiente manera:

Primera CLR:

- Realiza la operación XOR entre las entradas x_1 y x_2 .
- Los pesos utilizados son: $w_1 = 4$, $w_2 = 3$ y un *offset* en la entrada de $w_0 = -1$.
- El umbral está definido como $K_+ = 5 V$ y $K_- = 0 V$.
- La salida de esta CLR se conecta a la segunda CLR.

Segunda CLR:

- Procesa la salida de la primera CLR con un peso de $w = 2$.
- También se considera la señal C_{in} con un peso de $w = 5$ y un *offset* de $w_0 = -1$.
- Los umbrales son los mismos: $K_+ = 5 V$ y $K_- = 0 V$.

Los valores de la suma ponderada para cada una de estas CLR se muestran en las Tablas IV y V.

TABLA IV
SUMA PONDERADA PARA LA OPERACIÓN XOR ENTRE x_1 y x_2 .

offset	x	x	$g(x) = -1 + 4x + 3x$	Función
1	0	0	-1	
1	0	1	3	XOR
1	1	0	2	
1	1	1	6	

TABLA V
SUMA PONDERADA PARA LA OPERACIÓN XOR TRIPLE.

offset	$x_1 \oplus x_2$	C_{in}	$g(x) = -1 + 2(x_1 \oplus x_2) + 5C_{in}$	Función
1	0	0	-1	
1	1	1	6	XOR
1	1	0	1	$x_1 \oplus x_2 \oplus C_{in}$
1	0	1	4	

Para cada acarreo de salida, se utiliza una CLR.

El acarreo de la suma tiene los pesos asignados de: $w_1 = 4$, $w_2 = 2$ y $w_3 = 4$ para C_{in} y un umbral de $K_+ = 15 V$ y $K_- = 5 V$. Mientras que el acarreo de la resta tiene los pesos asignados de $w_1 = -2$, $w_3 = 2$ para C_{in} , -2 para la señal $\overline{x_1 \oplus x_2 \oplus C_{in}}$ y 6 para un *offset* y un umbral de $K_+ = 15 V$ y $K_- = 5 V$. La Tabla VI y la Tabla VII muestran los valores que tomaran las CLR's encargadas de los acarreos de salida.

La Tabla VIII muestra los valores que tomara la CLR encargada de las operaciones OR y XOR con los umbrales de $K_+ = 5 V$ y $K_- = 0.5 V$.

TABLA VI
ACARREO DE SALIDA DE LA SUMA.

C_{in}	x_2	x_1	$g(x) = 4x_1 + 2x_2 + 4C_{in}$	Función
0	0	0	0	
0	0	1	4	
0	1	0	2	
0	1	1	6	C_{out}
1	0	0	4	
1	0	1	8	
1	1	0	6	
1	1	1	10	

TABLA VII
ACARREO DE SALIDA DE LA RESTA.

offset	$\overline{x_1 \oplus x_2 \oplus C_{in}}$	C_{in}	x_1	$g(x)$	Función
1	1	0	0	4	
1	0	0	1	4	
1	0	0	0	6	
1	1	0	1	2	
1	0	1	0	8	C_{out}
1	1	1	1	4	
1	1	1	0	6	
1	0	1	1	6	

$$g(x) = 6 - 2x_1 + 2C_{in} - 2(x_1 \oplus x_2 \oplus C_{in})$$

TABLA VIII
SUMA PONDERADA PARA LAS OPERACIONES XOR Y OR.

S0	x_2	x_1	$g(x) = 3x_1 + 3x_2 - 2S_0$	Función
0	0	0	0	
0	0	1	3	XOR
0	1	0	3	
0	1	1	6	
1	0	0	-2	
1	0	1	1	OR
1	1	0	1	
1	1	1	4	

Por último, la CLR faltante de la sección de operaciones por explicar, es la que obtiene la operación de x_2 y $\overline{x_2}$. La Tabla IX muestra los valores que tomara la suma ponderada, resaltando en color lo que entra en el umbral utilizado de $K_+ = 5.5 V$ y $K_- = 0.5 V$.

TABLA IX
SUMA PONDERADA PARA LA OPERACIÓN x_2 y $\overline{x_2}$.

S3	S0	x_2	$g(x) = 5x_2 + 5S_0 - 4S_3$	Función
0	0	0	0	
0	0	1	5	x_2
1	0	0	-4	
1	0	1	1	
0	1	0	5	
0	1	1	10	
1	1	0	1	$\overline{x_2}$
1	1	1	6	

Para la última sección de la ALU, se tienen 4 CLR's, una para la salida Bit 1 y el resto para la salida Bit 0, en las Fig. 10 y 11 se exponen las conexiones para la obtención de las salidas Bit 0 y Bit 1, la primera CLR de la Fig. 10, es habilitada cuando el selector S3 está en un nivel alto, junta las operaciones lógicas, pero utilizando ganancias negativas, por lo tanto, todas las entradas se niegan, de aquí que se obtengan las operaciones negativas, $\overline{x_2} > x_1$, $\overline{x_1} > x_2$, $\overline{x_1}$, *NAND*, *XNOR*, *NOR*. Mientras que la segunda CLR habilitada al tener el selector S3 en un nivel bajo une las operaciones lógicas sin modificarlas junto con la operación de la suma y la resta. Ambas salidas se unen en la tercera CLR con ganancias unitarias dando lugar a la salida Bit 0. Para la obtención de la salida Bit 1, se utiliza la CLR de la Fig. 11, donde se unen los acarreos de salida de la suma y la resta.

