

Diseño de una Unidad Aritmética Lógica (ALU)

ALU capaz de realizar suma, resta y multiplicación.

Autores: Anthony Hurtado, Einar Barraza

anthony-hurtado-11@hotmail.com, enarbarraza@hotmail.com;

Escuela de Ingeniería Biomédica, Universidad Latina de Panamá, Ciudad de Panamá, Panamá

Correo-e: web@ulatina.edu.pa

Resumen— En todo sistema computacional, existen unidades funcionales encargadas de realizar las operaciones aritméticas, éstas son llamadas unidades de aritmética y lógica. Las ALU fue creada debido a que se señaló que una función esencial de muchas de las computadoras y de las calculadoras es la realizar operaciones aritméticas. La unidad aritmética lógica contiene alrededor de dos registros de flip-flops: el registro B y el registro acumulador. La finalidad primordial consiste en aceptar datos binarios que están almacenados en la memoria y ejecutar operaciones aritméticas con estos datos, de acuerdo con instrucciones que provienen de la unidad de control. ALU es un requisito fundamental para una computadora porque está garantizado que tendrá que efectuar operaciones matemáticas básicas, incluyendo adición, sustracción, multiplicación, y división. La mayoría de las operaciones se basan en el mismo circuito del sumador pero complementados de diferentes formas, por ejemplo, la resta tiene circuito sumador negado.

Palabras clave— Algoritmos, Aritmética, Bits, Registros, Transición.

Abstract— In every computer system, there are functional units in charge of performing arithmetic operations, these are called arithmetic and logic units. The ALU was created because it was pointed out that an essential function of many computers and calculators is to perform arithmetic operations. The arithmetic logic unit contains about two flip-flop registers: The B register and the accumulator register. The primary purpose is to accept binary data that is stored in memory and perform arithmetic operations on this data, according to instructions from the control unit. ALU is a fundamental requirement for a computer because it is guaranteed that it will have to perform basic mathematical operations, including addition, subtraction, multiplication, and division. Most of the operations are based on the same adder circuit but complemented in different ways, for example, subtraction has negated adder circuit.

Key Word — Algorithms, Arithmetic, Bits, Registers, Transition.

I. INTRODUCCIÓN

El matemático John von Neumann propuso el concepto de la ALU en 1945, cuando escribió un informe sobre las fundaciones para un nuevo computador llamado EDVAC (Electronic Discrete Variable Automatic Computer) (Computador Automático Variable Discreto Electrónico). Más adelante, en 1946, trabajó con sus colegas diseñando un

computador para el Princeton Institute of Advanced Studies (IAS) (Instituto de Princeton de Estudios Avanzados). El IAS computer se convirtió en el prototipo para muchos computadores posteriores. En la propuesta, von Neumann esbozó lo que él creyó sería necesario en su máquina, incluyendo una ALU. Von Neumann explicó que una ALU es un requisito fundamental para una computadora porque está garantizado que tendrá que efectuar operaciones matemáticas básicas, incluyendo adición, sustracción, multiplicación, y división.[1]

Por lo tanto, creyó que era "razonable que una computadora debería contener los órganos especializados para estas operaciones"

En la actualidad, las operaciones que realiza una ALU entre los distintos datos, que deben ser iguales a los que emplea el circuito digital, es la representación del número binario de complemento a dos. Sin embargo, no siempre fue así: en sus comienzos, las distintas computadoras utilizaron una amplia gama de sistemas numéricos, como el Complemento a uno, o el sistema decimal, adecuando el diseño de las ALU a cada sistema. A través del tiempo, el Complemento a dos resultó ser el preferido por la industria, al ser el de mayor simplicidad para la diagramación del circuito de la ALU. [2]

II. CONTENIDO

En todo sistema computacional, existen unidades funcionales encargadas de realizar las operaciones aritméticas, éstas son llamadas unidades de aritmética y lógica (ALU). Comúnmente trabajan a numeración de punto fijo o punto flotante, utilizando la representación binaria de los números de base decimal. El funcionamiento de una unidad aritmética convencional se basa en la utilización de sumadores binarios, que están compuestos por compuertas que cumplen funciones lógicas sencillas.

Las ALU fue creada debido a que se señaló que una función esencial de muchas de las computadoras y de las calculadoras es la realizar operaciones aritméticas. Estas operaciones se llevan a cabo mediante la unidad aritmética-lógica de una computadora, donde se combinan diversas compuertas lógicas y flip-flops de tal forma en que puedan sumar, restar, multiplicar y dividir números binarios.

