

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
ORGANIZACIÓN COMPUTACIONAL
SECCIÓN N

“Práctica 2”

Daniel Estuardo Cuque Ruíz	202112145
Damián Ignacio Peña Afre	202110568
Alvaro Norberto García Meza	202109567
José Manuel Ibarra Pirir	202001800

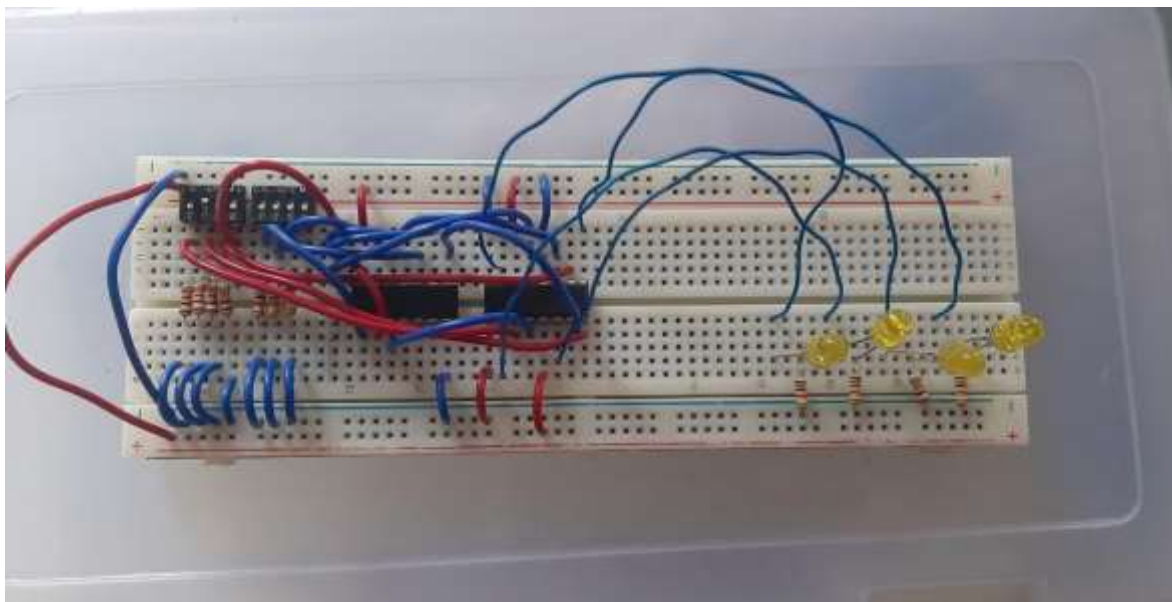
Grupo 1

Guatemala, diciembre 2022

Circuito Físico

Equipo utilizado para el circuito físico

1. XOR 7486
2. Sumador 7483
3. 2 dip Switch
4. 4 leds
5. 10 resistencias 1k ohm



Funciones booleanas y mapas de Karnaugh

Suma:

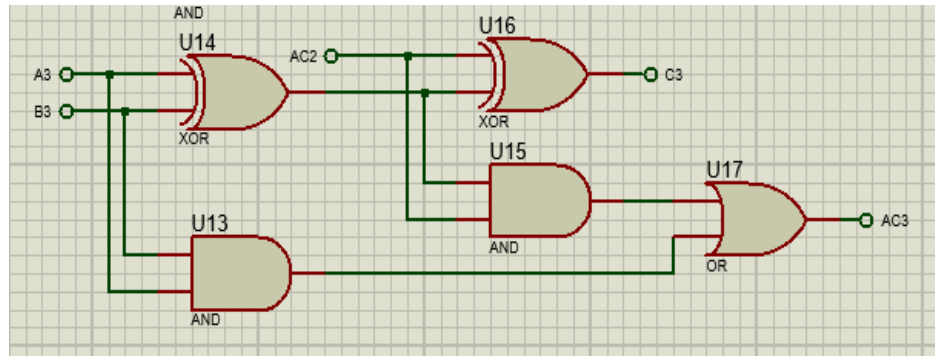
Tabla de verdad sumador completo				
In			Out	
A	B	Ci	Cout	Suma
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$\text{Suma} = A'B'Ci + A'BCi' + AB'Ci' + ABC$$

$$A \text{ XOR } B \text{ XOR } C$$

$$\text{Cout} = A'BCin + AB'Ci' + ABC' + ABCin$$

$$AB + Cin(A \text{ XOR } B)$$



Resta:

