



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA

INFORME 2, ARQUITECTURA Y ORGANIZACIÓN DE  
COMPUTADORES.

# IMPLEMENTACIÓN LOGICA DE BOOL Y CIRCUITOS COMBINACIONALES PARA CONSTRUIR CIRCUITOS.

Gabriela Acuña, rol 201973504-7.  
Profesor Mauricio Solar.

6 de Junio, 2021

# Índice

<b>1</b>	<b>Resumen</b>	<b>1</b>
<b>2</b>	<b>Introducción</b>	<b>1</b>
<b>3</b>	<b>Desarrollo</b>	<b>2</b>
3.1	Circuito de $T$ . . . . .	3
3.2	Circuito de Suma . . . . .	3
3.3	Circuito de $f$ . . . . .	3
3.4	Circuito de $g$ . . . . .	4
3.5	Circuito para Display Hexadecimal . . . . .	5
3.6	Circuito en General . . . . .	7
<b>4</b>	<b>Resultados</b>	<b>8</b>
<b>5</b>	<b>Análisis</b>	<b>8</b>
<b>6</b>	<b>Conclusión</b>	<b>9</b>
<b>7</b>	<b>Anexo</b>	<b>10</b>

# 1 Resumen

Para el desarrollo de esta tarea se abordaron cada uno de los circuitos a realizar por separado, para luego juntarlos y revisar el circuito general. Para cada circuito se realizó un respectivo intento de minimización, con mapas de Karnaugh y Lógica de Bool.

El circuito logrado, realiza la suma completa entre dos números de 4 bits, o ejecutan los circuitos de las funciones  $f$  y  $g$  expuestas en el enunciado.

## 2 Introducción

El objetivo de esta tarea fue desarrollar lógicamente las funciones entregadas en el enunciado para luego crear un circuito con la mínima cantidad posible de compuertas lógicas. En algunos casos, se realizaron los mapas de Karnaugh y álgebra de Bool en los resultados, para llegar a la expresión mínima solo con eso. En otros casos, una vez que existió un entendimiento completo de la función del circuito se pudo reducir a circuitos ya estudiados que son mínimos. Para desarrollar esta tarea se utilizaron Mapas de Karnaugh para reducir las funciones entregadas, en base a sus tablas de verdad. Un Mapa de Karnaugh es un método gráfico para simplificar ecuaciones booleanas de hasta 4 variables, con solo un bit de salida <sup>1</sup>. Al realizar la representación gráfica en una tabla como la que se puede ver en la *figura 17* se agrupan los 1 siguiendo las siguientes reglas:

- Usar la menor cantidad de círculos posibles para cubrir todos los 1, mientras estos son lo mas grandes posibles y solo contienen unos en su interior.
- Cada circulo debe abarcar un bloque rectangular que es una potencia de 2 en ambas direcciones.
- Un circulo puede dar la vuelta a los bordes del mapa.
- Un 1 puede pertenecer a varios grupos si con esto se minimiza la cantidad de estos.

Por otro lado, se utilizaron técnicas de minimización a través de axiomas del álgebra de bool. Para factorizar las expresiones multiplicadas, y así poder realizar circuitos multinivel cuando fue posible. Los Axiomas más utilizados fueron, las leyes de Morgan (*ecuación 1*, distribución (*ecuación 2*) y combinación (*ecuación 3*). Estos se combinaron entre si y con otros para desarrollar las expresiones obtenidas de los mapas de Karnaugh.

$$\overline{B_0 \cdot B_1 \cdot B_2 \dots} = \overline{B_0} \cdot \overline{B_1} \cdot \overline{B_2} \dots \quad (1)$$

$$(B \cdot C) + (B \cdot D) = B \cdot (C + D) \quad (2)$$

$$(B \cdot C) + (B \cdot \overline{C}) = B \quad (3)$$

El circuito previamente conocido que se utilizó para la implementación fue el de Suma Completa (Full Adder), al que se puede llegar desarrollando la ecuacion resultante del mapa de

---

<sup>1</sup>Digital Design and Computer Architecture ARM Edition, Harris & Harris

