

Problemas propuestos

- 1.1. Diseñar un circuito multiplexor con prioridad de 4 bits. El circuito tiene 4 entradas de datos (I_3 - I_0), 4 entradas de selección (S_3 - S_0) y dos salidas Z y G . Cuando una o más de las entradas S están a 1, Z toma el valor de la entrada I_i , siendo i es el índice más alto de las entradas S_i que están a 1; si todas las entradas S_3 - S_0 están a 0, entonces Z toma el valor 0. La salida G se fija a 1 si al menos alguna entrada S_i está a 1, en caso contrario se fija a 0.
- a) Mostrar en una tabla el comportamiento lógico del circuito. Encontrar las ecuaciones lógicas de la salidas Z y G expresándolas en dos niveles y en forma factorizada.
 - b) Implementar la expresión factorizada de Z utilizando multiplexores de 2 entradas.
 - c) Diseñar un multiplexor con prioridad de 16 bits en base a los multiplexores con prioridad de 4 bits diseñados.
 - d) Realizar una descripción VHDL del multiplexor con prioridad.

Problemas propuestos

- 2.1. Construir un multiplexor de 5 entradas
- a) utilizando puertas lógicas.
 - b) utilizando multiplexores de dos entradas.
- 2.2. Un circuito de “desplazamiento en barril” (“barrel-shifter”) mueve los datos de entrada de forma que aparezcan en la salida girados el número de posiciones marcados por las señales de control. Construir utilizando multiplexores un “barrel-shifter” de 4 bits de entrada ($a_3a_2a_1a_0$) y 4 bits de salida ($z_3z_2z_1z_0$) con 4 posibles desplazamientos (dos señales de control c_1c_0):
- $(c_1c_0) = 0 \Rightarrow (z_3z_2z_1z_0) = (a_3a_2a_1a_0),$
 - $(c_1c_0) = 1 \Rightarrow (z_3z_2z_1z_0) = (a_2a_1a_0a_3),$
 - $(c_1c_0) = 2 \Rightarrow (z_3z_2z_1z_0) = (a_1a_0a_3a_2),$
 - $(c_1c_0) = 3 \Rightarrow (z_3z_2z_1z_0) = (a_0a_3a_2a_1).$
- Realizar la descripción VHDL de este circuito.

Problemas propuestos

- 3.1. Diseñar un multiplexor de 16 entradas utilizando 4 multiplexores triestado de 4 entradas con habilitador (el circuito deshabilitado queda en alta impedancia) y un decodificador 2 a 4. Indicar como debería diseñarse el circuito con dos chips 74'153, un chip 74'139 y una puerta lógica.
- 3.2. Se quiere diseñar un decodificador de 40 direcciones de 0 a 39 utilizando decodificadores binarios (2 a 4, 3 a 8, 4 a 16, etc). Indicar cuál es el número mínimo de decodificadores binarios que hay que utilizar y realizar el diseño del decodificador utilizando los decodificadores binarios y las puertas lógicas que sean necesarias (un inversor).
- 3.3. Diseñar un circuito decodificador del código de Hamming capaz de recuperar un error simple en un código (M0M1M2M3) con bits de paridad (P0P1P2) par utilizando puertas EXOR (para determinar F2F1F0 la dirección del bit erróneo, y para complementar dicho bit) y un 3 a 8 DEC (para indicar el bit erróneo en función de F2F1F0).

1	2	3	4	5	6	7	
P0	P1	M0	P2	M1	M2	M3	
×		×		×		×	F0
	×	×			×	×	F1
			×	×	×	×	F2

Problemas propuestos

- 4.1. Realizar un multiplicador para números A (a_1a_0) y B (b_1b_0) de 2 bits en complemento-2, utilizando decodificadores 74LS138 y puertas NAND de 4 entradas (74LS20) o de 8 entradas (74LS30). El resultado P ($p_3p_2p_1p_0$) es de 4 bits en complemento-2.
- 4.2. Diseñar un circuito que realice simultáneamente:
- a) La suma de dos números positivos X ($X_1.H, X_0.H$) e Y ($Y_1.H, Y_0.H$) de dos bits.
 - b) El producto de dos números positivos M ($M_1.H, M_0.L$) e N ($N_1.L, N_0.H$) de dos bits.

