

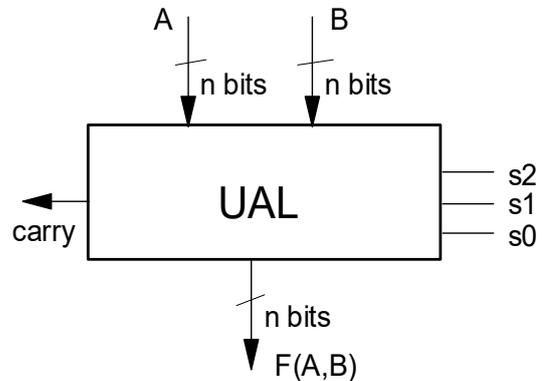


PROBLEMAS DE FUNDAMENTOS DE COMPUTADORES I

TEMA 4

Problemas básicos:

- Los computadores disponen de un circuito combinacional capaz de realizar las operaciones más elementales AND, OR, Suma Aritmética, etc. sobre dos configuraciones binarias de n bits. Dicho circuito se denomina Unidad Aritmético Lógica (UAL). Diseñe una UAL como la mostrada en la figura, capaz de realizar las siguientes funciones: A AND B, A OR B, C1 B, A+B (suma aritmética), desplazamiento de B un bit a la derecha, desplazamiento de B un bit a la izquierda.



El diseño se realizará para n=4 bits. La función que en cada caso realiza la UAL se selecciona mediante las entradas de control s₂, s₁, s₀.

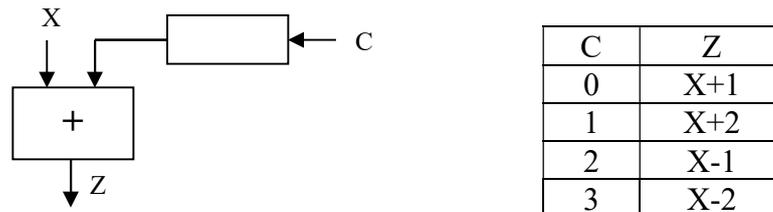
- Diseñe un multiplexor de 8 a 1 usando:
 - 4 multiplexores de 2 a 1 y el mínimo número de puertas.
 - 1 decodificador 3 a 8 y el menor número de puertas.
- Considere la implementación en una ROM de un sumador de números naturales codificados en binario puro con 3 bits que codifica el resultado en BCD. Discuta razonadamente el tamaño mínimo de la ROM e indique el contenido de la ROM en las direcciones (3E)₁₆ y (1D)₁₆.
- Usando un sumador y puertas lógicas diseñe un convertor de MyS a C2 de 8 bits. Ídem para un convertor de C2 a MyS que tenga una salida adicional de overflow.

Problemas adicionales:

- Para la construcción de un teclado de 24 teclas (T₀, T₁, ... T₂₃) se dispone de codificadores de prioridad de 8 entradas. Diseñe el circuito de codificación del teclado usando dichos codificadores más las puertas lógicas que se consideren necesarias, teniendo en cuenta que el circuito funcionará de la siguiente forma:
 - Si sólo se pulsa la tecla T_i, entonces la salida tomará el valor i.
 - Si se pulsan simultáneamente varias teclas, entonces la salida tomará el valor correspondiente al mayor subíndice de las teclas pulsadas.
- Implemente un sistema que tiene 6 entradas de datos (x₅...x₀), 2 entradas de control (s₁,s₀), 4 salidas de datos (z₃...z₀) y cuyo comportamiento viene descrito de la forma siguiente:

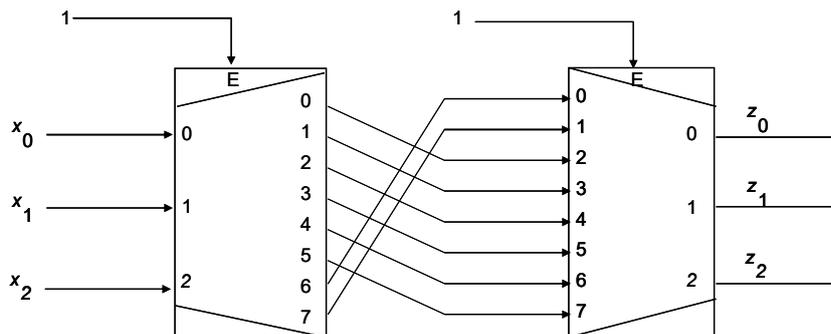
$$\begin{aligned}
 (z_3 \dots z_0) &= (x_5 \dots x_2) & \text{si } (s_1, s_0) &= (01) \\
 &= (x_3 \dots x_0) & \text{si } (s_1, s_0) &= (10) \\
 &= (x_4 \dots x_1) & \text{si } (s_1, s_0) &= (11) \\
 &= (0000) & \text{si } (s_1, s_0) &= (00)
 \end{aligned}$$

