amsmath

INSTITUTO TECNOLÓGICO DE BUENOS AIRES

22.12 - Electrónica III

Trabajo Práctico $N°1$

Grupo 4

Bertachini, Germán	58750
Dieguez, Manuel	56273
Galdeman, Agustín	59827
Laguingue, Juan Martín	57430

Profesores
Dewald, Kevin
Wundes, Pablo

Presentado el 5 de Septiembre de 2019

Índice

1. Algebra booleana y compuertas lógicas

En el álgebra booleana se conoce como término canónico de una función lógica a todo producto o suma en la cual aparecen todas las variables en su forma directa o inversa. Cualquier función lógica puede expresarse de forma canónica utilizando los conceptos de min y maxtérminos. El primero, se refiere a todas las filas para las cuales la función lógica es igual a uno en la tabla de verdad correspondiente, mientras que el segundo se corresponde con todos los valores para los cuales la función es igual a cero.

En el siguiente ejercicio se simplificarán dos funciones lógicas aplicando ambos conceptos. Luego se utilizarán mapas de Karnaugh para llegar a la misma expresión.A continuación se dibujará el circuito lógico resultante utilizando compuertas AND, OR y NOT, y por último, se usarán compuertas NAND exclusivamente.

1.1. Primera Expresión: suma de productos

La expresión de la cual se parte está dada por la siguiente fórmula:

$$f(e,d,c,b,a) = \sum m(0,2,4,7,8,10,12,16,18,20,23,24,25,26,27,28) \tag{1}$$

La ecuación 1 nos indica cuales entradas en la tabla de verdad son iguales a uno. A partir de ella podemos plantear la siguiente ecuación, la cual se trabajará para llegar a su respectiva expresión canónica.

$$\bar{e} \cdot \bar{d} \cdot \bar{c} \cdot \bar{b} \cdot \bar{a} \cdot + \bar{e} \cdot \bar{d} \cdot \bar{c} \cdot b \cdot \bar{a} + \bar{e} \cdot \bar{d} \cdot c \cdot \bar{b} \cdot \bar{a} +$$

(2)

1.1.1. Simplificación mediante álgebra booleana

1.1.2. Simplificacioón mediante mapas de Karnaugh

1.1.3. Implementación mediante compuertas AND, OR y NOT

1.1.4. Implementación mediante compuertas NAND

1.2. Segunda expresión: producto de sumas

1.2.1. Simplificación mediante álgebra booleana

1.2.2. Simplificacioón mediante mapas de Karnaugh

1.2.3. Implementación mediante compuertas AND, OR y NOT

1.2.4. Implementación mediante compuertas NAND

2. Ejercicio 4 - Conversor a codigo de Gray

Para esté ejercicio, realizamos el desarrollo de un circuito lógico capaz de convertir un número binario de 4 bits a su equivalente de código de Gray, esto resulta en la siguiente tabla de verdad:

Entrada				Salida			
X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

De la tabla de verdad obtenemos las siguientes ecuaciones en función de los mintérminos:

$$Y_4 = m_1 + m_2 + m_5 + m_6 + m_9 + m_{10} + m_{13} + m_{14}$$
$$Y_3 = m_2 + m_3 + m_4 + m_5 + m_{10} + m_{11} + m_{12} + m_{13}$$
$$Y_2 = m_4 + m_5 + m_6 + m_7 + m_8 + m_9 + m_{10} + m_{11}$$
$$Y_1 = m_8 + m_9 + m_{10} + m_{11} + m_{12} + m_{13} + m_{14} + m_{15}$$

Que al reemplazar cada mintérmino por su correspondiente expresión obtenemos:

$$Y_4 = \overline{X_1} \cdot \overline{X_2} \cdot \overline{X_3} \cdot X_4 + \overline{X_1} \cdot \overline{X_2} \cdot X_3 \cdot \overline{X_4} + \overline{X_1} \cdot X_2 \cdot \overline{X_3} \cdot X_4 + \overline{X_1} \cdot X_2 \cdot X_3 \cdot \overline{X_4} + X_1 \cdot \overline{X_2} \cdot \overline{X_3} \cdot X_4 + X_1 \cdot \overline{X_2} \cdot X_3 \cdot \overline{X_4} + X_1 \cdot X_2 \cdot \overline{X_3} \cdot X_4 + X_1 \cdot X_2 \cdot X_3 \cdot \overline{X_4}$$