El circuito debe tener sólo cuatro bits de entrada y dos grupos diferenciados de salidas para cada una de las operaciones, todas las salidas deben ser de polaridad positiva. El diseño debe realizarse utilizando el menor número de decodificadores 74'138, y el menor número de puertas NAND (suponer que es posible cualquier número de entradas en las puertas).

Problemas propuestos

- 5.1. Implementar las siguientes funciones utilizando el menor número de decodificadores 3 a 8, y 2 a 4.
- a) $F(A,B,C,D) = \sum(0,1,4,7,12,13,14)$ con A.H, B.H, C.H y D.H.
 - b) $F(A,B,C,D,E) = \sum(0,2,4,5,7,8,9,10,13,17,26,27)$ con A.H, B.L, C.L, D.L y E.H.
- 5.2. Implementar las siguientes funciones lógicas utilizando un único multiplexor, lo más pequeño posible.
- a) $F(A,B,C,D) = \sum(0,1,4,7,12,13,14) + \sum\emptyset(5,10,15)$ con A.H, B.H, C.H y D.H, F.H.
 - b) $F(A,B,C,D) = \sum(0, 1, 3, 4, 6, 10) + \sum\emptyset(2, 7, 8, 15)$ para A.L, B.H, C.H, D.L, F.H.
- 5.3. Implementar utilizando un multiplexor de cuatro entradas y el menor número de multiplexores de dos entradas la función lógica
- $$F.H = F(A, B, C, D) = \prod(1, 5, 9, 11, 12, 13, 14) \cdot \prod\emptyset(7, 15),$$
- para entradas A.H, B.L, C.H y D.L.

Problemas propuestos

- 6.1. Obtener las expresiones lógicas minimizadas que permiten encontrar cuál de 7 líneas de entrada A1, A4, A6, A8, A9, A13, A14 está puesta a valor lógico 1, dando como resultado su correspondiente codificación binaria: por ejemplo, A8 daría como resultado 8 en la salida (codificado en binario).
- a) Solo puede estar una línea a valor lógico 1.
 - b) Varias líneas de entrada están simultáneamente a 1, pero la salida tomará el valor binario de la línea de índice más bajo.
- 6.2. Encontrar las ecuaciones lógicas que permiten definir un circuito codificador con prioridad baja de 8 bits de entrada (I7-I0) y salidas en código Gray (de más a menos significativas: A B C).
- 6.3. Construir un circuito codificador binario de 8 a 3 con prioridad baja tomando como base el circuito codificador 74LS148, y el menor número posible de puertas lógicas que sean necesarias. Se permite definir como mejor convenga la polaridad de las entradas y de las salidas.

Problemas propuestos

- 7.1. Diseñar un circuito multiplexor con prioridad de 4 bits. El circuito tiene 4 entradas de datos (I_3 - I_0), 4 entradas de selección (S_3 - S_0) y dos salidas Z y G . Cuando una o más de las entradas S están a 1, Z toma el valor de la entrada I_i , siendo i es el índice más alto de las entradas S_i que están a 1; si todas las entradas S_3 - S_0 están a 0, entonces Z toma el valor 0. La salida G se fija a 1 si al menos alguna entrada S_i está a 1, en caso contrario se fija a 0. Utilizar en el diseño circuitos MSI convencionales: un 74LS148 (8 a 3 HPRI COD) y un circuito 74LS153 (4-input MUX).
- 7.2. Realizar un circuito conversor del código BCD con pesos (8, 7, -2, -4) al código NBCD (8, 4, 2, 1) usando únicamente circuitos 74'138 (decodificador 3 a 8) y 74'147 (codificador con prioridad alta 10 a 4).
- 7.3. Se quiere realizar un circuito de 8 entradas (I_7 - I_0) y 8 salidas (O_7 - O_0), tal que la salida muestra la entrada pero eliminando todos los unos menos el más significativo. Por ejemplo, si $I = "01101101"$, $O = "01000000"$; si $I = "00010110"$, $O = "00010000"$, etc. Si todos los bits de la entrada son 0, los de la salida también: $I = "00000000"$, $O = "00000000"$.
- a) Realizar un código VHDL para la descripción del problema.
 - b) Implementar el circuito con un circuito codificador 8 a 3 74LS148 y un circuito decodificador 3 a 8 74LS138. Suponer las entradas y salidas $I_7.L, \dots I_0.L; O_7.L, \dots O_0.L$ en polaridad negativa.