Karnaugh de la tabla de verdad de la suma completa, como se puede ver en la *tabla 2*. Con las que se llega a las siguientes ecuaciones:

$$Sum = \overline{cin} \cdot \overline{B} \cdot A + \overline{cin} \cdot B \cdot \overline{A} + cin \cdot \overline{B} \cdot \overline{A} + cin \cdot B \cdot A \quad (4)$$

$$cout = B \cdot A + cin \cdot A + cin \cdot B \quad (5)$$

De estas se puede notar que la ecuación de la suma es idéntica a la del *xor* luego de realizar su mapa de Karnaugh (*figura 18*).

$$y = \overline{a} \cdot \overline{b} \cdot c + \overline{a} \cdot b \cdot \overline{c} + a \cdot \overline{b} \cdot \overline{c} + a \cdot b \cdot c \quad (6)$$

Por lo tanto la ecuación para la suma entre dos elementos de un bit y un carry queda de la siguiente forma:

$$Sum = cin \oplus A \oplus B \quad (7)$$

La ecuación de *cout* es considerablemente concisa así que se implementó literalmente con compuertas para crear el full adder que se ve en la *figura 1*. Igualmente se realizó el mapa de Karnaugh para esta, como se puede ver en la *figura 19*.

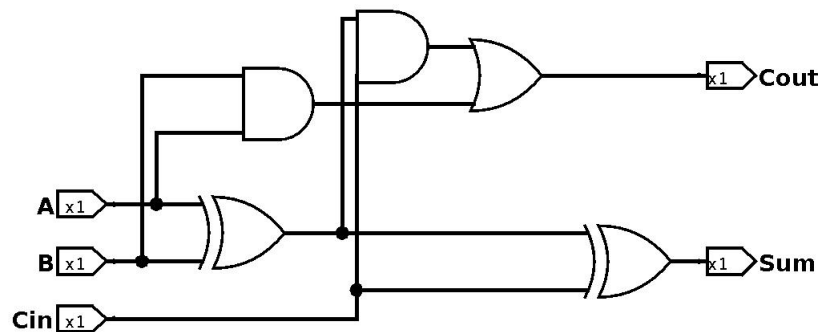


Figure 1: Circuito Full Adder

De forma análoga se utilizó el circuito para la suma sin Carry, el cual es simplemente realizar la operación *xor* entre sus componentes.

Otro concepto crucial para el desarrollo de esta tarea fueron los multiplexores. Estos en base a un bit *selector* deciden qué input utilizar, para transferir a la salida, que gráficamente, generalmente se encuentra a la derecha del símbolo del multiplexor.

Respecto a la transformación de binario a hexadecimal, ya que el rango del número binario de 4 bits es de  $0_{10}$  a  $15_{10}$  se puede expresar con solo un dígito en hexadecimal, por esta razón se realizó una transformación directa en el display led de 7 segmentos.

Por último, se utilizó la herramienta de análisis combinacional de Logisim para traspasar los mapas de Karnaugh a latex y traspasar las ecuaciones, que no se podían minimizar con álgebra de bool, a circuitos.

### 3 Desarrollo

Para realizar lo solicitado para esta tarea, se trabajó por partes, en un subcircuito a la vez, para por último realizar un análisis general del circuito resultante.

### 3.1 Circuito de T

Ya que  $T$  fue presentado en el enunciado como un selector, que si es 0 se realiza una cosa y si es 1 otra, se tomo la decisión de implementarlo utilizando un multiplexor. El circuito en si se puede ver en la *figura 2*. En este se ve un pin para suma, a modo de ejemplo ya que en la implementación en lugar del pin se encuentra el circuito implementado para la suma. Y en el pin llamado funciones, se encontraría el circuito implementado para la función que corresponda según al output. Por la forma de implementación, esto se repitió 4 veces, una para cada bit de salida binaria.

