

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Organización Computacional



PRÁCTICA #2

LogicCalc

GRUPO 5		
No.	NOMBRE	CARNÉ
1	Samuel Alejandro Pajoc Raymundo. (Coordinador)	201800665
2	Kevin Manuel Veliz Galvez	201901441
3	Oscar David Padilla Vásquez	202103250
4	Aarón Emanuel Trujillo Ibarra	201801608
5	Byron Enrique Rumpich Sal	201907769
6	Aarón Abdam Saravia Martinez	202105212

INTRODUCCIÓN

En el campo de la electrónica digital, la construcción de circuitos complejos requiere la utilización eficiente y confiable de bloques de medio escala de integración (MSI, por sus siglas en inglés). Dentro de esta categoría, los bloques MSI tipo aritmético desempeñan un papel crucial al permitir la realización de operaciones aritméticas básicas, como sumas, restas, multiplicaciones y potencias, utilizando compuertas lógicas, registros y otros componentes digitales.

Este documento se centra en la descripción detallada de una práctica específica en la que se implementan tres unidades distintas en un circuito digital: la Unidad Aritmética, la Unidad Lógica y la Unidad Comparativa. Cada una de estas unidades tiene un propósito definido y opera de manera independiente en el circuito.

La Unidad Aritmética está diseñada para realizar operaciones aritméticas, como sumas, restas, multiplicaciones y potencias, utilizando únicamente compuertas lógicas para garantizar la eficiencia y confiabilidad del proceso. Se especifican claramente los procedimientos para cada operación y se establecen condiciones de validación para asegurar resultados precisos.

La Unidad Lógica, por otro lado, se encarga de realizar operaciones lógicas, como AND, OR, XOR y NOT, entre dos números binarios de entrada. Los resultados de estas operaciones se presentan en 4 LEDs que representan los 4 bits resultantes.

Finalmente, la Unidad Comparativa determina y muestra en dos displays de 7 segmentos el número mayor y menor de la operación dada. Se incluye la indicación visual del estado de la unidad que está ejecutando la operación mediante un LED de color diferente para cada unidad.

El documento proporciona una descripción detallada de cada unidad, incluyendo sus funciones, operaciones admitidas, métodos de implementación y criterios de validación. Además, se enfatiza la importancia del orden y la claridad en el diseño del circuito para garantizar una ejecución correcta y una evaluación adecuada de la práctica.

OBJETIVOS

Objetivo General

1. Construir una Unidad Aritmética Lógica Básica (ALU). Objetivos Específicos
2. Aprender el funcionamiento de Multiplexores, Demultiplexores, Comparadores y Decodificadores.
3. Construir un diseño óptimo, logrando utilizar la menor cantidad de dispositivos.
4. Aprender el funcionamiento de Operaciones Lógicas, Aritméticas y Comparativas con números binarios.

Descripción del Problema

El problema consiste en el diseño e implementación de un circuito digital que incluya tres unidades funcionales: Aritmética, Lógica y Comparativa. Cada unidad tiene su propio conjunto de operaciones y requisitos específicos de funcionamiento.

La Unidad Aritmética debe ser capaz de realizar operaciones aritméticas básicas como suma, resta, multiplicación y potenciación de números binarios. Se deben implementar estas operaciones utilizando únicamente compuertas lógicas, sin utilizar semi-sumadores o sumadores completos.

La Unidad Lógica debe realizar operaciones lógicas como AND, OR, XOR y NOT entre dos números binarios de entrada. Los resultados de estas operaciones se muestran en 4 LEDs representando los 4 bits resultantes.

La Unidad Comparativa determina y muestra en dos displays de 7 segmentos el número mayor y menor de la operación dada. Si las entradas son iguales, se muestra el mismo número en ambos displays.

Se debe garantizar que cada unidad funcione de manera independiente y que los resultados se muestren correctamente según la operación seleccionada. Además, se debe indicar visualmente qué unidad está ejecutando la operación en un momento dado.

CONTENIDO

a) Tablas de verdad, mapas de Karnaugh y sus Funciones.

- Suma

Tabla de verdad				
Cin (Z)	A	B	Cout	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Mapas Karnaught				
F_Cout	AB	00	01	11
Z		10		
0				1
1		1	1	1

Mapas Karnaught				
F_S	AB	00	01	11
Z		10		
0		1		
1	1		1	

$F_{Cout} = ZB + AB + ZA$
 $F_{Cout} = Z(A+B) + AB$

$F_S = ZA'B' + Z'A'B + ZAB + Z'AB'$
 $F_S = (ZA'B' + Z'A'B) + (ZAB + Z'AB')$
 $F_S = A'(ZB' + Z'B) + A(ZB + Z'B')$
 $F_S = A'(Z \text{ xor } B) + A(Z \text{ xnor } B)$
 $F_S = A'(Y) + A(Y')$
 $Z \text{ xor } B = Y$
 $Z \text{ xnor } B = Y'$
 $F_S = A \text{ xor } Y$
 $F_S = A \text{ xor } (Z \text{ xor } B)$

- Resta

Tabla de verdad				
Z (Cin)	A	B	Cout	S
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

Mapas Karnaught		ZU		
F_Cout	AB	A1	B0	
Z (Cin)	00 01 11 10			
0		1		$F_{Cout} = A'B + ZA' + ZB$
1	1 1 1 1			$F_{Cout} = A'(B + Z) + ZB$

F_S		AB	ZU	
Z (Cin)	00 01 11 10			
0		1		$F_S = Z'(A'.B + A.B') + Z(A'.B' + A.B)$
1	1	1	1	$F_S = Z'(A \text{ xor } B) + Z(A \text{ xnor } B)$
				$F_S = Z'(A \text{ xor } B) + Z(A \text{ xor } B)'$
				$F_S = Z'(Y) + Z(Y)'$
				$F_S = (Z \text{ xor } Y)$
				$F_S = (Z \text{ xor } (A \text{ xor } B))$

- Multiplicación

Tabla de verdad		
A	B	R
0	0	0
0	1	0
1	0	0
1	1	1

Mapas Karnaught		
F_R	B	ZU
A	0 1	
0		
1		$F_R = A.B$

- Exponencial 2 (Cuadrado)

ENTRADA "B" EXPONENETE = 2					PESOS PONDERADOS							
Tabla de verdad					S	64	32	16	8	4	2	1
No.	A3	A2	A1	A0	S	S6	S5	S4	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0	0	0	0	1
2	0	0	1	0	4	0	0	0	0	1	0	0
3	0	0	1	1	9	0	0	0	1	0	0	1
4	0	1	0	0	16	0	0	1	0	0	0	0
5	0	1	0	1	25	0	0	1	1	0	0	1
6	0	1	1	0	36	0	1	0	0	1	0	0
7	0	1	1	1	49	0	1	1	0	0	0	1
8	1	0	0	0	64	1	0	0	0	0	0	0
9	1	0	0	1	81	1	0	1	0	0	0	1
10	1	0	1	0	E	X	X	X	X	X	X	X
11	1	0	1	1	E	X	X	X	X	X	X	X
12	1	1	0	0	E	X	X	X	X	X	X	X
13	1	1	0	1	E	X	X	X	X	X	X	X
14	1	1	1	0	E	X	X	X	X	X	X	X
15	1	1	1	1	E	X	X	X	X	X	X	X

E = Error

Mapas Karnaugh							
F_S6	A1 A0	00	01	11	10	F_S6 = A3	
A3 A2	00						
00							
01							
11		X	X	X	X		
10		1	1	X	X		

F_S4	A1 A0	00	01	11	10	F_S4 = A2.A1' + A2.A0 + A3.A0
A3 A2	00					F_S4 = A2(A1' + A0) + A3.A0
00						
00		1	1	1		
01		1	1	1		
11		X	X	X		
10		1	1	X		

F_S2	A1 A0	00	01	11	10	F_S2 = A1.A0'
A3 A2	00					F_S2 = A1.A0'
00						
00					1	
01					1	
11					X	
10					X	

F_S0		A1 A0				
A3 A2		00	01	11	10	F_S0 = A0
00			1	1		
01			1	1		
11			X	X		
10			1	X		

F_S5		A1 A0				
A3 A2		00	01	11	10	F_S5 = A2.A1
00						
01				1	1	
11				X	X	
10						

F_S3		A1 A0				
A3 A2		00	01	11	10	F_S3 = A2'.A1.A0 + A2.A1'.A0
00				1		F_S3 = A0(A2'.A1 + A2.A1')
01			1			F_S3 = A0(A2 xor A1)
11			X			
10				X		

F_S1		A1 A0				
A3 A2		00	01	11	10	F_S1 = 0 (Cable directo TIERRA)
00		0	0	0	0	
01		0	0	0	0	
11		0	0	0	0	
10		0	0	0	0	

- Exponencial 3 (Cubo)

ENTRADA "B" EXPONENETE = 3				PESOS PONDERADOS							
Tabla de verdad				64	32	16	8	4	2	1	
No.	A2	A1	A0	S	S6	S5	S4	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	1
2	0	1	0	8	0	0	0	1	0	0	0
3	0	1	1	27	0	0	1	1	0	1	1
4	1	0	0	64	1	0	0	0	0	0	0
5	1	0	1	E	X	X	X	X	X	X	X
6	1	1	0	E	X	X	X	X	X	X	X
7	1	1	1	E	X	X	X	X	X	X	X

