

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
ORGANIZACION COMPUTACIONAL
ESCUELA DE VACACIONES JUNIO 2025
ING: FERNANDO PAZ
AUX: JORGE DE LEÓN

PRÁCTICA #2

Vacas - ALU

Carnet	Nombre integrante
201610975	Michael Cristian Itzep Ixcayau
202210483	Angely Lucrecia García Martínez
202112092	Brandon Eduardo Pablo Garcia

INTRODUCCIÓN

En el ámbito de la electrónica digital, los bloques MSI (Medium Scale Integration, por sus siglas en inglés) desempeñan un papel crucial en la construcción de circuitos complejos. Dentro de esta categoría, los bloques MSI tipo aritmético son especialmente importantes para realizar operaciones aritméticas básicas, como sumas, restas y multiplicaciones, de manera eficiente y confiable. Los bloques digitales combinacionales MSI tipo aritmético están diseñados para realizar operaciones matemáticas utilizando compuertas lógicas, registros y otros componentes digitales. Estos bloques están configurados para recibir dos o más operandos y generar una salida que representa el resultado de la operación.

OBJETIVOS

Objetivo General

- Construir una Unidad Aritmética Lógica Básica (ALU).

Objetivos Específicos

- Aprender el funcionamiento de Multiplexores, Demultiplexores, Comparadores y Decodificadores.
- Construir un diseño óptimo, logrando utilizar la menor cantidad de dispositivos.
- Aprender el funcionamiento de Operaciones Lógicas, Aritméticas y Comparativas con números binarios

DESCRIPCIÓN

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).

FUNCIONES BOOLEANAS Y MAPAS DE KARNAUGH

Bloque1: Controlador de operación (Selector)

A	B	Operación	Suma
0	0	Suma	1
0	1	Resta	0
1	0	AND	0
1	1	OR	0

A	B	Operación	Resta
0	0	Suma	0
0	1	Resta	1
1	0	AND	0
1	1	OR	0

A	B	Operación	AND
0	0	Suma	0
0	1	Resta	0
1	0	AND	1
1	1	OR	0

A	B	Operación	OR
0	0	Suma	0
0	1	Resta	0
1	0	AND	0
1	1	OR	1

Bloque Aritmético: Suma										
A			B			CARRY	RESULTADO			DECIMAL
A2	A1	A0	B2	B1	B0	S3	S2	S1	S0	A+B
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1	1	1
0	0	0	0	1	0	0	0	1	0	2
0	0	0	0	1	1	0	0	1	1	3
0	0	0	1	0	0	1	0	0	0	4
0	0	0	1	0	1	1	0	1	1	5
0	0	0	1	1	0	0	1	1	0	6
0	0	0	1	1	1	0	1	1	1	7
0	0	1	0	0	0	0	0	0	1	1
0	0	1	0	0	1	0	0	1	0	2
0	0	1	0	1	0	0	0	1	1	3
0	0	1	0	1	1	0	1	0	0	4
0	0	1	1	0	0	0	1	0	1	5
0	0	1	1	0	1	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	0	1	1	1	1	0	1	1	0	8
0	1	0	0	0	0	0	0	1	0	2
0	1	0	0	0	1	0	0	1	1	3
0	1	0	0	1	0	0	1	0	0	4
0	1	0	0	1	1	0	1	0	1	5
0	1	0	1	0	0	0	1	1	0	6
0	1	0	1	0	1	0	1	1	1	7
0	1	0	1	1	0	0	1	0	0	8
0	1	0	1	1	1	0	1	0	1	9
0	1	1	0	0	0	0	0	1	1	3
0	1	1	0	0	1	0	1	0	0	4
0	1	1	0	1	0	0	1	0	1	5
0	1	1	0	1	1	0	1	1	0	6
0	1	1	1	0	0	0	1	1	1	7
0	1	1	1	0	1	0	1	0	0	8
0	1	1	1	1	0	1	0	0	1	9
0	1	1	1	1	1	1	0	1	0	10
1	0	0	0	0	0	0	1	0	0	4
1	0	0	0	0	1	0	1	0	1	5
1	0	0	0	1	0	0	1	1	0	6
1	0	0	0	1	1	0	1	1	1	7
1	0	0	1	0	0	1	0	0	0	8
1	0	0	1	0	1	1	0	0	1	9
1	0	1	0	0	0	0	1	0	1	5
1	0	1	0	0	1	0	1	1	0	6
1	0	1	0	1	0	0	1	1	1	7
1	0	1	0	1	1	1	0	0	0	8
1	0	1	1	0	0	1	0	0	1	9
1	0	1	1	0	1	1	0	1	0	10
1	0	1	1	1	0	1	0	1	1	11
1	0	1	1	1	1	1	1	0	0	12
1	1	0	0	0	0	0	1	1	0	6
1	1	0	0	0	1	0	1	1	1	7
1	1	0	0	1	0	1	0	0	0	8
1	1	0	0	1	1	1	0	0	1	9
1	1	0	1	0	0	1	0	1	0	10
1	1	0	1	0	1	1	0	1	1	11
1	1	0	1	1	0	1	1	0	0	12
1	1	0	1	1	1	1	1	0	1	13
1	1	1	0	0	0	0	1	1	1	7
1	1	1	0	0	1	1	0	0	0	8
1	1	1	0	1	0	1	0	0	1	9
1	1	1	0	1	1	1	0	1	0	10
1	1	1	1	0	0	1	0	1	1	11
1	1	1	1	0	1	1	1	0	0	12