7. Diseñe un conversor de código BCD a código Exceso-3 usando:
- Un codificador y un decodificador.
 - Un decodificador y puertas OR.
 - Multiplexores de 16 a 1.
 - Multiplexores de 4 a 1 y puertas lógicas.
 - Una ROM de tamaño mínimo.
8. Un sistema combinacional tiene una entrada X de 4 bits y una salida Z de 4 bits. Los datos, tanto a la entrada como a la salida, son números enteros codificados en C2. El sistema tiene otra entrada de control C que determina la función del sistema de acuerdo con la siguiente tabla:



Diseñe el circuito que tenga la estructura mostrada en la figura y use únicamente un sumador binario de 4 bits, multiplexores y puertas. Se valorará que el número y tamaño de los multiplexores sea el menor posible.

9. Se desea implementar un conversor de números de 8 bits en binario puro a números en BCD mediante una memoria ROM:
- Determine el tamaño mínimo necesario de la memoria, y el esquema de implementación. Indicando claramente el significado de las líneas de direcciones y salida de la ROM.
 - Obtenga el contenido de la ROM en las direcciones $(41)_{10}$ y $(2F)_{16}$
10. Dado el circuito de la figura, obtenga la especificación de Z en función de X . Justifique la respuesta.



11. Calcule el número de puertas AND, OR y NOT necesarias para diseñar implementaciones directas de un:
- Decodificador 3 a 8.
 - Multiplexor 8 a 1.
 - Codificador de prioridad 16 a 4.
12. Calcule el número de:
- Multiplexores 4 a 1 necesarios para implementar en árbol uno de 256 a 1.
 - Multiplexores 2 a 1 necesarios para implementar en árbol uno de 16 a 1.

- c) Multiplexores 2 a 1 necesarios para implementar uno vectorial de 4 a 1 de 8b.
 - d) Decodificadores 1 a 2 necesarios para implementar en árbol uno de 4 a 16.
 - e) Decodificadores 2 a 4 necesarios para implementar en árbol uno de 8 a 256.
13. Diseñe una ROM de $2^{10} \times 16$ bits (2 KiB) usando un decodificador 2 a 4 y el número de módulos ROM de $2^8 \times 8$ bits (256 B) que estime necesarios.
14. Diseñe 3 módulos combinatoriales que calculen el máximo de 2 números de 8 bits considerando, en cada caso, que ambos números están codificados:
- a) En binario puro.
 - b) En MyS.
 - c) En C2.

En todos ellos se utilizará un multiplexor vectorial 2 a 1, un restador binario y el menor número posible de puertas lógicas.

15. Diseñe un multiplicador combinatorial de números binarios sin signo de 4 bits. El multiplicador tiene 2 entradas de 4 bits para los operandos y una salida de 8 bits para el resultado. Use únicamente puertas AND de 2 entradas para calcular los productos parciales y sumadores completos de 4 bits para sumar en cascada dichos productos.
16. Diseñe un conversor de códigos BCD y EX3 usando un sumador/restador de 4 bits, un multiplexor 2 a 1 y las puertas que estime necesarias. El conversor tendrá una entrada de datos de 4 bits, una señal de control para seleccionar si el dato a la entrada debe convertirse de BCD a EX3 o de EX3 a BCD, una salida de datos de 4 bits y una salida de error para indicar si el código que hay a la entrada no puede convertirse por ser inválido. Es decir, esta salida de error deberá activarse cuando habiendo seleccionado una conversión BCD a EX3, en la entrada de datos haya, por ejemplo, 1011 (dado que no es un código BCD válido), sin embargo no deberá hacerlo si la conversión EX3 a BCD ha sido seleccionada (dado que representa a un 8 en EX3).
17. Diseñe un módulo combinatorial que ordene 3 números enteros codificados en binario puro con 8 bits. El módulo tendrá 3 entradas de 8 bits para los operandos y 3 salidas de 8 bits para los resultados (por una de ellas siempre saldrá el mayor de los tres operandos, por otra siempre el intermedio y por la última siempre el menor). Utilice el menor número posible de comparadores de magnitud, multiplexores vectoriales 2 a 1 de 8 bits y puertas lógicas.
18. Calcule el coste, el retardo de propagación (máximo) y el retardo de contaminación (mínimo) de un sumador binario de 4 bits utilizando los datos de la biblioteca de celdas presentada en teoría. Ídem para un sumador de 8 bits. Obtenga expresiones genéricas de coste y retardo para un sumador de n bits.