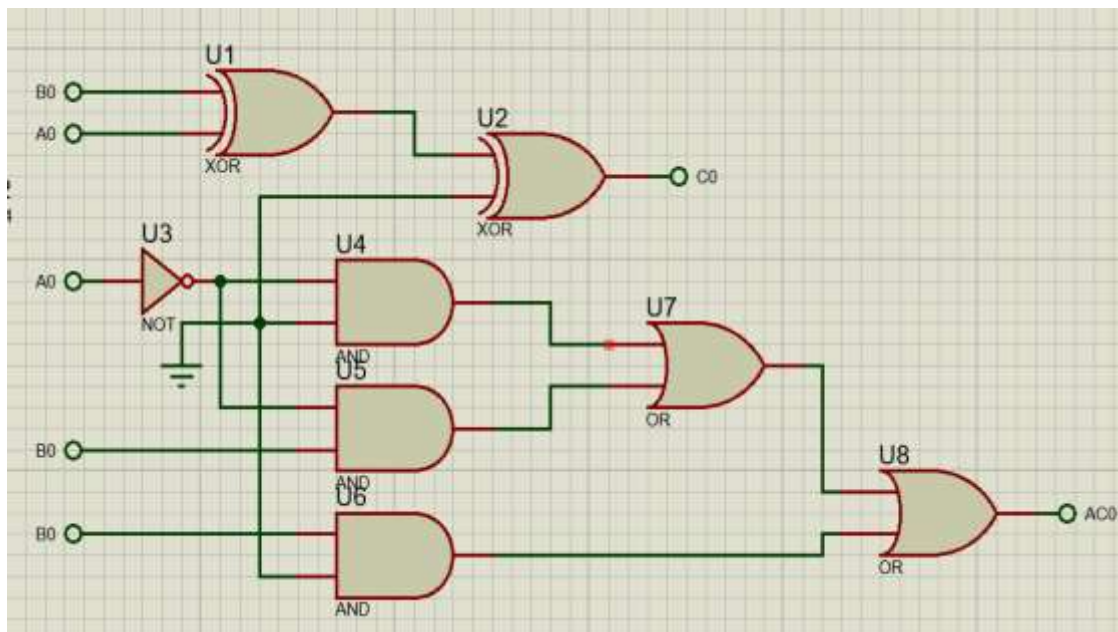
Tabla de verdad restador completo

In			Out	
A	B	Ci	Cout	Resta
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

$$\text{Resta} = A'Ci + A'B + BCi$$

$$A \text{ XOR } (B \text{ XOR } Ci)$$

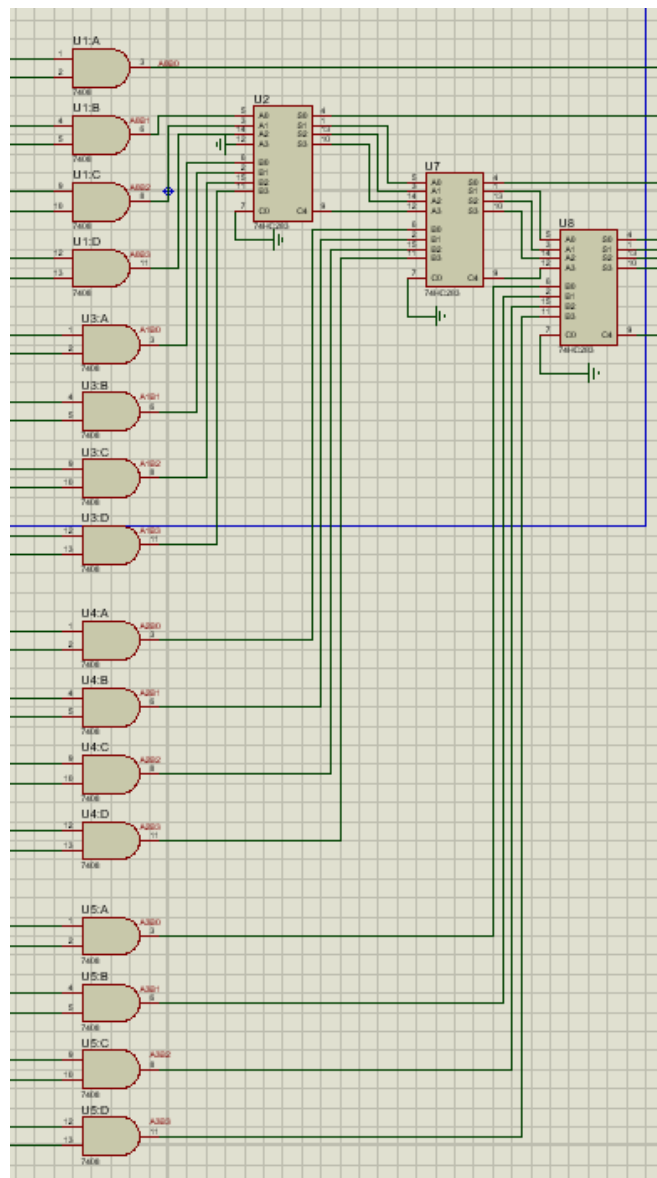
$$\text{Cout} = A'B + A'Ci + Bci$$



Multiplicación:

No maneja una tabla ya al manejar 8 variables sería una tabla de 256 unidades, así que haciendo una multiplicación normal obtenemos la suma por cada bit de cada variable. Usando compuertas and y sumadores.

