



Tecnología



62

Supuestos prácticos del tema 62

Supuestos prácticos. TEMA 62

PUERTAS LÓGICAS. TÉCNICAS DE DISEÑO Y SIMPLIFICACIÓN DE FUNCIONES LÓGICAS.

62.1.- Exámen de oposición de la Comunidad Valenciana. 2009

Enunciado

Una función lógica F está definida por la tabla de verdad que se adjunta.

| X | Y | Z | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

- a) Obtener las formas canónicas de la función como suma de productos (minterms) y como producto de sumas (maxterms).
- b) Simplificar la expresión lógica mediante el método de Karnaugh.
- c) Obtener los circuitos lógicos con puertas NAND de la expresión lógica anterior.

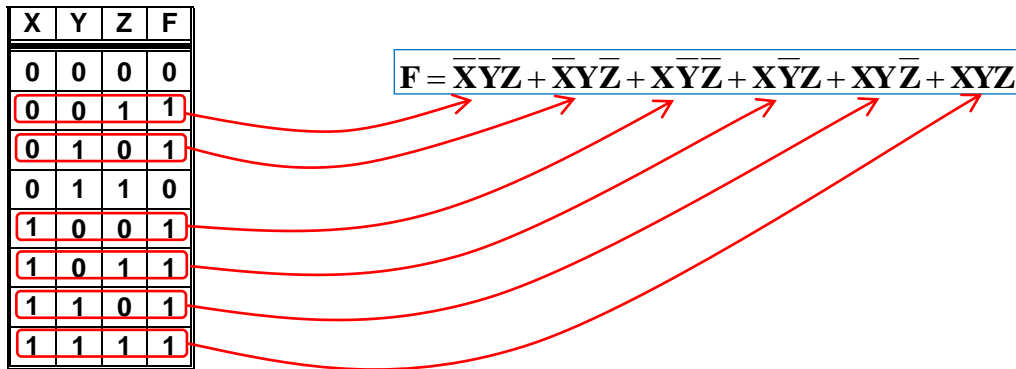
* * * * *

Solución

- a) Obtener las formas canónicas de la función como suma de productos (minterms) y como producto de sumas (maxterms).

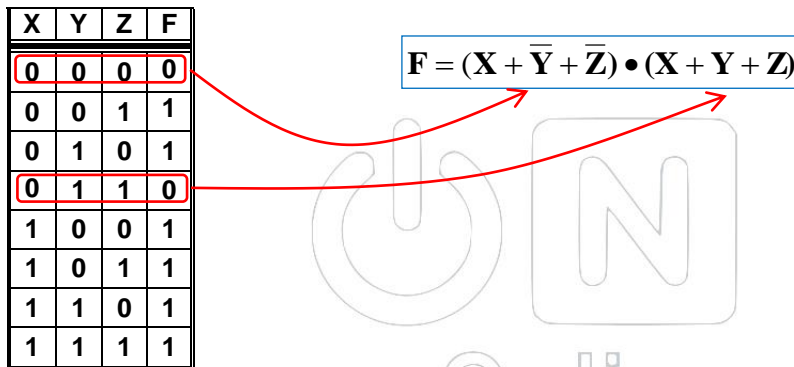
1ª Forma canónica (minterms)

Los minterms se obtienen como suma lógica de los términos para los cuales la función vale 1. Cada término estará formado por el producto de las variables, que estarán negadas si valen 0 y sin negar si valen 1:



2ª Forma canónica (maxterms)

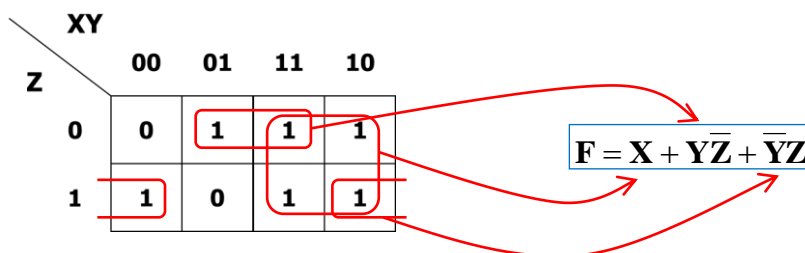
Los maxterms se obtienen como producto lógico de los términos para los cuales la función vale 0. Cada término estará formado por la suma de las variables, que estarán negadas si valen 1 y sin negar si valen 0:



b) Simplificar la expresión lógica mediante el método de Karnaugh.

Aunque se puede usar el mapa de Karnaugh tanto con los minterm como con los maxterms, si no se indica específicamente en el enunciado del problema, se procederá a usar la 1ª forma canónica, es decir, usaremos los minterms.

