UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Práctica 2

Construcción de funciones booleanas en la plataforma Quartus II, en modo gráfico.

ALUMNOS

Ríos Lira, Gamaliel
Vélez Grande Cinthya

PROFESOR

Flores Olvera, Vicente

ASIGNATURA

Laboratorio de Diseño Digital Moderno

GRUPO

1

5 de marzo de 2023

Índice

1. Objetivo **2**

2. Materiales **2**

3. Introducción **3**

4. Desarrollo **4**

4.1. Previo 4

4.2. Parte A 9

4.2.1. Instrucciones 9

4.2.2. Análisis 9

4.2.3. Implementación en Quartus 11

4.3. Parte B 14

4.3.1. Instrucciones 14

4.3.2. Análisis 14

4.3.3. Implementación en Quartus 16

4.4. Parte C 20

4.4.1. Propuesta 20

4.4.2. Análisis 20

4.4.3. Implementación en Quartus 21

5. Conclusiones **23**

6. Bibliografía **24**

1. Objetivo

El alumno diseñara mediante un bloque estructurado la implementación de funciones lógicas utilizando el modo grafico de la plataforma *Quartus II*, así como su respectiva simulación.

2. Materiales

Equipo del laboratorio

- Computadora

Software

- *Quartus Prime Lite* 18 (o superior)

3. Introducción

Una función booleana puede representarse mediante una ecuación booleana compuesta por una variable binaria que identifica a la función, en seguida, se tiene el símbolo de igualdad y una expresión booleana; esta describe la forma en la cual se obtiene la salida de un sistema digital a partir de las variables de entrada; estas son especificadas en seguida de la variable que identifica la función, dentro de paréntesis y separadas por comas, generalmente.

La expresión booleana se entiende como una expresión algebráica conformada por variables binarias; es decir, sus posibles valores se encuentran acotados dentro del conjunto $\{0, 1\}$; así como también se conforman de operadores lógicos y paréntesis.

Una función booleana puede tener una única salida o tener una salida múltiple. En caso de tener una única salida, el resultado de la función se obtiene a partir de evaluar las posibles combinaciones de los valores 0 y 1 entre las variables de entrada. En caso de ser una función con salida múltiple, el resultado se obtiene de la forma anterior, pero evaluando a su vez las combinaciones posibles entre las salidas de la función.

En la presente práctica, se implementan diversas funciones booleanas en la plataforma de desarrollo *Quartus II*. A partir de la representación gráfica de las funciones mediante los símbolos correspondientes de las compuertas que conforman la función, se generan componentes que implementan dichas funciones, para ser empleados en otros diseños, obteniendo como ventaja la reducción del circuito a un símbolo que especifica únicamente las entradas y la salida generada, sin especificar los componentes internos.

4. Desarrollo

4.1. Previo

Pregunta 1

¿Qué es un minitérmino y un maxitérmino?

Minitérmino. También conocido como *producto estándar*. Se trata de un término lógico formado por la conjunción de las n variables de una función lógica (negada o sin negar).

Maxitérmino. También conocido como *suma estándar*. Se trata de un término lógico formado por la disyunción de las n variables de una función lógica (negada o sin negar).

Para cada uno de los términos anteriores, existen 2^n combinaciones posibles con todas las variables lógicas.

Pregunta 2

¿Qué es una forma suma de productos y producto de sumas?

Suma de minitérminos. Se trata de una forma de representar a cualquier función lógica como una suma de minitérminos. Los minitérminos que se utilizan para la suma son aquellos que en la tabla de verdad producen una salida lógica de 1.

Producto de maxitérminos. De forma análoga, se trata de un método para representar a cuaquier función booleana como un producto de maxitérminos. Los maxitérminos que se utilizan para el producto son aquellos que en la tabla de verdad producen una salida lógica de 0.

Pregunta 3

De las siguientes expresiones, deduce las formas SOP y POS canónicas para cada una de ellas.

1. $f(abc) = ab + (\bar{b} + \bar{c}) + a\bar{c}$
2. $f(abcd) = (a + b)(\bar{c} + d)(a + c + d)$
3. $f(xywz) = xy\bar{z} + y\bar{w} + y(w\bar{z})$
4. $f(cabo) = \bar{c}(a + \bar{a}o) + \bar{b}o$

Expresión 1. Se comenzará con la obtención de la forma *SOP*.

$$\begin{aligned}
f(abc) &= ab + (\bar{b} + \bar{c}) + a\bar{c} \\
&= ab + \bar{b} + \bar{c}(1 + a) \\
&= ab1 + \bar{b}11 + \bar{c}11 \\
&= ab(c + \bar{c}) + \bar{b}(a + \bar{a})(c + \bar{c}) + \bar{c}(a + \bar{a})(b + \bar{b}) \\
&= abc + ab\bar{c} + a\bar{b}c + a\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c} + \cancel{abc} + \cancel{a\bar{b}\bar{c}} + \bar{a}b\bar{c} + \cancel{\bar{a}\bar{b}\bar{c}} \\
&= abc + ab\bar{c} + a\bar{b}c + a\bar{b}\bar{c} + \bar{a}\bar{b}c + \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c}
\end{aligned}$$

Continuando con la obtención de la forma *POS*:

$$\begin{aligned}
f(abc) &= ab + (\bar{b} + \bar{c}) + a\bar{c} \\
&= ab + \bar{b} + \bar{c}(1 + a) \\
&= ab + (\bar{b} + \bar{c}) \\
&= (a + \bar{b} + \bar{c})(b + \bar{b} + \bar{c}) \\
&= (a + \bar{b} + \bar{c})
\end{aligned}$$

$$\boxed{\therefore f(abc) = \sum(m_0, m_1, m_2, m_4, m_5, m_6, m_7)}$$

$$\boxed{\therefore f(abc) = \prod(M_3)}$$

Expresión 2. Se comenzará con la obtención de la forma *POS*.

$$\begin{aligned}
f(abcd) &= (a + b)(\bar{c} + d)(a + c + d) \\
&= (a + b + 0 + 0)(\bar{c} + d + 0 + 0)(a + c + d + 0) \\
&= (a + b + c\bar{c} + d\bar{d})(\bar{c} + d + a\bar{a} + b\bar{b})(a + c + d + b\bar{b}) \\
&= (a + b + c + d)(a + b + c + \bar{d})(a + b + \bar{c} + d)(a + b + \bar{c} + \bar{d})\cancel{(a + b + \bar{c} + d)} \\
&(a + \bar{b} + \bar{c} + d)(\bar{a} + b + \bar{c} + d)(\bar{a} + \bar{b} + \bar{c} + d)\cancel{(a + b + c + d)}(a + \bar{b} + c + d) \\
&= (a + b + c + d)(a + b + c + \bar{d})(a + b + \bar{c} + d)(a + b + \bar{c} + \bar{d})(a + \bar{b} + c + d) \\
&(a + \bar{b} + \bar{c} + d)(\bar{a} + b + \bar{c} + d)(\bar{a} + \bar{b} + \bar{c} + d)
\end{aligned}$$

Prosiguiendo con la obtención de la forma *SOP*.