F_S6	A1 A0	00	01	11	10	F_S6 = A2
A2	00	01	11	10		
0				1		
1	1	X	X	X		

F_S4	A1 A0	00	01	11	10	F_S4 = A1.A0
A2	00	01	11	10		
0				1		
1				X		

F_S2	A1 A0	00	01	11	10	F_S2 = 0 (cable directo TIERRA)
A2	00	01	11	10		
0	0	0	0	0		
1	0	0	0	0		

F_S0	A1 A0	00	01	11	10	F_S0 = A0
A2	00	01	11	10		
0		1	1			
1		X	X			

F_S5	A1 A0	00	01	11	10	F_S5 = 0 (cable directo TIERRA)
A2	00	01	11	10		
0	0	0	0	0		
1	0	0	0	0		

F_S3	A1 A0	00	01	11	10	F_S3 = A1
A2	00	01	11	10		
0			1	1		
1			X	X		

F_S1	A1 A0	00	01	11	10	F_S1 = F_S4
A2	00	01	11	10		
0			1			
1			X			

- Decoder

S4 = C4 (Cin4) = Cout3 = Z3					Display 2				Display 1			
S4	S3	S2	S1	S0	A	B	C	D	a	b	c	d
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1					0			1
0	0	0	1	0					0		1	
0	0	0	1	1					0		1	1
0	0	1	0	0					0	1		
0	0	1	0	1					0	1		1
0	0	1	1	0					0	1	1	
0	0	1	1	1					0	1	1	1
0	1	0	0	0					0	1		
0	1	0	0	1	0	0	0	0	0	1		1
0	1	0	1	0	0	0	0	0	1	0	0	0
0	1	0	1	1					1			1
0	1	1	0	0					1			
0	1	1	0	1					1		1	
0	1	1	1	0					1		1	
0	1	1	1	1					1		1	1
1	0	0	0	0					1		1	
1	0	0	0	1					1		1	1
1	0	0	1	0					1	1		
1	0	0	1	1					1	1		1
1	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1					1			1
1	0	1	1	0					1		1	
1	0	1	1	1					1		1	1
1	1	0	0	0					1		1	
1	1	0	0	1					1		1	1
1	1	0	1	0					1		1	
1	1	0	1	1					1		1	1
1	1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1					1		1	1
1	1	1	1	0					0	0	0	0
1	1	1	1	1	X	X	X	X	X	X	X	X

Display 2 (Decenas)

FA	S2 S1 S0	000	001	011	010	110	111	101	100	FA = 0 (cable TIERRA)
S4 S3		000	001	011	010	110	111	101	100	
00										
01										
11										
10										

FB	S2 S1 S0	000	001	011	010	110	111	101	100	FB = 0 (cable TIERRA)
S4 S3		000	001	011	010	110	111	101	100	
00										
01										
11										
10										

FC	S2 S1 S0	000	001	011	010	110	111	101	100	FC = S4.S3 + S4.S2
S4 S3		000	001	011	010	110	111	101	100	FC = S4(S3 + S2)
00										
01										
11	1	1	1	1		1	X	1	1	
10						1	1	1	1	

FD	S2 S1 S0	000	001	011	010	110	111	101	100	FD = S4'.S3.S2 + S4'.S3.S1 + S4.S3'.S2' + S3.S2.S1
S4 S3		000	001	011	010	110	111	101	100	FD = (S4'.S3.S2 + S4'.S3.S1 + S3.S2.S1) + S4.S3'.S2'
00										
01						1	1	1	1	FD = S3(S4'.S2 + S4'.S1 + S2.S1) + S4.S3'.S2'
11						1	X			FD = S3(S4'(S2 + S1) + S2.S1) + S4.S3'.S2'
10	1	1	1	1						FD = S3(S4'(S2 + S1) + S2.S1) + S4(S3 + S2)'

Display 1 (Unidades)								
Fa	S2	S1	S0	000	001	011	010	110
S4	S3							
00								
01		1	1					
11								
10				1	1			
						1	1	

$F_a = S_4.S_3.S_2.S_1' + S_4'.S_3.S_2'.S_1' + S_4.S_3'.S_2'.S_1$
 $F_a = (S_4.S_3.S_2.S_1' + S_4'.S_3.S_2'.S_1') + S_4.S_3'.S_2'.S_1$
 $F_a = S_3.S_1'(S_4.S_2 + S_4'.S_2') + S_4.S_3'.S_2'.S_1$
 $F_a = S_3.S_1'(S_4 \text{ xnor } S_2) + S_4(S_3'.S_2').S_1$
 $F_a = S_3.S_1'(S_4 \text{ xor } S_2)' + S_4(S_3 + S_2).S_1$

Display 1 (Unidades)								
Fb	S2	S1	S0	000	001	011	010	110
S4	S3							
00								
01								
11		1	1					
10		1	1					
		1	1					
		1	1					

$F_b = S_4'.S_3'.S_2 + S_4'.S_2.S_1 + S_4.S_3.S_2' + S_4.S_2'.S_1'$
 $F_b = S_4'.S_2(S_3' + S_1) + S_4.S_2'(S_3 + S_1)$

Display 1 (Unidades)								
Fc	S2	S1	S0	000	001	011	010	110
S4	S3							
00								
01								
11			1	1				
10			1	1				
		1	1					
		1	1					

$F_c = S_4'.S_3'.S_1 + S_4'.S_3.S_2.S_1' + S_4.S_3.S_2'.S_1 + S_4.S_3'.S_2'.S_1' + S_3'.S_2.S_1$
 $F_c = (S_4'.S_3'.S_1 + S_3'.S_2.S_1) + S_4'.S_3.S_2.S_1' + (S_4.S_3.S_2'.S_1 + S_4.S_3'.S_2'.S_1')$
 $F_c = S_3'.S_1(S_4' + S_2) + S_4'.S_3.S_2.S_1' + S_4.S_2'(S_3.S_1 + S_3'.S_1')$
 $F_c = S_3'.S_1(S_4' + S_2) + S_4'.S_3.S_2.S_1' + S_4.S_2'(S_3 \text{ xnor } S_1)$
 $F_c = S_3'.S_1(S_4' + S_2) + S_4'.S_3.S_2.S_1' + S_4.S_2'(S_3 \text{ xor } S_1)$
 $F_c = S_3'.S_1(S_4' + S_2) + S_4'.S_1.S_3.S_2 + S_4.S_2'(S_3 \text{ xor } S_1)$
 $F_c = S_3'.S_1(S_4' + S_2) + (S_4'.S_1').S_3.S_2 + S_4.S_2'(S_3 \text{ xor } S_1)$
 $F_c = S_3'.S_1(S_4' + S_2) + (S_4 + S_1).S_3.S_2 + S_4.S_2'(S_3 \text{ xor } S_1)'$

