Grupo 22:
Jorge Andrés Hiler Ricardo C.C 1152220165
Laura Daniela Erazo Santiago C.C 1036681364

Laboratorio 1, ALU de 16 bits.

En el presente informe se exponen las decisiones de diseño, tablas de verdad, mapas de simplificación, funciones lógicas minimizadas, esquemáticos de circuitos combinacionales, que se realizaron para la construcción de una ALU de 16 bits, que permite hacer las operaciones: AND, OR, NOR, Suma, Resta, SLT, SLR y SLL. Además de mostrar el proceso de construcción de la salida en displays 7 segmentos.

El componente principal es una unidad aritmético-lógica (ALU) de 16 bits, en el que ingresan como entradas A y B de 16 bits considerando operandos con signos, para ello se emplea una representación en complemento A2.

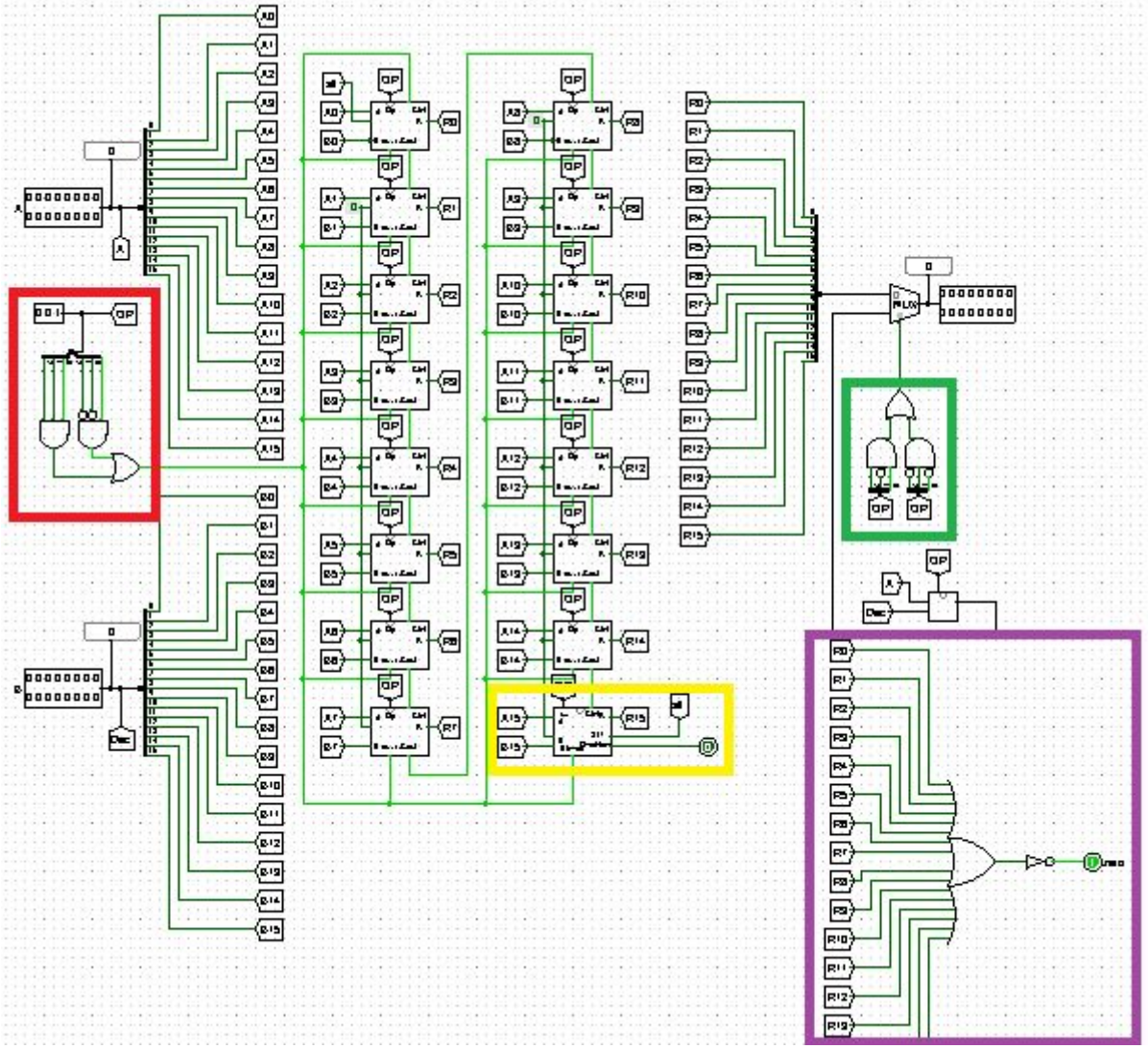


figura 1

Con el circuito encerrado por el recuadro verde en la figura 1, se toma la decisión en el multiplexor de cuál entrada habilitar, en caso de que la operación que se desee realizar sea SLR (010) o SLL (101) el circuito retorna 1 y habilita la entrada que realiza estas operaciones.

ALU de un bit.

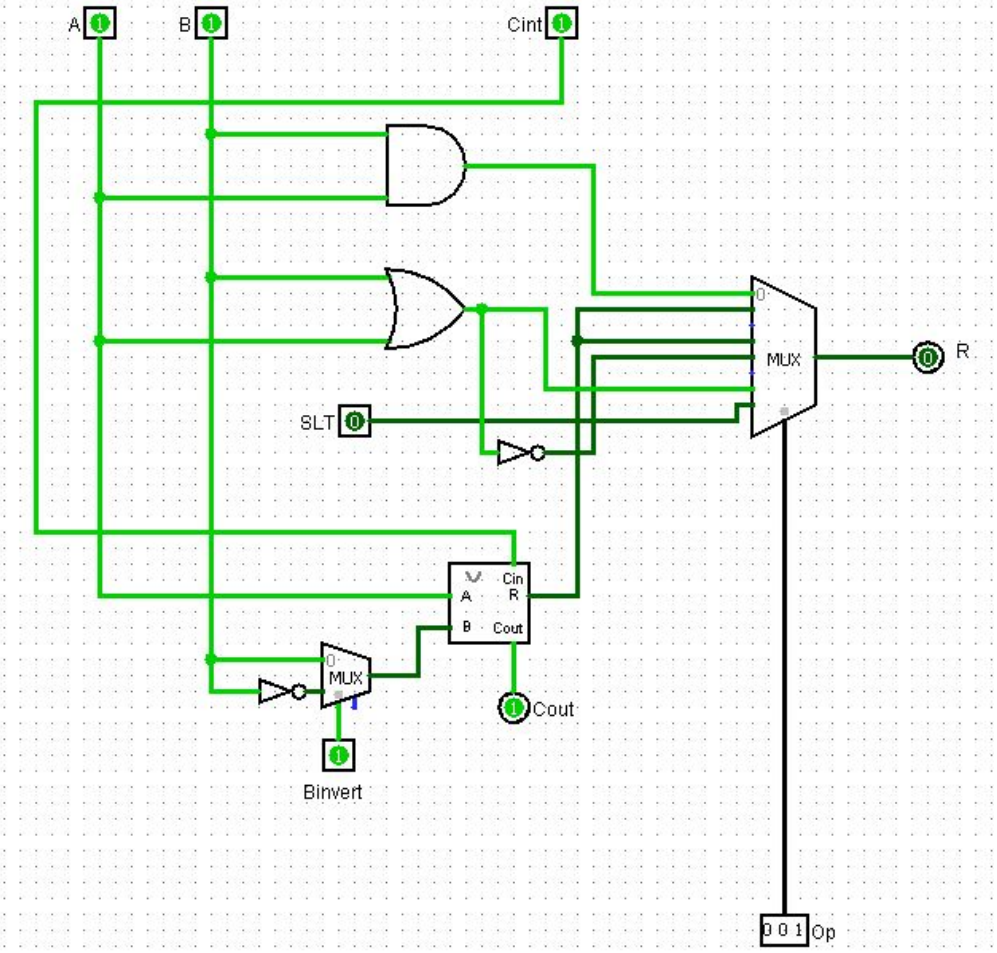
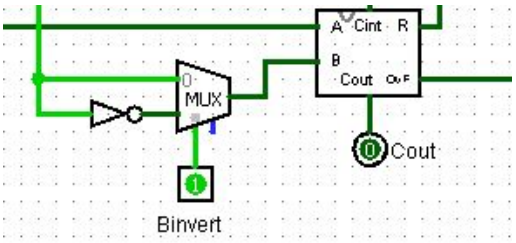