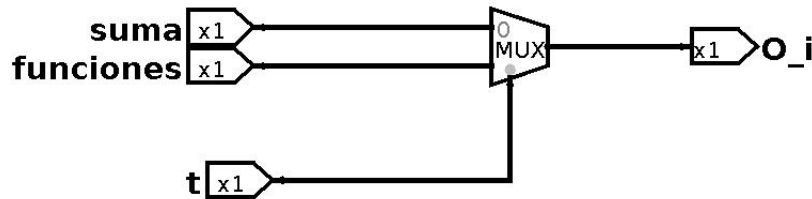


Figure 2: Circuito para T

### 3.2 Circuito de Suma

Para realizar este circuito, se intento iniciar generando una tabla de verdad en base a los resultados de la suma binaria sin carry de salida entre dos números de 4 bits. Pero al ser 256 resultados posibles, se decidió iniciar calculando una ecuación de bool para cada bit a sumar. Las ecuaciones desarrolladas fueron las siguientes,

$$O_0 = w_A \oplus w_B \oplus ((x_A \oplus x_B) \cdot ((y_A \oplus y_B) \cdot z_A \cdot z_B + y_B \cdot y_A) + x_A \cdot x_B) \quad (8)$$

$$O_1 = x_A \oplus x_B \oplus ((y_A \oplus y_B) \cdot z_A \cdot z_B + y_B \cdot y_A) \quad (9)$$

$$O_2 = y_A \oplus y_B \oplus (z_A \cdot z_B) \quad (10)$$

$$O_3 = z_A \oplus z_B \quad (11)$$

Se puede notar que aumentan considerablemente en dificultad a medida que se acercan al bit mas importante, por lo que se tomo la decisión de utilizar la implementación descrita en la introducción, de *full adder*. A esta se le aplicaron ciertos ajustes para que encajara con lo necesitado para la implementación. En primer lugar, para calcular el resultado que le corresponde al bit menos importante ( $O_3$ ) no se necesita el carry de la suma anterior (*cin* en la *figura 1*) y la suma a realizar sería entre  $z_A$  y  $z_B$ . El carry de salida de la suma del bit menos importante es utilizado en la suma del siguiente bit ( $O_2$ ) y así sucesivamente hasta llegar al bit mas importante ( $O_0$ ).

### 3.3 Circuito de $f$

En este caso, se empezó realizando la tabla de verdad de la ecuación entregada (*tabla 3*), y su mapa de Karnaugh asociado. De este se dedujo que la ecuación no se podía reducir más. Pero

$x$	$y$	$z$	$f(x,y,z)$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Figure 3: Tabla de verdad de la función  $f$ .

		$y, z$			
		$x$	00	01	11
$f$	0	0	1	0	1
	1	1	0	1	0

Figure 4: Mapa de Karnaugh para la ecuación  $f(x,y,z)$ .

luego de analizar la tabla de verdad, se puede notar que lo que realiza la función  $f$  es la suma binaria incompleta para tres números de 1 bit.

$$f(x,y,z) = \bar{x} \cdot \bar{y} \cdot z + \bar{x} \cdot y \cdot \bar{z} + x \cdot \bar{y} \cdot \bar{z} + x \cdot y \cdot z \quad (12)$$

Luego de esta realización, se puede desarrollar la ecuación encontrada con el Mapa de Karnaugh (figura 4, y simplificarla utilizando la definición del operador *xor* en base a *and* y *or*, y la ley de morgan ecuación 1. Luego del desarrollo se llega a la ecuación 11 y se paso a puertas lógicas como se puede ver en la figura

$$A \text{ xor } B = A \oplus B = (A + B) \cdot (\bar{A} + \bar{B}) = \bar{A}B + A\bar{B} \quad (13)$$