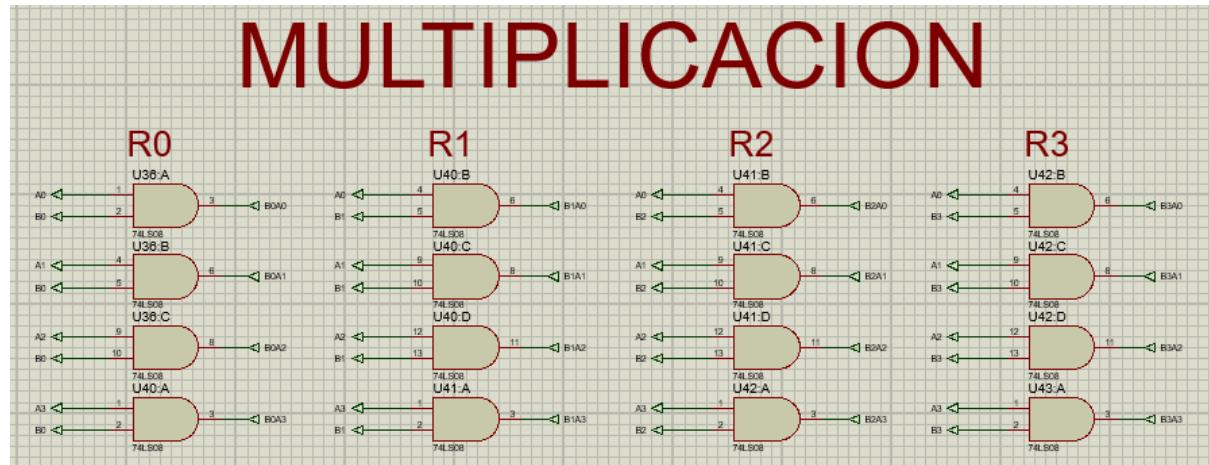
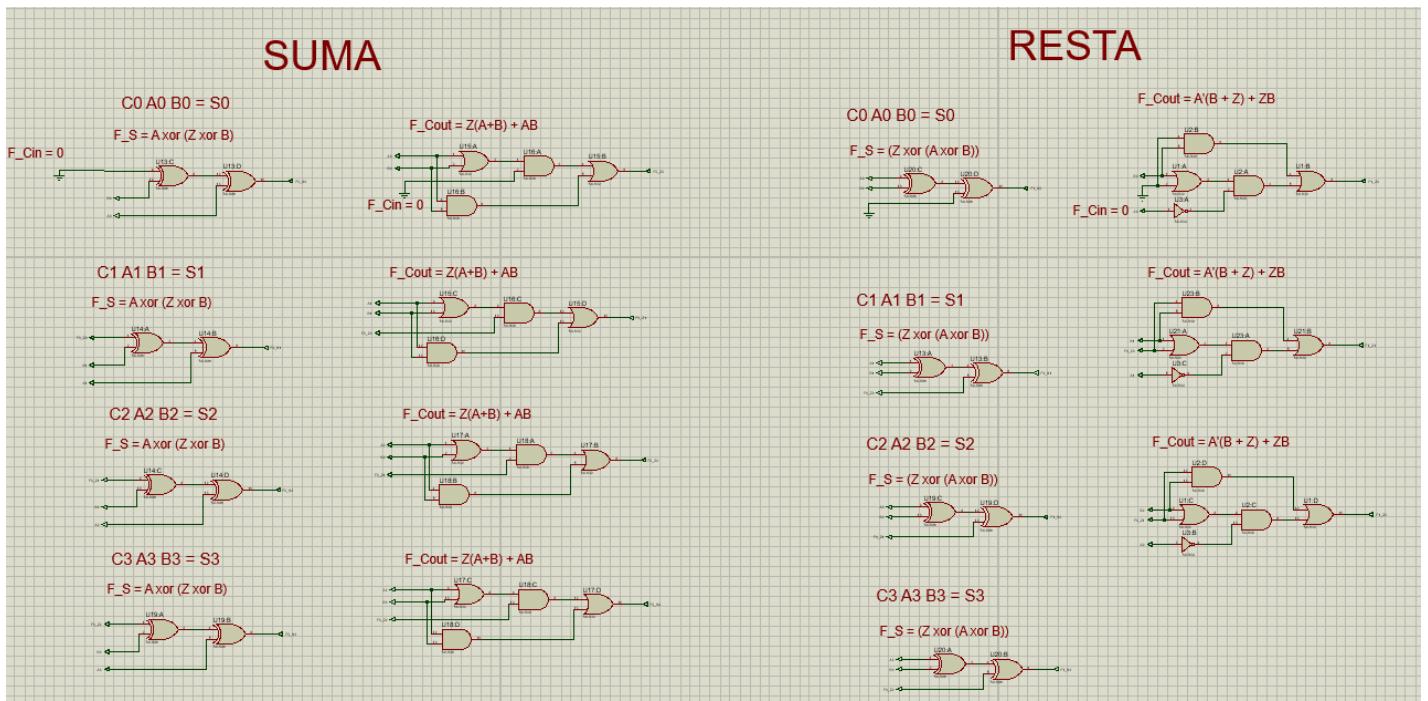
Display 1 (Unidades)								
Fd	S2	S1	S0	000	001	011	010	110
S4	S3							
00								
01		1	1					
11		1	1					
10		1	1					
		1	1					
		1	1					

$F_d = S_2'.S_0 + S_2.S_0$
 $F_d = S_0(S_2' + S_2)$
 $F_d = S_0(1)$
 $F_d = S_0$

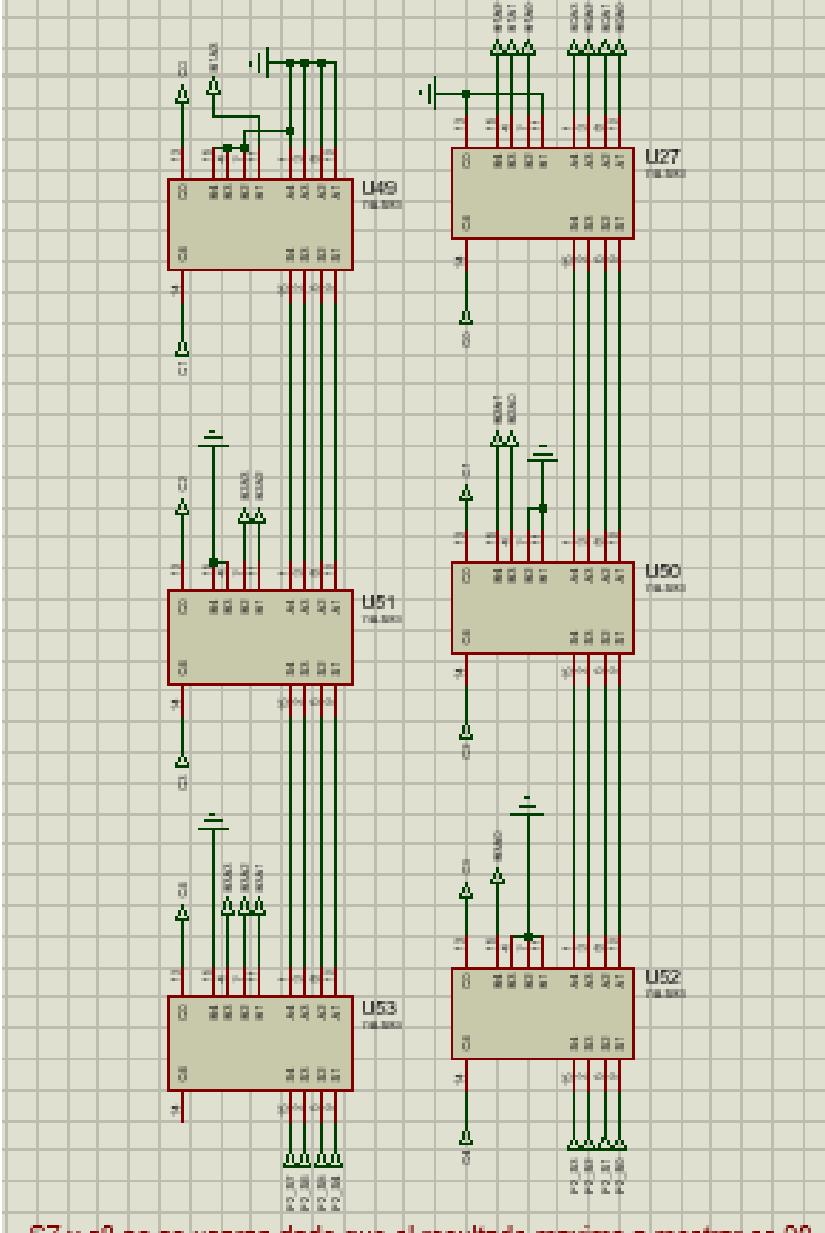
- Tabla Selector.

C	B	A	Operación
0	0	0	Suma
0	0	1	Resta
0	1	0	Multiplicación
0	1	1	Número Elevado
1	0	0	XOR
1	0	1	OR
1	1	0	AND
1	1	1	NOT

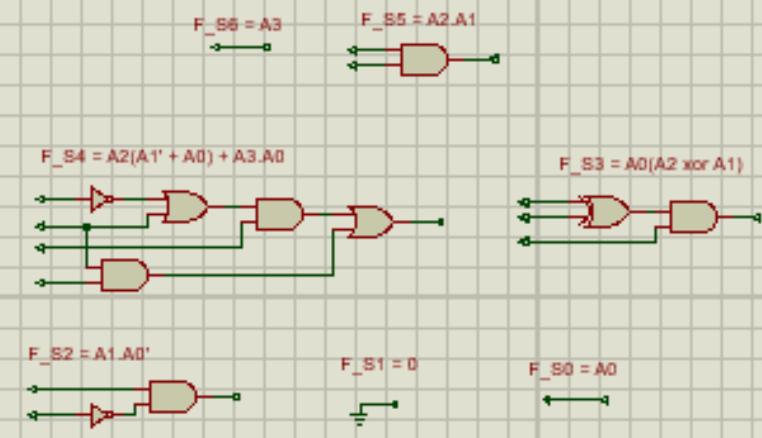
b) Diagramas del diseño del circuito en proteus.



