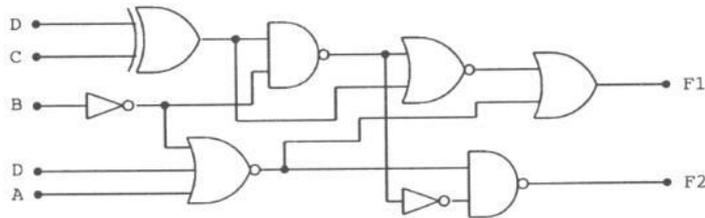


EJERCICIOS

- 1.- Dado el diagrama lógico de la figura



- Obtener las expresiones canónicas de F_1 y F_2 .
 - Obtener las expresiones simplificadas en suma de productos y en producto de sumas para F_1 y F_2 .
 - Implementar F_1 y F_2 con el mínimo número de puertas NAND de dos entradas.
 - Implementar F_1 y F_2 con el mínimo número de puertas NOR de dos entradas.
- 2.- Demostrar algebraicamente las siguientes propiedades de la función XOR:

$$A \oplus B = A' \oplus B', \quad (A \oplus B)' = A' \oplus B = A \oplus B' \quad \text{Y} \quad A \cdot (B \oplus C) = (A \cdot B) \oplus (A \cdot C).$$

NO obtener un código Hamming con posibilidad de corrección de errores en un bit, para un código fuente Gray de cuatro bits. Implementar el emisor Hamming y el detector de errores con posibilidad de corrección automática.

NO dibujar el diagrama lógico y la tabla de verdad de un codificador binario sin prioridad de 8×3 .

- Diseñar con puertas lógicas un codificador con prioridad de 4×2 , que además esté dotado de las siguientes características:
 - Entrada de habilitación.
 - Salida que indique petición de servicio.
 - Posibilidad de conexión entre bloques similares para obtener codificadores con mayor número de líneas.
- Utilizando el bloque diseñado en el ejercicio anterior, obtener un codificador con prioridad de 8×3 con las mismas características.
- Se desea diseñar un circuito para monitorizar el estado de un BUS (conjunto de líneas binarias de información) de 16 líneas (B_0 - B_{15}). Si cualquier línea del BUS toma el valor "1", el circuito debe activar con un "1" una línea llamada "uno presente". También debe generar una palabra Z de cuatro bits en binario natural, indicando la línea B_i del BUS activada. Si aparecen varios unos simultáneamente en sendas líneas del BUS, sólo ha de indicarse la línea B_i que tenga el índice más alto.

Realizar el diseño utilizando como elemento base el codificador con prioridad de la figura 6.3 y las puertas lógicas necesarias.

8.- Realizar las siguientes extensiones de decodificadores:

- a) Obtener un decodificador 4x16 con unidades 3x8.
- b) Obtener un decodificador 5x32 con unidades 3x8.
- c) Obtener un decodificador 5x32 con unidades 4x16.

NO 9.- Diseñar un conversor de binario natural de cuatro bits a BCD natural mediante decodificadores 3x8 como el que se muestra en la figura 7.5 (74138) y las puertas lógicas necesarias.

10.- Utilizando cuatro bloques 74138 (figura 7.5) y las puertas lógicas necesarias, obtener un decodificador 5x32.

NO 11.- Diseñar un conversor de binario natural de cuatro bits a Gray, utilizando un decodificador 4x16 con entradas y salidas activas a nivel alto y las puertas lógicas necesarias.

12.- Realizar las siguientes extensiones de multiplexores:

- a) Obtener un multiplexor 16x1 con unidades 8x1.
- b) Obtener un doble multiplexor 16x1 con unidades 8x1.
- c) Obtener un multiplexor 64x1 con unidades 8x1.

13.- Diseñar un detector de paridad impar para un código de cuatro bits utilizando un multiplexor de ocho canales.

NO 14.- Además de las conexiones estáticas y con bus, existen otros mecanismos intermedios para interconectar distintos elementos (procesadores, memorias, etc.).

Uno de ellos, la red Crossbard, permite conectar dos grupos de unidades $P=\{P_1, P_2, \dots, P_m\}$ y $M=\{M_1, M_2, \dots, M_n\}$, de manera que cada elemento de P puede comunicarse con cualquier elemento de M de forma programada. Esta estructura permite varias transferencias simultáneas de información entre distintas unidades de P y M . Evidentemente para ello se deben evitar colisiones, es decir, que un mismo canal de salida P_i sea requerido por dos o más canales de entrada M_j (o viceversa) en el mismo instante.

Por ejemplo, P puede ser un conjunto de (micro)procesadores y M un conjunto de módulos de memoria principal.

- a) Diseñar una red Crossbard 2x2 a partir de multiplexores y demultiplexores de tamaño adecuado. En la figura siguiente se indican las posibles conexiones entre los elementos de P y M .

EJERCICIOS

NO Deducir las expresiones algebraicas para C_1 , C_2 , C_3 y C_4 en un sumador paralelo de cuatro bits con acarreo anticipado. Dar también las expresiones para el grupo propagador $P(3-0)$ y el grupo generador $G(3-0)$.

- 2.- Diseñar un circuito sumador/restador de números binarios expresados en el convenio signo-magnitud con el formato de seis bits más signo. El diseño debe evitar la posibilidad de desbordamiento aritmético. El resultado vendrá expresado en el mismo convenio de representación.
- 3.- A partir de un cuádruple sumador total, diseñar un conversor de BCD natural a BCD exceso tres o viceversa. El sentido de la conversión se indicará con una señal de control.

NO 4.- A partir de un cuádruple sumador total, diseñar un comparador de números de cuatro bits codificados en binario natural. El comparador tendrá las salidas $A > B$, $A = B$ y $A < B$.

- 5.- Diseñar un circuito aritmético que calcule en valor absoluto la diferencia de dos números codificados en binario natural.
- 6.- Dados tres números A, B y C de cuatro bits codificados en binario natural, diseñar un circuito que realice la suma de A con el mayor de B y C: Si $B = C$ el resultado debe ser A.
- 7.- Dados dos números binarios A y B de cinco bits expresados en valor absoluto y signo, diseñar un circuito que opere según la siguiente tabla y que elimine la posibilidad de desbordamiento aritmético.

C_2	C_1	SALIDA
0	0	A + B en complemento a uno
0	1	A - B en complemento a uno
1	0	A + B en complemento a dos
1	1	A - B en complemento a dos

NO 8. Utilizando como elemento base los sumadores, diseñar un conversor de BCD natural de dos cifras a binario natural.

NO 9. En el teclado hexadecimal de la figura, aparece un uno lógico en la fila y el la columna en cuya intersección se encuentra la tecla pulsada. Diseñar un circuito que indique en BCD la tecla pulsada. Si son activadas varias teclas al mismo tiempo, se indicará la más alta.