figura 2

Las funciones AND, OR, NOR, SUMA, RESTA se realizan bit a bit usando el ALU de un bit que es usado para realizar las funciones de los bits 0-14, el número 15 se hace con el ALU de un bit que contiene las mismas operaciones pero adicionalmente tiene otra salida para el overFlow. Para la función resta se utilizó la función suma, hallando el complemento A2. Para ello se tuvo en cuenta la siguiente consideración:

> *Si N es un número positivo, NC2 es su negativo en complemento a dos, y es igual al complemento a uno de N más uno: NC2 = ((2n – 1) – N) + 1 = 2n - N, en donde n es el número de bits de la representación.*

Para la operación Suma se sacan, la tabla de verdad y los mapas de Karnaugh para obtener las expresiones simplificadas y diseñar el circuito. Estos se muestran en la figura 12.

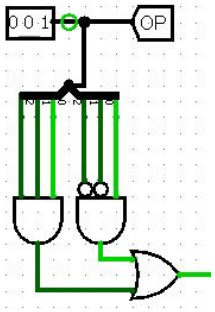


Se halla el complemento A1 del bit correspondiente a la entrada B y se pasa como entrada al multiplexor con el que se decidirá si se selecciona el valor de B ingresado o su complemento. El 1 que debe ser sumado al complemento A1 se pasa como acarreo de entrada.

Binvert corresponde a la selección de la operación, en caso de ser 0 realiza la suma de forma normal y en caso de ser 1 rea realiza el complemento A1.

En este circuito ALU de un bit. También se utiliza un multiplexor que selecciona la operación realizada en el ALU de un bit de acuerdo al código de operación asignado.

El circuito que se encuentra señalado con el recuadro rojo de la figura 1.1 se utiliza para obtener la salida de 1 que entrará como acarreo al primer ALU de un bit para realizar la operación de resta y obtener el complemento A2.



Se observa que para la operación suma que corresponde a la entrada 001 el circuito produce una salida de 1. Este mismo circuito será útil para hacer la función SLT, la cual se implementó realizando una resta, haciendo necesario realizar la entrada de 1 bit para el complemento A2.
Como se muestra en la figura ppp, para la suma del primer bit recibe como acarreo de entrada un 1 y se realiza el complemento de A2 de la entrada B dando un resultado correcto para las entradas 1 y 1 de A y B respectivamente.

ALU de un bit con overflow.

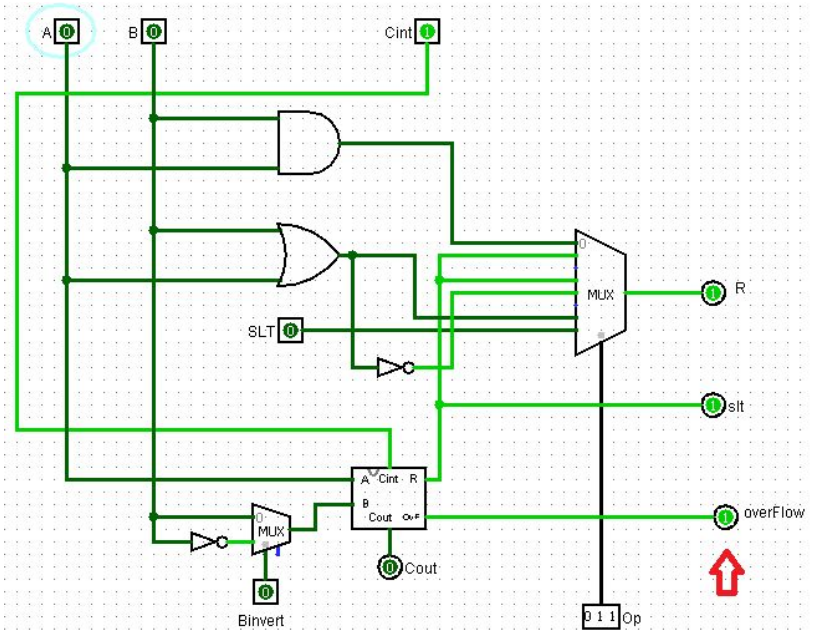
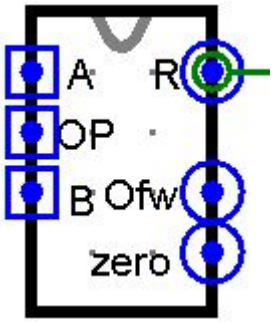