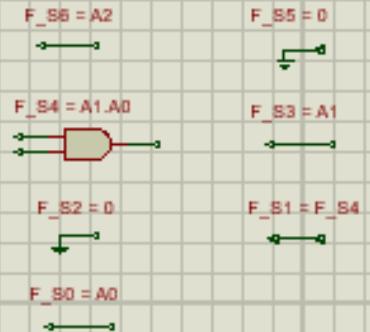
SUMA-RESULTADOS



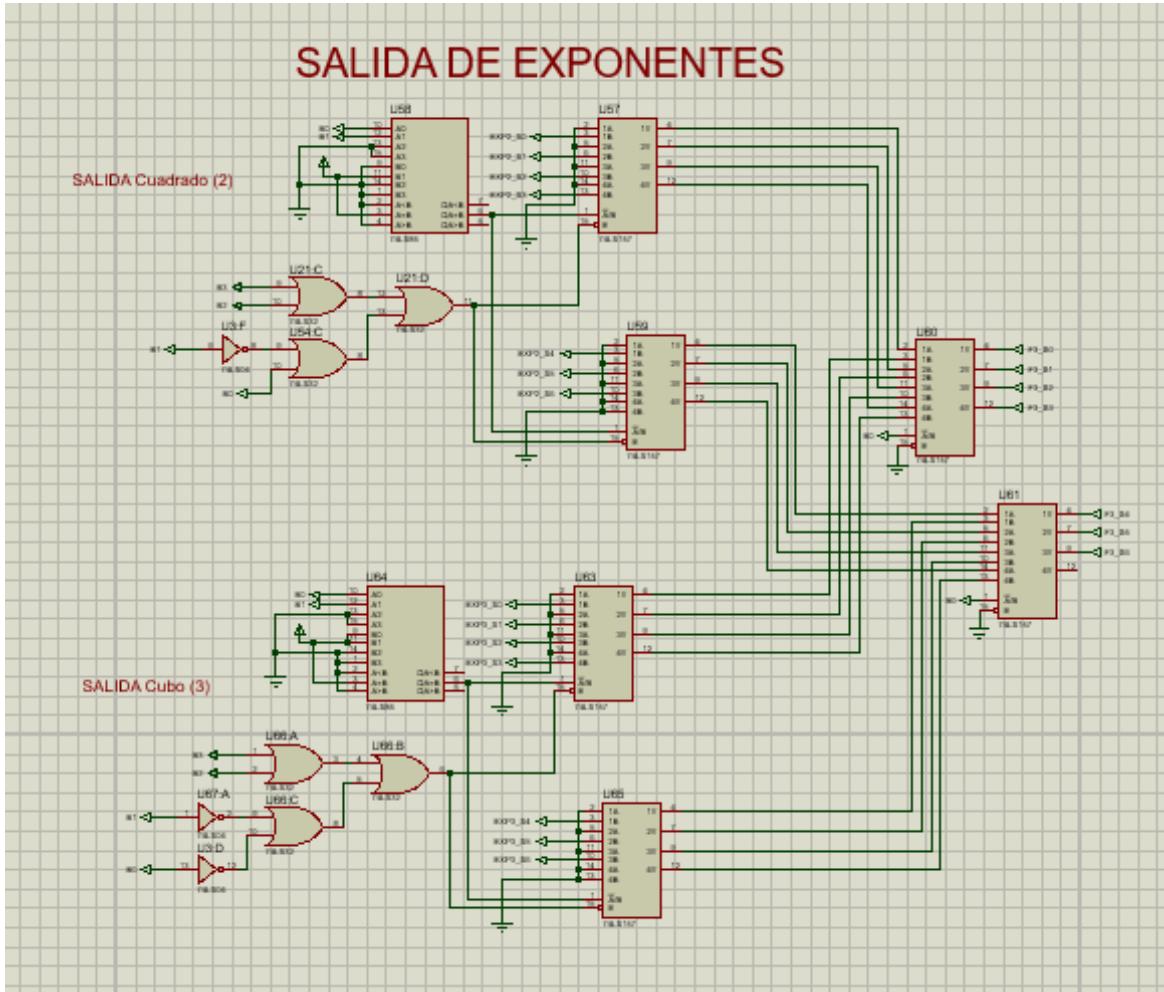
CUADRADO



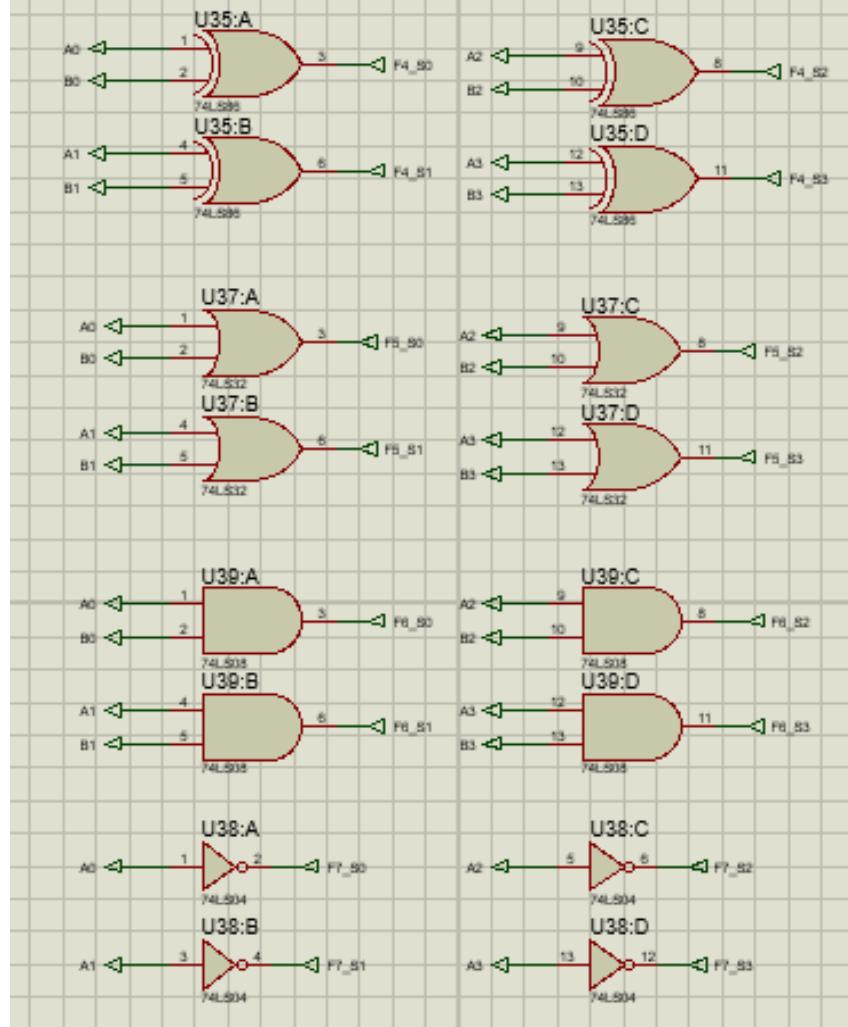
CUBO



SALIDA DE EXPONENTES

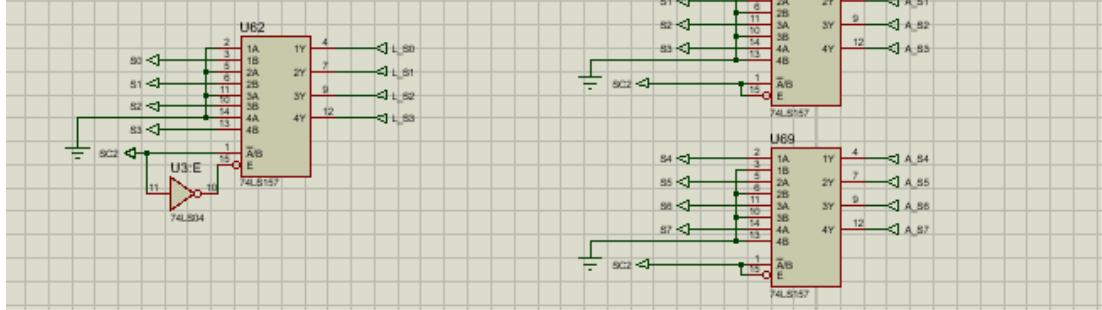


LOGICA

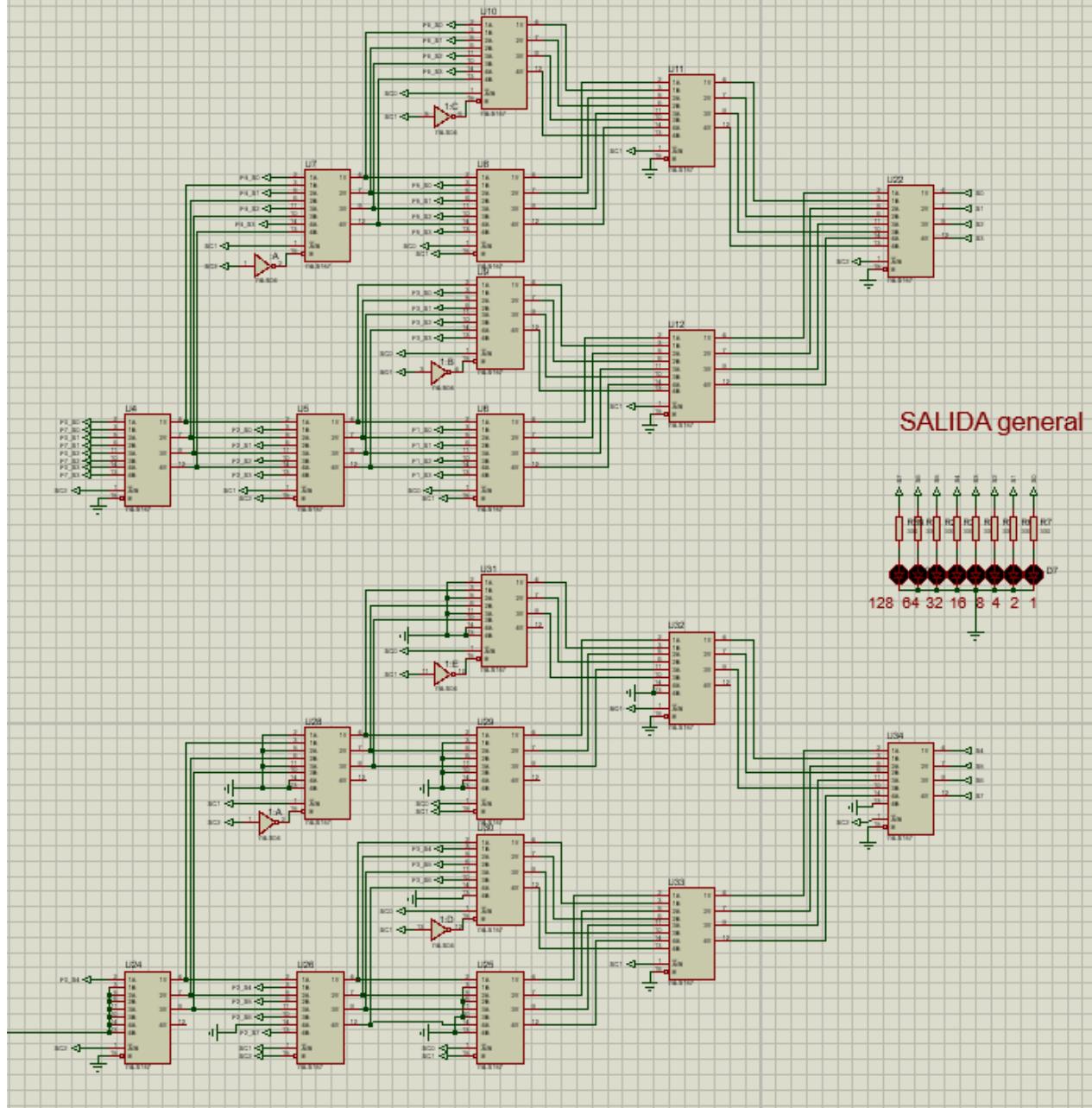