En el mapa de Karnaugh, situaremos los minterms (combinación de variables donde F=1) en la posición adecuada y realizaremos lazos de 2ⁿ celdas con valor 1. Eliminandose en cada lazo las variables que cambian de valor:

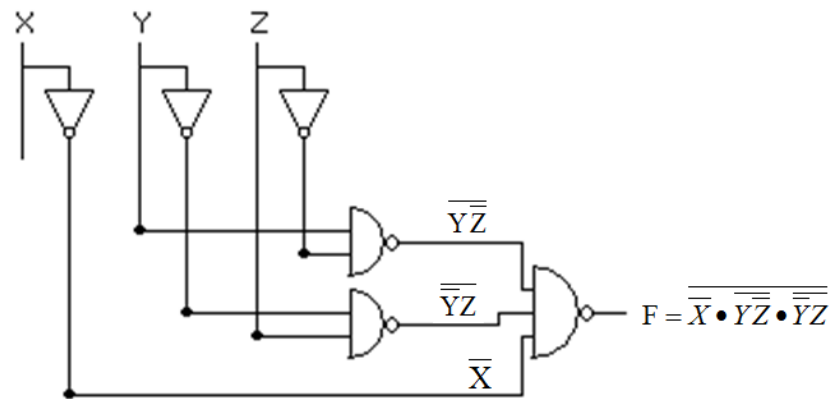


- c) Obtener los circuitos lógicos con puertas NAND de la expresión lógica anterior.

Para representar la función homeneizada sólo con puertas NAND, tendremos que aplicar a la función obtenida con la 1ª forma canónica los Teoremas de Morgan, convirtiendo las sumas de cada término en productos:

$$F = X + Y\bar{Z} + \bar{Y}Z = \overline{\overline{X + Y\bar{Z} + \bar{Y}Z}} = \overline{\overline{X} \cdot \overline{Y\bar{Z}} \cdot \overline{\bar{Y}Z}} = \overline{\overline{X} \cdot \overline{Y\bar{Z}} \cdot \overline{\bar{Y}Z}}$$

Una vez conseguida la función solo con productos realizamos el circuito lógico:



epOnline

62.2.- Exámen de oposición de la Comunidad de Murcia. 1998

Enunciado

Diseñar un circuito comparador de 2 palabras de 3 bits con puertas lógicas y que nos dé 3 salidas:

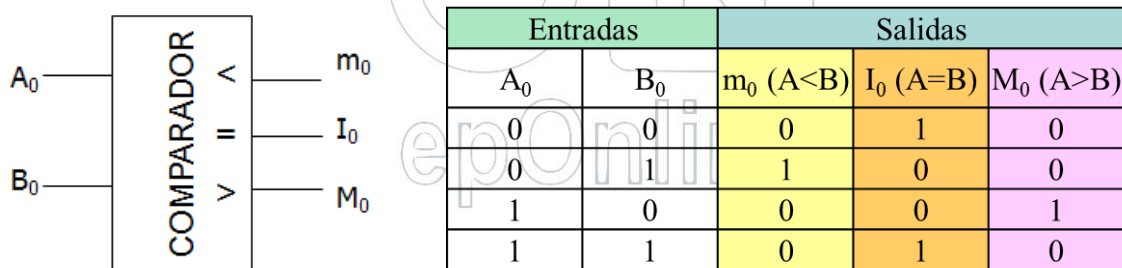
- I cuando A=B
- M cuando A>B
- m cuando A<B

* * * * *

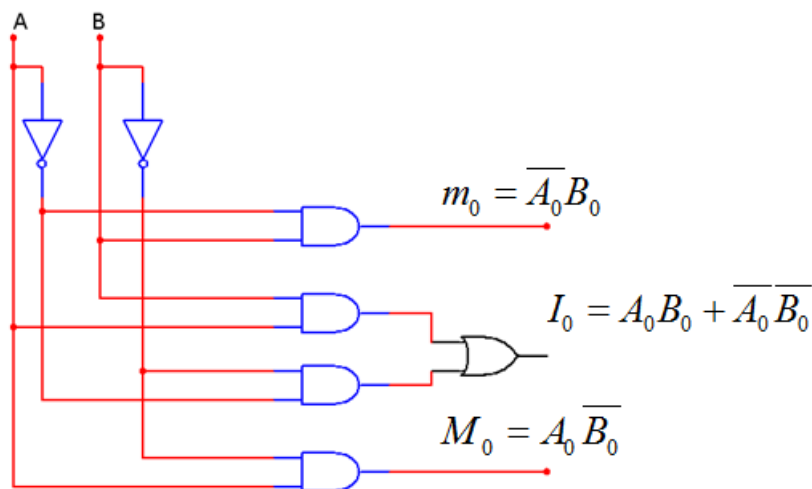
Solución

Al tener 6 bits de entrada (3 por cada palabra), si intentamos hacer la tabla de verdad para el comparador completo, nos saldrán $2^6=64$ combinaciones, siendo muy compleja su simplificación.