$$f(x,y,z) = (x \oplus y) \oplus z \quad (14)$$

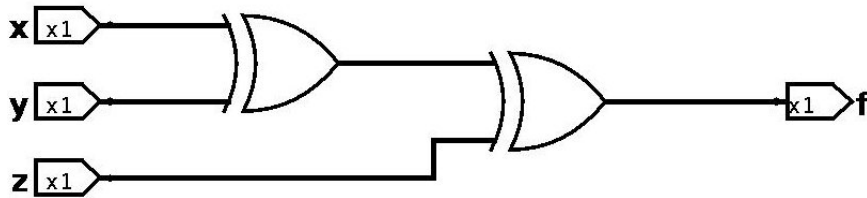


Figure 5: Implementación función  $f(x,y,z)$

### 3.4 Circuito de $g$

En este caso, muy parecido a la implementación de  $T$ , se presenta la función  $g(w,x,y,z)$  como una composición de la función  $f(x,y,z)$  y su negación, utilizando a  $w$  como su selector. Por esta razón se tomo la decisión de utilizar un multiplexor con un selector de un bit,  $w$ . En la posición 0 del multiplexor se encuentra el resultado de la función  $f(x,y,z)$  y en la posición 1 la negación de la anterior. La salida va a su respectivo bit objetivo, ya sea  $O_1$  u  $O_3$ . Debido a estas dos posibilidades de salida, se utilizaron dos multiplexores, uno para  $O_1$  y otro para  $O_3$ .

Cabe notar que en la figura 6 'fx' es el circuito para  $f(x,y,z)$ , figura 5, que se encapsulo ya que no es relevante su forma para la implementación de  $g$ , puesto que se utilizó el resultado de salida de  $f(x,y,z)$  para esta.

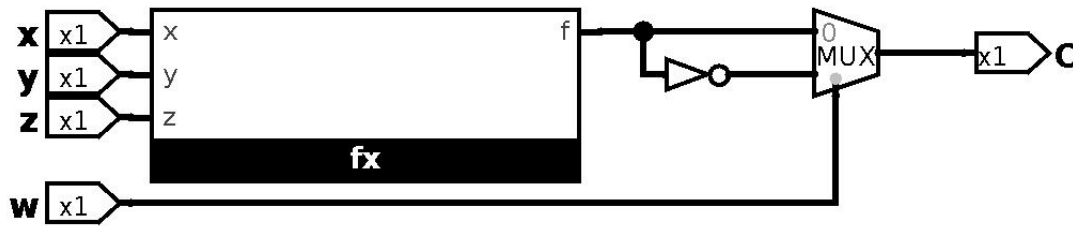
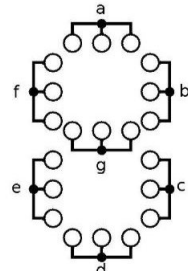


Figure 6: Circuito para  $g(w,x,y,z)$

### 3.5 Circuito para Display Hexadecimal

Para realizar este circuito, en primer lugar, en base al tipo de display solicitado se diseñaron las salidas de la transformación de binario a hexadecimal, agrupando los leds de salida de a 3, siguiendo la *figura 7*. Luego de realizar esta guía, se creo la tabla de verdad para transformar los numero binarios de 4 bits a números hexadecimales de un bit. Para crear esta tabla de verdad se reviso individualmente como se expresaría cada dígito hexadecimal en el display y se puso un 1 en los segmentos que era necesario encender.



$o0$	$o1$	$o2$	$o3$	$a$	$b$	$c$	$d$	$e$	$f$	$g$
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	1	1	1	0	1	1	1
1	0	1	1	0	0	1	1	1	1	1
1	1	0	0	1	0	0	1	1	1	0
1	1	0	1	0	1	1	1	1	0	1
1	1	1	0	1	0	0	1	1	1	1
1	1	1	1	1	0	0	0	1	1	1

Figure 7: Display de 7 segmentos led y tabla de transformación entre binario y hexadecimal

Ya con los 7 outputs posibles, se realizaron 7 mapas de Karnaugh e intentos de minimización con álgebra de bool. Pese a que se lograban minimizaciones con álgebra de bool y mapas de karnaugh, estas no disminuían el circuito en más de una compuerta. Por esta razón se tomo la decisión de utilizar el análisis combinacional provisto por *logisim*. De todas formas en las figuras 8 a 14 se pueden ver las minimizaciones realizadas. El axioma mas utilizado fue la ley de morgan y la definicion de *xor* en base a *and* y *or*.