Resultado - Salida Aritmetica

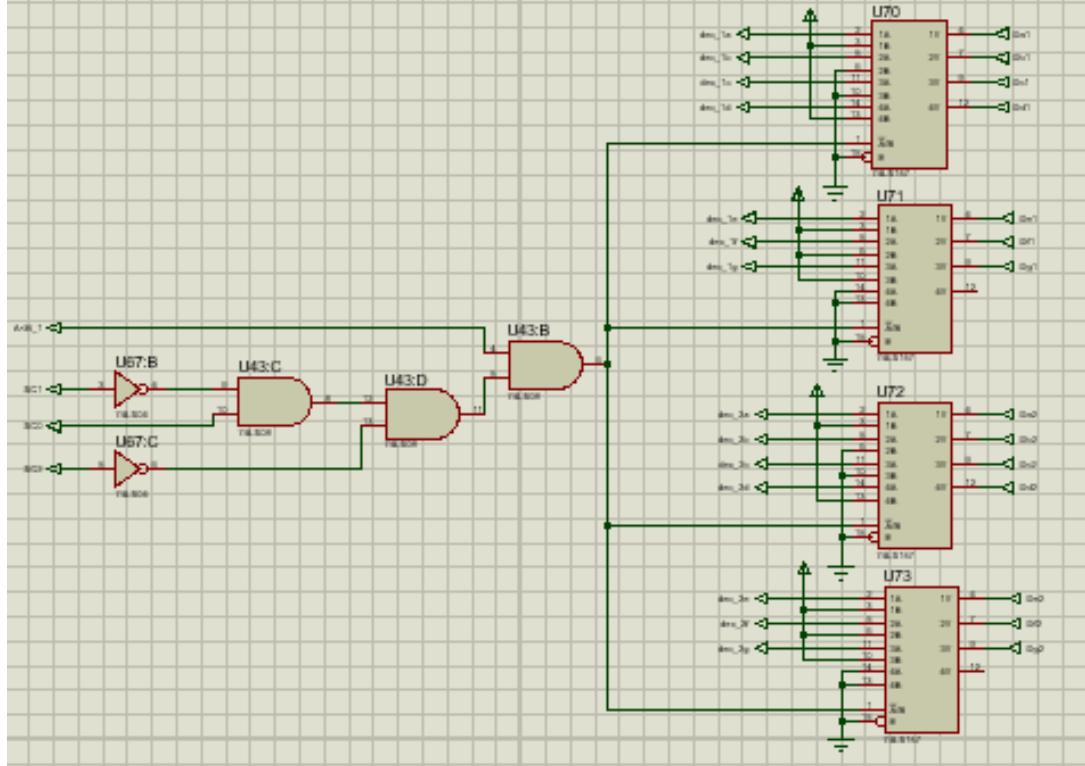
Resultado - Salida Logico



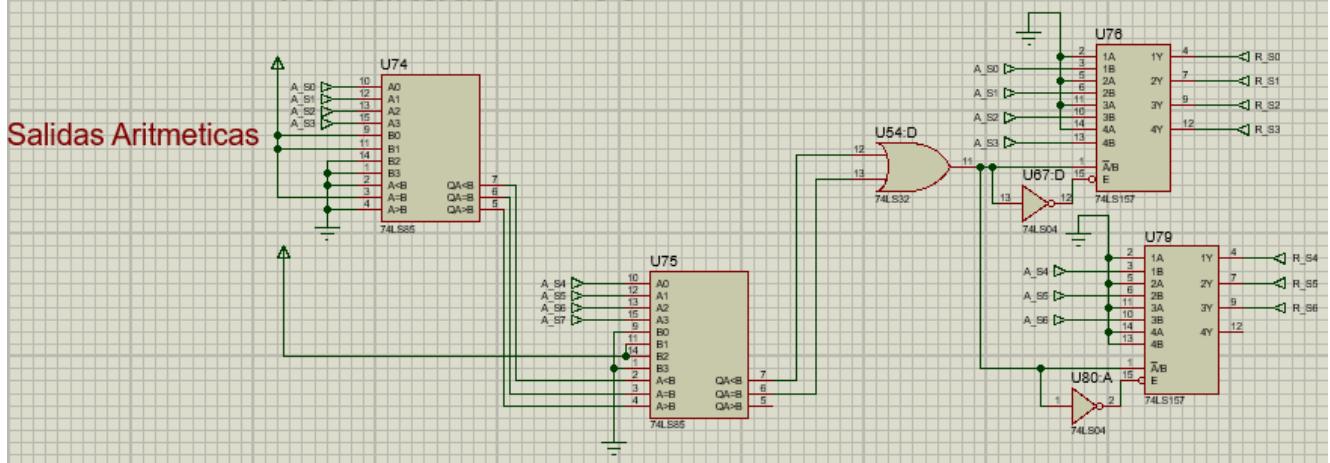
SELECTOR



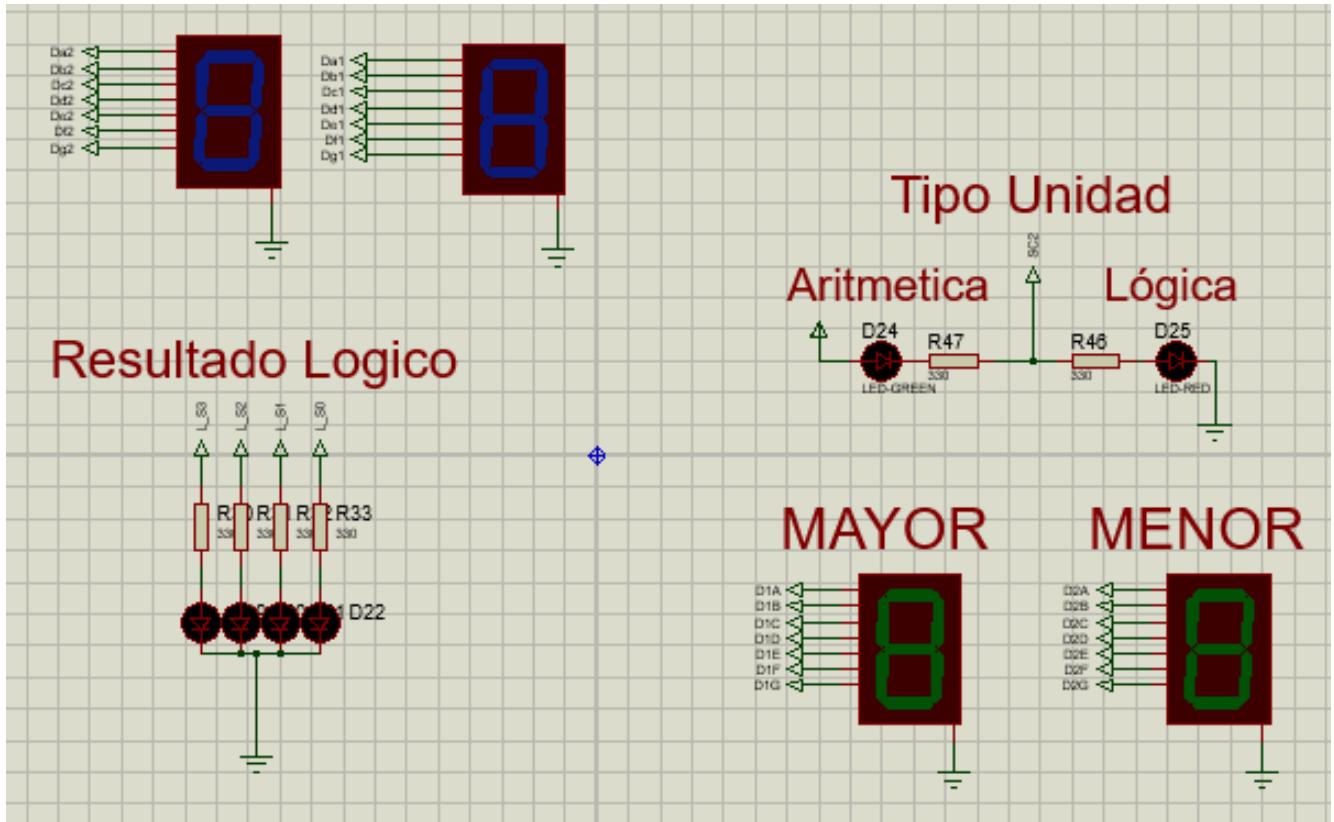
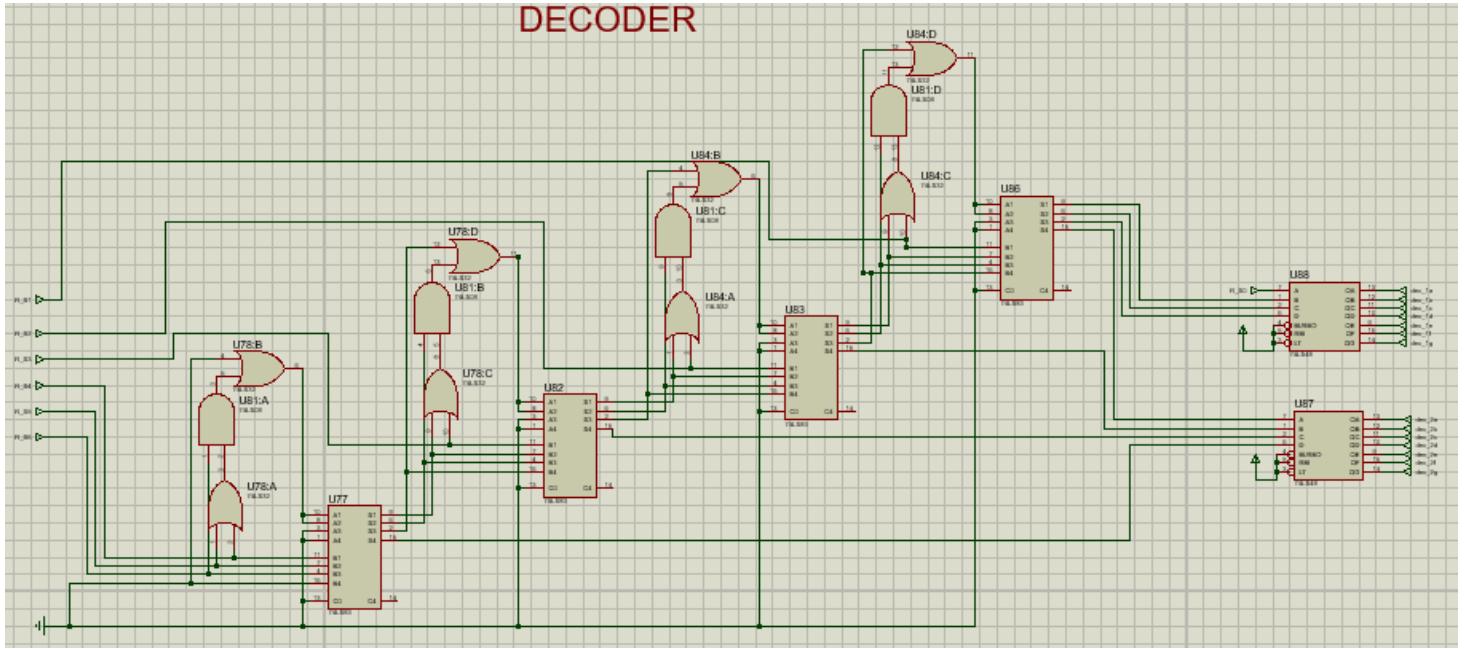
Resultado Aritmetico / Error A<B