						A3	A2	A1	A0
					x	B3	B2	B1	B0
						B0A3	B0A2	B0A1	B0A0
				B1A3		B1A2	B1A1	B1A0	
			B2A3	B2A2		B2A1	B2A0		
		B3A3	B3A2	B3A1		B3A0			
Salidas:	M7	M6	M5	M4	M3	M2	M1	M0	



Potencia:

Elevado al cuadrado

POTENCIA DE UN NÚMERO AL CUADRADO												
ENTRADAS					SALIDAS							
					2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	
No	A0	A1	A2	A3	M6	M5	M4	M3	M2	M1	M0	
0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	1	0	0	0	0	0	0	1	
2	0	0	1	0	0	0	0	0	1	0	0	
3	0	0	1	1	0	0	0	1	0	0	1	
4	0	1	0	0	0	0	1	0	0	0	0	
5	0	1	0	1	0	0	1	1	0	0	1	
6	0	1	1	0	0	1	0	0	1	0	0	
7	0	1	1	1	0	1	1	0	0	0	1	
8	1	0	0	0	1	0	0	0	0	0	0	
9	1	0	0	1	1	0	1	0	0	0	1	

Mapas de Karnaugh

A0A1\A2A3	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	0	0	0
10	0	1	0	0
M0	$A0'A3 + A1'A2'A3$			

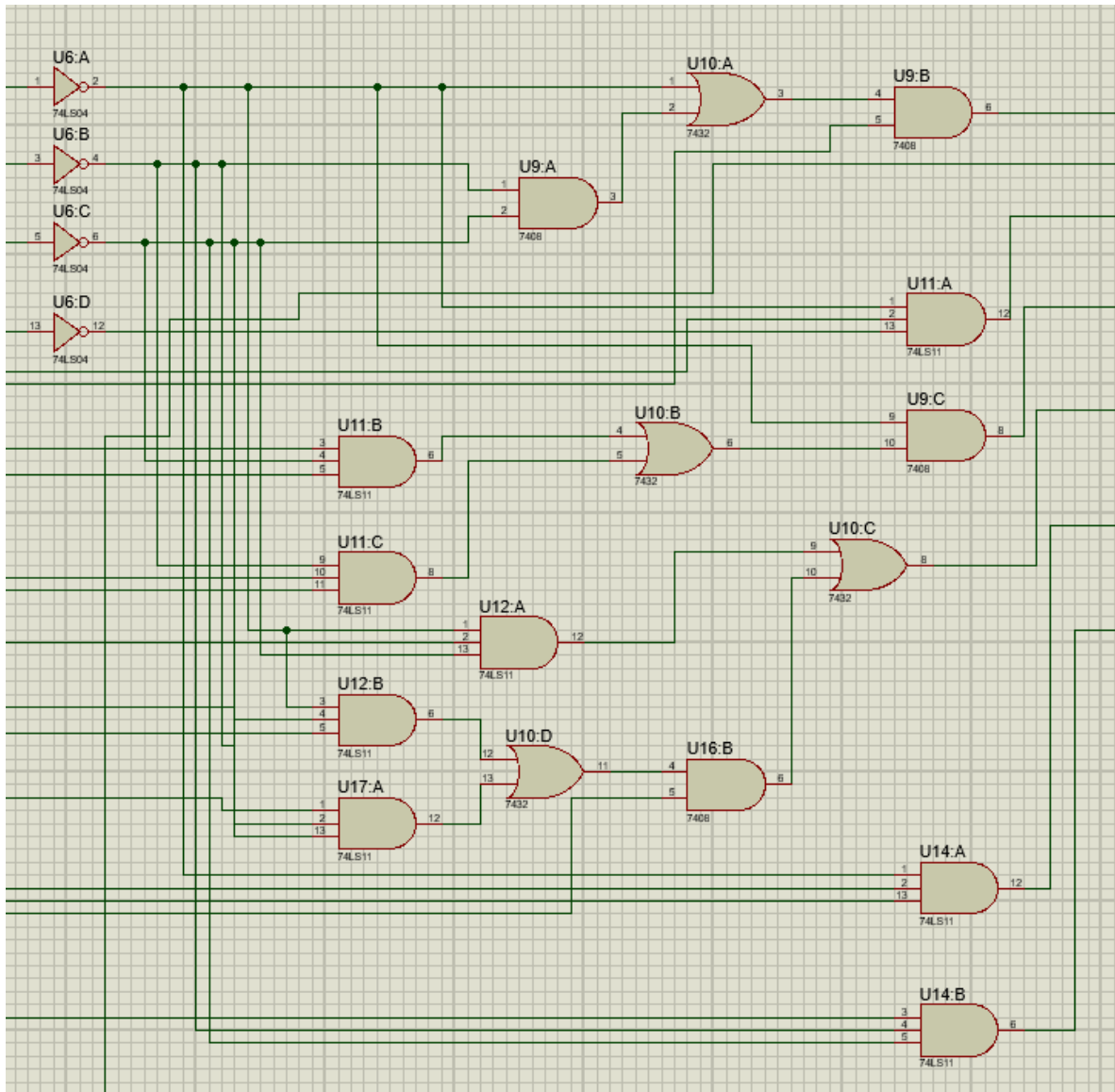
A0A1\A2A3	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	0	0	0	0
10	0	0	0	0
M2	$A0'A2A3'$			