*a*

		$o_2, o_3$			
		00	01	11	10
$o_0, o_1$	00	1	0	1	1
	01	0	1	1	1
	11	1	0	1	1
	10	1	1	0	1

$$a = \overline{o_1} \cdot \overline{o_3} + \overline{o_0} \cdot o_2 + \overline{o_0} \cdot o_1 \cdot o_3 + o_1 \cdot o_2 + o_0 \cdot \overline{o_1} \cdot \overline{o_2} + o_0 \cdot \overline{o_3} \quad (15)$$

*b*

		$o_2, o_3$			
		00	01	11	10
$o_0, o_1$	00	1	1	1	1
	01	1	0	1	0
	11	0	1	0	0
	10	1	1	0	1

$$b = \overline{o_0} \cdot \overline{o_1} + \overline{o_0}(o_2 \oplus o_3) + \overline{o_1} \cdot \overline{o_3} + o_0 \cdot \overline{o_2} \cdot o_3 \quad (16)$$

Figure 8: Mapa de Karnaugh y ecuación minimizada para el segmento *a*.

Figure 9: Mapa de Karnaugh y ecuación minimizada para el segmento *b*.

En la *figura 8*, se ve el análisis realizado para el segmento *a*, pese a que se intentaron diferentes formas de minimización la mejor lograda fue a través del mapa de Karnaugh. Respecto a *b*, en este caso si se logró la minimización, pero solo se disminuye en una compuerta lógica respecto a la minimización por mapa de Karnaugh.

*c*

		$o_2, o_3$			
		00	01	11	10
$o_0, o_1$	00	1	1	1	0
	01	1	1	1	1
	11	0	1	0	0
	10	1	1	1	1

$$c = \overline{o_0} \cdot \overline{o_2} + \overline{o_0} \cdot o_3 + \overline{o_2} \cdot o_3 + (o_0 \oplus o_1) \quad (17)$$

*d*

		$o_2, o_3$			
		00	01	11	10
$o_0, o_1$	00	1	0	1	1
	01	0	1	0	1
	11	1	1	0	1
	10	1	1	1	0

$$d = \overline{o_0} \cdot \overline{o_1} \cdot \overline{o_3} + o_2(o_1 \oplus o_3) + o_0 \cdot \overline{o_2} \quad (18)$$

Figure 10: Mapa de Karnaugh y ecuación minimizada para el segmento *c*.

Figure 11: Mapa de Karnaugh y ecuación minimizada para el segmento *d*.

En *c* se pudo realizar una minimización con transformación a *xor*, esto redujo 3 compuertas a 1, pero el resto de la ecuación se dejó como quedó desde el mapa de Karnaugh. En *d* también se logró una minimización con un procedimiento u resultados parecidos a los de *c*.



$e$		$o_2, o_3$			
		$o_0, o_1$	00	01	11
$o_0, o_1$	00	1	0	0	1
	01	0	0	0	1
	11	1	1	1	1
	10	1	0	1	1

$$e = \overline{o_3}(\overline{o_1} + o_2) + o_0(o_1 + o_2) \quad (19)$$

Figure 12: Mapa de Karnaugh y ecuacion minimizada para el segmento  $e$ .

$f$		$o_2, o_3$			
		$o_0, o_1$	00	01	11
00	1	0	0	0	
01	1	1	0	1	
11	1	0	1	1	
10	1	1	1	1	

$$f = \overline{o_2} \cdot \overline{o_3} + \overline{o_0} \cdot o_1 \cdot \overline{o_2} + o_1 \cdot \overline{o_3} + o_0 \cdot \overline{o_1} + o_0 \cdot o_2 \quad (20)$$

Figure 13: Mapa de Karnaugh y ecuacion minimizada para el segmento  $f$ .

En cuanto a la minimizacion de  $e$ , lo logado utiliza la misma cantidad de compuertas que la solucion del mapa de Karnaugh. Y en  $f$  se utilizó la ecuación derivada del Mapa de Karnaugh.

$g$		$o_2, o_3$			
		$o_0, o_1$	00	01	11
00	0	0	1	1	
01	1	1	0	1	
11	0	1	1	1	
10	1	1	1	1	

$$g = \overline{o_1} \cdot o_2 + o_2 \cdot \overline{o_3} + \overline{o_0} \cdot o_1 \cdot \overline{o_2} + o_0 \cdot \overline{o_1} + o_0 \cdot o_3 \quad (21)$$

Figure 14: Mapa de Karnaugh y ecuacion minimizada para el segmento  $g$ .