La ALU o unidad aritmética lógica consiste en un circuito digital que permite realizar operaciones aritméticas y lógicas

entre dos números. ALU viene del inglés y es acrónimo de Arithmetic Logic Unit. En español, la unidad aritmética lógica, [1] fueron implementadas por primera vez en el año 1945 cuando John Von Neumann redactó un informe relacionado a un nuevo computador llamado EDVAC, posteriormente se llegó a la conclusión de que es un requisito indispensable para cualquier computadora el poder efectuar operaciones matemáticas básicas.

La finalidad primordial de la ALU consiste en aceptar datos binarios que están almacenados en la memoria y ejecutar operaciones aritméticas con estos datos, de acuerdo con instrucciones que provienen de la unidad de control. Podemos mencionar algunas características básicas de un ALU:

- El circuito ALU es simplemente un operador, es decir, sólo realiza operaciones.
- La ALU no toma decisiones.
- Las entradas deben contener tanto la magnitud como el signo que corresponda a la operación.
- La ALU requiere de un mecanismo de control que le permita saber el tipo de operación a realizar.

2.1. Elementos principales de un ALU:

La unidad aritmética lógica contiene alrededor de dos registros de flip-flops: el registro B y el registro acumulador. Además, contiene lógica combinatoria, que se encarga de llevar a cabo las operaciones aritméticas sobre los números binarios que se encuentran almacenados en el registro B y el acumulador (ver figura 1). Una secuencia común de operaciones puede ocurrir de la manera siguiente:

1. La unidad de control recibe una instrucción de la unidad de memoria, especificando que un número almacenado en la memoria se va a sumar al número que está almacenado en ese momento en el registro acumulador.
2. El número que se va a sumar es transferido de la memoria al registro B.
3. El número contenido en el registro B y el número en el registro acumulador se suman en los circuitos lógicos por el comando establecido desde la unidad de control. La suma resultante se envía entonces al acumulador para luego ser almacenada.
4. El nuevo número en el acumulador puede permanecer para que se pueda sumar otro número a él, o, si el proceso aritmético particular llega a su fin, puede ser transferido a la memoria para ser almacenado. [3]

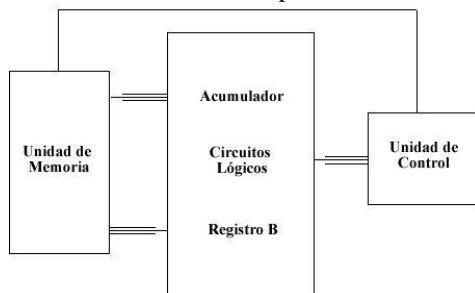


Figura 1. Diagrama de bloques de elementos principales de un ALU.

2.2. Composición básica de un ALU

- ✓ Circuito Operacional
- ✓ Registros de Entradas
- ✓ Registro Acumulador y un
- ✓ Registro de Estados
- ✓ Conjunto de registros que hacen posible la realización de cada una de las operaciones.

La mayoría de las acciones de la computadora son realizadas por la ALU. La ALU toma datos de los registros del procesador. Estos datos son procesados y los resultados de esta operación se almacenan en los registros de salida de la ALU. Otros mecanismos mueven datos entre estos registros y la memoria.[1] Una unidad de control controla a la ALU, al ajustar los circuitos que le señala a la ALU qué operaciones realizar.

2.3. Operaciones básicas que pueden realizar un ALU

La mayoría de las ALU pueden realizar las siguientes operaciones:

- Operaciones aritméticas de números enteros (adición, sustracción, y a veces multiplicación y división)
- Operaciones lógicas de bits (AND, NOT, OR, XOR, XNOR)
- Operaciones de desplazamiento de bits (Desplazan o rotan una palabra en un número específico de bits hacia la izquierda o la derecha, con o sin extensión de signo). Los desplazamientos pueden ser interpretados como multiplicaciones o divisiones por 2.

Pueden realizar operaciones más complejas como el cálculo de la raíz cuadrada, emulación de software, entre otros.