Bloque Aritmético: Resta

A			B			RESULTADO				DECIMAL
A2	A1	A0	B2	B1	B0	R3	R2	R1	R0	A-B
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	E	E	E	E	E
0	0	0	0	1	0	E	E	E	E	E
0	0	0	0	1	1	E	E	E	E	E
0	0	0	1	0	0	E	E	E	E	E
0	0	0	1	0	1	E	E	E	E	E
0	0	0	1	1	0	E	E	E	E	E
0	0	0	1	1	1	E	E	E	E	E
0	0	1	0	0	0	0	0	0	1	1
0	0	1	0	0	1	0	0	0	0	0
0	0	1	0	1	0	E	E	E	E	E
0	0	1	0	1	1	E	E	E	E	E
0	0	1	1	0	0	E	E	E	E	E
0	0	1	1	0	1	E	E	E	E	E
0	0	1	1	1	0	E	E	E	E	E
0	0	1	1	1	1	E	E	E	E	E
0	1	0	0	0	0	0	0	1	0	2
0	1	0	0	0	1	0	0	0	1	1
0	1	0	0	1	0	0	0	0	0	0

0	1	0	1	0	0	E	E	E	E	E
0	1	0	1	0	1	E	E	E	E	E
0	1	0	1	1	0	E	E	E	E	E
0	1	0	1	1	1	E	E	E	E	E
0	1	1	0	0	0	0	0	1	1	3
0	1	1	0	0	1	0	0	1	0	2
0	1	1	0	1	0	0	0	0	1	1
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	E	E	E	E	E
0	1	1	1	0	1	E	E	E	E	E
0	1	1	1	1	0	E	E	E	E	E
0	1	1	1	1	1	E	E	E	E	E
1	0	0	0	0	0	0	1	0	0	4
1	0	0	0	0	1	0	0	1	1	3
1	0	0	0	1	0	0	0	1	0	2
1	0	0	0	1	1	0	0	0	1	1
1	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	E	E	E	E	E
1	0	0	1	1	0	E	E	E	E	E
1	0	0	1	1	1	E	E	E	E	E
1	0	1	0	0	0	0	1	0	1	5
1	0	1	0	0	1	0	1	0	0	4
1	0	1	0	1	0	0	0	1	1	3

1	0	1	0	1	1	0	0	1	0	2
1	0	1	1	0	0	0	0	0	1	1
1	0	1	1	0	1	0	0	0	0	0
1	0	1	1	1	0	E	E	E	E	E
1	0	1	1	1	1	E	E	E	E	E
1	1	0	0	0	0	0	1	1	0	6
1	1	0	0	0	1	0	1	0	1	5
1	1	0	0	1	0	0	1	0	0	4
1	1	0	0	1	1	0	0	1	1	3
1	1	0	1	0	0	0	0	1	0	2
1	1	0	1	0	1	0	0	0	1	1
1	1	0	1	1	0	0	0	0	0	0
1	1	0	1	1	1	E	E	E	E	E
1	1	1	0	0	0	0	1	1	1	7
1	1	1	0	0	1	0	1	1	0	6
1	1	1	0	1	0	0	1	0	1	5
1	1	1	0	1	1	0	1	0	0	4
1	1	1	1	0	0	0	0	1	1	3
1	1	1	1	0	1	0	0	1	0	2
1	1	1	1	1	0	0	0	0	1	1
1	1	1	1	1	1	0	0	0	0	0