A0A1\A2A3	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0
M3	$A0'(A1A2'A3+A1'A2A3)$			

A0A1\A2A3	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	0	0	0	0
10	0	1	0	0
M4	$A0'A1(A2'+A3)+A1A2'A3$			

A0A1\A2A3	00	01	11	10
00	0	0	0	0
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0
M5	$A0'A1A2$			

A0A1\A2A3	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	1	1	0	0
M6	A0A1'A2'			



Elevado al cubo

ENTRADAS					SALIDAS						
No	A0	A1	A2	A3	2 ⁶ M6	2 ⁵ M5	2 ⁴ M4	2 ³ M3	2 ² M2	2 ¹ M1	2 ⁰ M0
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	1	0	0	0
3	0	0	1	1	0	0	1	1	0	1	1
4	0	1	0	0	1	0	0	0	0	0	0

Mapas de Karnaugh

A0A1\A2A3	00	01	11	10		A0A1\A2A3	00	01	11	10
00	0	1	1	0		00	0	0	1	0
01	0	0	0	0		01	0	0	0	0
11	0	0	0	0		11	0	0	0	0
10	0	0	0	0		10	0	0	0	0
M0	A0'A1'A3					M1	A0'A1'A2A3			

A0A1\A2A3	0 0	0 1	1 1	1 0
0 0	0	0	0	0
0 1	0	0	0	0
1 1	0	0	0	0
1 0	0	0	0	0

M2	cable a tierra
----	----------------

A0A1\A2A3	0 0	0 1	1 1	1 0
0 0	0	0	0	0
0 1	0	0	0	0
1 1	0	0	0	0
1 0	0	0	0	0

M3	A0'A1'A2
----	----------

A0A1\A2A3	0 0	0 1	1 1	1 0
0 0	0	0	1	0
0 1	0	0	0	0
1 1	0	0	0	0
1 0	0	0	0	0

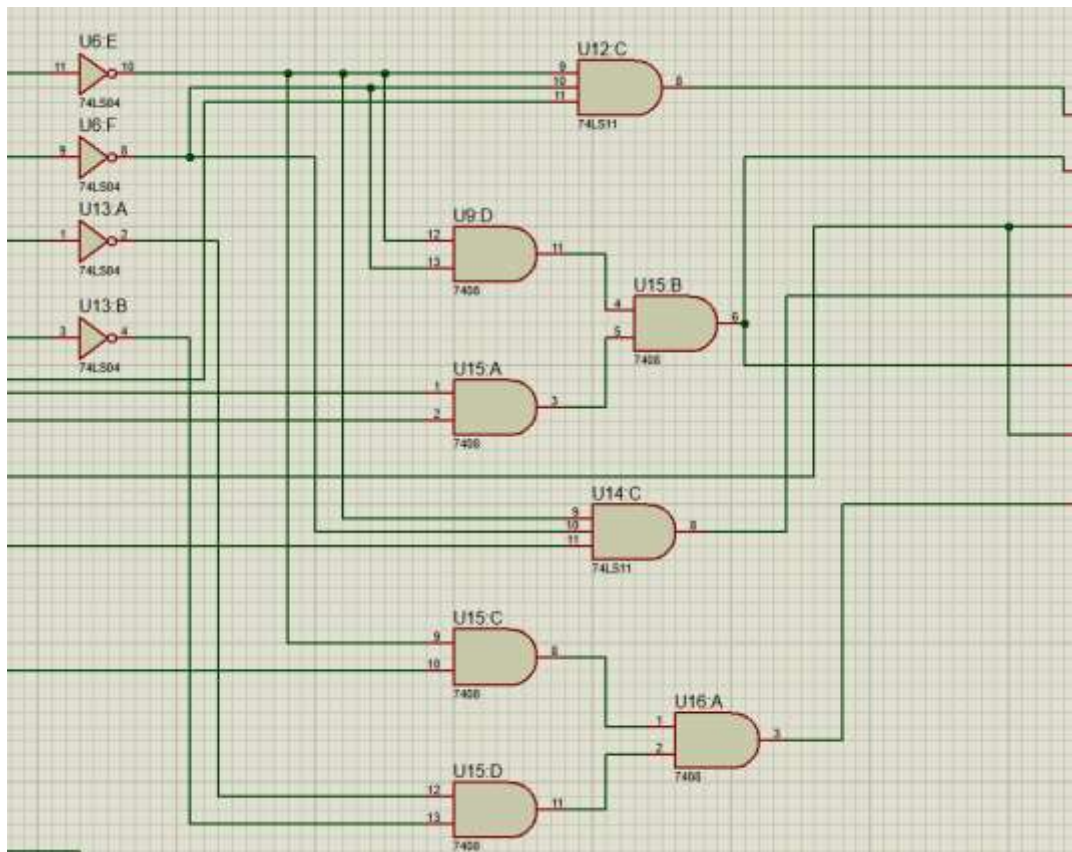
M4	A0'A1'A2A3
----	------------