Problemas de examen:

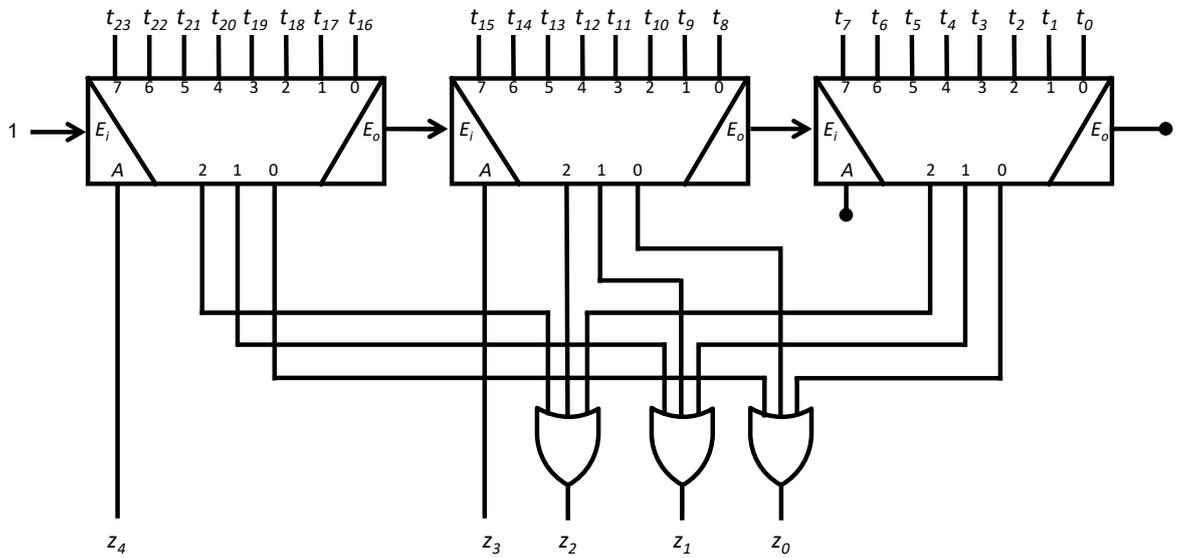
19. (Septiembre 2017) Implementar un sumador/incrementador/decrementador utilizando un sumador binario de 8 bits, multiplexores y puertas lógicas. El sistema tendrá una señal de control S y otra I/D, de manera que si S=1 el sistema realiza una suma de 8 bits, si S=0 el sistema incrementará cuando I/D=0 y decrementará cuando I/D=1.
20. (Septiembre 2018) Sean dos números A y B enteros positivos representados en C1. Implementar un sistema en el que si $A > B$ $Z = -A$, si $A < B$ $Z = -B$, si $A = B$ $Z = 2A$, utilizando sumadores binarios, multiplexores y puertas lógicas.

- 21.** (Enero 2020) Diseñar un conversor combinacional de complemento a 1 a complemento a 2 de 8 bits utilizando multiplexores, sumadores binarios, y las puertas lógicas que necesites.

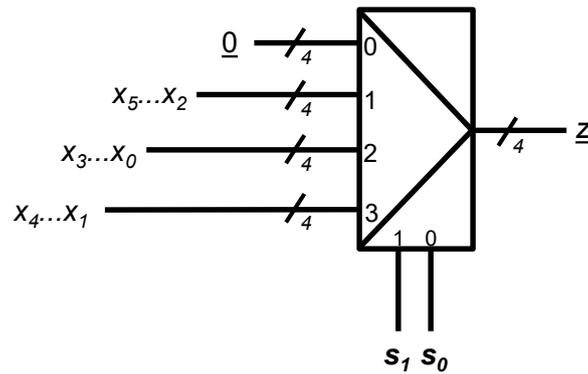
Soluciones

1. al 4. Véanse las transparencias.