• Bloque Lógico: Resultado Bit a bit

A			B			AND			OR			XNOR			NAND		
A2	A1	A0	B2	B1	B0	AND2	AND1	AND0	OR2	OR1	OR0	XNOR2	XNOR1	XNOR0	NAND2	NAND1	NAND0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	1	1	1
0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	1	1	1
0	0	0	0	1	1	0	0	0	0	1	1	1	0	0	1	1	1
0	0	0	1	0	0	0	0	0	1	0	0	0	1	1	1	1	1
0	0	0	1	0	1	0	0	0	1	0	1	0	1	0	1	1	1
0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	1	1	1
0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1
0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1
0	0	1	0	0	1	0	0	1	0	0	1	1	1	1	1	1	0
0	0	1	0	1	0	0	0	0	0	1	1	1	0	0	1	1	1
0	0	1	0	1	1	0	0	1	0	1	1	1	0	1	1	1	0
0	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	1	1
0	0	1	1	0	1	0	0	1	1	0	1	0	1	1	1	1	0
0	0	1	1	1	0	0	0	0	1	1	1	0	0	1	1	1	0
0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1
0	1	0	0	0	1	0	0	0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	0	1	0	0	1	0	1	1	1	1	0	1

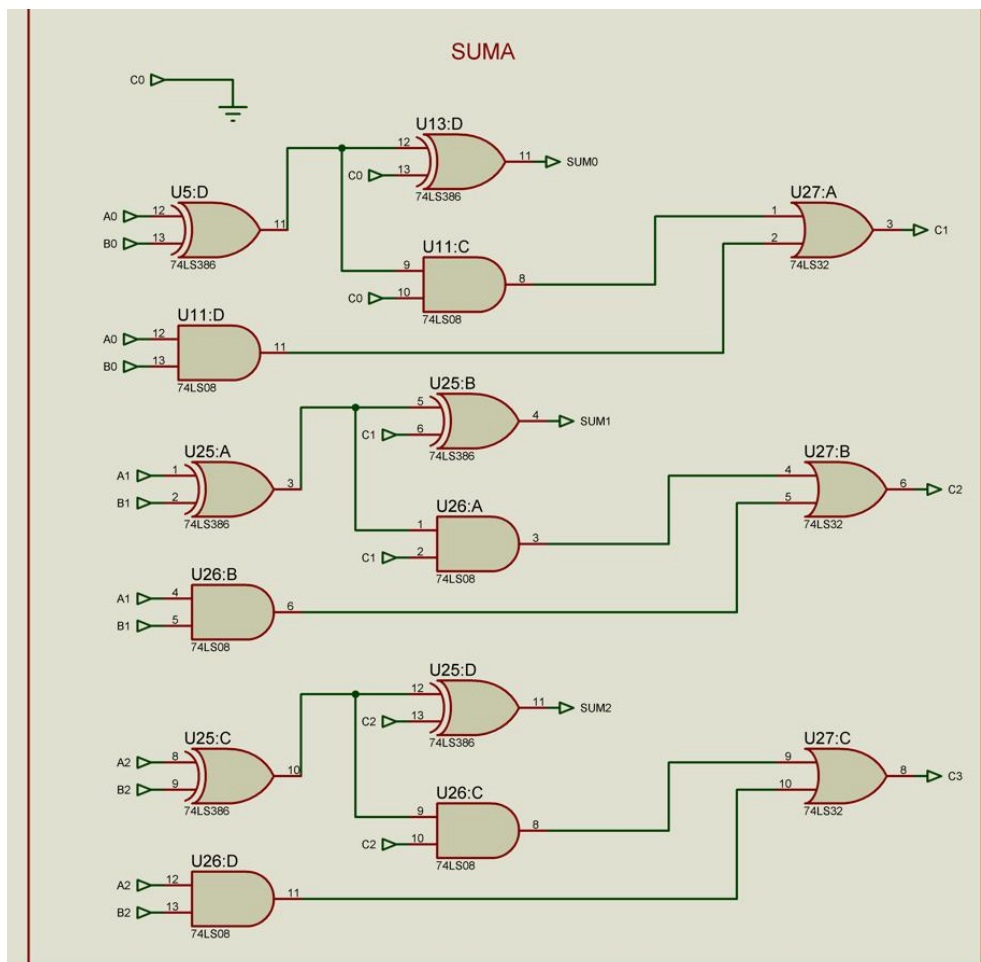
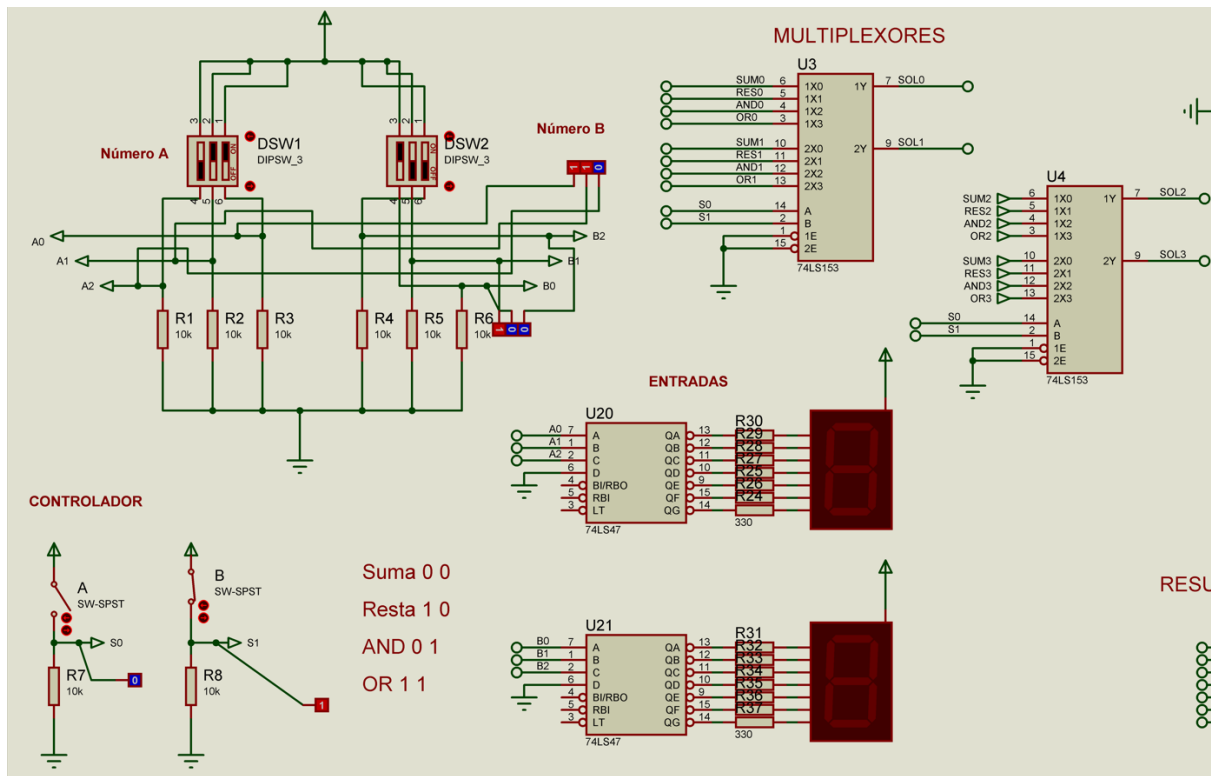
0	1	0	1		0	0	0	0	0	1	1	0	0	0	1	1	1	1
0	1	0	1	0	1	0	1	0	0	1	1	1	0	0	1	1	1	1
0	1	0	1	1	0	0	1	0	1	1	1	0	0	1	1	0	1	
0	1	0	1	1	1	0	1	0	1	1	1	1	0	1	0	1	1	
0	1	1	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1	
0	1	1	0	0	1	0	0	1	0	1	1	1	1	0	1	1	0	
0	1	1	0	1	0	0	1	0	0	1	1	1	1	1	0	1	1	
0	1	1	0	1	1	0	1	0	1	0	1	1	1	1	1	1	0	
0	1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