Problemas propuestos

- 8.1. Diseñar utilizando únicamente semisumadores y sumadores completos un circuito digital que realice la multiplicación de un número binario de dos bits por otro de tres bits.
- 8.2. Diseñar un circuito que calcule la distancia de Hamming de dos palabras A y B de 7 bits usando el menor número posible de puertas lógicas y/o de dispositivos MSI.
- 8.3. Diseñar utilizando elementos MSIs (sumadores y decodificadores) un circuito que calcule el resultado de una votación de siete votos. Cada voto aparece codificado mediante dos bits I_1I_0 , de forma que la abstención se representa por 00, 'Si' por 01, 'No' por 10 y los votos nulos aparecen como 11. El resultado de la votación debe darse indicando el número de votos de cada tipo. Realizar lo mismo para una votación de ocho votos.
- 8.4. Realizar la suma de cuatro números de dos bits A (a_2a_1), B (b_2b_1), C (c_2c_1) y D (d_2d_1) y tres números de 1 bit, E (e_1), F (f_1) y G (g_1) utilizando el menor número posible de sumadores completos ("full-adders").

Problemas propuestos

9.1. Diseñar un circuito que realice la operación aritmética:

$$O = 5X + 2Y + Z$$

para operandos X (x_1x_0), Y (y_1y_0) y Z (z_1z_0) de dos bits, utilizando el menor número posible de semisumadores de dos bits de operandos de entradas A (a_1a_0) y B (b_1b_0).

9.2. Un sistema digital accede a los elementos de una matriz de 12×12 (144 elementos) que están almacenados en memoria. Para acceder a un elemento de la matriz el sistema utiliza la posición de filas F y la posición de columnas C , ambas de 4 bits (F_3-F_0 , C_3-C_0) en código binario con valores entre 0 y 11. Sin embargo, la memoria utilizada sólo tiene un bus de direcciones D .

Diseñar un circuito que genere la dirección D (8 bits D_7-D_0 , valores entre 0 y 143) del elemento de la matriz en memoria de la forma $D = 12 \cdot F + C$. Utilizar el menor número posible de sumadores (preferentemente 74LS83) para realizar la implementación.

9.3. Realizar el diseño de un circuito que sume dos dígitos NBCD, dando el resultado en código NBCD, utilizando puertas lógicas cuando sea necesario. Indicar como puede utilizarse este circuito para sumar números NBCD de más de un dígito,

Ayuda: Hay que sumar 6.

Problemas propuestos

- 10.1. Realizar el diseño de un comparador de dos números A y B de cuatro bits tomando como base el sumador de números binarios de cuatro bits 74'83, utilizando puertas lógicas cuando sea necesario. El circuito debe generar tres salidas: O1 ($A = B$), O2 ($A > B$), O3 ($A < B$).
- 10.2. Diseñar un circuito que realice la operación aritmética $Z = A + 1$ cuando A es igual B, y la operación aritmética $Z = (A - B) - 1$ cuando A es mayor que B, donde A y B son números binarios de cuatro bits, siendo siempre A mayor o igual que B. Implementar el circuito utilizando como base el sumador 74'83 y otros elementos MSI y puertas lógicas.
- 10.3. Diseñar un circuito comparador COMP16 de números binarios de 16 bits utilizando únicamente cuatro comparadores 74'85 de cuatro bits y un máximo de dos niveles de lógica. El circuito debe tener dos entradas A y B de 16 bits, tres entradas de expansión I($A=B$), I($A>B$) y I($A<B$), y tres salidas O($A=B$), O($A>B$) y O($A<B$), como el circuito 74'85. Indicar razonadamente el número máximo de bits de los números que se pueden comparar mediante un circuito formado por 3 COMP16 y 3 circuitos 74'85, y un máximo de tres niveles de comparadores 74'85 (sabiendo que en los circuitos COMP16 ya hay dos niveles).