A0A1\A2A3	0 0	0 1	1 1	1 0
0 0	0	0	0	0
0 1	0	0	0	0
1 1	0	0	0	0
1 0	0	0	0	0

M5	cable a tierra
----	----------------

A0A1\A2A3	0 0	0 1	1 1	1 0
0 0	0	0	0	0
0 1	1	0	0	0
1 1	0	0	0	0
1 0	0	0	0	0

M6	A0'A1A2'A3'
----	-------------



Compuerta AND:

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

Mapa de Karnaugh AND:

A\B	0	1
0	0	0
1	0	1
Función: AB		

Compuerta OR:

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

Mapa de Karnaugh OR:

A\B	0	1
0	0	1
1	1	1
Función: $A + B$		

Compuerta NOT:

A	Z
0	1
1	0

Compuerta XNOR:

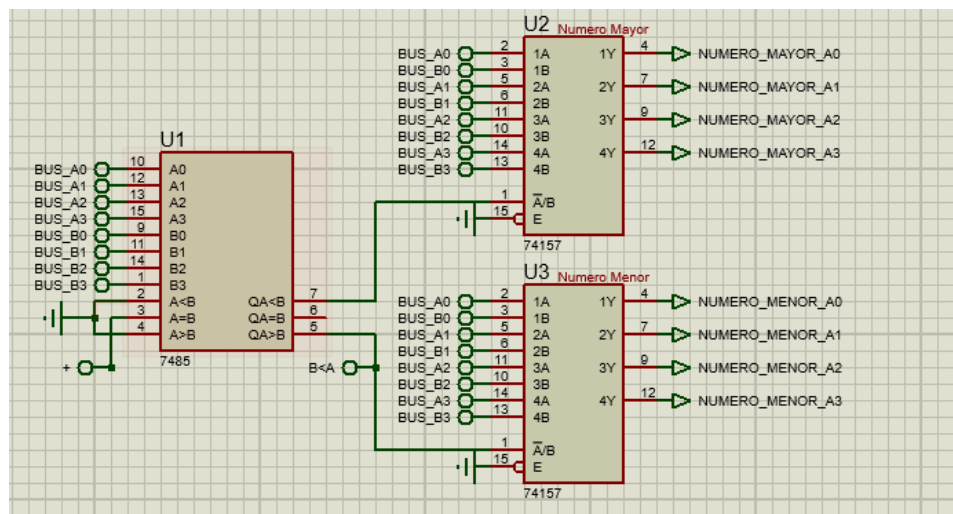
A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

Mapa de Karnaugh XNOR

A\B	0	1
0	1	0
1	0	1
Función: $A'B' + AB$		

Operaciones comparativas:

En esta sección se hace uso del circuito integrado 7485, que no es más que un comparador de 4 bits. Su implementación resulta sencilla, recibe como entrada 2 números de 4 bits (A y B) e indica si A es mayor que B, si A es igual a B o si A es menor que B. Tomando este resultado y haciéndolo pasar por dos multiplexores es posible mostrar la señal o bits de entrada del número que sea mayor o menor. Este arreglo quedaría de la siguiente manera:

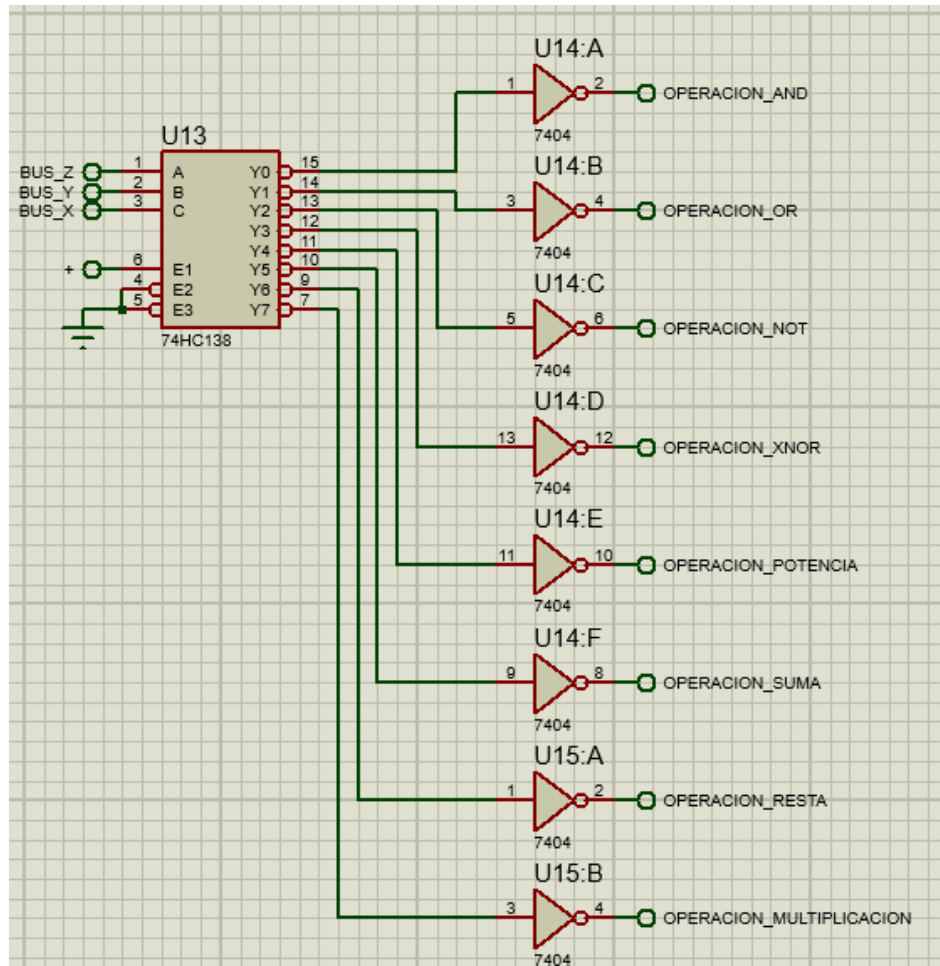


Controlador:

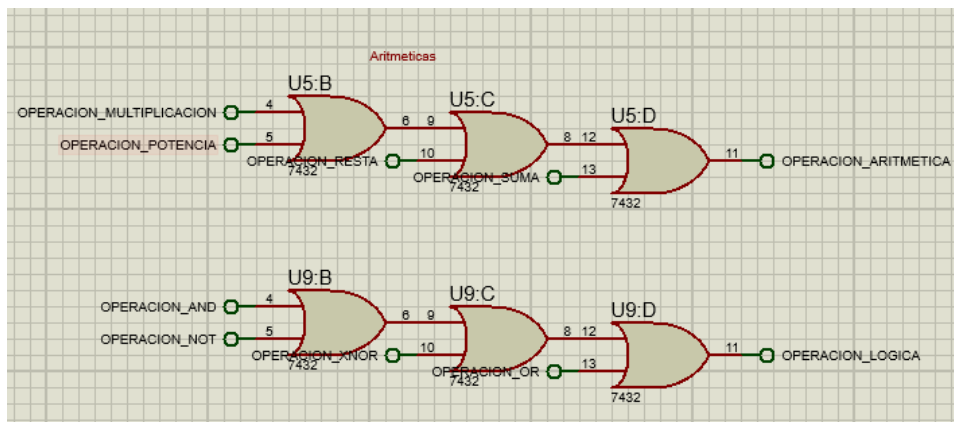
Para poder escoger entre 8 tipos diferentes de operaciones posibles es necesario convertir 3 bits de selección a 8 resultados. Se describe a continuación:

X	Y	Z	F
0	0	0	AND
0	0	1	OR
0	1	0	NOT
0	1	1	XNOR
1	0	0	Potencia(A^B)
1	0	1	Suma
1	1	0	Resta
1	1	1	Multiplicación

Para no requerir de un número excesivo de compuertas lógicas se utilizó un demultiplexor/decodificador, en este caso el circuito integrado 74138. Para su uso es necesario habilitar sus 3 entradas de enable, para que cuando corresponda una entrada al tipo de operación se envíe un 1 como señal para dicha operación y un 0 para el resto de operaciones. Es oportuno mencionar que este circuito integrado trabaja con lógica negativa a la salida, por lo que será necesario negar las salidas para obtener el funcionamiento adecuado. El arreglo quedaría de la siguiente manera:

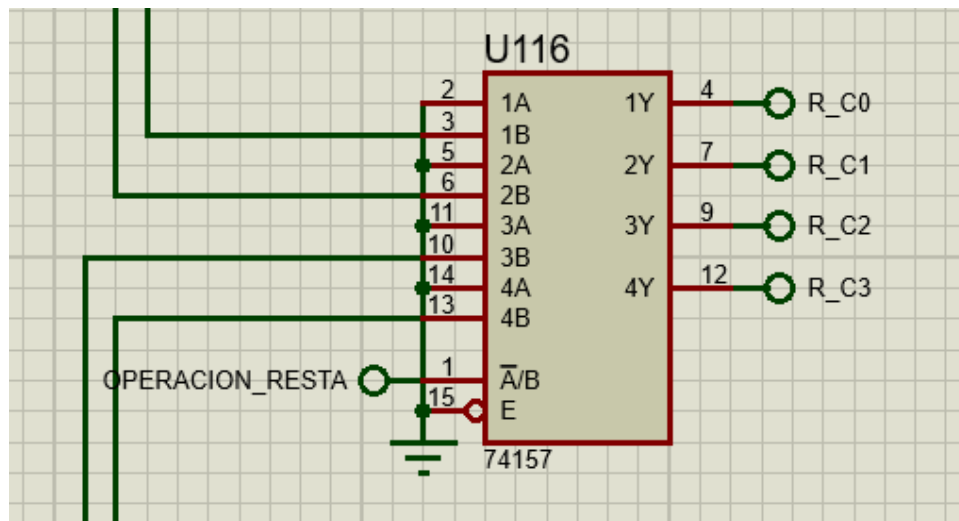


Haciendo uso de estas salidas se determinó el tipo operación que se seleccionó. Se propone este arreglo de compuertas OR.

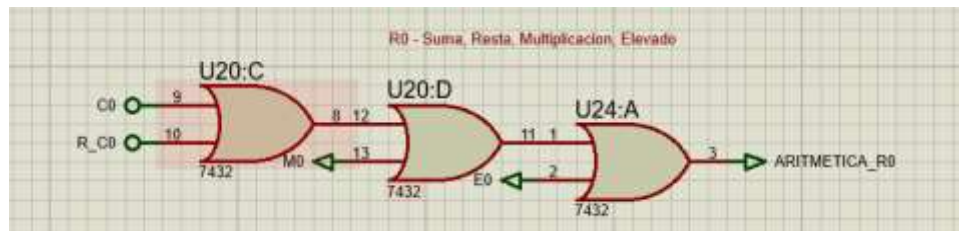


Unificación de resultados:

Para hacer efectivo el uso del controlador se hace uso de multiplexores que en dado caso se realice una operación distinta a la seleccionada envía en todas sus salidas 0 (elemento neutro de la suma). Esto se realiza con el objetivo de posteriormente sumar todas las salidas y que únicamente afecte al resultado final aquella operación seleccionada.



Multiplexor encargado de habilitar la resta.

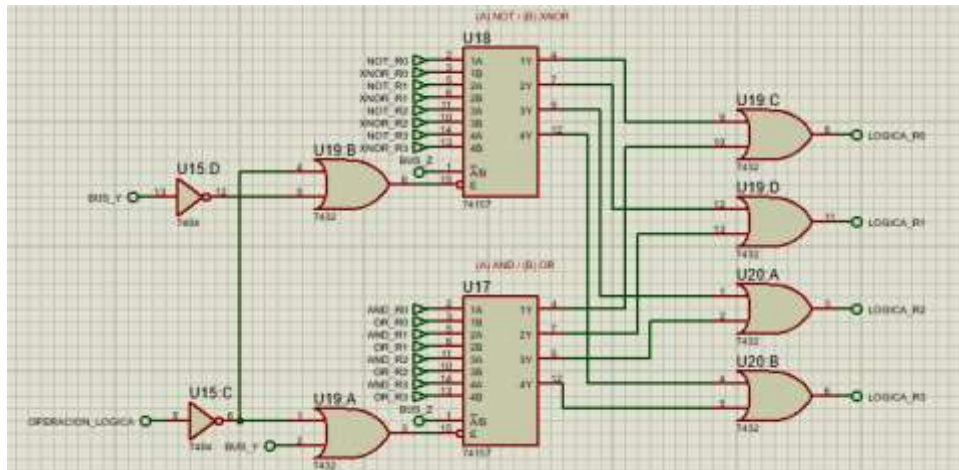


Suma del bit menos significativo de las operaciones aritmeticas.