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

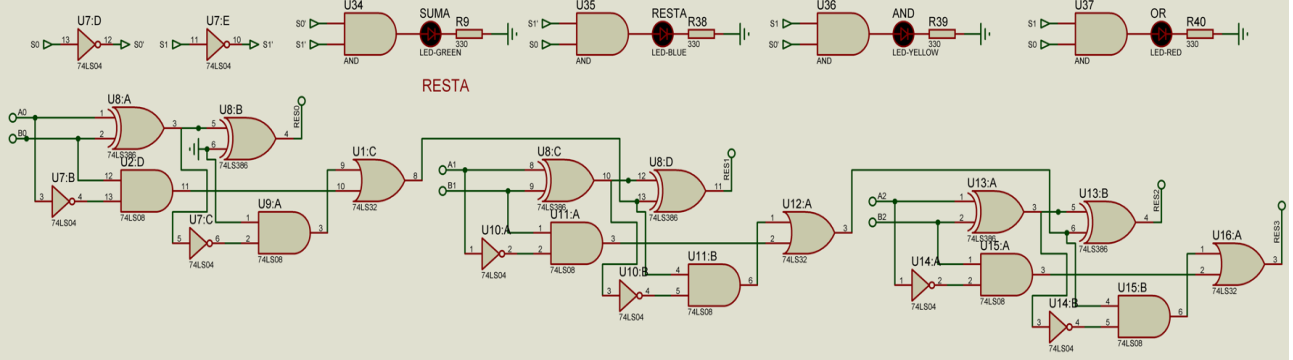
Bloque Comparador: Compara los valores												
A			B			MAYOR			MENOR			DECIMAL
A2	A1	A0	B2	B1	B0	Mayor2	Mayor1	Mayor0	Menor2	Menor1	Menor0	Valor
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1	0	0	0	1
0	0	0	0	1	0	0	1	0	0	0	0	2
0	0	0	0	1	1	0	1	1	0	0	0	3
0	0	0	1	0	0	1	0	0	0	0	0	4
0	0	0	1	0	1	1	0	1	0	0	0	5
0	0	0	1	1	0	1	1	0	0	0	0	6
0	0	0	1	1	1	1	1	1	0	0	0	7
0	0	1	0	0	0	0	0	1	0	0	0	0
0	0	1	0	0	1	0	0	1	0	0	1	1
0	0	1	0	1	0	0	1	0	0	0	1	2
0	0	1	0	1	1	0	1	1	0	0	1	3
0	0	1	1	0	0	1	0	0	0	0	1	4
0	0	1	1	0	1	1	0	1	0	0	1	5
0	0	1	1	1	0	1	1	0	0	0	1	6
0	0	1	1	1	1	1	1	1	0	0	1	7
0	1	0	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	1	0	1	0	0	0	1	1
0	1	0	0	1	0	0	1	0	0	1	0	2

Suma	A = 0	A = 1		Minterm		Maxterm	
B = 0	1	0		Suma = B' * A'		Suma = (B + A')(B' + A)(B' + A')	
B = 1	0	0					
Resta	A = 0	A = 1		Minterm		Maxterm	
B = 0	0	1		Resta = B' * A		Resta = (B + A)(B' + A)(B' + A')	
B = 1	0	0					
AND	A = 0	A = 1		Minterm		Maxterm	
B = 0	0	0		And = B * A'		And = (B + A)(B + A')(B' + A')	
B = 1	1	0					
OR	A = 0	A = 1		Minterm		Maxterm	
B = 0	0	0		Or = B * A		Or = (B + A)(B + A')(B' + A)	
B = 1	0	1					