$$\begin{aligned}
f(abcd) &= (a+b)(\bar{c}+d)(a+c+d) \\
&= aa\bar{c} + a\bar{c}c + a\bar{c}d + aad + acd + add + ab\bar{c} + b\bar{c}c + b\bar{c}d + abd + bcd + bdd \\
&= a\bar{c} + 0 + a\bar{c}d + ad + acd + ad + ab\bar{c} + 0 + b\bar{c}d + abd + bcd + bd \\
&= a\bar{c} + a\bar{c}d + ad + acd + ad + ab\bar{c} + b\bar{c}d + abd + bcd + bd \\
&= a\bar{c}(1+d) + ad(1+c) + ab\bar{c} + b\bar{c}d + ad(1+b) + bd(c+1) \\
&= a\bar{c} + ad + ab\bar{c} + b\bar{c}d + \cancel{ad} + bd \\
&= a\bar{c} + ad + ab\bar{c} + bd(\bar{c}+1) \\
&= a\bar{c} + ad + ab\bar{c} + bd
\end{aligned}$$

Con la forma deducida anteriormente es posible obtener los minitérminos de forma más sencilla.

$$\begin{aligned}
f(abcd) &= a\bar{c} + ad + ab\bar{c} + bd \\
&= a\bar{c}11 + ad11 + ab\bar{c}1 + bd11 \\
&= a\bar{c}(b+\bar{b})(d+\bar{d}) + ad(b+\bar{b})(c+\bar{c}) + ab\bar{c}(d+\bar{d}) + bd(a+\bar{a})(c+\bar{c}) \\
&= ab\bar{c}d + ab\bar{c}\bar{d} + a\bar{b}\bar{c}d + a\bar{b}\bar{c}\bar{d} + abcd + \cancel{ab\bar{c}\bar{d}} + a\bar{b}cd + \cancel{a\bar{b}\bar{c}d} + \cancel{ab\bar{c}\bar{d}} + \cancel{ab\bar{c}\bar{d}} \\
&+ \cancel{abc\bar{d}} + \cancel{ab\bar{c}\bar{d}} + \bar{a}bcd + \bar{a}b\bar{c}d \\
&= abcd + ab\bar{c}d + ab\bar{c}\bar{d} + a\bar{b}cd + a\bar{b}\bar{c}d + a\bar{b}\bar{c}\bar{d} + \bar{a}bcd + \bar{a}b\bar{c}d
\end{aligned}$$

$$\boxed{\therefore f(abcd) = \sum(m_5, m_7, m_8, m_9, m_{11}, m_{12}, m_{13}, m_{15})}$$

$$\boxed{\therefore f(abcd) = \prod(M_0, M_1, M_2, M_3, M_4, M_6, M_{10}, M_{14})}$$

Expresión 3. Se comenzará con la obtención de la forma *SOP*:

$$\begin{aligned}
f(xywz) &= xy\bar{z} + y\bar{w} + y(w\bar{z}) \\
&= xy\bar{z}(w+\bar{w}) + y\bar{w}(x+\bar{x})(z+\bar{z}) + y(w\bar{z})(x+\bar{x}) \\
&= xyw\bar{z} + xy\bar{w}\bar{z} + xy\bar{w}z + \cancel{xy\bar{w}\bar{z}} + \bar{x}y\bar{w}z + \bar{x}y\bar{w}\bar{z} + \cancel{xyw\bar{z}} + \bar{x}ywz \\
&= xyw\bar{z} + xy\bar{w}z + xy\bar{w}\bar{z} + \bar{x}ywz + \bar{x}y\bar{w}z + \bar{x}y\bar{w}\bar{z}
\end{aligned}$$

Con lo anterior, ahora se puede continuar con la obtención de la forma *POS*.

$$
\begin{aligned}
f(xywz) &= xy\bar{z} + y\bar{w} + y(w\bar{z}) \\
&= y(x\bar{z} + \bar{w} + w\bar{z}) \\
&= y(\bar{z}(x + w) + \bar{w}) \\
&= y(\bar{z} + \bar{w})(x + w + \bar{w}) \\
&= y(\bar{z} + \bar{w}) \\
&= (y + x\bar{x} + w\bar{w} + z\bar{z})(\bar{z} + \bar{w} + x\bar{x} + y\bar{y}) \\
&= (x + y + w + z)(x + y + w + \bar{z})(x + y + \bar{w} + z)(x + y + \bar{w} + \bar{z}) \\
&(\bar{x} + y + w + z)(\bar{x} + y + w + \bar{z})(\bar{x} + y + \bar{w} + z)(\bar{x} + y + \bar{w} + \bar{z}) \\
&\cancel{(x + y + \bar{w} + \bar{z})}(x + \bar{y} + \bar{w} + \bar{z})\cancel{(\bar{x} + y + \bar{w} + \bar{z})}(\bar{x} + \bar{y} + \bar{w} + \bar{z}) \\
&= (x + y + w + z)(x + y + w + \bar{z})(x + y + \bar{w} + z)(x + y + \bar{w} + \bar{z}) \\
&(x + \bar{y} + \bar{w} + \bar{z})(\bar{x} + y + w + z)(\bar{x} + y + w + \bar{z})(\bar{x} + y + \bar{w} + z) \\
&(\bar{x} + y + \bar{w} + \bar{z})(\bar{x} + \bar{y} + \bar{w} + \bar{z})
\end{aligned}
$$

$$\boxed{\therefore f(xywz) = \sum(m_4, m_5, m_6, m_{12}, m_{13}, m_{14})}$$

$$\boxed{\therefore f(xywz) = \prod(M_0, M_1, M_2, M_3, M_7, M_8, M_9, M_{10}, M_{11}, M_{15})}$$

Expresión 4. Se comenzará con la obtención de la forma *SOP*:

$$
\begin{aligned}
f(cabo) &= \bar{c}(a + \bar{a}o) + \bar{b}o \\
&= \bar{c}a + \bar{c}\bar{a}o + \bar{b}o \\
&= \bar{c}a(b + \bar{b})(o + \bar{o}) + \bar{c}\bar{a}o(b + \bar{b}) + \bar{b}o(c + \bar{c})(a + \bar{a}) \\
&= \bar{c}abo + \bar{c}ab\bar{o} + \bar{c}a\bar{b}o + \bar{c}a\bar{b}\bar{o} + \bar{c}\bar{a}bo + \bar{c}\bar{a}\bar{b}o + ca\bar{b}o + c\bar{a}\bar{b}o + \cancel{\bar{c}a\bar{b}o} + \cancel{\bar{c}\bar{a}\bar{b}o} \\
&= ca\bar{b}o + c\bar{a}\bar{b}o + \bar{c}abo + \bar{c}ab\bar{o} + \bar{c}a\bar{b}o + \bar{c}a\bar{b}\bar{o} + \bar{c}\bar{a}bo + \bar{c}\bar{a}\bar{b}o
\end{aligned}
$$