La ALU realiza por separado las operaciones aritméticas de las lógicas, pero los circuitos se construyen integrados en un mismo chip. Por lo tanto, usa señales para identificar el tratamiento que debe dar a las entradas para saber, primero: el tipo de modo ya sea una operación aritmética o lógica; y segundo el tipo de función para la operación seleccionada suma, resta, multiplicación, etc.

Un ejemplo del diseño de un ALU (fig. 2) El modelo posee cuatro entradas para A y cuatro para B, cada entrada es un bit, lo que quiere decir que este modelo hace operaciones de 4 bits. Donde, posee otra entrada y una salida para el acarreo de las operaciones aritméticas (Cin Cout) y otras entradas de un bit para la selección de función que corresponda S0 S1, y una última para el modo, en este caso S2.

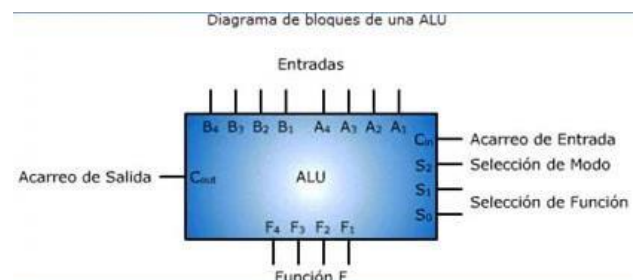


Figura 1. Diagrama de bloques de un ALU.

2.4. Entradas y Salidas de un ALU

- ✓ Las entradas a la ALU: son los datos en los que se harán las operaciones llamados operandos y un código desde la unidad de control indicando qué operación realizar.
- ✓ Su salida: es el resultado del cómputo de la operación.

En muchos diseños la ALU también toma o genera como entradas o salidas un conjunto de códigos de condición desde o hacia un registro de estado. Estos códigos son usados para indicar casos como acarreo entrante o saliente, overflow, división por cero, etc

2.5. ALU De 4 Bits de Entrada

2.5.1. Suma

El componente más básico es el que realiza la suma, ya que con este es posible realizar las demás operaciones como: la resta, multiplicación y división.

Utilizamos un sumador con propagación de acarreo, también denominado sumador de acarreo serie, el cual se encuentra conformado por varios sumadores completos en los cuales el acarreo de salida se conecta a la entrada de acarreo del sumador siguiente. Puede notarse que el primero de los sumadores puede ser reemplazado por un semisumador suponiendo que el acarreo de entrada C_i es cero.

Para que funcione el sumador, se deben recibir las entradas A y B, luego sus salidas son el resultado de la suma binaria. Para realizar la suma se debe colocar cada entrada y el valor de acarreo se pasa al circuito siguiente para hacer parte de la operación por ejemplo se realiza la siguiente suma:

Binario /Decimal

A=> 1001 = 9 (1)

B=> 0110 = 6 (2)

S=> 1112 = 15 (3)

A continuación (figura 3), vemos el diagrama en bloque de un sumador de 4 bit y ejemplo de una suma de los componentes. Tenemos un sumador (a la derecha) y tres sumadores completos, todos colocados en serie.

- El primer semisumador no necesita la entrada del carry-in al estar arracando desde cero.

- Las salidas S0, S1, S2, S3 y S4 son los resultados de la suma binaria, S5 es el carry-out de la última suma.

Gracias a esto, no se presentará un overflow, donde no se ignora el último Carry-out. El rango del sumador es 00000 a 11110 (0 a 30).

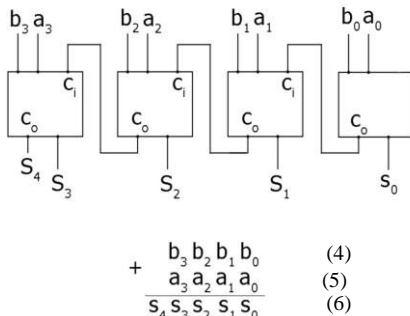


Figura 3: Diagrama en bloque de una suma de 4 bits.