figura 3

Este ALU de un bit se realiza con las mismas operaciones del anterior pero usando el sumador de un bit del que se obtiene como salida el overflow. En la imagen se puede observar que la salida marcada en el overflow es correcta para las entradas A, B, Binvert abajo y Cint arriba.

La salida “slt” corresponde al bit más significativo. Esta salida es usada para realizar la función SLT.

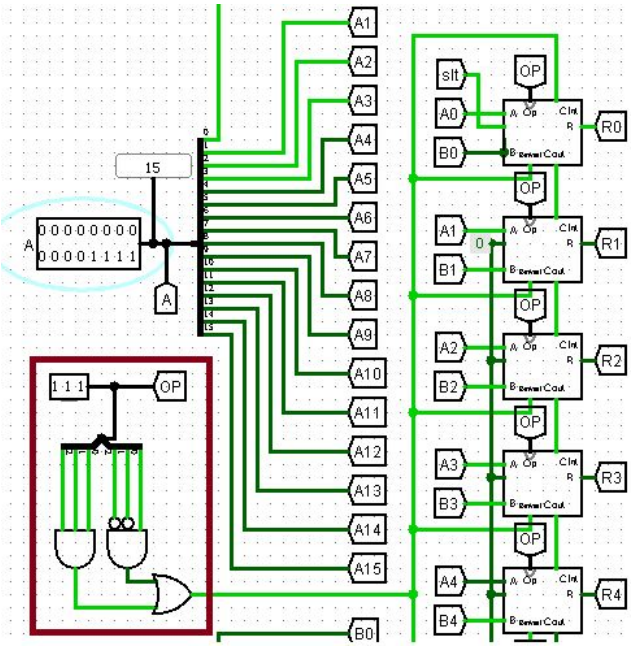
ALU de 16 bits.



El ALU de 16 bits tiene como entradas los números A y B en complemento en caso de ser negativo. OP para seleccionar la operación del ALU y como salidas a R, ofw (overFlow) y zero.

Una vez seleccionada la función, si esta corresponde a **SUMA, RESTA, AND, OR o NOR** se realizan bit a bit con el ALU de un bit simple y con overflow. Si la función seleccionada es SLT, SLR o SLL se opera directamente utilizando los operandos y resultados en 16 bits.

Para los bits de entradas se utilizó el elemento túnel para simplificar el diseño.



Se puede notar que para el primer ALU de un bit que en el dato de entrada SLT depende del túnel slt, a diferencia de los demás ALUs que reciben un 0 en esta entrada. Esto por la solución que se implementó para la función SLT.

Función SLT.

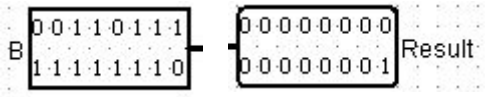
Al restar las entradas de 16 bits A y B, si A es menor que B el bit más significativo tomará valor de 1. Entonces el resultado de la función será 1 (0 en el caso contrario) para el primer bit y 0 para los otros 15. Tendiendo esto en cuenta:

Cuando se selecciona la entrada 111 correspondiente a SLT en el circuito marcada con el recuadro rojo da una salida 1, que será de entrada al Binvert (para realizar el complemento A1) y al Cint (Para sumar 1 al complemento A1) y así obtener el complemento A2, esto solo para el primer ALU, que también recibirá como entrada el bit etiquetado con “slt” proveniente del último ALU como el bit más significativo. De esta manera el primer bit del result quedará con la salida SLT que será cero o uno y las tomarán siempre el valor de 0.