Por último, en  $g$  se utilizo la minimización obtenida del mapa de Karnaugh. Se puede ver el circuito resultante completo en el link presente en el anexo en la *figura 23*.

### 3.6 Circuito en General

Debido a la cantidad de inputs y outputs del circuito completo, no se pueden realizar mapas de Karnaugh ni tablas de verdad, ya que existen  $2^7$  posibles input. Por lo que se asume que con las minimizaciones realizadas, el circuito ya es mínimo. Todo lo mencionado anteriormente se conecta con sus respectivos inputs y outputs. Se puede ver el circuito completo en la *figura 15*, con el subcircuito de transformación de binario a led resumido para que sea visible, este se puede ver completo en el link en la *figura 23*.

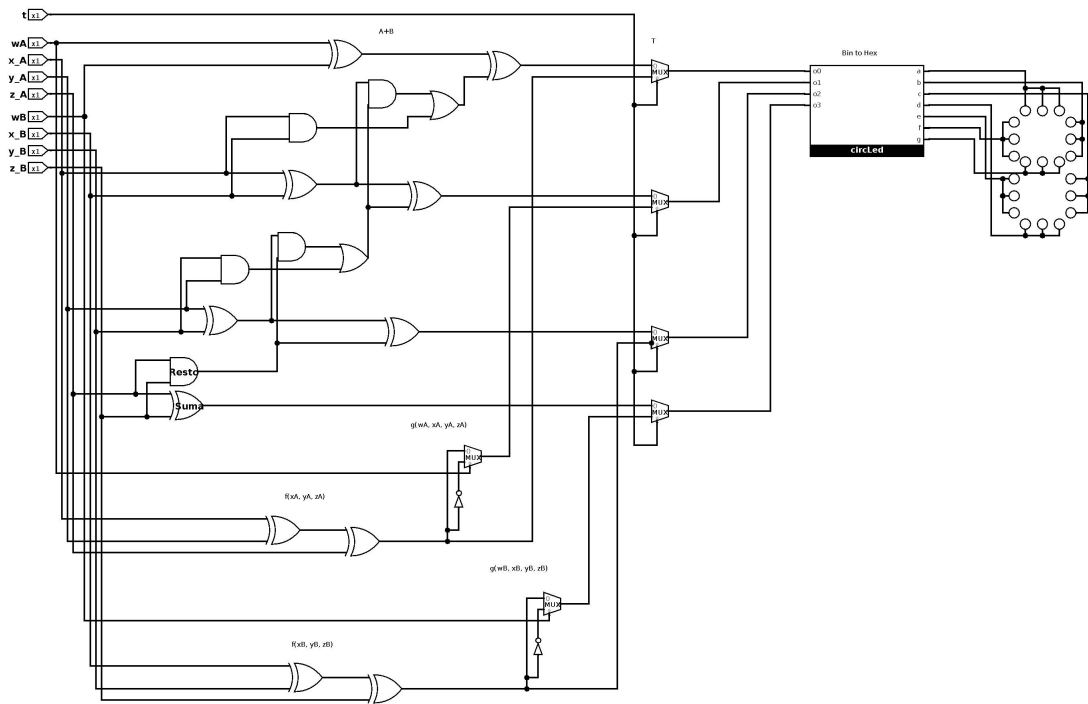


Figure 15: Circuito completo con subcircuito para leds resumidos.

## 4 Resultados

Se comprobaron todos los ejemplos entregados en el enunciado. Aun así en la *tabla 1* se pueden ver resultados de otros ejemplos. Se incluye una captura de pantalla del circuito en sí funcionando para mayor claridad en la *figura 16*.

T A B	Bin	Hex
0 1010 1100	0110	6
1 1010 1100	1010	A
0 1110 1101	1011	b
0 1110 1101	0101	5

Table 1: Resultados del circuito.

## 5 Análisis

Se obtuvieron todos los resultados esperados y estos se comprueban lógicamente siguiendo las respectivas funciones a aplicar según los bits marcados previamente como selectores ( $T$ ,  $w_A$  y  $w_B$ ).