Prosiguiendo con la obtención de la *POS*:

$$
\begin{aligned}
f(cabo) &= \bar{c}(a + \bar{a}o) + \bar{b}o \\
&= (\bar{c} + \bar{b}o)(a + \bar{a}o + \bar{b}o) \\
&= (\bar{c} + \bar{b})(\bar{c} + o)(a + (\bar{a} + \bar{b})o) \\
&= (\bar{c} + \bar{b})(\bar{c} + o)(a + \bar{a} + \bar{b})(a + o) \\
&= (\bar{c} + \bar{b})(\bar{c} + o)(1 + \bar{b})(a + o) \\
&= (\bar{c} + \bar{b})(\bar{c} + o)(a + o) \\
&= (\bar{c} + \bar{b} + a\bar{a} + o\bar{o})(\bar{c} + o + a\bar{a} + b\bar{b})(a + o + c\bar{c} + b\bar{b}) \\
&= (\bar{c} + a + \bar{b} + o)(\bar{c} + a + \bar{b} + \bar{o})(\bar{c} + \bar{a} + \bar{b} + o)(\bar{c} + \bar{a} + \bar{b} + \bar{o}) \\
&(\bar{c} + a + b + o)\cancel{(\bar{c} + a + \bar{b} + \bar{o})}(\bar{c} + \bar{a} + b + o)\cancel{(\bar{c} + \bar{a} + \bar{b} + \bar{o})} \\
&(c + a + b + o)(c + a + \bar{b} + o)\cancel{(\bar{c} + a + b + \bar{o})}\cancel{(\bar{c} + a + \bar{b} + \bar{o})} \\
&= (c + a + b + o)(c + a + \bar{b} + o)(\bar{c} + a + b + o)(\bar{c} + a + \bar{b} + o)(\bar{c} + a + \bar{b} + \bar{o}) \\
&(\bar{c} + \bar{a} + b + o)(\bar{c} + \bar{a} + \bar{b} + o)(\bar{c} + \bar{a} + \bar{b} + \bar{o})
\end{aligned}
$$

$$\boxed{\therefore f(cabo) = \sum(m_1, m_3, m_4, m_5, m_6, m_7, m_9, m_{13})}$$

$$\boxed{\therefore f(cabo) = \prod(M_0, M_2, M_8, M_{10}, M_{11}, M_{12}, M_{14}, M_{15})}$$

4.2. Parte A

4.2.1. Instrucciones

Implantar las siguientes funciones:

- $f_1(abc) = a\bar{b}c + \bar{a}$
- $f_2(abc) = ab + \bar{b}a\bar{c} + a \odot c$
- $f_3(abc) = (a + \bar{b} + \bar{c})(b + c)$

4.2.2. Análisis

Antes de comenzar a implementar la solución en *Quartus*, es importante deducir el comportamiento esperado, con la finalidad de detectar errores en el funcionamiento. Para realizar esto, haremos uso de las tablas de verdad de las expresiones.

Para el caso de las función $f_1(abc)$, se tiene la siguiente tabla de verdad:

a	b	c	$a\bar{b}c$	$\bar{a}$	f_1
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	0	1	1
1	0	0	0	0	0
1	0	1	1	0	1
1	1	0	0	0	0
1	1	1	0	0	0

Tabla 1: Tabla de verdad de $f_1(abc)$ (Parte A)

Para el caso de la función $f_2(abc)$, es un poco más cómodo hacer una representación a través de la forma equivalente de la compuerta XNOR, recordando que esta tiene más precedencia que la compuerta AND u OR, por lo que queda representada de la siguiente forma:

$$f_2(abc) = ab + \bar{b}a\bar{c} + ac + \bar{a}\bar{c}$$

Entonces, la tabla de verdad queda representada de la siguiente forma:

a	b	c	ab	$\bar{b}a\bar{c}$	ac	$\bar{a}\bar{c}$	f_2
0	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0
0	1	0	0	0	0	1	1
0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	1	0	0	1	0	1
1	1	0	1	0	0	0	1
1	1	1	1	0	1	0	1

Tabla 2: Tabla de verdad de $f_2(abc)$ (Parte A)

Prosiguiendo con la obtención de las tablas de verdad, se tiene la la tabla de verdad de la función lógica $f_3(abc)$, para la cual se tiene lo siguiente:

a	b	c	$a + \bar{b} + \bar{c}$	$b + c$	f_3
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

Tabla 3: Tabla de verdad de $f_3(abc)$ (Parte A)

Combinando los resultados de la Tablas 1, 2 y 3, se tiene lo siguiente:

a	b	c	f_1	f_2	f_3	HEX
0	0	0	1	1	0	6
0	0	1	1	0	1	5
0	1	0	1	1	1	7
0	1	1	1	0	0	4
1	0	0	0	1	0	2
1	0	1	1	1	1	7
1	1	0	0	1	1	3
1	1	1	0	1	1	3

Tabla 4: Resumen del comportamiento de f_1, f_2 y f_3 (Parte A)

4.2.3. Implementación en Quartus

Para la implementación de estas compuertas en *Quartus*, se creó un diagrama de bloques para cada una de ellas, tal como se muestra a continuación:

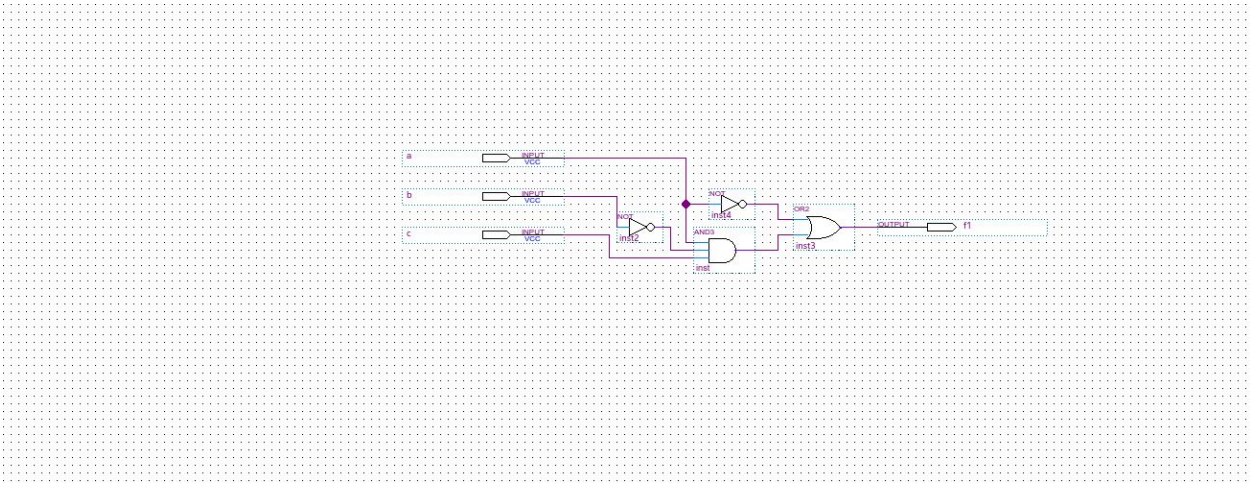

Figura 1: Implementación de $f_1(abc)$ (Parte A)