$$Y_3 = \overline{X_1} \cdot \overline{X_2} \cdot X_3 \cdot \overline{X_4} + \overline{X_1} \cdot \overline{X_2} \cdot X_3 \cdot X_4 + \overline{X_1} \cdot X_2 \cdot \overline{X_3} \cdot \overline{X_4} + \overline{X_1} \cdot X_2 \cdot \overline{X_3} \cdot X_4 + X_1 \cdot \overline{X_2} \cdot X_3 \cdot \overline{X_4} + X_1 \cdot \overline{X_2} \cdot X_3 \cdot X_4 + X_1 \cdot X_2 \cdot \overline{X_3} \cdot \overline{X_4} + X_1 \cdot \overline{X_2} \cdot X_3 \cdot X_4$$

$$Y_2 = \overline{X_1} \cdot X_2 \cdot \overline{X_3} \cdot \overline{X_4} + \overline{X_1} \cdot X_2 \cdot \overline{X_3} \cdot X_4 + \overline{X_1} \cdot X_2 \cdot X_3 \cdot \overline{X_4} + \overline{X_1} \cdot X_2 \cdot X_3 \cdot X_4 + X_1 \cdot \overline{X_2} \cdot \overline{X_3} \cdot \overline{X_4} + X_1 \cdot \overline{X_2} \cdot \overline{X_3} \cdot X_4 + X_1 \cdot \overline{X_2} \cdot X_3 \cdot \overline{X_4} + X_1 \cdot \overline{X_2} \cdot X_3 \cdot X_4$$

$$Y_1 = X_1 \cdot \overline{X_2} \cdot \overline{X_3} \cdot \overline{X_4} + X_1 \cdot \overline{X_2} \cdot \overline{X_3} \cdot X_4 + X_1 \cdot \overline{X_2} \cdot X_3 \cdot \overline{X_4} + X_1 \cdot \overline{X_2} \cdot X_3 \cdot X_4 + X_1 \cdot X_2 \cdot \overline{X_3} \cdot \overline{X_4} + X_1 \cdot X_2 \cdot \overline{X_3} \cdot X_4 + X_1 \cdot X_2 \cdot X_3 \cdot \overline{X_4} + X_1 \cdot X_2 \cdot X_3 \cdot X_4$$

Tenemos unas funciones muy larga y como las tenemos expresadas en mintérminos podemos simplificarlas por medio del mapa de Karnaugh. Ésto nos da a lugar a los siguientes mapas de Karnaugh y funciones de salida simplificadas:

X_3X_4 \ X_1X_2	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$Y_4 = X_3 \cdot \overline{X_4} + \overline{X_3} \cdot X_4$

Mapa de Karnaugh y formula de Y_4

X_3X_4 \ X_1X_2	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	1	0	0	1
10	1	0	0	1

$Y_3 = X_2 \cdot \overline{X_3} + \overline{X_2} \cdot X_3$

Mapa de Karnaugh y formula de Y_3

X_3X_4 \ X_1X_2	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$Y_2 = X_1 \cdot \overline{X_2} + \overline{X_1} \cdot X_2$

Mapa de Karnaugh y formula de Y_2

X_3X_4 \ X_1X_2	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	1	1
10	0	0	1	1

$Y_1 = X_1$

Mapa de Karnaugh y formula de Y_1

De los valores obtenidos podemos realizar el siguiente circuito conformado por compuertas OR, AND y NOT:

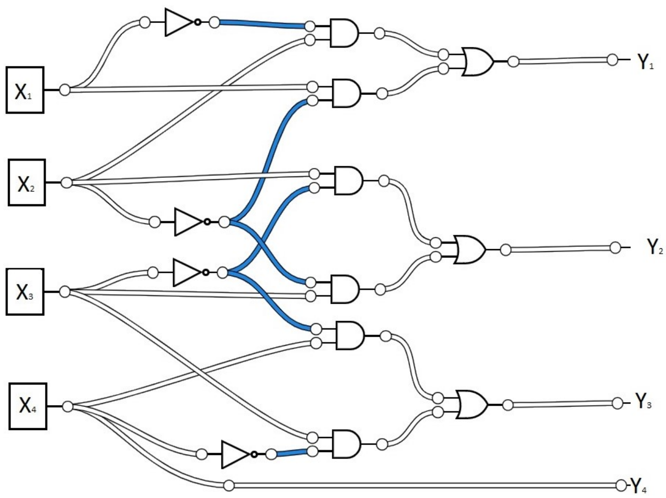

Figura 1: Implementación del conversor a código de Gray