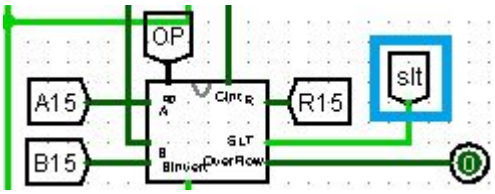
En conclusión, después de realizar la resta, el bit más significativo será el que da el resultado de la función SLT.

La prueba descrita y mostrada en la imagen anterior corresponde al valor de A mostrado y a el de B:

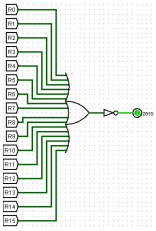
siendo A<B. . Al realizar el proceso para A>B se puede evidenciar que el túnel “slt” toma valor 0.



- **Acerca del último ALU de 1 bit con overflow**
  En la figura 1, el último bit marcado con el recuadro amarillo cumple un papel importante en la realización de la función SLT. La salida slt contiene el bit más significativo que es aprovechado cuando se realiza la resta.

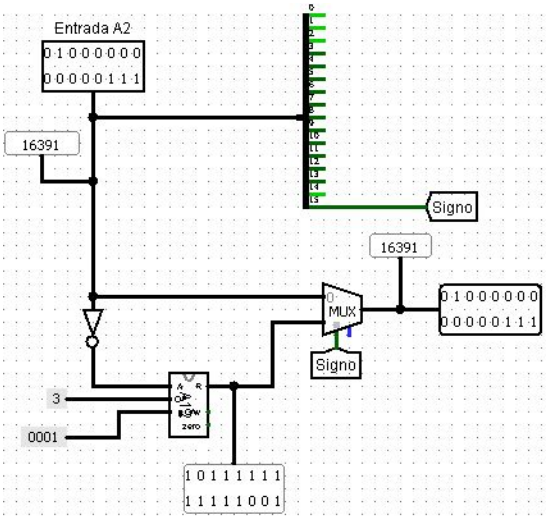


ZERO



En la figura 1, el recuadro morado contiene una compuerta OR y NOT equivalente a la compuerta NOR, recibe 16 entradas correspondientes a los result. Cuando todas las entradas sean cero la salida correspondiente será 1.

ComplementoA2.



Este circuito fue diseñado para que al momento de representar los dígitos decimales en los display de 7 segmentos, recibiera solo la magnitud del result.

Para ello se toma el bit más significativo para obtener el signo; si es negativo, el multiplexor toma la entrada de 16 bits a la que se le obtuve el complemento A2 usando el ALU de 16 bits, de lo contrario, si es positivo, lo pasa tal cual como lo recibe.

BCD a 7 segmentos.

Para los resultados de las operaciones de la ALU se utilizan 5 displays de 7 segmentos, en los cuales se muestra el resultado de la operación seleccionada, esto tomando el número binario y por medio de la división, se separa en decenas de mil, unidades de mil, centenas, decenas y unidades.

