Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización Computacional

PRÁCTICA #2
LogicCalc

Grupo 9

Nombre	Carnet	% Participación
Tulio Jafeth Pirir Schuman	201700698	20
Carlos Sebastian del Cid Ramirez	202300625	20
José Emanuel Monzón Lémus	202300539	20
Ottoniel Fabricio Vásquez Pineda	202307820	20
Diego Alejandro Vasquez Alonzo	202300638	20

Guatemala 08/03/2025

Introducción

En el ámbito de la electrónica digital, la capacidad de realizar operaciones aritméticas de manera eficiente y confiable es fundamental. Aquí es donde los bloques MSI (Integración a Media Escala) de tipo aritmético entran en juego, desempeñando un papel crucial en la construcción de circuitos complejos. Estos componentes, que representan un hito en la evolución de la electrónica digital, permiten a los ingenieros diseñar sistemas capaces de realizar cálculos matemáticos fundamentales, como sumas, restas y multiplicaciones, con una precisión y velocidad sorprendentes.

Los bloques MSI aritméticos, construidos a partir de compuertas lógicas y registros, son la columna vertebral de numerosos dispositivos electrónicos, desde computadoras y procesadores de señales digitales hasta sistemas de control y equipos de comunicación. Su capacidad para procesar operaciones y generar resultados precisos es esencial para el funcionamiento de una amplia gama de aplicaciones tecnológicas.

En esta exploración, profundizaremos en el funcionamiento interno de estos bloques, examinando cómo las compuertas lógicas y los registros se combinan para realizar operaciones aritméticas. También analizaremos los diferentes tipos de bloques MSI aritméticos, sus aplicaciones y su importancia en el contexto de la electrónica digital moderna.

Objetivos

General

1. Construir una Unidad Aritmética Lógica Básica (ALU).

Objetivos Específicos

1. Aprender el funcionamiento de Multiplexores, Demultiplexores,Comparadores y Decodificadores.

2. Construir un diseño óptimo, logrando utilizar la menor cantidad de dispositivos.

3. Aprender el funcionamiento de Operaciones Lógicas, Aritméticas y Comparativas con números binarios.

Descripción del Problema

Como estudiantes del curso Organización Computacional, han sido contratados por Intel Corporation para desarrollar un prototipo de calculadora llamado "LogicCalc". Intel busca una solución óptima basada en lógica combinacional que sea capaz de realizar cálculos aritméticos y lógicos. Para cumplir con estos requisitos, Intel ha proporcionado las especificaciones únicas para una Unidad Aritmética Lógica Básica (ALU).

Se debe comprender cómo se construyen y funcionan los circuitos complejos. Especialmente el papel de los bloques MSI (Integración a Media Escala) en este proceso, ya que son componentes fundamentales para realizar operaciones aritméticas básicas.

Definir cómo estos bloques MSI aritméticos, que utilizan compuertas lógicas y registros, son capaces de llevar a cabo operaciones como sumas, restas y multiplicaciones de manera eficiente y confiable.

Proceso Teórico

Comparación de dos números binarios

$$B = b_0 b_1 b_2 b_3$$

$$A = a_0 a_1 a_2 a_3$$

A > B

A>B = (a3>b3)+(a3=b3)(a2>b2)+(a3=b3)(a2=b2)(a1>b1)+(a3=b3)(a2=b2)(a1=b1)(a0>b0)

An	Bn	An>Bn
0	0	0
0	1	0
1	0	1
1	1	0

An	Bn	An=Bn
0	0	1
0	1	0
1	0	0
1	1	1

$A>B=(a3 \wedge \neg b3) \vee [(a3 \oplus b3) \wedge (a2 \wedge \neg b2)] \vee [(a3 \oplus b3) \wedge (a2 \oplus b2) \wedge (a1 \wedge \neg b1)] \vee [(a3 \oplus b3) \wedge (a2 \oplus b2) \wedge (a1 \oplus b1) \wedge (a0 \wedge \neg b0)]$

A < B

$A>B = (a3<b3)+(a3=b3)(a2<b2)+(a3=b3)(a2=b2)(a1<b1)+(a3=b3)(a2=b2)(a1=b1)(a0<b0)$