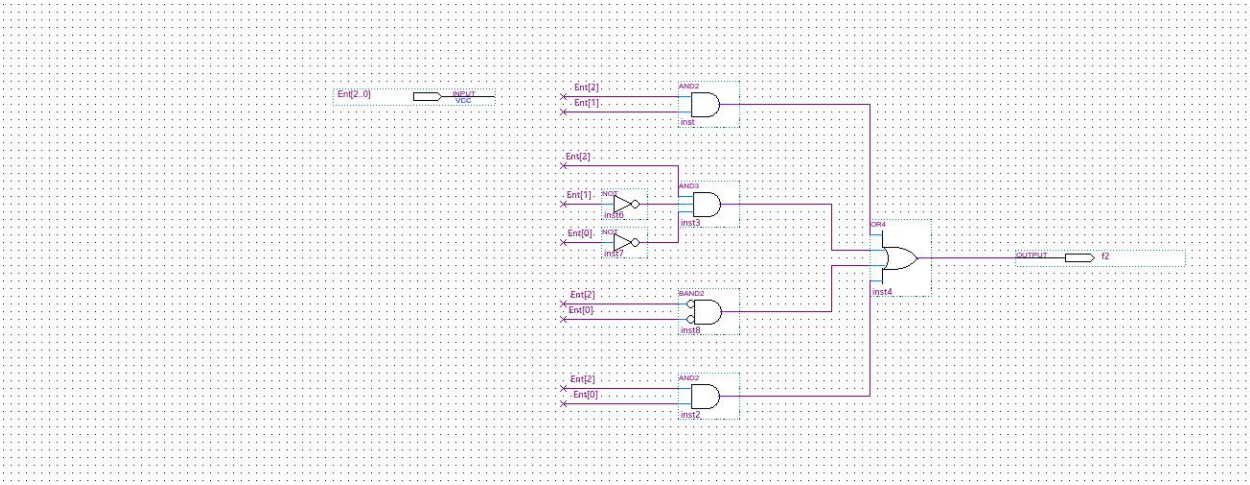

Figura 2: Implementación de $f_2(abc)$ (Parte A)

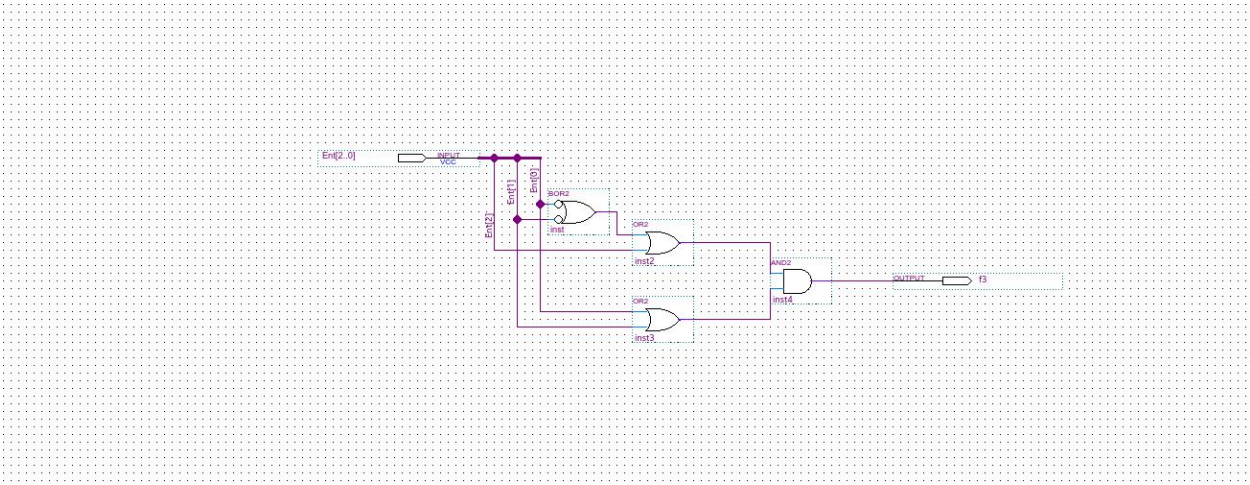

Figura 3: Implementación de $f_3(abc)$ (Parte A)

Tal como se puede apreciar en la Figura 1, las entradas se manejaron como valores independientes (a, b y c). Una aproximación bastante similar a la que se tuvo en la práctica anterior.

Por otro lado, en las Figuras 2 y 3 se hizo uso de dos conceptos nuevos:

1. Por una parte, los valores de a, b y c se agruparon a través de una variable de tipo *array* con una longitud de **tres elmentos**, en la que se tiene la siguiete estructura:
   - $a = Ent[2]$
   - $b = Ent[1]$
   - $c = Ent[0]$
2. Y por el otro, se manejó un alambrado a través de etiquetas en los cables, con la finalidad de evitar cruces inesperados cuando las conexiones se vuelve más complicadas. Además, cuando se quiere pasar de un cable que trae consigue muchas señales (como el caso de las señales de arreglos) y se quiere pasar a un cable con una señal, es obligatorio poner el elemento exacto del cual se tomará el valor.

Para comprobar el comportamiento de estas funciones lógicas, se tuvo que implementar un apartado especial (haciendo uso de los símbolos creados anteriormente para cada una de las funciones) en un archivo especial. Se configuró la entrada como un arreglo de tres elementos y la salida también se implementó como otro arreglo, tal como se muestra a continuación:

1. **Ent**: El arreglo de valores de entrada
   - $a = Ent[2]$
   - $b = Ent[1]$
   - $c = Ent[0]$
2. **Sal**: El arreglo de valores de salida

- $f_1 = Sal[2]$
- $f_2 = Sal[1]$
- $f_3 = Sal[0]$

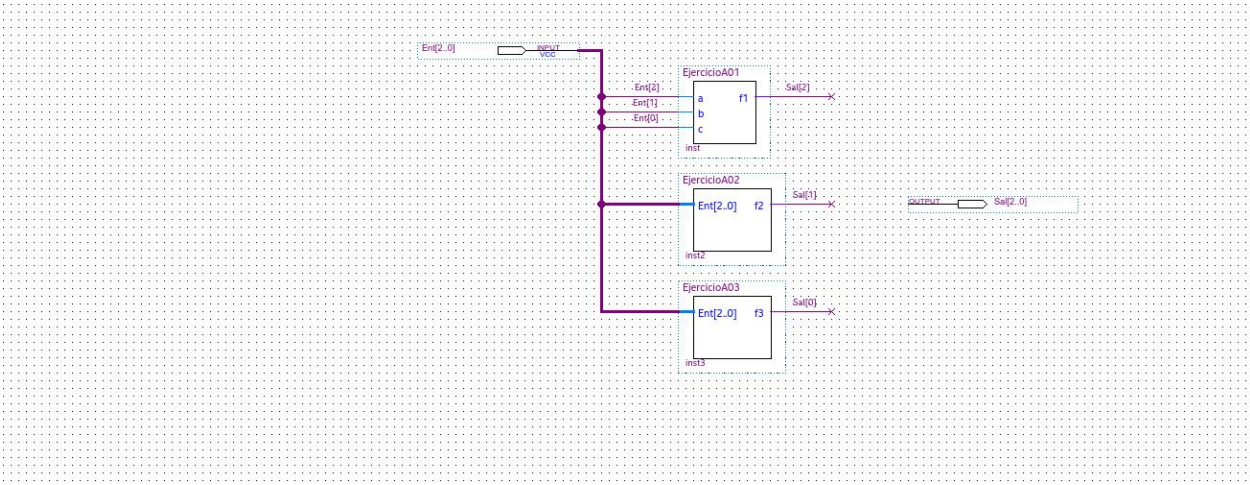

Figura 4: Implementación de la solución (Parte A)

Con lo anterior, es posible realizar la simulación con ayuda de un archivo de tipo *University Program VWF*. Se configuraron las señales a visualizar, y se configuraron para tener las entradas mostradas en la Tabla 4. Su utilizó la *Funcional Simuation*, y el resultdo fue el siguiente:

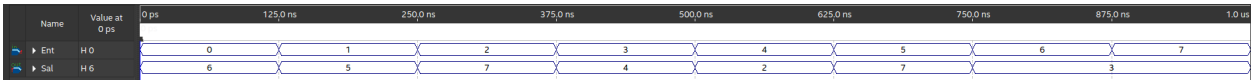

Figura 5: Simulación (Parte A)

Tal como se puede ver en la Figura 5, los resultados de la simulación fueron los mismos que los pre-calculados en la Tabla 4.