Problemas propuestos

- 11.1. Se quieren diseñar circuitos digitales que realicen la comparación de dos números binarios con signo X e Y . Se deben obtener tres salidas que indiquen cuando $X = Y$, $X > Y$ o $X < Y$. Se debe utilizar en lo posible comparadores comerciales como el 74'85, y otros elementos lógicos cuando sea necesario. Indicar en cada caso el razonamiento o las ecuaciones lógicas que llevan al diseño final.
Recordar: los números positivos siempre son mayores que los negativos; entre números positivos el mayor es el de mayor valor absoluto ($5 > 3$), entre números negativos el mayor es el de menor valor absoluto ($-3 > -5$).
- a) Suponer X ($x_3x_2x_1x_0$) e Y ($y_3y_2y_1y_0$) de 4 bits en complemento-2.
 - b) Suponer X ($S_x x_3x_2x_1x_0$) e Y ($S_y y_3y_2y_1y_0$) de cinco bits en formato con bit de signo, donde S_x e S_y son los signos de X y de Y : 0 positivo, 1 negativo. Los otros bits contienen los módulos M_x y M_y de cada número en código binario. Para simplificar un poco el problema suponer que existe el $+0$ pero no existe el -0 (eso evita el problema de evaluar $-0 = +0$).
- 11.2. Indicar como se puede hacer un comparador de dos palabras de 4 bits X e Y que codifican números binarios sin signo cuando las entradas de datos están en lógica negativa, utilizando el comparador 74'85.

Problemas propuestos

- 12.1. Dos tanques de agua A y B tienen cada uno de ellos 8 sensores (entradas A7-A0, B7-B0, el índice 0 es el nivel inferior, el índice 7 el nivel superior) que indican el nivel de agua de cada tanque. El sensor está activado cuando hay agua en el nivel y desactivado en caso contrario. Los tanques tienen unas esclusas que permiten introducir agua (salidas AI, BI) o sacar agua (salidas AO, BO).

Diseñar un circuito para automáticamente sacar agua del tanque (activar AO o BO) que más tiene cuando la suma de los niveles de los tanques pase de 11, o cargar agua en el tanque (activar AI o BI) que menos tiene cuando la suma de los niveles de los tanques sea menor de 5. Usar elementos MSI estándar y/o puertas lógicas.

Calcular el tiempo de propagación máximo del circuito.

- 12.2. Diseñar un circuito que dado como entrada un número binario positivo A de 3 bits realice la operación $Z = 3 \cdot A$ si la entrada de control C está a 0 o la operación $Z = 5 \cdot A$ si la entrada de control C está a 1, usando 1 sumador completo 74LS83 y el menor número de puertas lógicas u otros circuitos MSI (74LS157).

Calcular el tiempo de propagación máximo del circuito.

Problemas propuestos

- 13.1. Realizar la operación X^2 ($X \cdot X$), siendo X un número binario sin signo de tres bits ($X_2X_1X_0$). Utilizar el menor número posible de puertas AND, y semisumadores y sumadores completos de 1 bit. Para simplificar el circuito se recuerda que se puede utilizar la propiedad conmutativa, el teorema de la idempotencia, y que $A \text{ PLUS } A = 2 \cdot A = (A_0)$.

Diseñar un circuito que realice la operación X^2 , siendo X de 4 bits ($X_3X_2X_1X_0$) usando el circuito X^2 para X de 3 bits diseñado anteriormente, el menor número de puertas AND y un único sumador 74LS83 (4-bit full-adder).

- 13.2. Se disponen de dos palabras de 3 bits A ($S_a a_1 a_0$) y B ($S_b b_1 b_0$) que representan números con signo en notación de bit de signo. Se quiere obtener la suma de A y B pero mostrando el resultado en complemento-2. La suma se hará convirtiendo los números A y B a complemento-2 usando decodificadores 74LS138 y puertas NAND de 4 entradas, y luego sumando los números convertidos usando un sumador completo de 4 bits como el 74LS83. Mostrar la implementación del circuito.