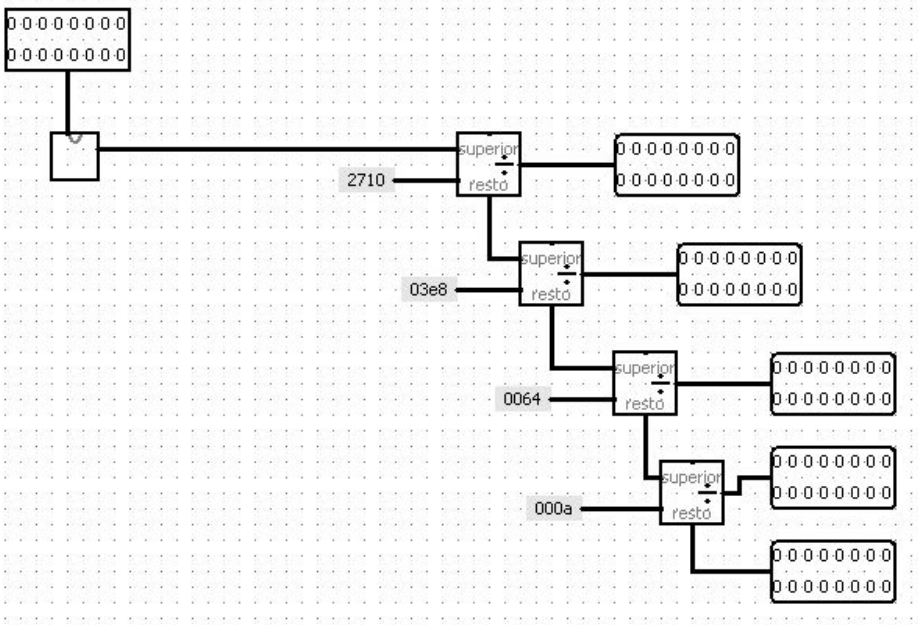


figura 4

Como se ve en la figura 4 el número se divide entre 10000, el residuo entre 1000, luego entre 100 y finalmente entre 10 para separarlo por cada unidad, el residuo final se toma como número unidad.

Para dibujar cada número en cada display se requirió hacer la tabla de verdad de la Tabla 1. Según el número recuperado y teniendo en cuenta la convención del display de 7 segmentos mostrada en la figura 5, se realizaron la tabla y los mapas de Karnaugh (figuras de la 7 a la 10) para simplificar las expresiones.

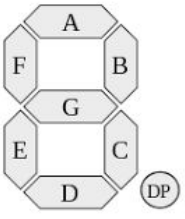


figura 5

Y con estas expresiones se formó el circuito de la figura 6.

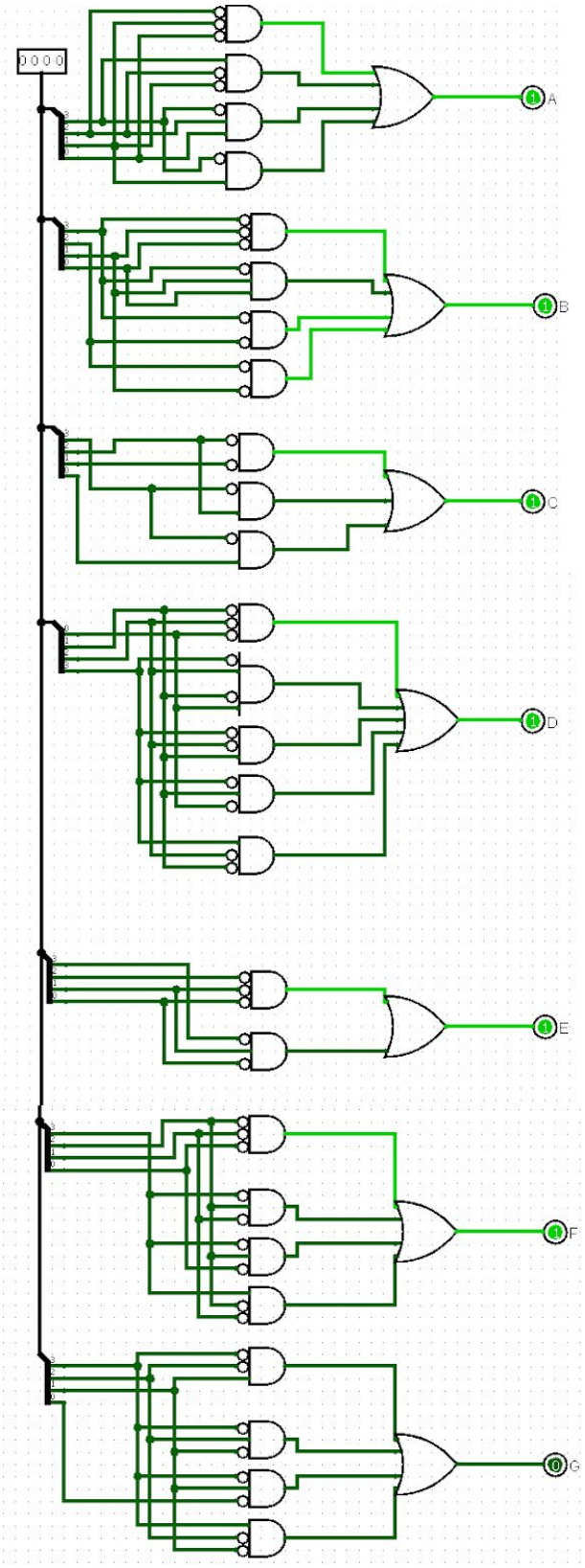


figura 6

Tabla de verdad.