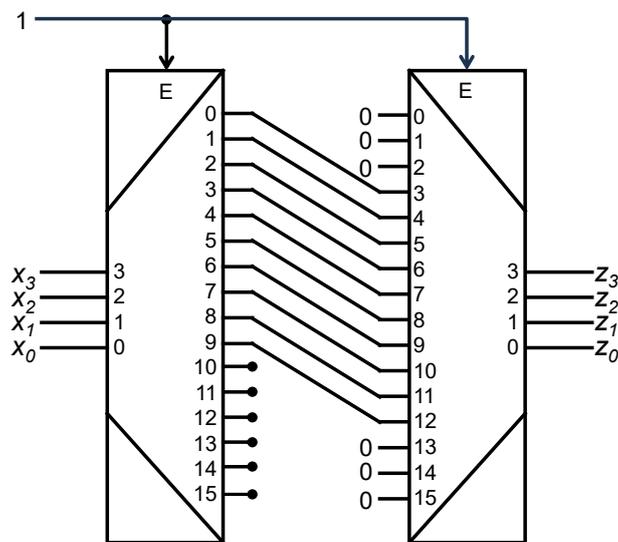
5.



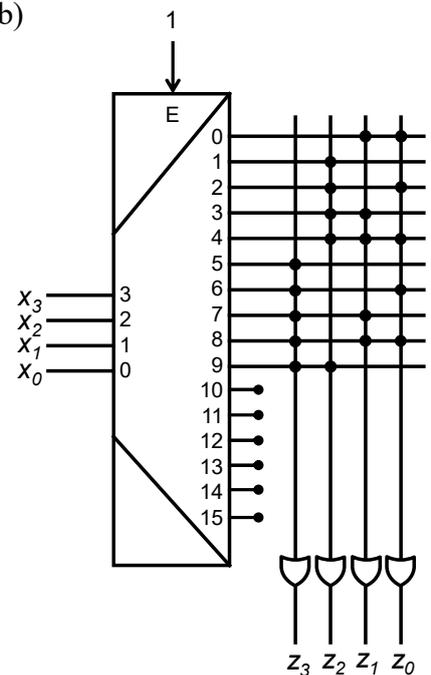
6.