4.3. Parte B

4.3.1. Instrucciones

A partir de las siguientes funciones lógicas, impleméntelas dentro del ambiente gráfico de la plataforma *Quatus II* de *Altera*.

- $F_1(xyz) = x(\bar{x} + y\bar{z} + z)$
- $F_2(xyz) = (x + y)(\bar{x} + y)(\bar{y} + \bar{z})$
- $F_3(xyz) = \bar{x}\bar{y} + x \oplus \bar{y}$
- $F_4(xz) = (x + z) \odot \bar{x}$

4.3.2. Análisis

De igual forma que en la Parte A, se comenzará haciendo el análsis del comportamiento de cada una de las funciones lógicas a través de us tabla de verdad. Comenzando con el comportamiento de $f_1(xyz)$.

x	y	z	$\bar{x}$	$y\bar{z}$	$\bar{x} + y\bar{z} + z$	f_1
0	0	0	1	0	1	0
0	0	1	1	0	1	0
0	1	0	1	1	1	0
0	1	1	1	0	1	0
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	0	1	1	1
1	1	1	0	0	1	1

Tabla 5: Tabla de verdad de $f_1(xyz)$ (Parte B)

Continuando con el comportamiento de $f_2(xyz)$, se tiene que:

x	y	z	$x + y$	$\bar{x} + y$	$\bar{y} + \bar{z}$	f_2
0	0	0	0	1	1	0
0	0	1	0	1	1	0
0	1	0	1	1	1	1
0	1	1	1	1	0	0
1	0	0	1	0	1	0
1	0	1	1	0	1	0
1	1	0	1	1	1	1
1	1	1	1	1	0	0

Tabla 6: Tabla de verdad de $f_2(xyz)$ (Parte B)

Se prosigue con el comportamiento de la función lógica $f_3(xyz)$, para esta, es primero hacer uso de la equivalencia lógica de la operación XOR, con la finalidad de simplificar sus cálculos. Por lo tanto, se obtiene lo siguiente:

$$\begin{aligned} f_3(xyz) &= \bar{x}\bar{y} + x\bar{\bar{y}} + \cancel{\bar{x}\bar{\bar{y}}} \\ &= \bar{x}\bar{y} + xy \end{aligned}$$

$$\therefore f_3(xyz) = \bar{x}\bar{y} + xy$$

Haciendo uso de la equivalencia anterior, se puede conformar la tabla de verdad de la siguiente forma:

x	y	z	$\bar{x}\bar{y}$	xy	f_3
0	0	0	1	0	1
0	0	1	1	0	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	0	0
1	0	1	0	0	0
1	1	0	0	1	1
1	1	1	0	1	1

Tabla 7: Tabla de verdad de $f_3(xyz)$ (Parte B)

Para concluir lo anterior, también se elaborará la tabla de verdad de la función $f_4(xy)$, la cual se puede simplificar a través de la equivalencia lógica de la compuerta XNOR.

$$\begin{aligned} f_4(xz) &= (x+z) \odot \bar{x} \\ &= \overline{(x+z)}\bar{\bar{x}} + (x+z)x \\ &= \cancel{(\bar{x}\bar{z})x} + (x+z)\bar{x} \\ &= (x+z)\bar{x} \end{aligned}$$

$$f_4(xz) = (x+z)\bar{x}$$

Con la equivalencia anterior se puede elaborar la tabla de verdad, tal como se muestra a continuación

x	z	$\bar{x}$	$x+z$	f_4
0	0	1	0	0
0	1	1	1	1
1	0	0	1	0
1	1	0	1	0

Tabla 8: Tabla de verdad de $f_4(xz)$ (Parte B)

Finalmente, se combinarán los resultados de las Tablas 5, 6, 7 y 8, con la finalidad de tener el resultado de cada una de ellas en una sola tabla.

x	y	z	f_1	f_2	f_3	f_4	HEX
0	0	0	0	0	1	0	2
0	0	1	0	0	1	1	3
0	1	0	0	1	0	0	4
0	1	1	0	0	0	1	1
1	0	0	0	0	0	0	0
1	0	1	1	0	0	0	8
1	1	0	1	1	1	0	E
1	1	1	1	0	1	0	A

Tabla 9: Resumen del comportamiento de f_1, f_2, f_3 y f_4 (Parte B)

4.3.3. Implementación en Quartus

Para la implementación en la plataforma *Quartus*, se siguió una metodología similar a la mostrada en la Sección 4.2.3, con la cual se hicieron cuatro archivos de bloques, uno para cada función lógica requirida. Los bloques implementados se muestran a continuación.

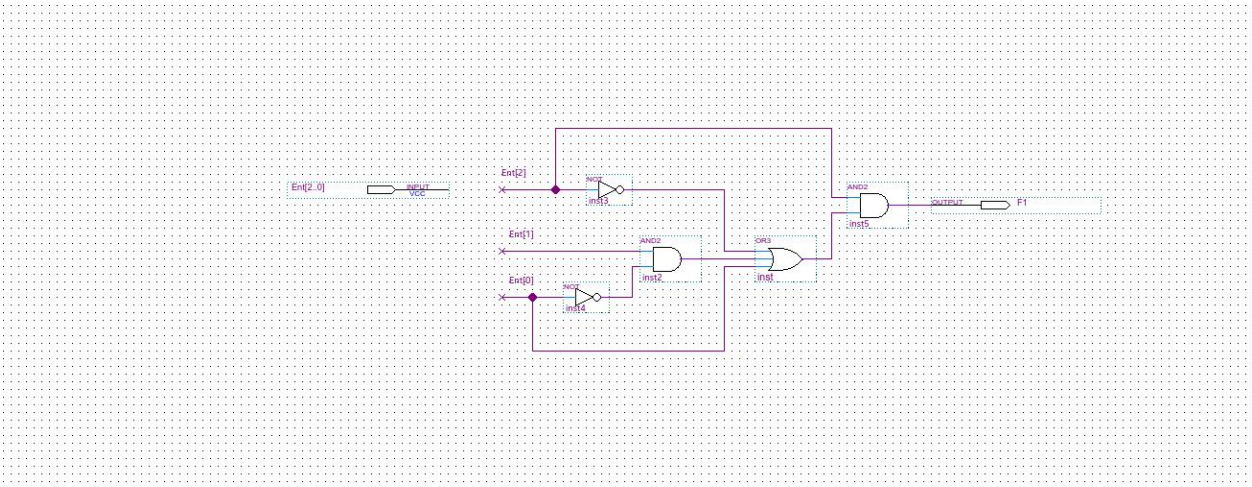

Figura 6: Implementación de $f_1(xyz)$ (Parte B)