An	Bn	An<Bn
0	0	0
0	1	1
1	0	0
1	1	0

$A<B=(b3 \wedge \neg a3) \vee [(a3 \oplus b3) \wedge (b2 \wedge \neg a2)] \vee [(a3 \oplus b3) \wedge (a2 \oplus b2) \wedge (b1 \wedge \neg a1)] \vee [(a3 \oplus b3) \wedge (a2 \oplus b2) \wedge (a1 \oplus b1) \wedge (b0 \wedge \neg a0)]$

Multiplicación de Dos Numero Binarios

			A3	A2	A1	A0
			B3	B2	B1	B0
			B0A3	B0A2	B0A1	B0A0
		B1A3	B1A2	B1A1	B1A0	
	B2A3	B2A2	B2A1	B2A0		
B3A3	B3A2	B3A1	B3A0			
Z6	Z5	Z4	Z3	Z2	Z1	Z0

- De acuerdo con con la lógica de multiplicación aritmética se llego al resultado de la función
    - Asumiendo un valor arbitrario 1111 y realizando una multiplicación aritmética

			1	1	1	1
			1	1	1	1
			1	1	1	1
		1	1	1	1	
	1	1	1	1		
1	1	1	1			

- Tenemos como resultado (asima 1 color rojo como acarreo)

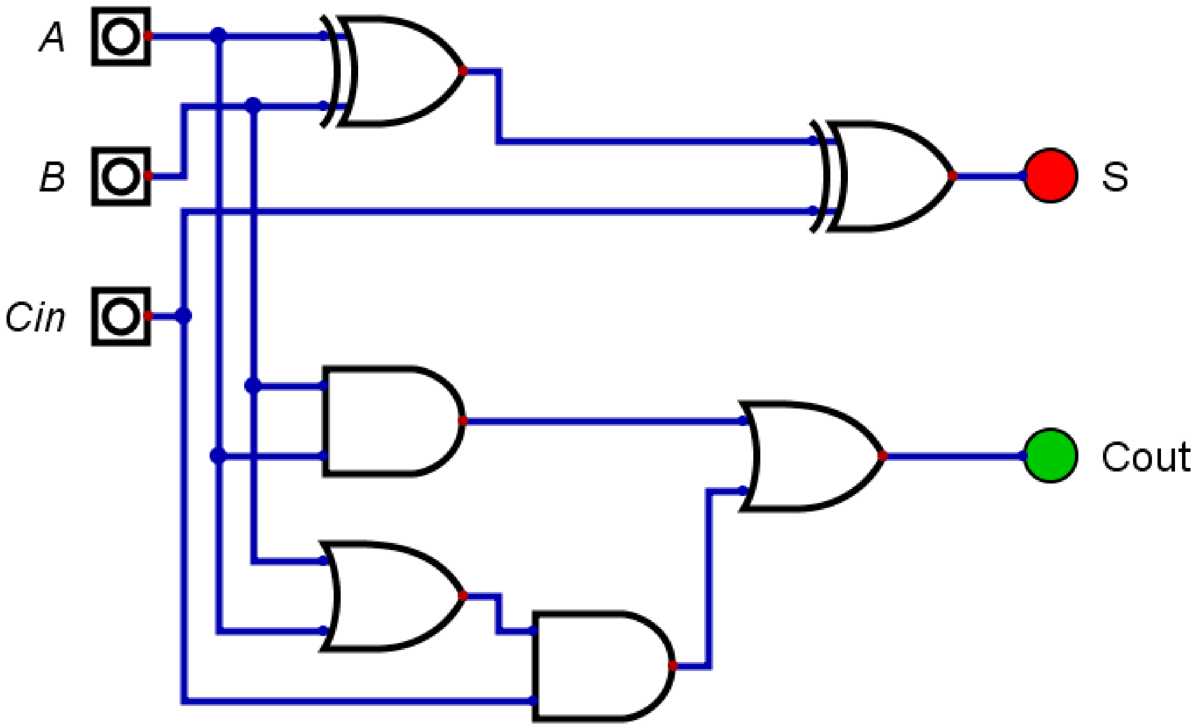
		1	1				
	1	1	1	1			
	1	1	1	1	1		
	0	0	0	1	1	1	1
	0	0	1	1	1	1	0
	0	1	1	1	1	0	0
1	1	1	1	1	0	0	0
1	1	1	0	0	0	0	1