A0	A1	A2	A3	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Tabla 1

Mapas de Karnaugh.

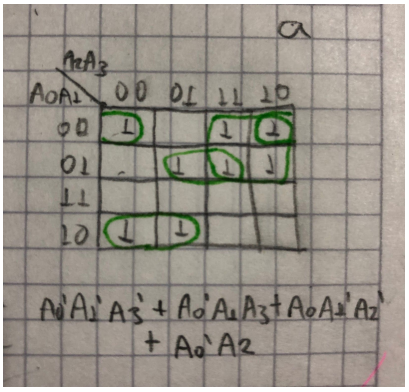


figura 7

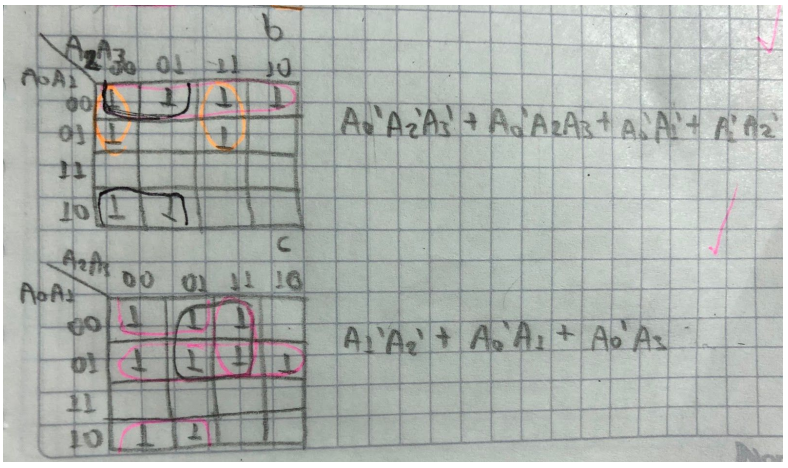


figura 8

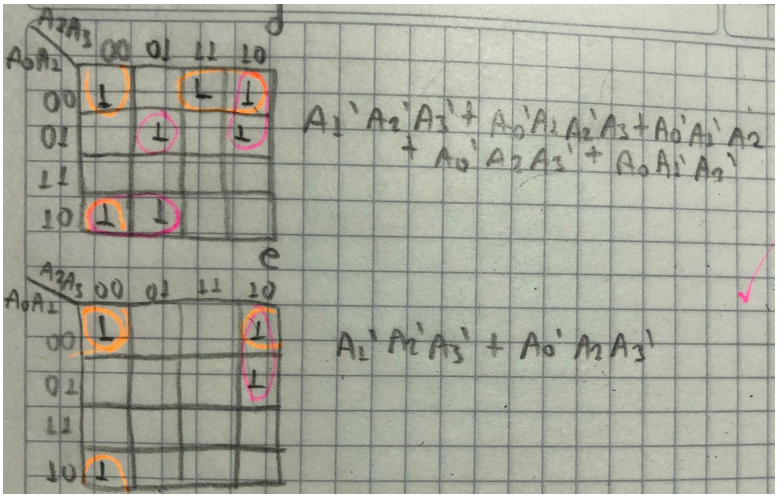


figura 9

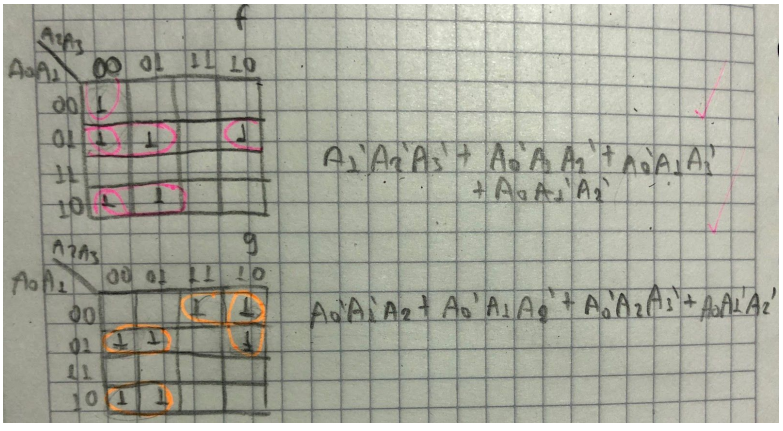


figura 10

Se hace un mapa por cada segmento del display y se hallan las expresiones simplificadas para formar el circuito. Finalmente en la figura 11 se muestra la conexión entre cada display con cada número por unidades del resultado.