2.5.1.1. Ecuaciones para un sumador

$$S = A \oplus B \oplus C_i \quad (7)$$

$$= A\bar{B}\bar{C}_i + \bar{A}B\bar{C}_i + \bar{A}\bar{B}C_i + ABC_i \quad (8)$$

$$C_o = AB + BC_i + AC_i \quad (9)$$

Formulas de reducción:

$$S = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}c + abc = c(\bar{a}\bar{b} + a\bar{b} + \bar{a}b + ab) = c(\bar{a}(\bar{b} + b) + a(\bar{b} + b)) = c(\bar{a} + a) = c \oplus m = c \oplus (a \oplus b) \quad (10)$$

$$C_o = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc = ab + c(a \oplus b) \quad (11)$$

2.5.1.2. Tabla de la verdad

A continuación, encontramos la tabla de transición (tabla 1) para un circuito o sumador de 4 bits, Donde tenemos A, B, y el C_i como entrada y S y C_0 como salidas

TABLA 1

TABLA DE TRANSICIÓN DE UN SUMADOR

A	B	C_i	S	C_o
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 1: tabla de transición de un sumador

En la siguiente figura (figura 4 y 5) se encuentra el circuito usando compuertas lógicas para un sumador de cuatros bits. Posee las entradas A y B, la entrada de acarreo, la salida. Se han utilizado compuertas And, OR y Xor.

El circuito comienza con un semisumador, como ya se mencionó, debido a que no es necesario la entrada del carry-in. Posteriormente, se añade un sumador completo y se coloca en serie con un semisumador. Se pasa como entrada el segundo para de dígitos a sumar y el carry-out del semisumador como carry-in.

Luego, se repite la misma lógica con los dos sumadores completos restantes. Las salidas de la suma van una a cada led junto con el carry-out del último sumador.

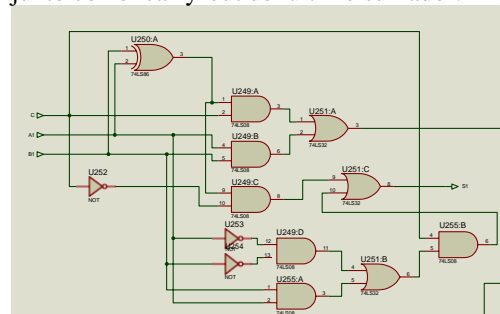


Figura 4. Circuito sumador.

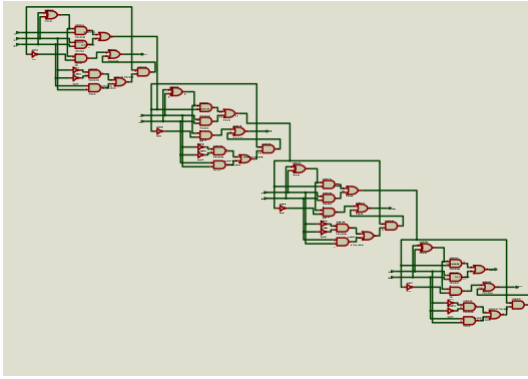


Figura 5. Circuito sumador para 4 bits.

2.5.2. Restador

La resta se implementa mediante un sumador. Los circuitos que se utilizan para restar dos números binarios se llaman Sustractores. Cuando los números binarios que se van a restar son de bits individuales, entonces podemos usar un medio restador para realizar la tarea, mientras que, si necesitamos restar tres números binarios de bits individuales entre los cuales dos serán generalmente entradas, mientras que el otro será el pedir prestado, tendremos que usar el restador completo. En nuestro caso necesitamos restar dos números binarios de n (4) bits, en nuestro caso exige el uso de n -bit. Un restador paralelo. Para lograr la resta, se puede usar el mismo circuito que para la suma siempre que tengamos el número para ser restado en su complemento de 2. Esta tarea de expresar el número en forma de complemento a 2 se puede llevar a cabo utilizando primero las puertas NOT para invertir los bits en el número binario. Sin embargo, para agregar 1 a LSB, se puede hacer uso del primer sumador en la secuencia de n sumadores completos utilizados en el circuito simplemente proporcionando una alta lógica en su acarreo (Cin). Como resultado, uno puede diseñar el n -bit restador paralelo. En la figura 6 podemos observar el diagrama en bloque de un restador con propagación de arrastre.

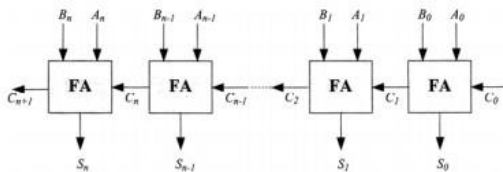


Figura 6: Diagrama en bloque para un restador.

El funcionamiento de este circuito es sencillo y es muy similar al de un sumador paralelo. Podemos observar el circuito diseñado en la figura 7 y 8.