Ya que se siguieron prolijamente todas las aplicaciones de mapas de Karnaugh y Álgebra de Bool, los resultados experimentales coinciden en todos los casos con los teóricos. Considerando a los teóricos como los ejecutados siguiendo las funciones entregadas en el enunciado y los experimentales los retornados por la implementación.

La complicación mas grande a lo largo de esta tarea fue la minimización del circuito para transformar de binario a hexadecimal en display de 7 segmentos. Ya que los mapas de Kar-

naugh retornaron expresiones relativamente extensas y el desarrollo algebraico no disminuyo la cantidad de compuertas.

## 6 Conclusión

La tarea se completo satisfactoriamente para todos los subcircuitos. Estos se minimizaron y unieron para formar el circuito completo que se ve en la *figura 15*.

Los usos de circuitos combinacionales en general, son bastante amplios y se utilizan en muchísimos campos. Por ejemplo existen circuitos de transmisión de datos, de conversión de códigos, entre otros <sup>2</sup>. Respecto a los Mapas de Karnaugh, se utilizan para minimizar gráficamente expresiones booleanas. Y los circuitos combinacionales junto con los Mapas de Karnaugh, realizan una dupla bastante útil, presto que las implementaciones físicas de los sistemas combinacionales tienen un costo económico y temporal asociado, y con los mapas de Karnaugh se pueden encontrar las versiones minimizadas de los circuitos para así disminuir costos.

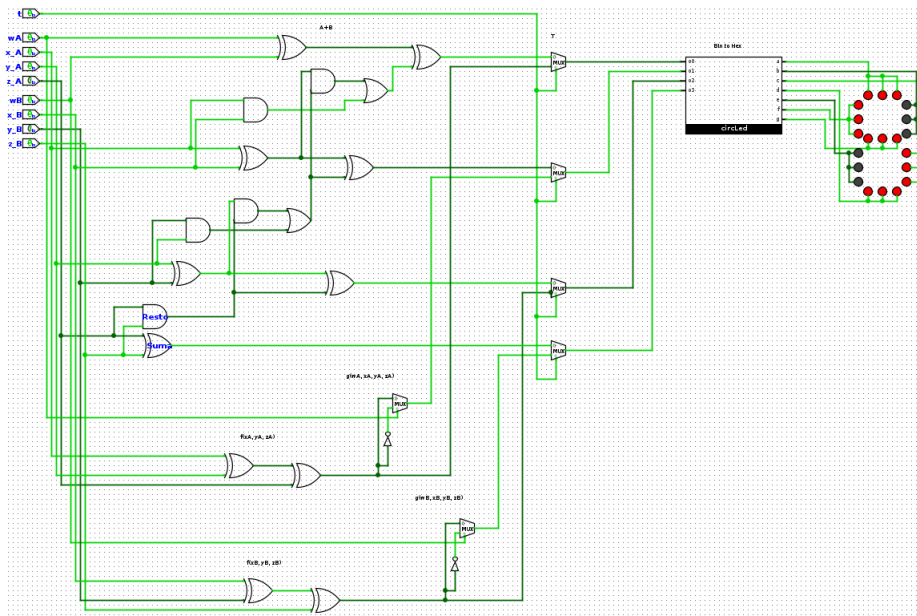


Figure 16: Circuito funcionando con input  $T = 1$ ,  $A = 1110$  y  $B = 1101$ .

<sup>2</sup>watelectronics

## 7 Anexo

$Y$		$y, z$			
		$x$	00	01	11
0	0	0	1	0	1
1	1	1	0	1	0

Figure 17: Suma sin carry, 3 bits

$cin$	$B$	$A$	$Sum$	$cout$
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

Table 2: Tabla de Verdad para suma de 1 bit con carry.

Sum		B, A			
		00	01	11	10
cin	0	0	1	0	1
	1	1	0	1	0

Figure 18: Mapa de Karnaugh para la Suma de 1 bit con carry.