Cuando los selectores no están configurados para la operación de suma ni de resta, la salida Bit 1 permanece en un nivel alto, excepto en el caso de las operaciones x_2 y \bar{x}_2 , donde se registra un nivel bajo.

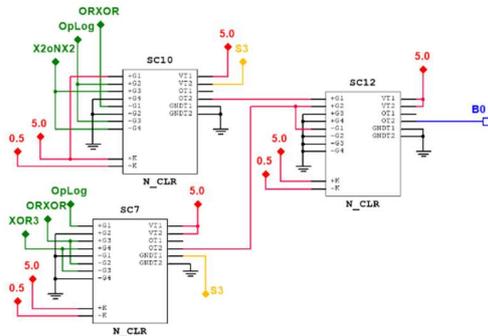


Fig. 10. CLR para la salida Bit 0.

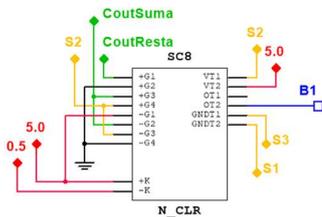


Fig. 11. CLR para la salida Bit 1.

V. IMPLEMENTACIÓN DE LA ALU

El armado de la ALU se ilustra en la Fig. 12 para la cual se utilizan 2 fuentes de voltaje, un osciloscopio y generadores de trenes de pulso para las señales de entrada x_1 , x_2 y C_{in} , mientras que los selectores se ajustaron manualmente a un nivel alto o a un nivel bajo.



Fig. 12 Implementación del ALU en laboratorio.

Desde la Fig. 13 a la Fig. 16, se presentan los resultados de 4 de las 16 funciones de la ALU, observados en el osciloscopio. En las capturas, la señal x_1 aparece en la parte superior, representada en color azul rey o negro, con una frecuencia que es el doble de la de x_2 . La señal x_2 , en color azul cian, tiene el doble de frecuencia que la señal C_{in} , la cual se muestra en color violeta. Por último, la salida de la ALU se visualiza en la parte inferior, en color verde. Para la operación de resta, también se incluye el acarreo de salida, representado en color azul rey.

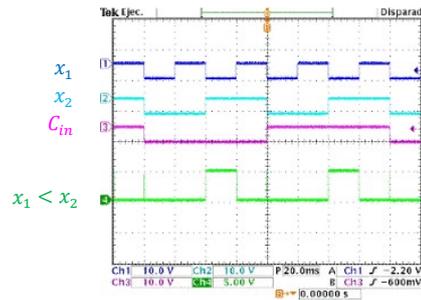


Fig. 13 Función $x_1 < x_2$.

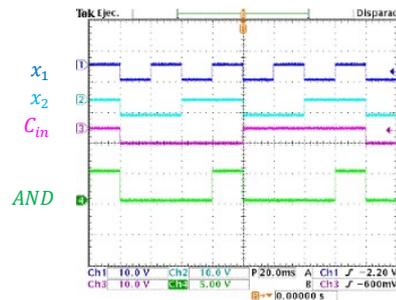


Fig. 14 Función AND.

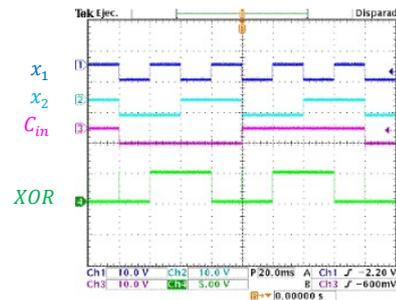


Fig. 15 Función XOR.

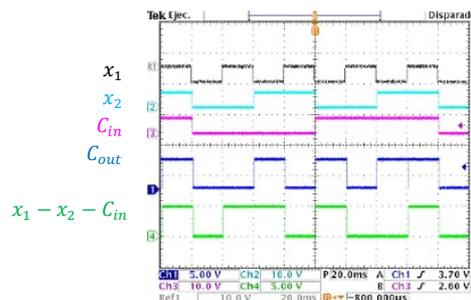


Fig. 16 Función $x_1 - x_2 - C_{in}$.

VI. CONCLUSIÓN

Como se evidenció en las Tablas 3 a la 9, los valores de voltaje utilizados para los 11 umbrales de acción en esta ALU fueron: 0 V, 0.5 V, 5 V, 5.5 V y 15 V. De estos, 0 V (GND), 5 V y 15 V se emplean también para la alimentación del sistema, junto con -15 V, lo que da un total de seis niveles de voltaje. Esta configuración contrasta con la ALU analógica de un bit presentada en [7], que emplea 11 niveles de voltaje distintos. A

pesar de ello, la ALU propuesta aquí logra implementar 16 funciones, el doble de las reportadas en [7], lo que sugiere una mayor eficiencia en la relación funciones/recursos.