Suma Dos Números Binarios

Tendremos tres variables de entrada, a las que llamaremos A, B y la propia C_{in} y dos de salida: S, la suma y C_{out}, el acarreo de la misma (si procede). Entonces, la tabla de verdad quedará:

A	B	Cin	S	Cout
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

Una vez simplificadas las funciones e implementadas, obtenemos el siguiente circuito:



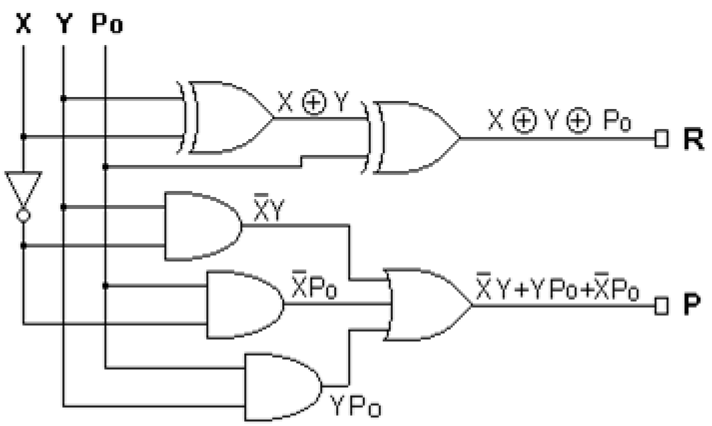
Resta Dos Números Binarios

Un restador completo es un circuito combinacional que lleva a cabo una sustracción entre dos bits, tomando en cuenta que en un 1 se ha tomado por una etapa significativa más baja. Este circuito tiene tres entradas y dos salidas. Las tres entradas x, y y z, denotan al minuendo, sustraendo y a la toma previa, respectivamente. Las dos salidas, D y B, representan la diferencia y la salida tomada, respectivamente. La tabla de verdad para el circuito es como sigue:

X	Y	Z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

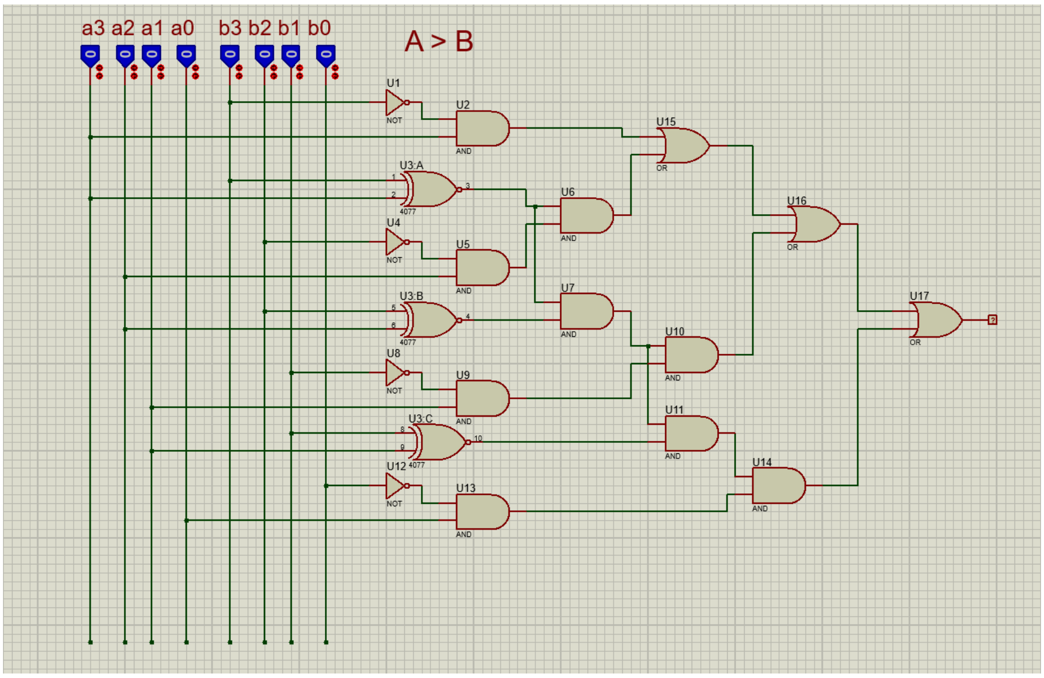
Los ocho renglones bajo las variables de entrada designan todas las combinaciones posibles de 1 y 0 que pueden tomar las variables binarias. Los 1 y 0 para las variables de salida están determinados por la sustracción de x – y – z. Las combinaciones que tienen salida de toma z = 0 se reducen a las mismas cuatro condiciones del medio sumador. Para x = 0, y = 0 y z = 1, tiene que tomarse un 1 de la siguiente etapa, lo cual hace B = 1 y añade 2 a x. Ya que 2 – 0 – 1, D = 1. Para x = 0 y yz = 11, necesita tomarse otra vez, haciendo B = 1 y x = 2. Ya que 2 – 1 – 1 = 0, D = 0. Para x = 1 y yz = 01, se tiene x – y – z = 0, lo cual hace B = 0 y D = 0. Por último, para x = 1, y = 1, z = 1, tiene que tomarse 1, haciendo B = 1 y x = 3 y, 3 –1 – 1 = 1, haciendo D = 1