El método consiste en llevar al minuendo a una de las entradas y el sustraendo en complemento 2 a la otra entrada. En la diferencia, cada bit del sustraendo se resta de su correspondiente bit del minuendo para formar el bit de la diferencia. El préstamo ocurre cuando el bit del minuendo es menor al bit del sustraendo, de tal forma que se presta un 1 de la siguiente posición significativa. Podemos observar la tabla 2 para la representación de un sumador/restador.

La resta en complemento a 2 se hace:

$$A - B = A + \bar{B} + 1$$

(12)

Procedimiento para restar:

- Se invierten los bits del sustraendo B
- Se pone la entrada de acarreo a 1 en un sumador
- Se suma A

TABLA 2

TABLA DE VERDAD (FUNCIONAMIENTO)

S	Función	Comentario
0	$A + B$	Suma
1	$A + B' + 1$	Resta

Tabla 2: Tabla de verdad del funcionamiento de un sumador/restador.

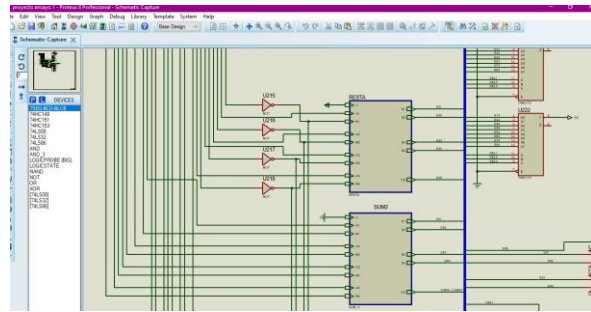


Figura 7. Entrada negada a un restador y comparación con la suma

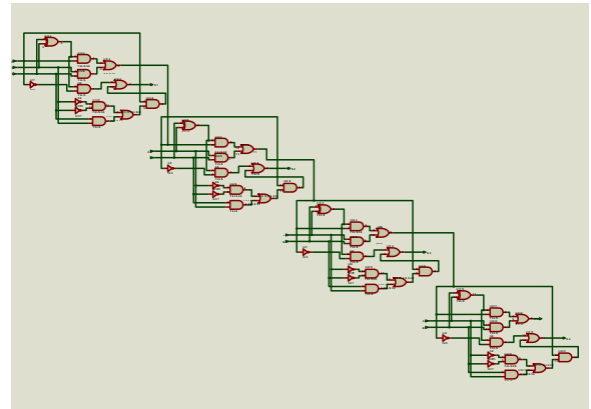


Figura 8: Circuito de un restador.

TABLA 3

TABLA DE LA VERDAD DE UN RESTADOR

A	B	Cin	S	Cout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Tabla 1. Tabla de transición de un restador.

2.5.3. Multiplicador

La multiplicación consiste en una serie de operaciones AND entre los distintos bits y una serie de sumas.

- Se requieren de 2^4 compuertas AND.
- Se requiere de N sumadores de N bits

En esta práctica se va a diseñar un circuito que multiplique dos números binarios de 4 bits utilizando el método natural de suma y desplazamiento. La multiplicación es una operación matemática que exige producto y desplazamiento. Al tratar con lógica digital, esta operación se realiza mediante una puerta lógica AND y el desplazamiento mediante un registro de desplazamiento. Cuando se realiza la multiplicación de dos números binarios mediante el procedimiento tradicional de multiplicaciones se puede observar como la multiplicación consiste en realizar sumas desplazadas a la izquierda. Al ser los datos binarios, los sumandos sólo pueden ser el primer operando de la multiplicación o cero. [1]

El algoritmo de multiplicación por sumas y desplazamientos propuesto consiste en realizar cada una de estas sumas parcialmente. La idea básica es sumar a un resultado parcial un operando cada vez que un bit del otro operando sea 1. En cada paso se realiza una suma parcial y se guarda en un registro separado el bit menos significativo del resultado obtenido. Esto se debe a que en la siguiente suma parcial este bit ya no interviene. Realizando sucesivamente la operación suma y desplazamiento a la derecha de un bit se obtiene el resultado de la multiplicación. Ver ejemplo en la figura 9 y 10. Consta de un sumador completo y una puerta AND lógica. (Figura 11 y 12)

$$\begin{array}{r}
 1010 \\
 \times 0101 \\
 \hline
 1010 \\
 0000 \\
 1010 \\
 + 0000 \\
 \hline
 0110010
 \end{array}$$

Procedimiento de multiplicación tradicional

Figura 9: Ejemplo de multiplicaciones tradicional.

