

**Profesor: Juan J Pombo**

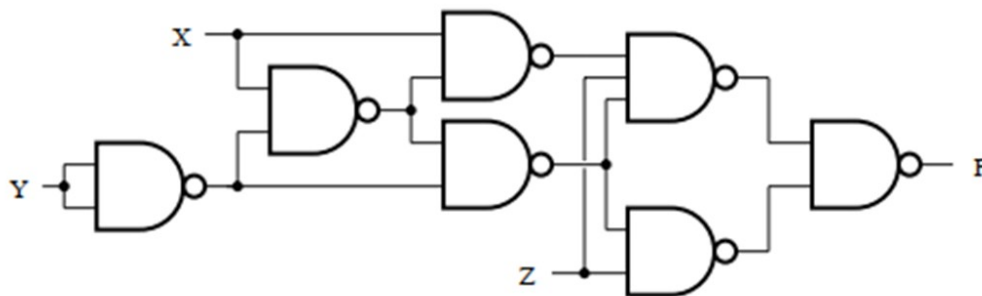
***Boletín 1***

1.- Durante los seminarios hemos desarrollado un circuito conversor de código binario a gray para codificaciones de 4-bit. El procedimiento utilizado ha consistido en desarrollar mapas de Karnaugh a partir de las tablas de verdad, simplificar las funciones y obtener, por inspección, una solución basada en puertas EXOR. Se propone ahora diseñar la operación inversa, **gray a binario**. El circuito que resulta debería ser bastante “similar” al de la conversión anterior.

a) Generar las tablas de verdad y los mapas de Karnaugh para simplificar la función. Obtener un circuito de conversión operativo. Probar el circuito en el simulador Logisim (se ha de entregar el fichero de Logisim)

b) Construir en Logisim un circuito más completo, como un “macro-bloque” que pueda ser utilizado para realizar las dos conversiones, cambiando de binario a gray o gray a binario, según el valor de un bit de control sea 1 o 0, reutilizando en la medida de lo posible el circuito anterior (evitar duplicar completamente el número de puertas EXOR).

2.- Para el circuito de la *figura 2.1* obtener la tabla de verdad y la expresión booleana de la función que realiza. Simplificar y obtener un circuito equivalente que también utilice solamente puertas NAND. Se valorará el nivel de minimización obtenido.



*Fig 2.1*

3.- Considera las funciones F y G que se derivan de la siguiente tabla de verdad:

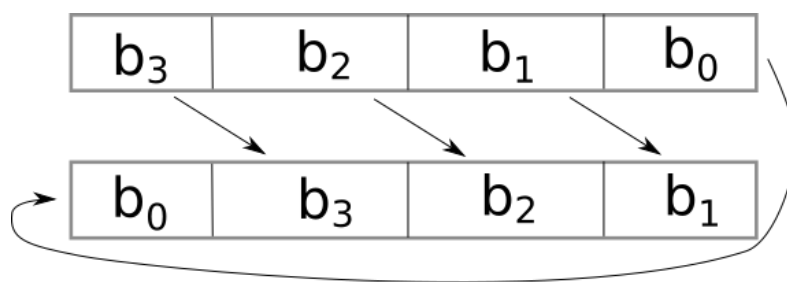
A	B	C	F(A,B,C)	G(A,B,C)
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	1
1	1	1	1	1

a) Deducir las expresiones lógicas para cada una de las funciones F y G.

b) Implementar cada una de las funciones F y G de las siguientes maneras:

- Con puertas lógicas cualesquiera
- Usando solamente puertas lógicas de 2 entradas (pueden usarse inversores)
- Usando solamente puertas NAND de 2 entradas.

4.- Un “desplazador de barril” (*barrel-shift*) es un circuito muy útil en unidades aritmético lógicas. En su versión combinacional es un circuito capaz de poner en la salida la palabra de bits de la entrada con sus bits desplazados de una manera peculiar: los bits que al desplazar hacia un lado quedan fuera de la representación se introducen por el lado contrario de la palabra. En la figura siguiente se muestra el ejemplo de un desplazador de 1 bit a la derecha. El bit 0 que desaparecería por la derecha se reintroduce por la izquierda de la palabra resultado.



En los seminarios de problemas se ha introducido la forma de implementar este circuito valiéndose de multiplexadores.

Se pide implementar un desplazador de 4 bits utilizando MUX 4:1. El desplazador debe ser capaz de implementar las siguientes funciones:

- Retener los bits de la entrada en la salida sin modificación alguna.
- Desplazar en modo “barril/circular” 1 bit a la derecha
- Desplazar en modo “barril/circular” 2 bit a la derecha
- Desplazar en modo “barril/circular” 1 bit a la izquierda.

Especificar los pasos seguidos en el diseño y probar el circuito en Logisim, entregando el diseño.