Problemas propuestos

- 14.1. Diseñar un circuito sumador para números binarios A y B de cinco bits descritos en código binario con signo, de la forma $(S_a a_3 a_2 a_1 a_0)$ y $(S_b b_3 b_2 b_1 b_0)$, donde S es el bit de signo (0 positivo, 1 negativo), y (a_3-a_0) , (b_3-b_0) la codificación binaria del módulo de los números. El circuito tiene que generar un número de seis bits como resultado $(S_f f_4 f_3 f_2 f_1 f_0)$, f_4-f_0 bits de módulo y S_f de signo). Utilizar como base del diseño un único circuito sumador/restador (S/R) de 4 bits, y otros circuitos lógicos MSI (comparadores, multiplexores, etc) y puertas lógicas. El circuito S/R utiliza dos operandos X e Y de cuatro bits, una señal de control C del tipo de operación (X PLUS Y, X MINUS Y), una salida Z de 4 bits y un bit de salida de acarreo C_o . Los operandos X e Y son números binarios positivos, con la restricción de que en el modo resta X debe ser mayor o igual que Y.
- Explicar en qué se basa el diseño realizado y representar el diseño en modo esquemático con notación de tipo bus para simplificar el dibujo final. Generar un circuito sumador/restador, añadiendo una señal de control K y sólo una puerta lógica al circuito anterior.

Problemas propuestos

15.1. La multiplicación de dos números binarios sin signo A y B puede hacerse utilizando puertas AND para generar los términos producto parciales ($X_{ij} = a_i \cdot b_j$), para sumar posteriormente estos términos mediante diferentes algoritmos (por filas, por columnas, etc). Cuando se trabaja con números con signo en complemento-2, el proceso se complica ya que los X_{ij} pueden tener peso positivo o negativo en el resultado final por lo que deben sumarse o restarse. En este problema se debe realizar la multiplicación de dos números de tres bits A ($a_2 \ a_1 \ a_0$) y B ($b_2 \ b_1 \ b_0$) en notación en complemento-2, cuyos pesos de más significativo a menos significativo son (-4, 2, 1).

¿Cuántos bits debe tener el resultado P (también en complemento-2) de la multiplicación?

Teniendo en cuenta que

$$P = A \cdot B = 16 X_{22} - 8(X_{21} + X_{12}) - 4(X_{20} + X_{02}) + 4X_{11} + 2(X_{01} + X_{10}) + X_{00}.$$

y ordenando de forma adecuada las operaciones de suma y resta a realizar, diseñar un circuito que realice la multiplicación propuesta utilizando el menor número de sumadores (completos o semisumadores, indicando el número de bits de cada sumador) y puertas lógicas básicas.

Problemas propuestos

16.1. La división A/B de dos números A ($a_3a_2a_1a_0$) y B ($b_3b_2b_1b_0$) de 4 bits, calculando el cociente Q ($q_3q_2q_1q_0$) y el resto R ($r_3r_2r_1r_0$), puede realizarse según el siguiente método:

- Tomar el dividendo como $(000a_3a_2a_1a_0)$ y el divisor como $(b_3b_2b_1b_0)$.
- Sea D_1 los 4 bits de la izquierda ($000a_3$) del dividendo, comparar D_1 con el divisor B , si $D_1 \geq B$ el bit del cociente q_3 es 1 y X ($x_3x_2x_1x_0$) = $D_1 - B$; si no el cociente q_3 es 0 y X ($x_3x_2x_1x_0$) = D_1 .
- Tomar D_2 como $(x_2x_1x_0a_2)$, si $D_2 \geq B$ el bit del cociente q_2 es 1 e Y ($y_3y_2y_1y_0$) = $D_2 - B$; si no el cociente q_2 es 0 e Y ($y_3y_2y_1y_0$) = D_2 .
- Tomar D_3 como $(y_2y_1y_0a_1)$, si $D_3 \geq B$ el bit del cociente q_1 es 1 y Z ($z_3z_2z_1z_0$) = $D_3 - B$; si no el cociente q_1 es 0 y Z ($z_3z_2z_1z_0$) = D_3 .
- Tomar D_4 como $(z_2z_1z_0a_0)$, si $D_4 \geq B$ el bit del cociente q_0 es 1 y se genera el resto R ($r_3r_2r_1r_0$) = $D_4 - B$; si no el cociente q_0 es 0 y el resto R ($r_3r_2r_1r_0$) = D_4 .

Diseñar un circuito digital que realice la división usando sumadores 74LS83 (4 chips), multiplexores 74LS157 (4 chips), y cuatro inversores 74LS04 (1 chip), utilizando el algoritmo anterior basado en operaciones de comparaciones y restas.

Calcular el tiempo de propagación máximo del circuito.