Si bien no se desarrolló una ALU completa de dos bits como en [9], se reportó exitosamente un sumador y un comparador de dos bits, lo cual demuestra el potencial de escalabilidad del sistema propuesto.

En comparación con dispositivos clásicos como el 74LS181 (ALU de 4 bits), esta ALU no solo permite realizar funciones lógicas, sino que ofrece reconfiguración mediante variaciones de voltaje, operación con señales analógicas, y escalabilidad en entradas y salidas, lo que la hace ideal para aplicaciones híbridas. A diferencia del 74LS83 (sumador de 4 bits) y el 74LS85 (comparador de 4 bits), ambos de propósito fijo, el sistema presentado no se limita a una sola tarea: puede implementar distintas funciones lógicas, aritméticas y de selección, como se evidenció con la implementación de un multiplexor de 6 a 1 en la sección 3 de la ALU (Fig. 10).

En conjunto, estos resultados destacan la versatilidad del enfoque reconfigurable propuesto, con ventajas notables en adaptabilidad y eficiencia frente a soluciones digitales tradicionales.

VII. TRABAJOS A FUTURO

- Explorar la posibilidad de utilizar sólo los 4 niveles requeridos para la alimentación (0 V, 5 V y ± 15 V) en las etapas de comparadores de ventanas de la ALU, buscando alternativas de pesos.
- Validar experimentalmente el funcionamiento de la CLR con diferentes encapsulados de amplificadores cuádruples, a pesar de que en simulación se obtuvieron resultados similares con diferentes modelos (LM324, TL074, LM348 Y TL084).
- Desarrollar una herramienta de software para agilizar el cálculo de pesos en la asignación, complementando el enfoque empírico utilizado en esta investigación.
- Extender el diseño a nivel transistor, permitiendo optimizar consumo y área.
- Explorar la implementación de lógicas no binarias, como la lógica ternaria, cuya viabilidad ha sido demostrada en trabajos recientes [8], con el objetivo de ampliar las capacidades de procesamiento de la CLR.

REFERENCIAS

- [1] A. S. Sedra, K. C. Smith, T. C. Carusone, and V. Gaudet, "Microelectronic Circuits, 8th Edition," *Oxford University Press*, 2020.
- [2] S. Huang, L. Waeijen, and H. Corporaal, "How Flexible is Your Computing System?," *ACM Transactions on Embedded Computing Systems*, vol. 21, no. 4, pp. 1–41, Jul. 2022, doi: 10.1145/3524861.
- [3] V. Aggarwal, U. Agrawal, R. Jain, and R. Kapoor, "Advances in Arithmetic Logic Units: A Comprehensive Review," in *2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, IEEE, Jul. 2024, pp. 1–11. doi: 10.1109/APCIT62007.2024.10673660.
- [4] Y. Liu, C. Zhang, and H. Zhang, "The Mechanism of The Arithmetic Logic Unit," *Highlights in Science, Engineering and Technology*, vol. 81, pp. 97–108, Jan. 2024, doi: 10.54097/qzqppap16.
- [5] H. Peng, Y. Yang, L. Li, and H. Luo, "Harnessing piecewise-linear systems to construct dynamic logic architecture," *Chaos*, vol. 18, no. 3, 2008, doi: 10.1063/1.2953494.
- [6] S. Sinha and W. L. Ditto, "Dynamics based computation," *Phys Rev Lett*, vol. 81, no. 10, 1998, doi: 10.1103/PhysRevLett.81.2156.
- [7] R. E. Lozoya, "Celdas Lógicas Reconfigurables," Universidad Autónoma de San Luis Potosí, San Luis Potosí, México, 2017.
- [8] A. Unutulmaz and C. Ünsalan, "Implementation and Applications of a Ternary Threshold Logic Gate," *Circuits Syst Signal Process*, vol. 43, no. 2, 2024, doi: 10.1007/s00034-023-02512-1.
- [9] M. Hayati and A. Rezaei, "Design of a New Optimized Universal Logic Gate for Quantum-Dot Cellular Automata," *IETE J Res*, vol. 68, no. 2, 2022, doi: 10.1080/03772063.2019.1643262.
- [10] I. Díaz and E. Campos, "Toward a Voltage Reconfigurable Logic Gate," *Memorias del Congreso Nacional de Control Automático*, vol. 6, no. 1, pp. 503–506, Oct. 2023, doi: 10.58571/CNCA.AMCA.2023.107.
- [11] R. Rivera, "Diseño e implementación de arreglos de compuertas dinámicas programables en campo y su aplicación en redes Booleanas," Instituto Potosino de Investigación Científica y Tecnológica, A.C., San Luis Potosí, 2018.
- [12] B. Cerda, "Diseño de un Circuito Electrónico para una Red No Linear Celular Reconfigurable," Universidad Autónoma de San Luis Potosí, San Luis Potosí, México, 2018.