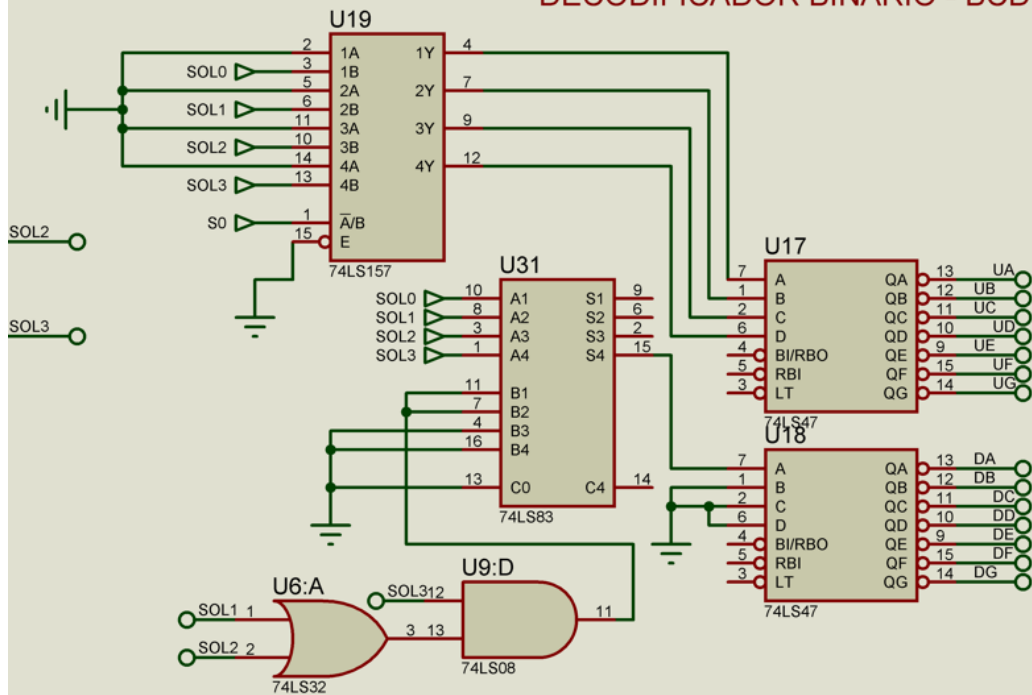
DIAGRAMAS DEL DISEÑO DEL CIRCUITO



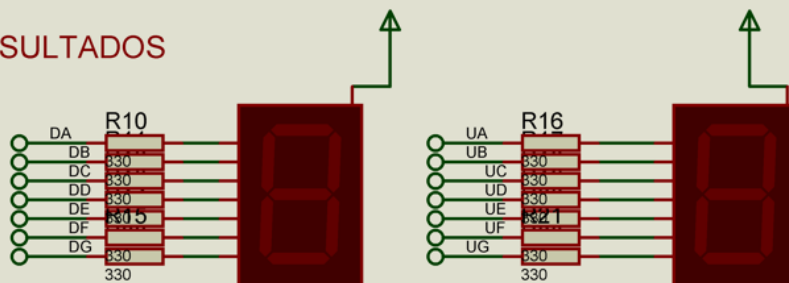
TIPO DE OPERACIÓN



DECODIFICADOR BINARIO - BCD



RESULTADOS



EQUIPO UTILIZADO

Herramienta utilizada para su elaboración:

Herramienta	Uso
Pelacables	Para preparar las puntas de los cables a usar
Corta cables o alicata de corte	Para cortar cables y terminales
Pinzas de punta fina	Para posicionar y ajustar componentes pequeños
Protoboard	Para pruebas previas
Computadora con Proteus	Para simulación

Componentes electrónicos utilizados:

Circuitos Integrados TTL

Componente	Descripción	Código
74LS08	Compuerta AND TTL	AND
74LS32	Compuerta OR TTL	OR
74LS04	Compuerta NOT TTL	NOT
74LS86	Compuerta XOR TTL	XOR
74LS85	Comparador	
74LS47	Decoder	
74LS153	Multiplexores	