Algoritmo de sumas y desplazamientos

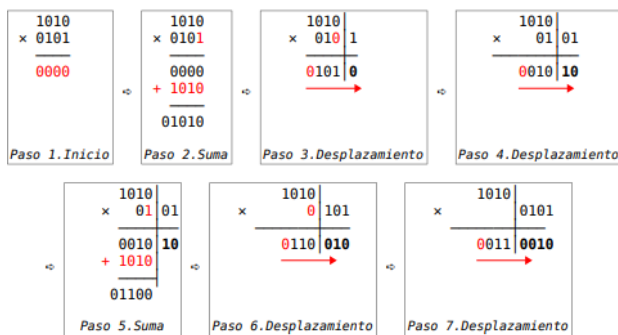


Figura 10: Ejemplo de algoritmo de sumas y desplazamientos.

TABLA 4
TABLA DE VERDAD DE UN MULTIPLICADOR

Entradas				Salida A*B			
Numero A		Numero B					
A1	A0	B1	B0	M3	M2	M1	M0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Tabla 4: Tabla de transición de un multiplicador.

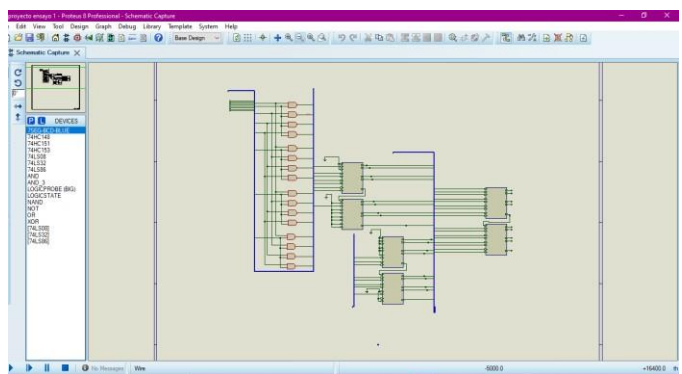


Figura 11: Circuito de un multiplicador

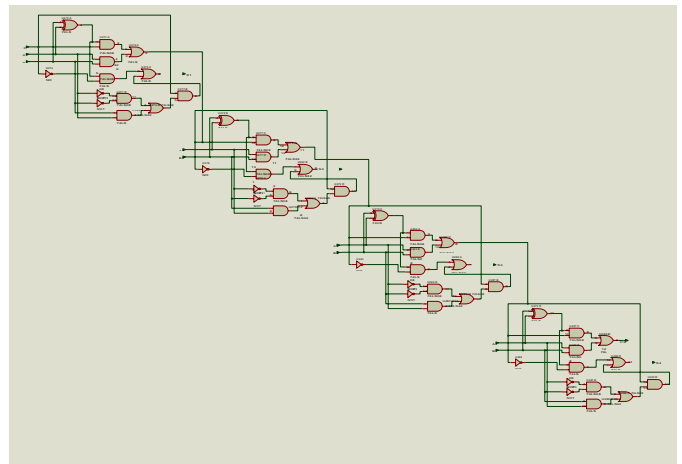


Figura 12: Circuito dentro de cada bloque de sumador del multiplicador.

III. CONCLUSIONES

La ALU o unidad aritmética lógica consiste en un circuito digital que permite realizar operaciones aritméticas y lógicas entre dos números. La mayoría de las acciones de la computadora son realizadas por la ALU. La ALU toma datos

de los registros del procesador los cuales son procesados y los resultados de esta operación se almacenan en los registros de salida de la ALU. El componente más básico es el que realiza la suma, ya que con este es posible realizar las demás operaciones como: la resta, multiplicación y división. En el caso de la resta se puede usar el mismo circuito que para la suma siempre que tengamos el número para ser restado en su complemento de 2. La multiplicación consiste en una serie de operaciones AND entre los distintos bits y una serie de sumas.