Resultado < 100



DECODER



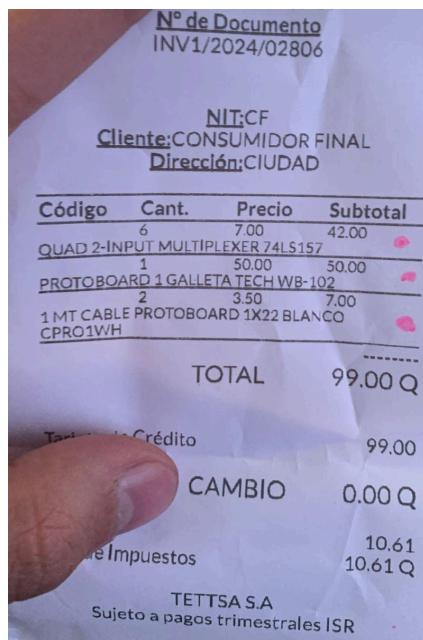
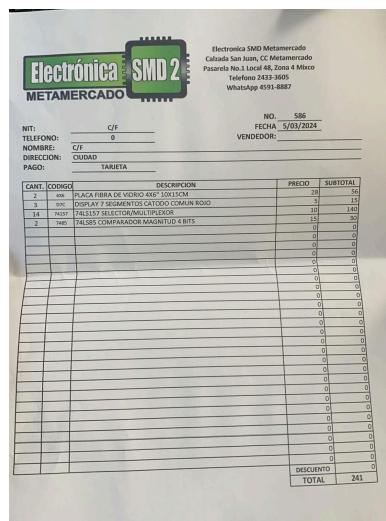
c) Equipo utilizado.

Descripcion	Codigo
Compuerta AND	7408
Compuerta NOT	7404
Compuerta OR	7486
Compuerta XOR	7432
Comparador	7485
Decoder	7447
Sumador	7483
Multiplexores	74157
Demultiplexor	74138
Transistores	2N2222

d) Tabla de Presupuesto.

Componente	Precio por unidad	Cantidad
Compuerta AND	Q.7.00	11
Compuerta NOT	Q.7.00	4
Compuerta OR	Q.7.00	6
Compuerta XOR	Q.7.00	6
Comparador	Q.10.00	7
Decoder	Q.15.00	6
Sumador	Q.10.00	10
Multiplexor	Q.10.00	35
Demultiplexor	Q.15.00	1
Transistores	Q.1.00	8
Protoboard	Q.36.00	9
Puentes	Q.0.75	120
Cables	3 Q * metro	15
Total		Q 936.00

e) Facturas



FEL - DOCUMENTO TRIBUTARIO ELECTRÓNICO
 FACTURA ELECTRÓNICA
 SERIE: 06318E19
 Nro. 1935885258

Referencia Interna: FEL02-88100
 Fecha: 08/03/2024
 Cliente NIT: 50395483
 Nombre: SANTIAGO RUMPICH XOYON
 Dirección: CIUDAD

DETALLE			
Cant.	Cod. /Descripción	Precio	Importe
1	EIC 830	Q 40.00	Q 40.00
PROTOBOARD TRANSPARENTE (HN)			
	SubTotal:	Q 40.00	
	Descuento:	Q 0.00	
	TOTAL:	Q 40.00	
Forma de pago:		Efectivo	

TOTAL EN LETRAS:
 CUARENTA QUETZALES EXACTOS

NO SE ACEPTAN CAMBIOS NI DEVOLUCIONES.
 Sujeto a retención definitiva ISR

Código Vendedor: 5
 Serie Interna: FEL02-88100
 Número Interno: null
 Datos Del Certificador:

VILLA NUEVA, GUATEMALA
 NIT.: 6455700

DOCUMENTO TRIBUTARIO ELECTRÓNICO
FEL
 FACTURA

AUTORIZACION NO.:
 AFD080DF-C9E9-497F-8920-3C6E9BA8D84C

REFERENCIA INTERNA: FE1-3131
 FECHA CERTIFICACION 2024-03-07T16:18:03.
 SERIE: AFD080DF NO : 3387509119
 FECHA EMISION: 07/MAR/2024

Vendedor: (O) N/D

NOMBRE: TRUJILLO, CONDE, SERGIO, GEHOVAN
 Y

NIT.: 6521118

DIRECCION:
 CIUDAD

CANT.	PRODUCTO	TOTAL Q.
1.00	COMPONENTES ELECTRONICOS	86.00
OCHENTA Y SEIS QUETZALES CON 00/100.		
TOTAL :		=Q. 86.00

G4S DOCUMENTA S.A
 NIT DEL CERTIFICADOR 600102027
 SUJETO A PAGOS TRIMESTRALES

FEL - DOCUMENTO TRIBUTARIO ELECTRÓNICO
 FACTURA ELECTRÓNICA
 SERIE: 1E53E183
 Nro. 2163032078

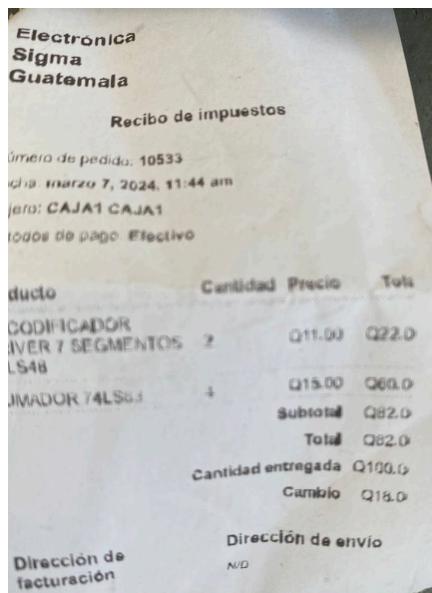
Referencia Interna: FEL02-38099
 Fecha: 08/03/2024
 Cliente NIT: 50395483
 Nombre: SANTIAGO RUMPICH XOYON
 Dirección: CIUDAD