Para unificar los resultados de las operaciones lógicas es posible reducir el uso de compuertas lógicas OR's puesto que el multiplexor utilizado es el 74157, este multiplexor convierte 2 buses de 4 bits a un solo bus de 4 bits. Se combinaron dos de estos multiplexores para obtener una entrada de 4 buses de 4 bits a un unico bus de 4 bits.

Para su correcto funcionamiento es necesario deshabilitar según corresponda uno de los multiplexores, para ello se utiliza en enable. El multiplexor funciona cuando

en su entrada es recibido un 0, por lo que se realizó una operación OR previa que únicamente resulta en 0 cuando ambas entradas sean 0, en este caso las entradas son si se está realizando una operación lógica y el bus de entrada Y, que representa específicamente el tipo de operación lógica.



Visualización en displays de 7 segmentos:

Luego de tener el resultado en código binario de las operaciones aritméticas, así como de las operaciones comparativas será necesario aplicar una transformación de la señal hacia código BCD y que este sea interpretado por un decodificador a 7 segmentos.

En el caso de las operaciones comparativas los números no superan los 4 bits, por lo que se puede comprobar si el número en cuestión es mayor o igual al número decimal 9 y si ese fuera el caso sumarle 6 para obtener el formato correcto. De esta forma es posible visualizar hasta el número 15.

Por otro lado, para los resultados de operaciones aritméticas se tendrán resultados no mayores a 81, por lo que es necesario convertir el código binario de 7 bits a BCD. Para obtener dicha transformación se sigue el siguiente procedimiento:

1. Se toma el número más significativo y se la desplaza hacia la izquierda
2. Se compara si dicho desplazamiento es Mayor o igual a 5
3. Si el número desplazado a la izquierda es mayor o igual a 5 entonces sumamos 3 (011) y se sigue desplazando

4. Se regresa al paso 1,2,3 hasta completar los bits necesarios para las Decenas y Centenas

Esto se realiza haciendo uso de compuertas lógicas para las comparaciones si dichos desplazamientos realizados son mayor o igual a 5 y con sumadores, que me permitan sumar 3 o 0, hasta obtener las unidades y decenas de forma que nos quede transformado a un número BCD, las unidades y decenas y poder usar el decodificador 7 Segmentos.

Para los últimos 3 bits del resultado se compara si su valor es mayor o igual a 5 con la siguiente tabla de verdad.

M6	M5	M4	C1						
0	0	0	0						
0	0	1	0						
0	1	0	0						
0	1	1	0						
1	0	0	0						
1	0	1	1						
1	1	0	1						
1	1	1	1						

M6 M5 \ M4	00	01	11	10
0	0	0	1	0
1	0	0	1	1

$$C1 = M6 M5 + M4 M6$$

$$C1 = M6 (M5 + M4)$$

Si en dado caso esta condición se cumpliera, se procede a sumar por medio del circuito integrado 7483. A la salida del circuito integrado se obtiene las salidas desplazadas, por lo que se itera el resultado en conjunto del siguiente bit de la entrada (F0), como se ejemplifica a continuación.

F3	F2	F1	F0	C2
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1
1	1	1	1	1

F1F0\F3F2	00	01	11	10
00	0	0	1	1
01	0	1	1	1
11	0	1	1	1
10	0	1	1	1
$C2 = F3 + F2F0 + F2F1$				
$C2 = F3 + F2(F0 + F1)$				

Esta función es utilizada nuevamente de manera iterativa hasta llegar al penúltimo bit menos significativo.

Para recuperar las unidades, se utilizaron las salidas del último sumador realizado en conjunto con bit más significativo.

Para obtener las decenas es necesario realizar una comprobación del 2do grupo de 4 bits aplicando las mismas reglas para determinar si es necesario sumarle 3. Para esta operación únicamente se utilizan los 3 primeros bits de ese grupo y se puede reutilizar la primera función propuesta para determinar si número de 3 bits es mayor a 5.

Link repositorio: https://github.com/AlvaroG13191704/PRACTICA_2_ORGA