Transistores y componentes asociados

Componente	Descripción	Notas
2N2222A	Transistor NPN	Para compuertas transistorizadas y switches
Resistencias	330Ω, 300Ω, 10kΩ	Para limitación de corriente en base, colector y segmentos
Dipswitch	Selector de entrada binaria (X1 y X2)	Ideal para pruebas
Display de 7 segmentos cátodo común	Salida visual para semáforo normal	Requiere conexión a tierra por cátodo
Display de 7 segmentos ánodo común	Salida para semáforo espejo	Requiere alimentación por ánodo
Protoboard	Para pruebas antes de soldar	Opcional si no se trabaja solo en placa
PCB de cobre o fibra de vidrio	Para la placa final	Con grabado o perforación manual
Zócalos DIP	Para proteger los integrados TTL en la placa	Recomendado para montaje seguro
Fuente regulada de 5V	Fuente de alimentación estable	Obligatoria para TTL

PRESUPUESTO

Producto	Cantidad	Precio Unitario	Total
COMPUERTA LOGICA AND 74LS08	8	Q5.00	Q40.00
COMPUERTA LOGICA OR 74LS32	5	Q5.00	Q25.00
COMPUERTA LOGICA NOT 74LS04	7	Q5.00	Q35.00
COMPUERTA LOGICA XOR 74LS86	7	Q6.00	Q42.00
DISPLAY DE 7 SEGMENTOS ANODO COMUN ROJO	8	Q5.00	Q40.00
DISPLAY DE 7 SEGMENTOS CATODO ROJO	4	Q5.00	Q20.00
RESISTENCIAS 330Ω 1/4W	20	Q0.50	Q10.00
RESISTENCIAS 10kΩ 1/4W	10	Q0.50	Q5.00
PROTOBOARD 2 GALLETAS CON TERMINALES	1	Q165.00	Q165.00
DIP SWITCH DE 3 POSICIONES	1	Q3.00	Q6.00
DIP SWITCH DE 2 POSICIONES	1	Q2.00	Q2.00
74LS153 SELECTOR Y MULTIPLEXOR	2	Q7.00	Q14.00
SUMADOR 74LS85	2	Q15.00	Q30.00
DECODIFICADOR DRIVER 7 SEGMENTOS 74LS48 CATODO COMUN	4	Q11.00	Q44.00
RESISTENCIAS 300Ω 1/4W	15	Q0.50	Q7.50

LED ROJO DIFUSO 5MM	6	Q1.00	Q6.00
PROTOBOARD 1 GALLETA BLANCO	2	Q34.00	Q68.00
ALAMBRE AZUL PARA PROTOBOARD	3	Q2.00	Q6.00
ALAMBRE BLANCO PARA PROTOBOARD	11	Q2.00	Q22.00
ALAMBRE NEGRO PARA PROTOBOARD	9	Q2.00	Q18.00
ALAMBRE ROJO PARA PROTOBOARD	3	Q2.00	Q6.00
CABLES DUPONT MACHO-MACHO 10 CM	15	Q0.50	Q7.50
CABLES DUPONT MACHO-MACHO 20 CM	15	Q0.75	Q11.25
CABLES DUPONT MACHO-MACHO 30 CM	15	Q0.75	Q11.25
CONECTOR PLUG DCA TERMINAL BLOCK	1	Q4.50	Q4.50
CONECTOR JACK DCA TERMINAL BLOCK	1	Q5.00	Q5.00
CABLES DUPONT MACHO-MACHO	2	Q0.50	Q1.00
74LS157 MULTIPLEXOR 2 ENTRADAS	1	Q7.00	Q7.00
TOTAL			Q643.00

APORTE DE CADA INTEGRANTE

El aporte de cada integrante fue de Q 213.00 quetzales exactos

CONCLUSIÓN

- La implementación del ALU en Proteus permitió simular y verificar el funcionamiento lógico de las operaciones básicas (suma, resta, AND y OR), lo cual facilitó la identificación de errores antes de pasar al montaje físico. Esta etapa fue clave para asegurar que el diseño funciona correctamente con componentes reales.
- El montaje físico del ALU demostró la viabilidad práctica del diseño digital simulado, confirmando que las operaciones aritméticas y lógicas se ejecutan correctamente mediante compuertas e integrados. Esta experiencia permitió afianzar conocimientos sobre lógica combinacional y cómo se integran múltiples funciones en una unidad funcional como el ALU.