DETALLE			
Cant.	Cod. /Descripción	Precio	Importe
4	7488	Q 10.00	Q 40.00
TTL 4 Compuertas OR Exclusivas 2 entradas(X-OR) (HN)			
9	74157	Q 10.00	Q 90.00
TTL 2 INPUT MULTIPLEXER NON INVERTING			
1	JUMPER MM X40	Q 15.00	Q 15.00
JUMPER MACHO #0 UNIDADES (H,HN)			
4	CAB 24G	Q 2.50	Q 10.00
CABLE DE PROTOBOARD # 24 NA			
	SubTotal:	Q 155.00	
	Descuento:	Q 0.00	
	TOTAL:	Q 155.00	
Forma de pago:		Efectivo	

TOTAL EN LETRAS:
 CIENTO CINCUENTA Y CINCO QUETZALES
 EXACTOS

NO SE ACEPTAN CAMBIOS NI DEVOLUCIONES.
 Sujeto a retención definitiva ISR

Código Vendedor: 5
 Serie Interna: FEL02-88099
 Número Interno: null
 Datos Del Certificador:
 G4S DOCUMENTA S.A NIT: 12521337



TOTAL: Q.936.00

APORTE INDIVIDUAL DE CADA INTEGRANTE

Integrante	Trabajo	Trabajo
Samuel Alejandro Pajoc Raymundo	Lógica Aritmética	Documentación y ensamblaje
Kevin Manuel Veliz Galvez	Lógica Aritmética	Documentación y ensamblaje
Oscar David Padilla Vásquez	Lógica Aritmética	Documentación y ensamblaje
Aarón Emanuel Trujillo Ibarra	Operación AND , OR Y NOT	Documentación y ensamblaje
Byron Enrique Rumpich Sal	Placas	Documentación y ensamblaje
Aarón Abdam Saravia Martinez	Lógica Aritmética	Documentación y ensamblaje

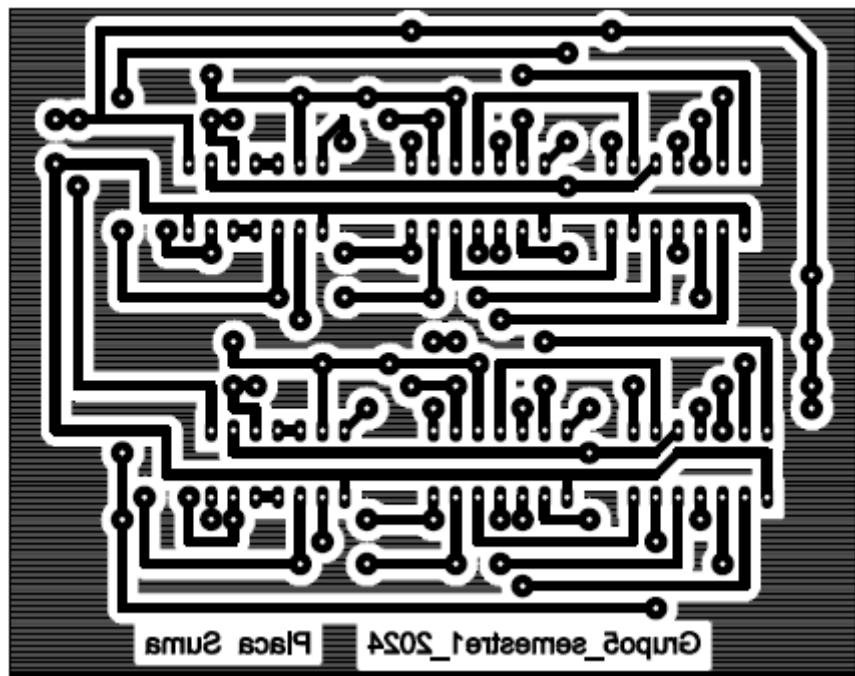
Conclusiones

1. La práctica proporciona una oportunidad para aplicar conocimientos teóricos de electrónica digital en un contexto práctico y realista. Los estudiantes pueden desarrollar habilidades en diseño de circuitos, implementación y resolución de problemas.
2. La especificación detallada de las operaciones y requisitos para cada unidad permite una comprensión clara de las funcionalidades requeridas y los métodos de implementación necesarios.
3. La práctica fomenta el desarrollo de habilidades en la manipulación de compuertas lógicas, registros y otros componentes digitales para lograr la funcionalidad deseada del circuito.
4. La importancia del orden y la claridad en el diseño del circuito se destaca como un aspecto crucial para el éxito de la implementación y la evaluación de la práctica. Un diseño claro y legible facilita la comprensión y la depuración del circuito.

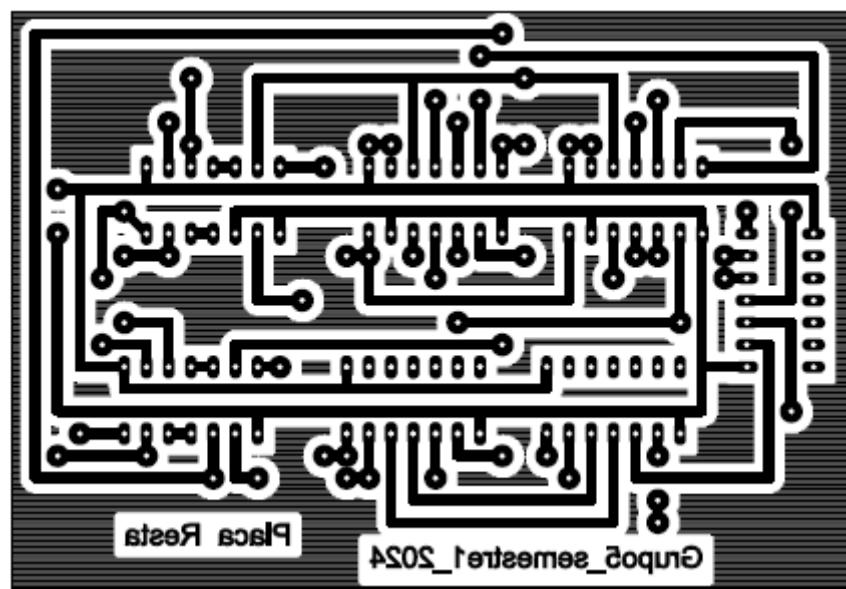
ANEXOS

- a) Diagrama del circuito impreso.

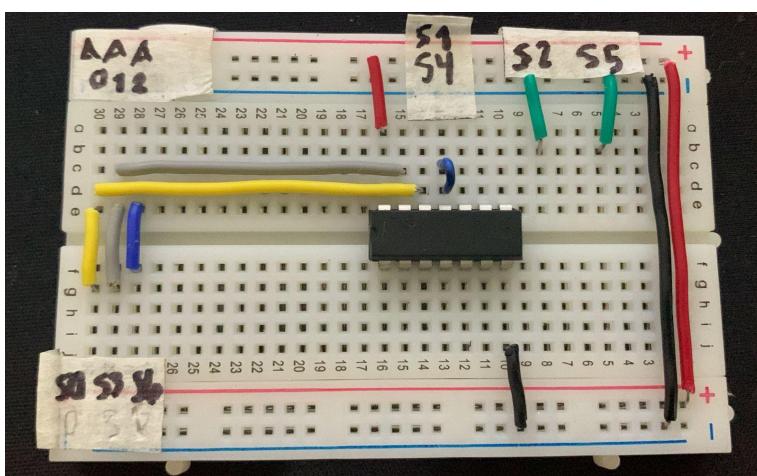
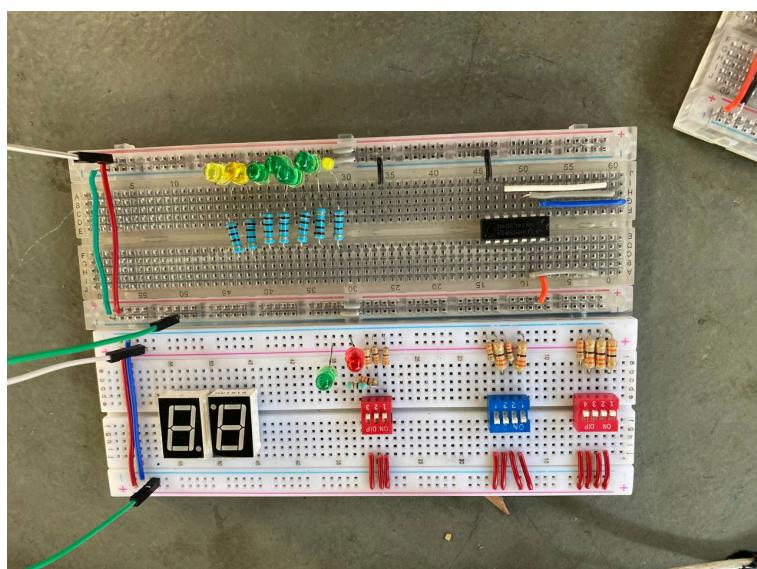
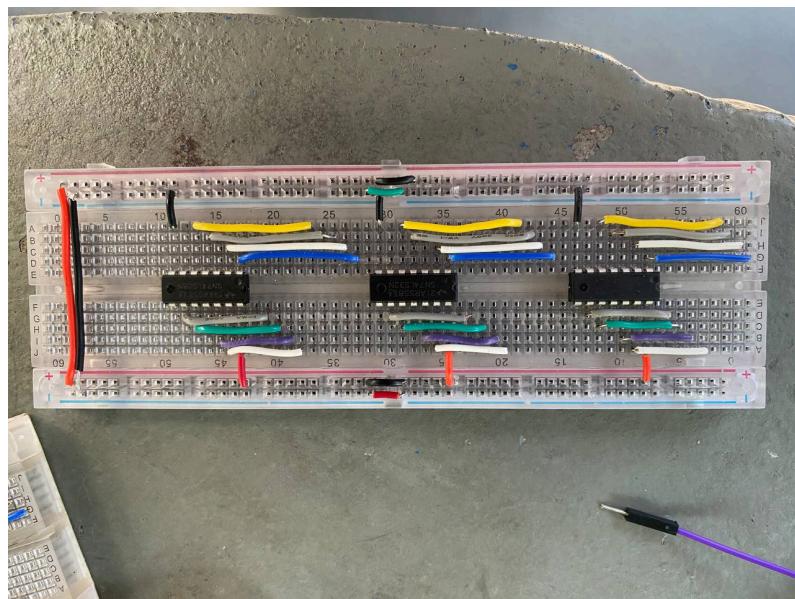
PLACA SUMA