Por ello, vamos a trabajar inicialmente con 1 bit de cada palabra por separado:



Siendo su circuito lógico el siguiente:



Para cada bit el circuito sería idéntico. Ahora solo nos falta establecer las condiciones que debe cumplir el comparador completo en función de los comparadores individuales:

Condición de IGUAL (I):

$$\left. \begin{array}{l} A_2 = B_2 \Rightarrow I_2 \\ \text{y} \\ A_1 = B_1 \Rightarrow I_1 \\ \text{y} \\ A_0 = B_0 \Rightarrow I_0 \end{array} \right\} I = I_2 I_1 I_0$$

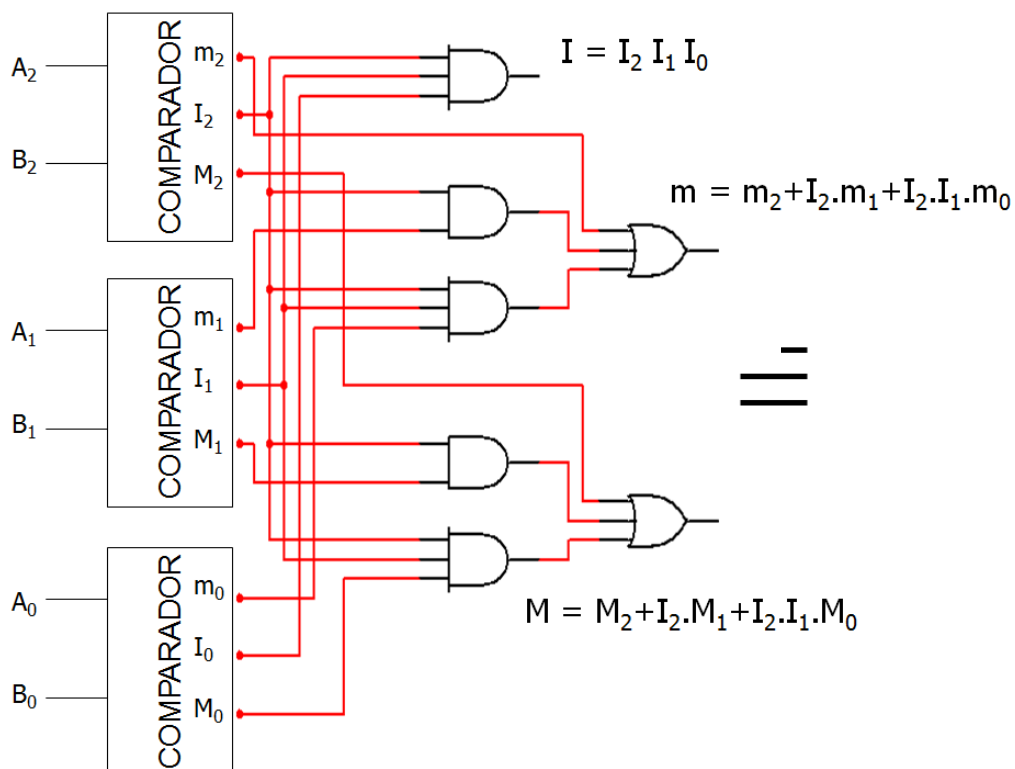
Condición de menor (m):

$$\left. \begin{array}{l} A_2 < B_2 \Rightarrow m_2 \\ \text{o} \\ A_2 = B_2 \text{ y } A_1 < B_1 \Rightarrow I_2 \cdot m_1 \\ \text{o} \\ A_2 = B_2 \text{ y } A_1 = B_1 \text{ y } A_0 < B_0 \Rightarrow I_2 \cdot I_1 \cdot m_0 \end{array} \right\} m = m_2 + I_2 \cdot m_1 + I_2 \cdot I_1 \cdot m_0$$

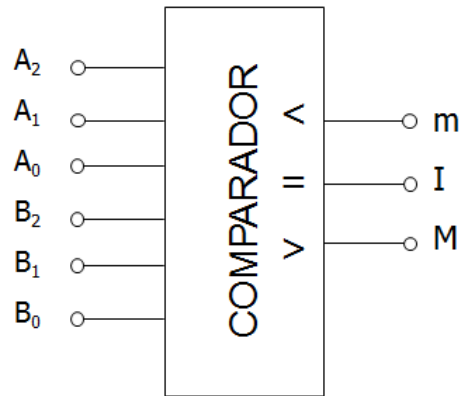
Condición de MAYOR (M):

$$\left. \begin{array}{l} A_2 > B_2 \Rightarrow M_2 \\ \text{o} \\ A_2 = B_2 \text{ y } A_1 > B_1 \Rightarrow I_2 \cdot M_1 \\ \text{o} \\ A_2 = B_2 \text{ y } A_1 = B_1 \text{ y } A_0 > B_0 \Rightarrow I_2 \cdot I_1 \cdot M_0 \end{array} \right\} M = M_2 + I_2 \cdot M_1 + I_2 \cdot I_1 \cdot M_0$$

Ya solo nos queda representar el circuito lógico completo para comparar las 2 palabras de 3 bits:



El circuito anterior, representado en forma de bloque compacto sería así:



* * * * *

Supuesto práctico demo de la especialidad de Tecnología del Cuerpo de Profesores de Enseñanza Secundaria. epOnline