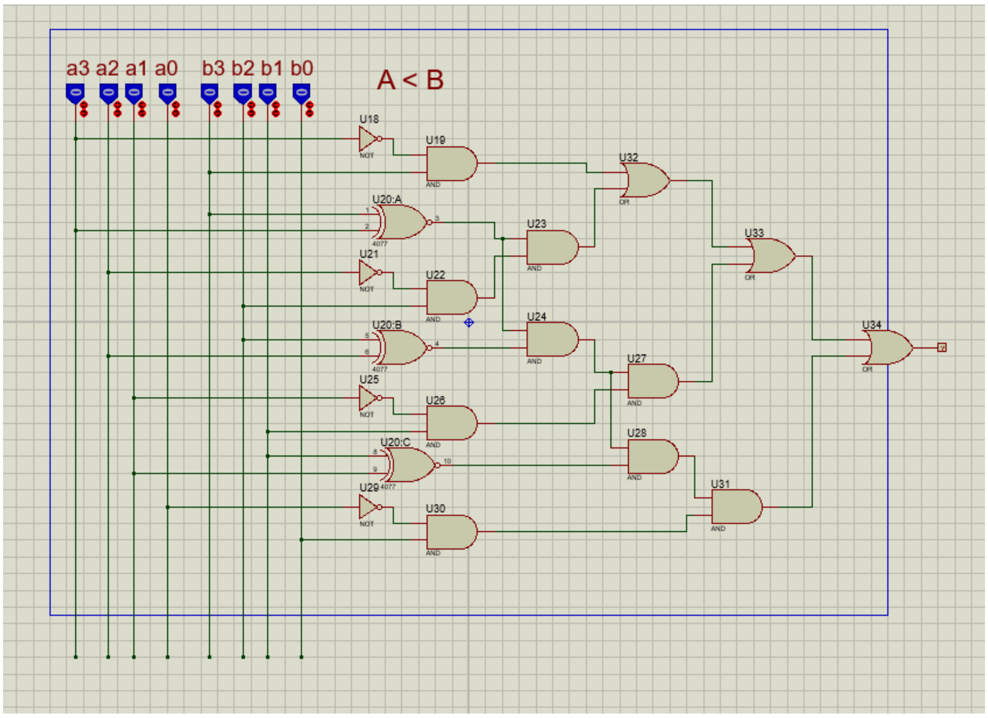
El circuito lógico implementado con compuertas es el siguiente:



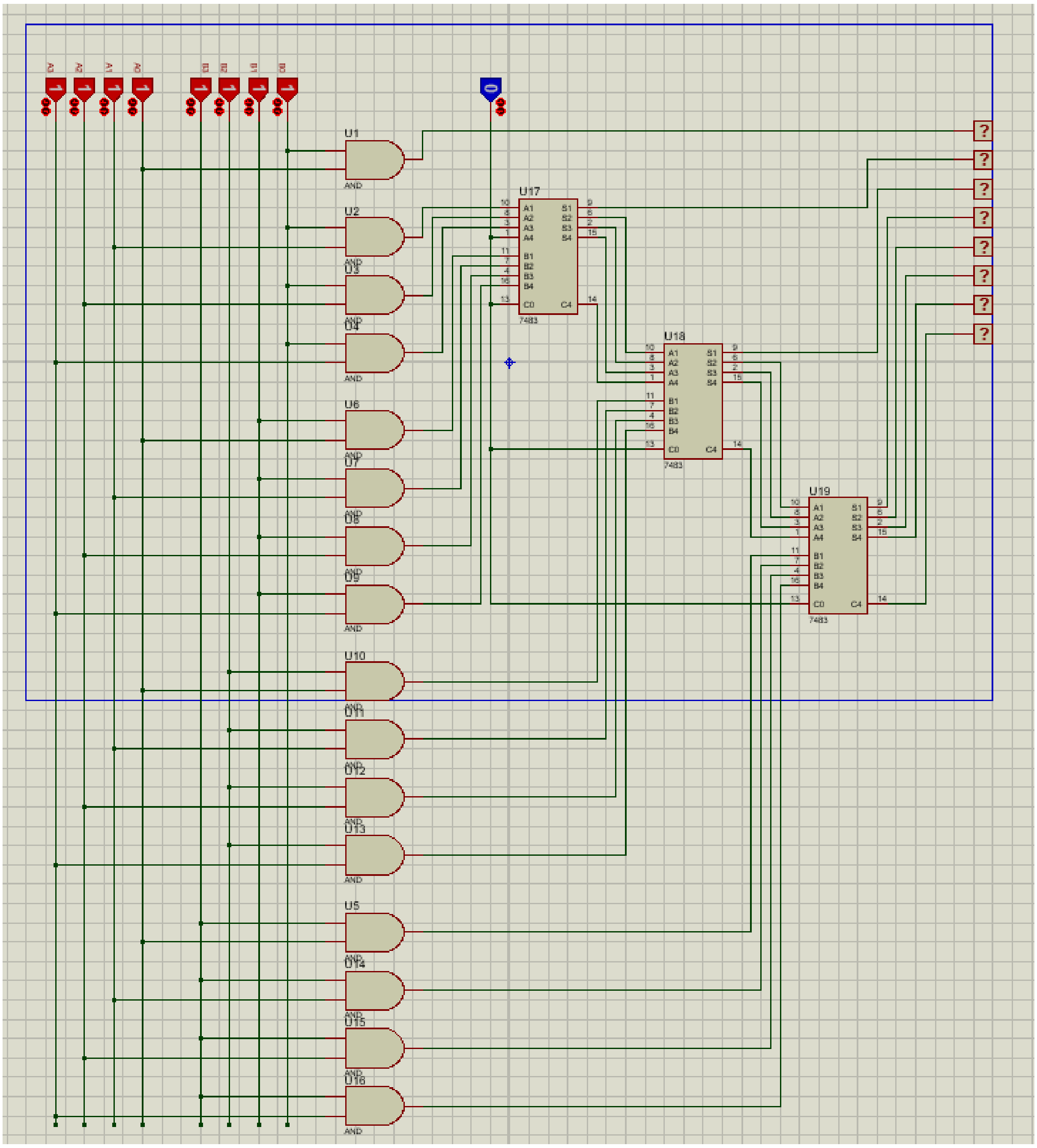
Diagramas de los diseños desarrollados

Comparadores

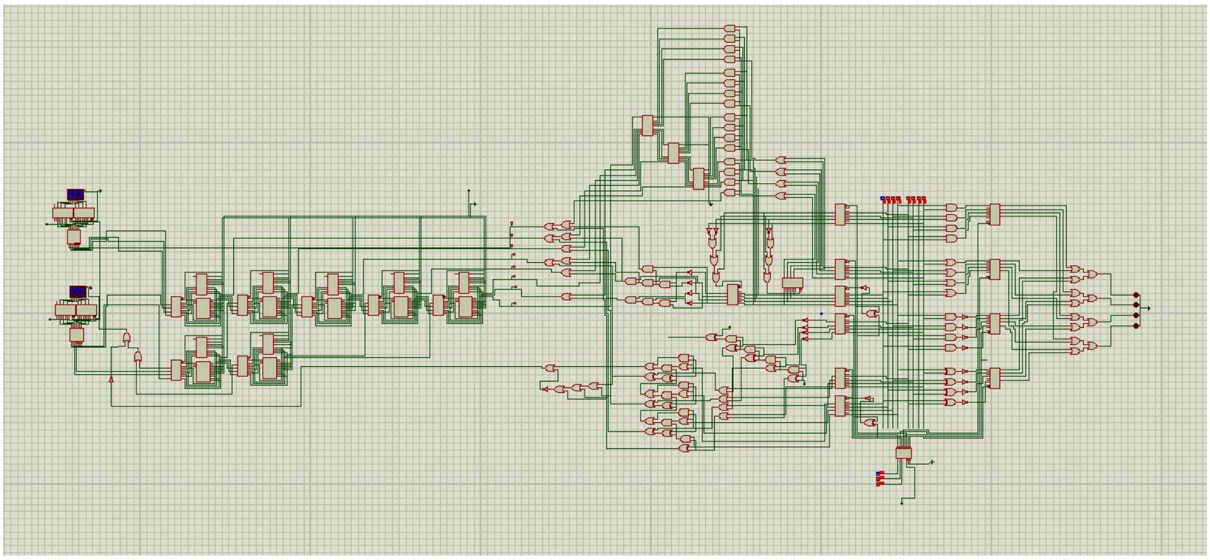




Multiplicación



Diseño Final Del Circuito



Equipo Utilizado.

Componentes	**Cantidad**
Alambre	21
Lija	1
Papel Termotransferible	1
Impresión Termotransferible	1
Placa de Cobre	1
Protoboard	5
Sumador 74LS83	10
Comparador 74LS85	7
Compuerta Lógica XOR 74LS86	8
Compuerta Lógica NOT 74LS04	13
Compuerta Lógica AND 74LS08	16
Multiplexor 74LS157	27
DIP Switch	2
Display 7 Segmentos	2
Capacitor	4
Compuerta Lógica OR 74LS32	4
Decoder Demux 74LS138	2
Decodificador 74LS18	2
Protoboard	4
LED	5

Presupuesto.

Componentes	Cantidad	Precio(Q)	Subtotal(Q)
Alambre	21	3	63
Lija	1	7	7
Papel Termotransferible	1	5	5
Impresión Termotransferible	1	3	3
Placa de Cobre	1	32	32
Protoboard	5	34	170
Sumador 74LS83	10	15	150
Comparador 74LS85	7	11	77
Compuerta Lógica XOR 74LS86	8	6	48
Compuerta Lógica NOT 74LS04	13	5	65
Compuerta Lógica AND 74LS08	16	5	80
Multiplexor 74LS157	27	7	189
DIP Switch	2	3.75	7.5
Display 7 Segmentos	2	5	10
Capacitor	4	0.75	3
Compuerta Lógica OR 74LS32	4	5	20
Decoder Demux 74LS138	2	6	12
Decodificador 74LS18	2	11	22
Protoboard	4	38	152
LED	5	1	5

Total **1120.5 Q**

Aportacion de Cada Integrante

Nombre	Aportación
Tulio Jafeth Pirir Schuman	224
Carlos Sebastian del Cid Ramirez	224
José Emanuel Monzón Lémus	224
Ottoniel Fabricio Vásquez Pineda	224
Diego Alejandro Vasquez Alonzo	224

Conclusiones.

- La construcción de una Unidad Aritmética Lógica (ALU) básica depende en gran medida del funcionamiento y la correcta implementación de los bloques MSI. Estos componentes permiten realizar las operaciones lógicas y aritméticas necesarias para el funcionamiento de la ALU.

- El proyecto "LogicCalc" destaca la importancia de diseñar circuitos eficientes utilizando la menor cantidad posible de bloques MSI. Esto implica una comprensión profunda de las funciones de cada bloque y su aplicación óptima en el diseño del circuito.

- El entendimiento de cómo se realizan las operaciones lógicas, aritméticas y comparativas con números binarios es crucial para trabajar con bloques MSI. Estos bloques están diseñados para operar con señales digitales, que se representan en forma binaria, por lo que su dominio es fundamental para el diseño y la implementación de circuitos digitales.