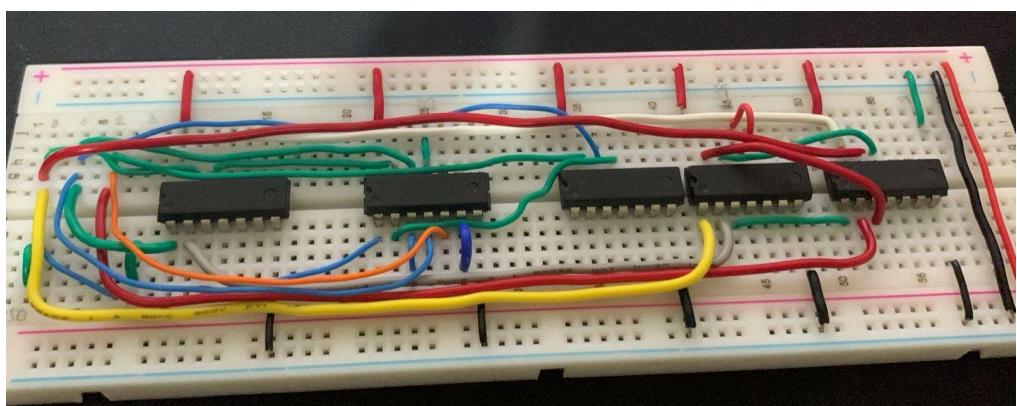
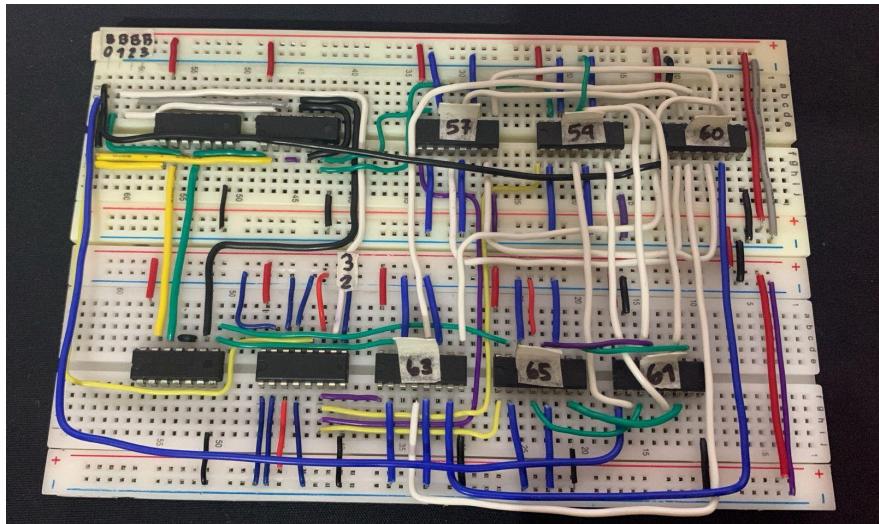


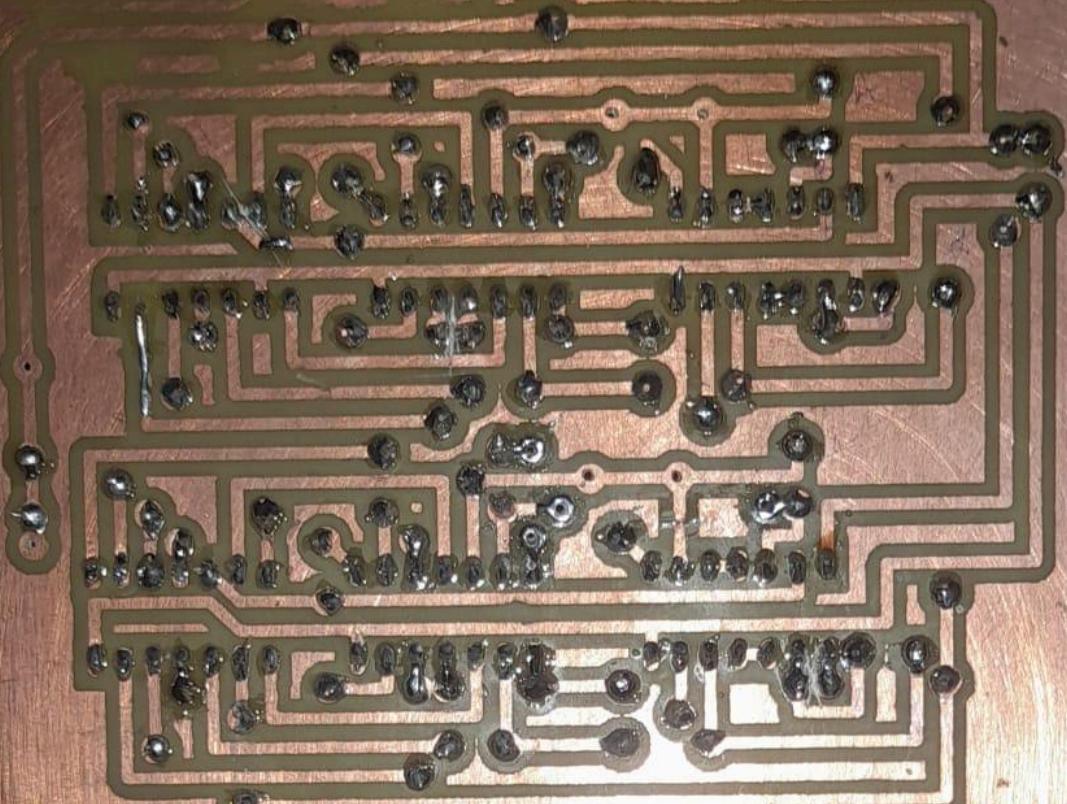
PLACA RESTA



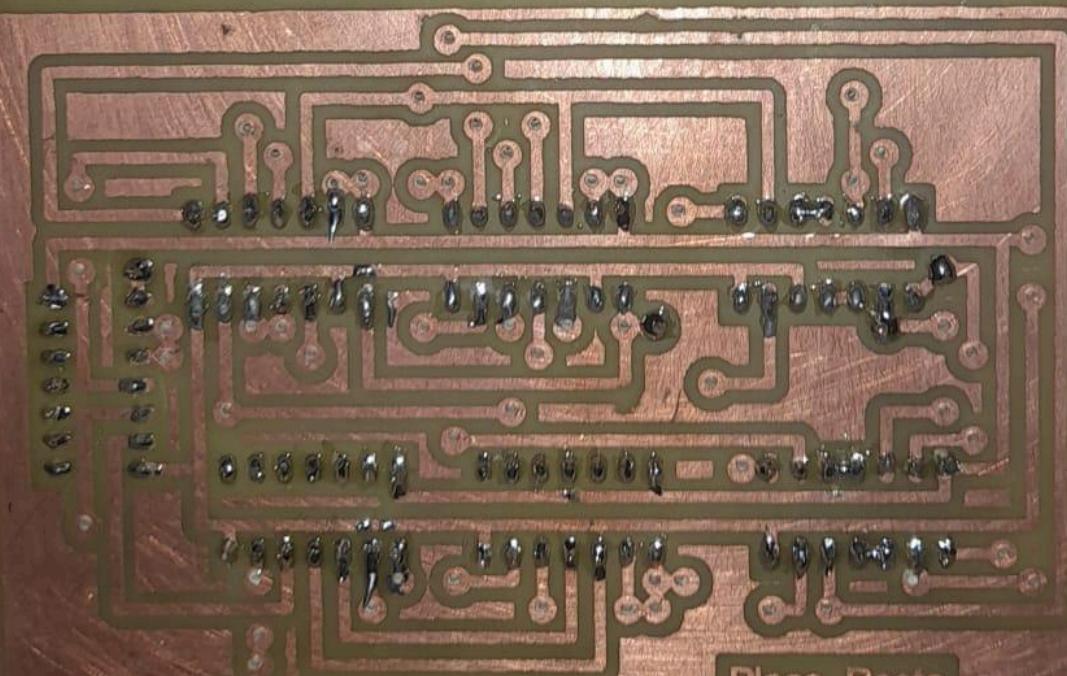
b) Fotografías de los circuitos físicos (Protoboard y Placa)







Grupo5_semestre1_2024 Placa Suma



Grupo5_semestre1_2024 Placa Rests