RECOMENDACIONES

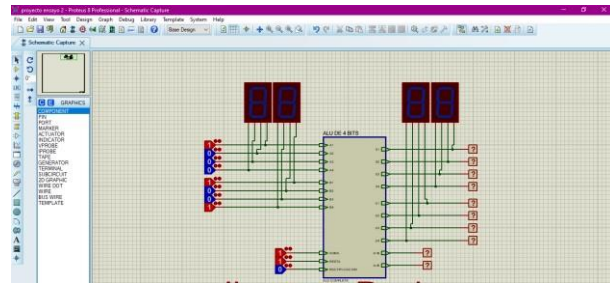
Finalmente, quiero agradecer al Ing. Efraín Becerra quien nos dio una explicación de cómo utilizar el programa Proteus y nos explicó el concepto general de como implementar las diversas operaciones dentro de un ALU. Por otro lado, queremos agradecer al Ing. Carlos Fernández quien nos explicó cada uno de los conceptos y aclaró las dudas que teníamos respecto a este proyecto.

REFERENCIAS

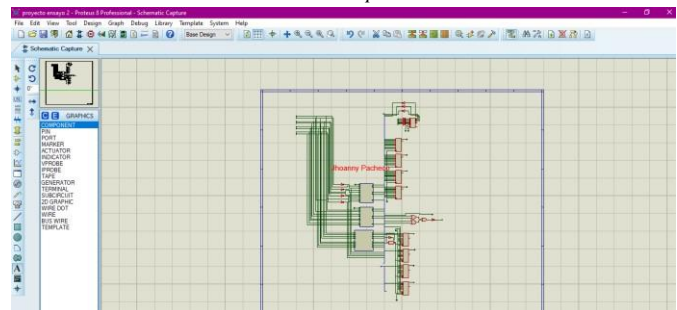
Referencias de documentos no publicados

- [1] Stallings, William (2006). *Computer Organization & Architecture: Designing for Performance* 7th ed (<http://williamstallings.com/COA/COA7e.html>). Pearson Prentice Hall. ISBN 0-13-185644-8.
- [2] V. Bembibre, «Definición ABC,» enero 2009. [En línea]. Available: <https://www.definicionabc.com/tecnologia/alu.php>. [Último acceso: 11 Agosto 2021].
- [3] R. J. Tocci, «clrueda.docentes.upbbga,» [En línea]. Available: http://clrueda.docentes.upbbga.edu.co/web_digitales/Tema_3/alu.html. [Último acceso: 11 Agosto 2021].
- [3] *IEEE Guide for Application of Power Apparatus Bushings*, IEEE Standard C57.19.100-1995, Aug. 1995.
- [4] A. Ramírez Reyes y G. y. S. S. E. G. Blanco Gómez, 2010. [En línea]. Available: <http://repositorio.unan.edu.ni/id/eprint/6162>.
- [5] J. Pardo, C. Barbosa y E. Carrillo, <<Calameo, >>2020.[En línea]. Available: <https://es.calameo.com/read/006155253ffe35705654f>

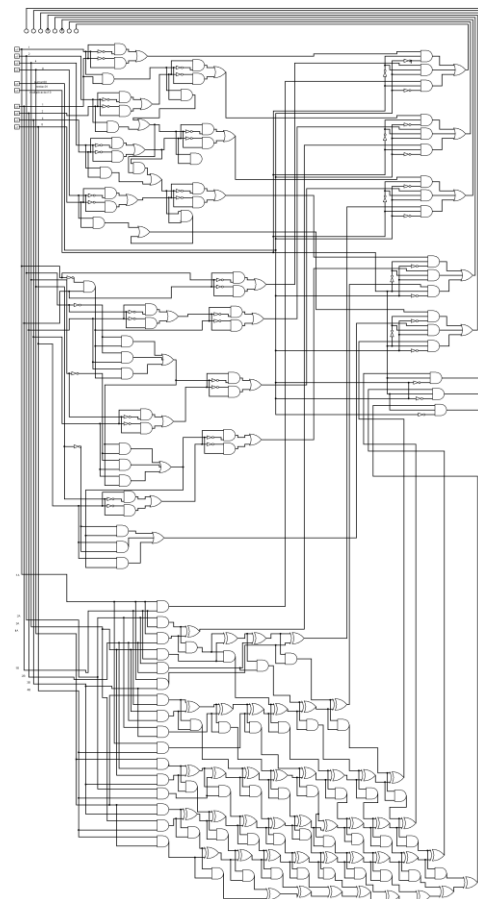
ANEXOS



ALU completa



Circuito interno del ALU



Circuito Final diseñado con Logisim (suma, resta y multiplicación)