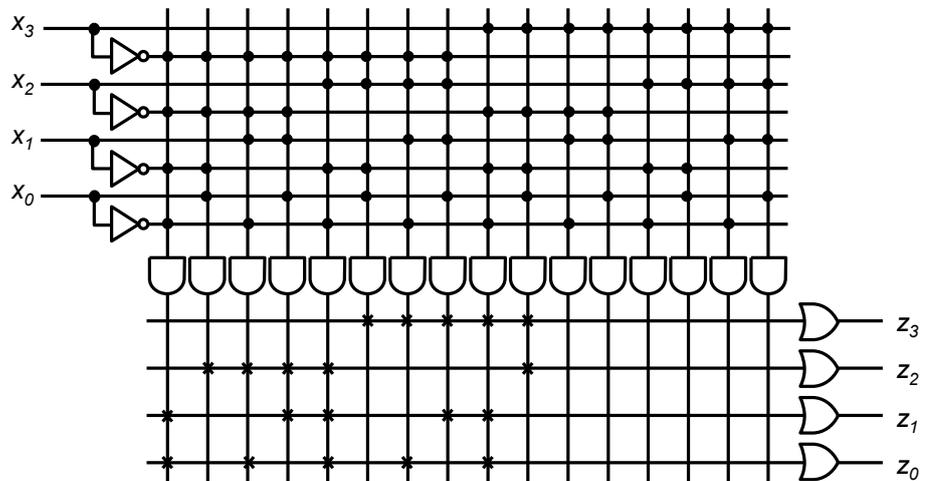
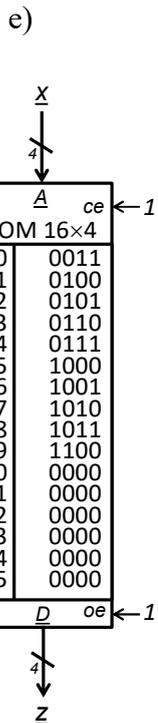
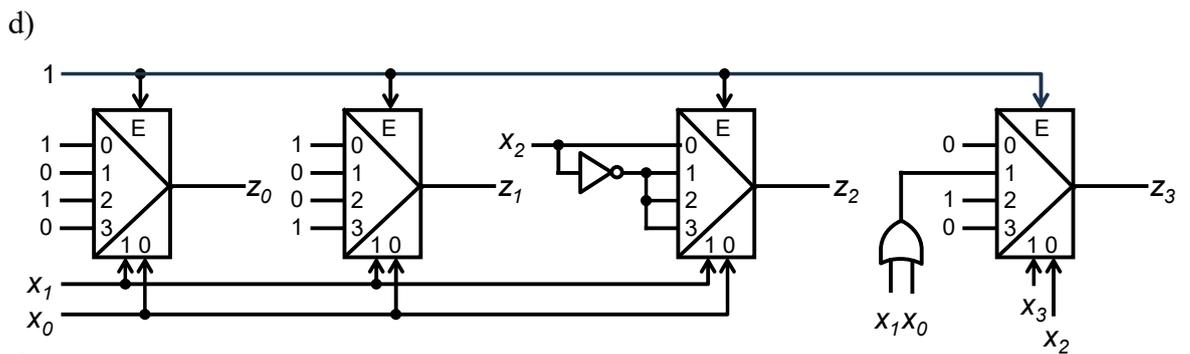
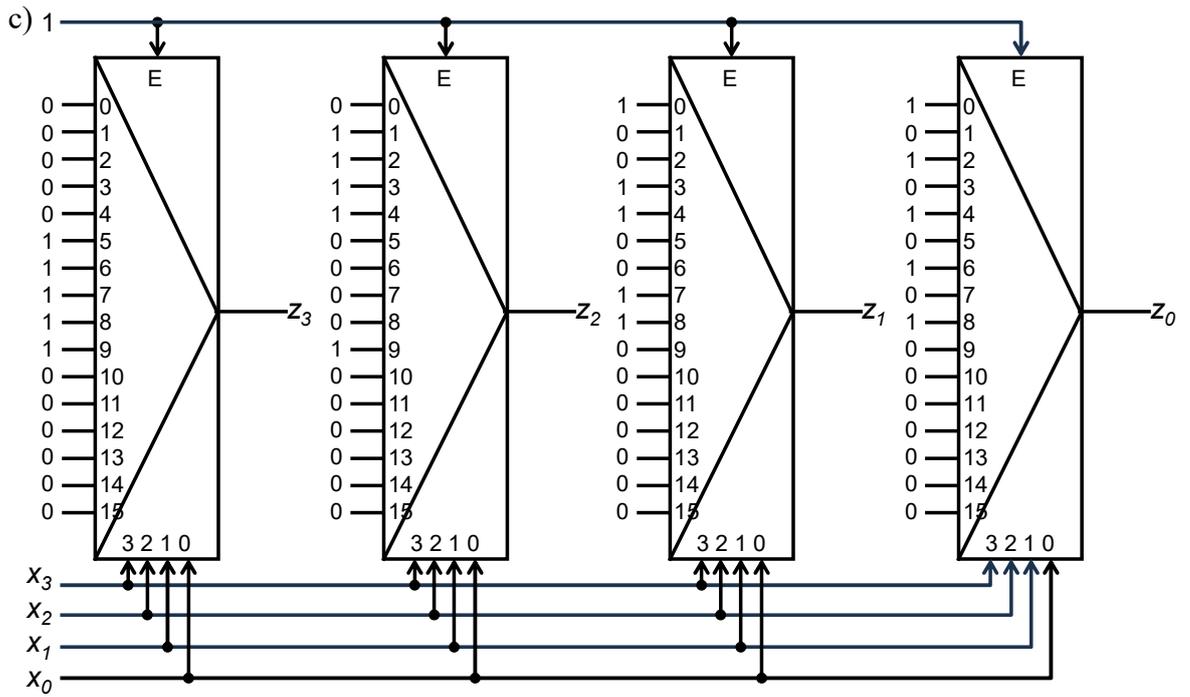


7. a)

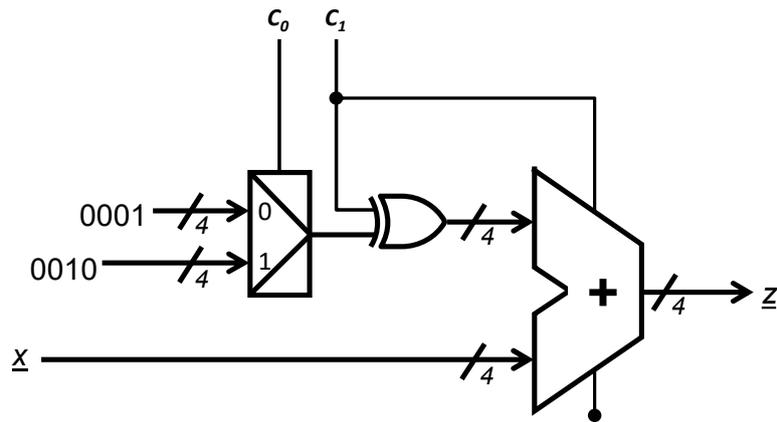


b)





8.