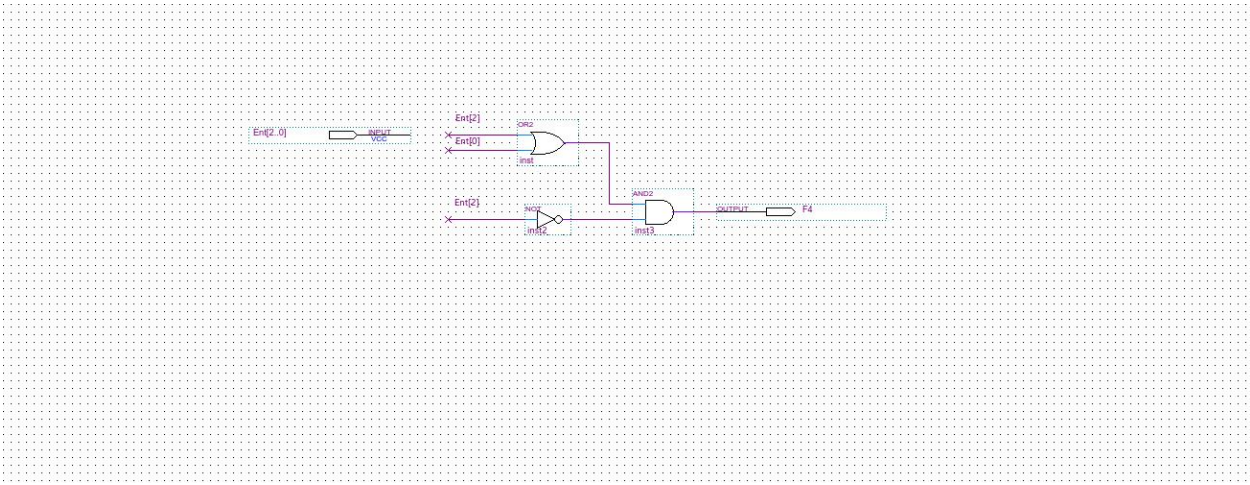

Figura 7: Implementación de $f_2(xyz)$ (Parte B)

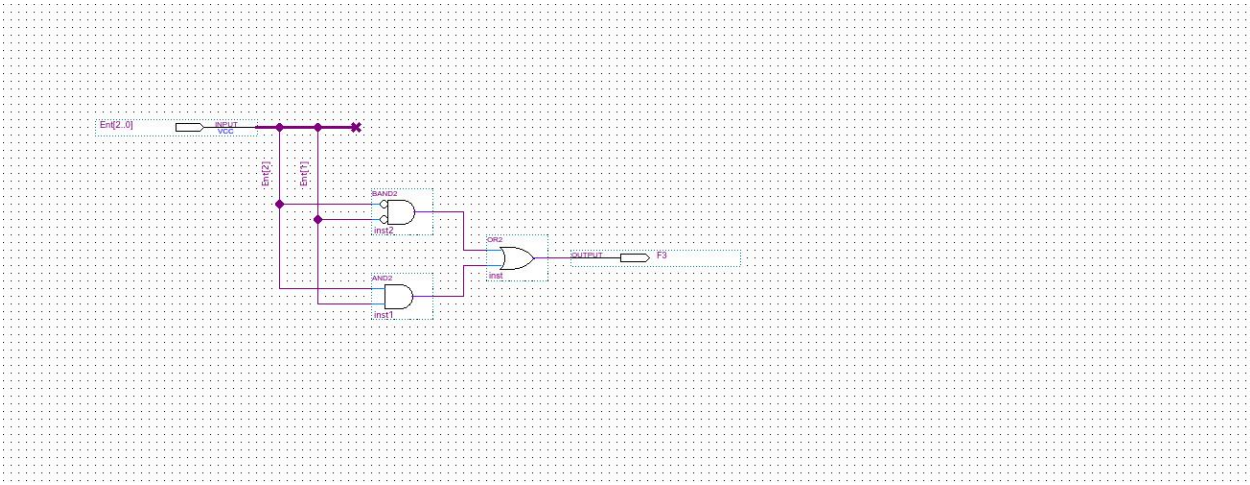

Figura 8: Implementación de $f_3(xyz)$ (Parte B)

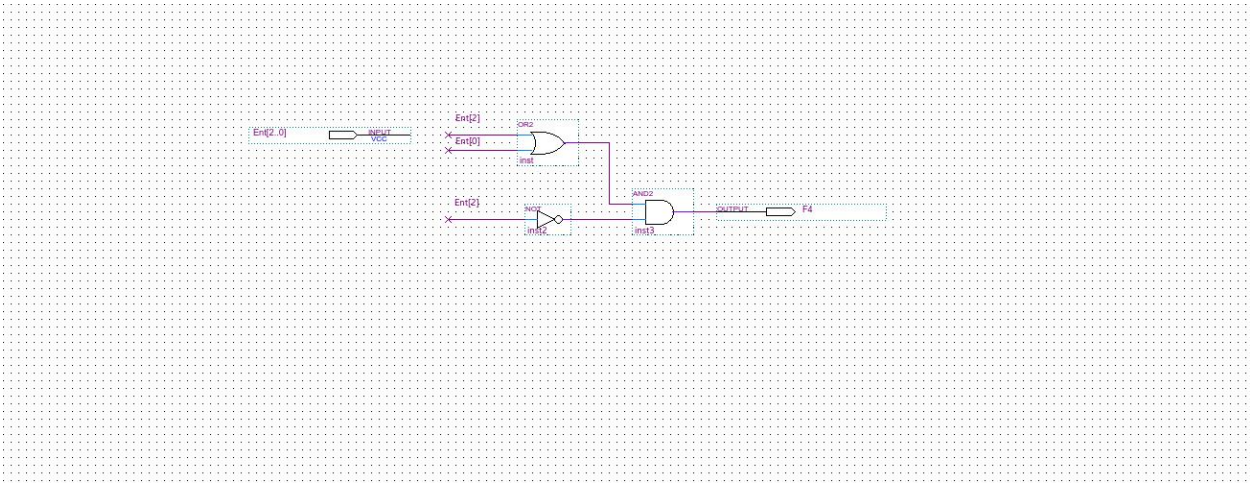

Figura 9: Implementación de $f_4(xyz)$ (Parte B)

Se hizo uso del concepto de arreglos de señales con la finalidad de simplificar la utilización de cada uno de los módulos y así mismo, comprobar más sencillamente el funcionamiento de los sistemas. De igual forma, muchas de las conexiones se llevaron a cabo a través de la colocación de etiquetas en los cables —tal como aprendimos en la Parte A.

De igual forma, en un archivo de bloques separado fue importante mandar a llamar a cada uno de los símbolos previamente creados con la finalidad de ponerlos a pruba con un vector de entrada y un vector de salida, tal como se muestra a continuación.