		<i>cout</i> <i>B, A</i>			
<i>cin</i>		00	01	11	10
	0	0	0	1	0
	1	0	1	1	1

Figure 19: Mapa de Karnaugh para el carry de la suma de 1 bit

		<i>xor</i> <i>b, c</i>			
<i>a</i>		00	01	11	10
	0	0	1	0	1
	1	1	0	1	0

Figure 20: Mapa de Karnaugh para *xor*

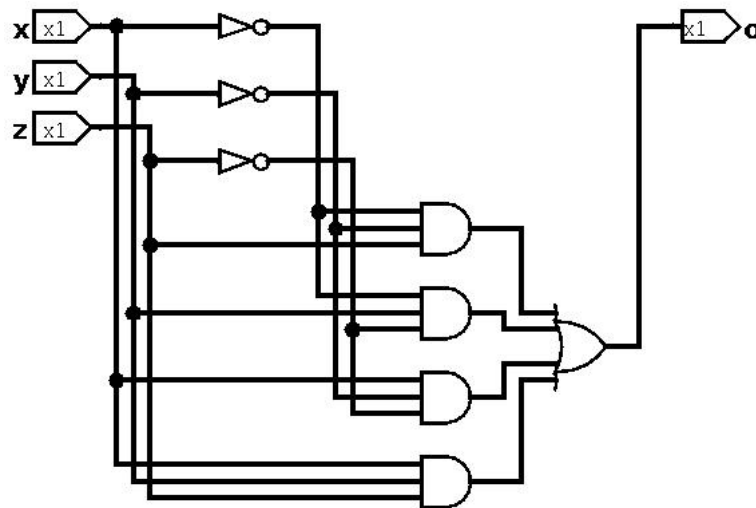


Figure 21:  $f(x, y, z)$  minimizada solo con mapa de Karnaugh

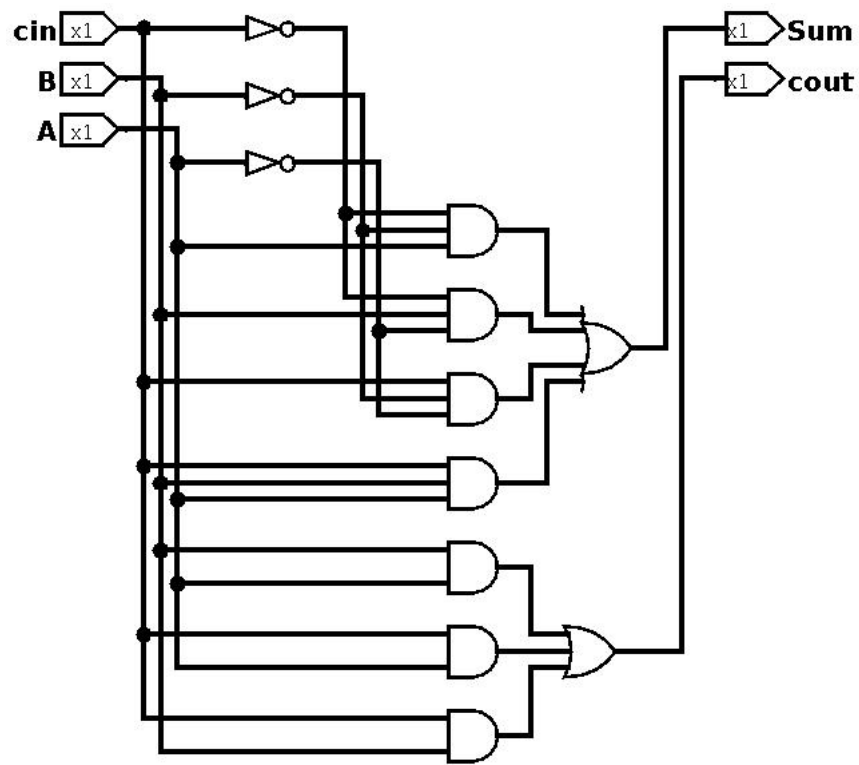


Figure 22: Circuito para la suma completa luego de mapa de Karnaugh

[Link a imagen del subcircuito](#)

Figure 23: Circuito completo para transformación de binario a hexadecimal expresado en display led de 7 segmentos