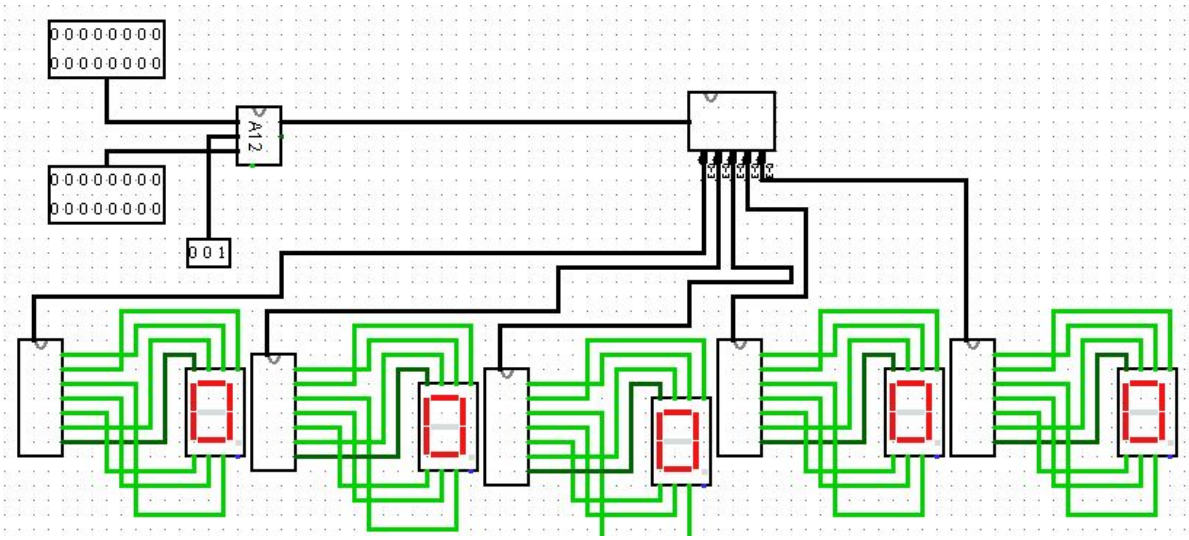


figura 11

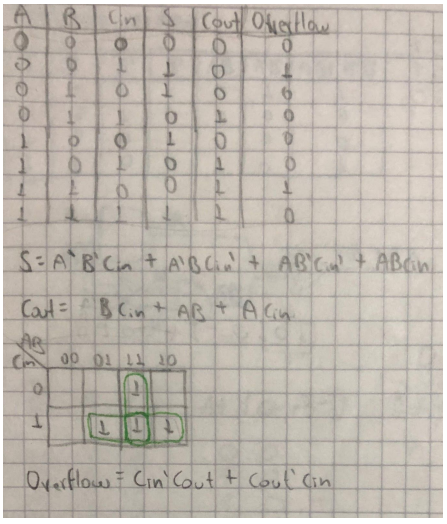


figura 12

CONCLUSIONES

La elaboración del ALU con las 8 operaciones llevó a la elaboración de un diseño de una estructura jerárquica a nivel de bloques, tomando las decisiones más convenientes para la elaboración del circuito.

La construcción de algunos circuitos implicó el uso de las tablas de verdad y manejo de expresiones booleanas para obtener una expresión más simple, esto requirió el uso de mapas de Karnaugh. Además del uso en diferentes ocasiones de multiplexores para seleccionar la salida deseada, ayudó a reforzar los conceptos aprendidos en clase.

Las limitaciones en cuanto a los componentes a usar implicó la elaboración de soluciones lo más simple posibles, que involucró el ejercicio de evaluar la mejor opción.