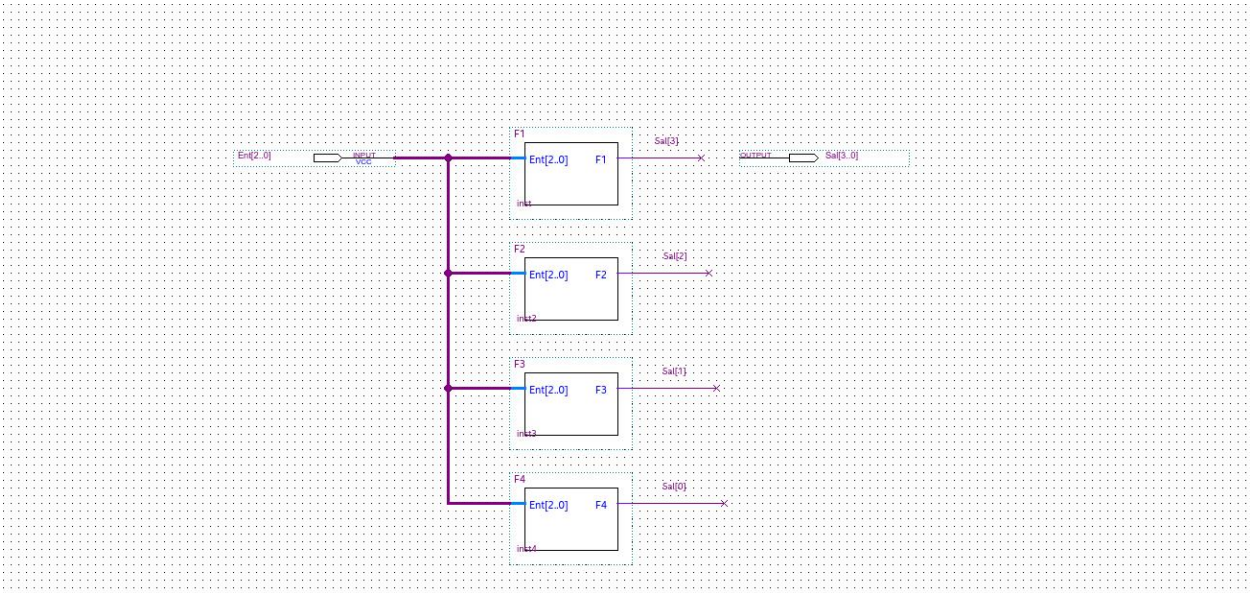

Figura 10: Implementación de la solución (Parte B)

Con lo anterior, fue posible realizar la simulación con ayuda de un archivo de tipo *University Program VWF*. Se configuraron las señales a visualizar, y se configuraron para tener las entradas mostradas en la Tabla 9. Su utilizó la *Funcional Simuation*, y el resultdo fue el siguiente:

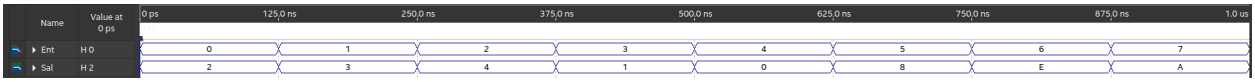

Figura 11: Simulación (Parte B)

Tal como se puede ver en la Figura 11, los resultados de la simulación fueron los mismos que los pre-calculados en la Tabla 9. Los valores del arreglo Ent (codificados en hexadecimal) son las señales de entrada; y los valores del arreglo Sal (codificados en hexadecimal) son las señales de salida de la función.

4.4. Parte C

4.4.1. Propuesta

Se desea implementar un sistema que convierta un número en base 2 de cuatro bits a su representación en *Código Grey* y otro sistema que haga el proceso inverso. De forma general, se tiene que un número $abcd_{(2)}$ tiene una representación $wxyz_{Grey}$ dada por la siguiente tabla:

HEX	a	b	c	d	w	x	y	z
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
A	1	0	1	0	1	1	1	1
B	1	0	1	1	1	1	1	0
C	1	1	0	0	1	0	1	0
D	1	1	0	1	1	0	1	1
E	1	1	1	0	1	0	0	1
F	1	1	1	1	1	0	0	0

Tabla 10: Relación entre Base 2 y Código Grey

4.4.2. Análisis

Por la forma en la que trabaja el *Código Grey*, se tiene el siguiente comportamiento:

- $w(abcd) = a$
- $x(abcd) = a \oplus b$
- $y(abcd) = b \oplus c$
- $z(abcd) = c \oplus d$

Y por otra parte, se tiene que:

- $a(wxyz) = w$
- $b(wxyz) = x \oplus a(wxyz)$

- $c(wxyz) = y \oplus b(wxyz)$
- $d(wxyz) = z \oplus c(wxyz)$

Lo interesante de este útimo conjunto de relaciones es que su definición es recurrente, es decir, $a(wxyz)$, $b(wxyz)$ y $c(wxyz)$ se comportan como salidas, pero a su vez también son entradas para calcular las sigueintes cifras.

4.4.3. Implementación en Quartus

Se crearon dos símbolos, uno para convertir de *Base 2* a *Código Grey* (f_1), y otro para convertir de *Código Grey* a *Base 2* (f_2). Las implementaciones se muestran a continuación:

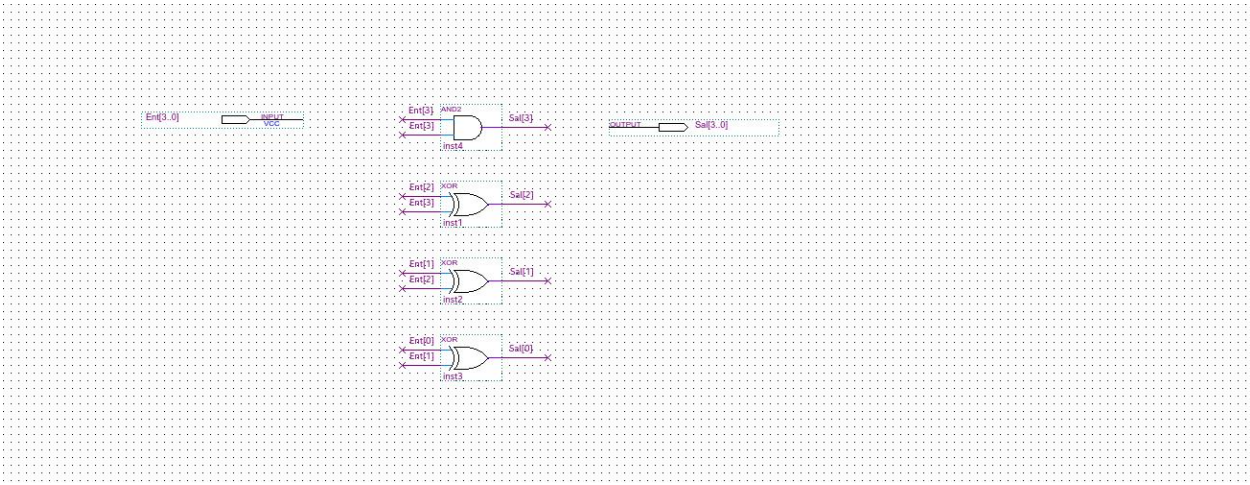

Figura 12: Implementación de $f_1(abcd)$ (Parte C)

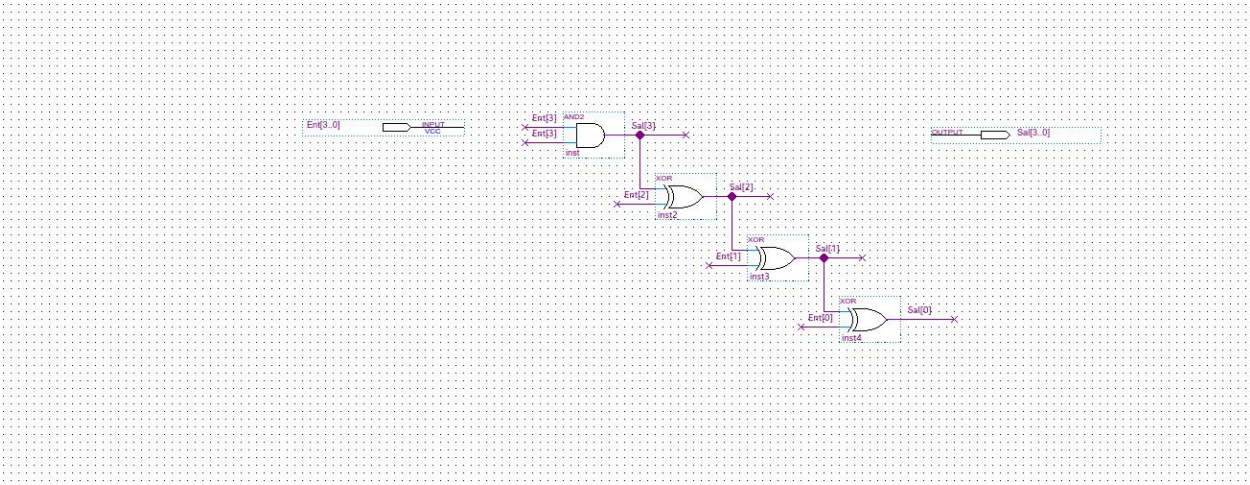

Figura 13: Implementación de $f_2(wxyz)$ (Parte C)

De forma general, cada uno de estos módulos ocupa etiquetas en los cables para reducir el número de conexiones físicas y así evitar errores; y arreglos para simplificar las entradas y las salidas. Ambos módulos se probaron de la siguiente forma:

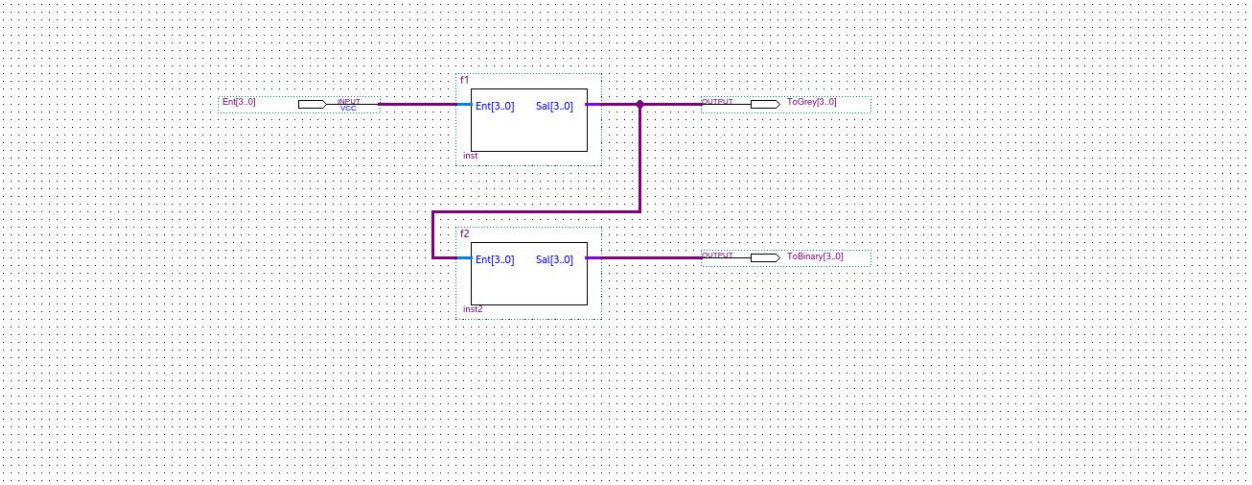

Figura 14: Implementación de la solución (Parte C)

El sistema tiene una entrada en forma de arreglo (representación en bits del número binario a convertir) y dos salidas (una en *Código Grey* y otra en *Base 2*). Lo que se pretende es obtener la representación en *Grey* (primera señal de salida) a través de $f_1(abcd)$ y posteriormente esa misma salida pasarla a $f_2(wxyz)$ para volver a obtener la misma señal de entrada. Si lo anterior sucede, entonces los sistemas implementados funcionarán correctamente.

La comprobación de ambos sistemas se llevó a cabo a través de un *diagrama de tiempos*, para el cual se configuró un archivo *University Program VWF*. Ls imulación resultó en lo siguiente:

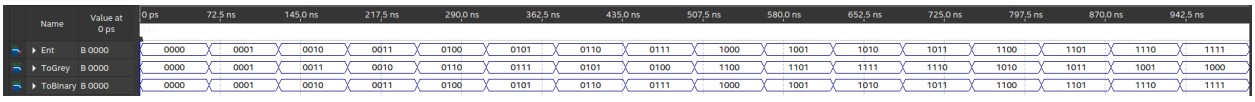

Figura 15: Simulación (Parte A)

Los resultados mostrados en la Figura 15 son los mismos que se describen en la Tabla 10. Además, la señal Ent conicide de forma exacta con la señal $ToBin$. Esto indica que están bien implementados ambos sitemas.

5. Conclusiones

Ríos Lira, Gamaliel. Con la realización de la práctica, se cumplió el objtivo planteado al inicio de la misma. Conocimos nuevas funcionalidades que nos proporciona el entorno de *Quartus II*, como la creación de sub-módulos a través de símbolos, el concepto de vectores y el uso de etiquetas para evitar conexiones complejas. Los resultados obtenidos a través de la protaforma se comprobaron con éxito a través del análisis previo anterior. Así mismo, ví la importancia de utilizar vectores para facilitar las simulaciones y el manejo de las variables. Finalmente, me pareció adecuado todo lo que aprendimos para implementar la Parte C de la práctica.

Vélez Grande, Cinthya. A partir del desarrollo de la práctica fue posible implementar diversas funciones booleanas en el entorno de desarrollo Quartus II, especificando la interconexión de las variables de entrada con las compuertas lógicas que la conforman y las salidas de los sistemas. Uno de los aprendizajes obtenidos fue la forma de abstraer las funciones en forma de símbolos que posteriormente se implementaron en otros archivos del tipo BDF para obtener en forma de única salida las múltiples salidas generadas; y a partir del cronograma generado de la evaluación de las funciones, se comprobó que el resultado obtenido era idéntico al obtenido mediante la evaluación tabular de las funciones.

6. Bibliografía

[1] M. Mano, *Digital Design*. Upper Saddle River, NJ, United States: Prentice Hall, 2002.

[2] K. E. Schwartz. "Component Creation: Quartus Tool Tip." (2018), dirección: `https://mil.ufl.edu/3701/docs/quartus/Component_Creation.pdf`.