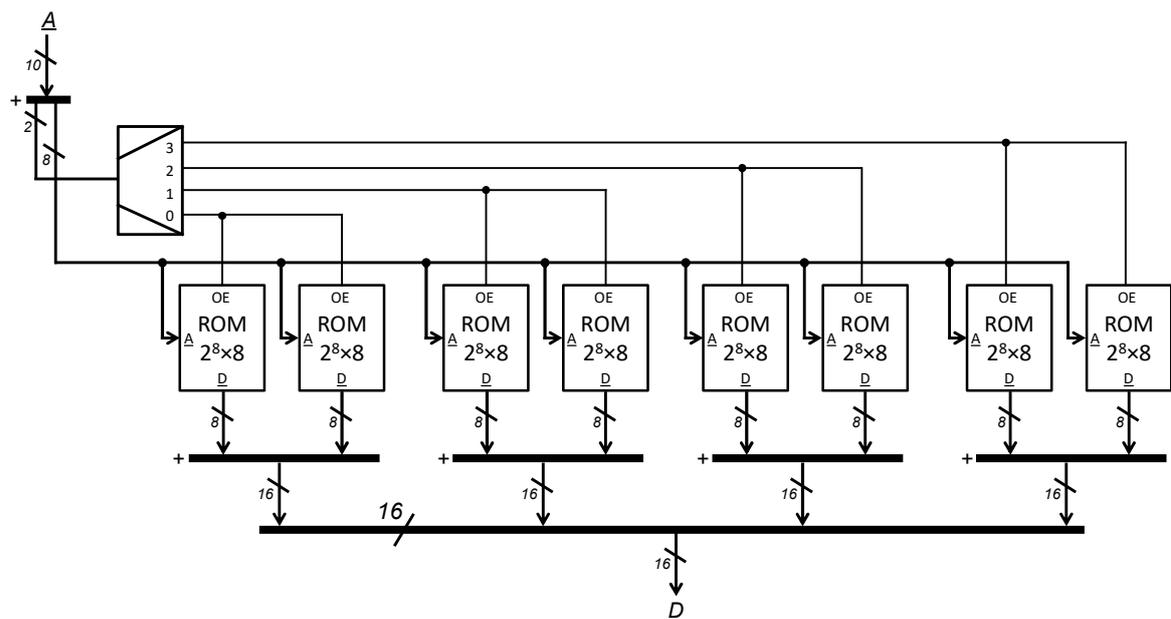
9. a) ROM 256×12
 b) ROM(41) = 0000 0100 0001 ROM(47) = 0000 0100 0111

10. $z(x) = (x + 2) \text{ mod } 8$

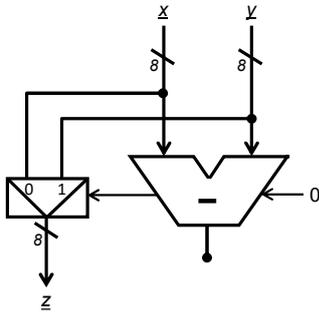
11. a) AND (4 entradas) = 8, OR = 0, NOT = 3
 b) AND (5 entradas) = 8, OR (8 entradas) = 1, NOT = 3
 c) codificador: AND (2 entradas) = 5, OR (8/16 entradas) = 5, NOT = 0
 resolutor: AND (2-16 entradas) = 15, NOT = 15

12. a) 85
 b) 15
 c) 24
 d) 15
 e) 85

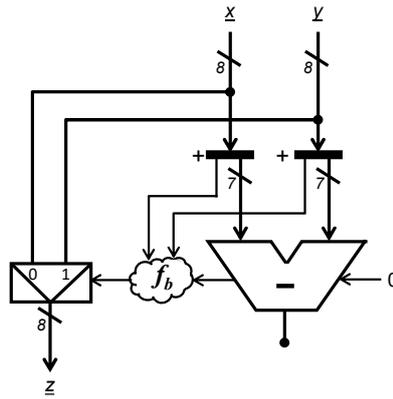
13.



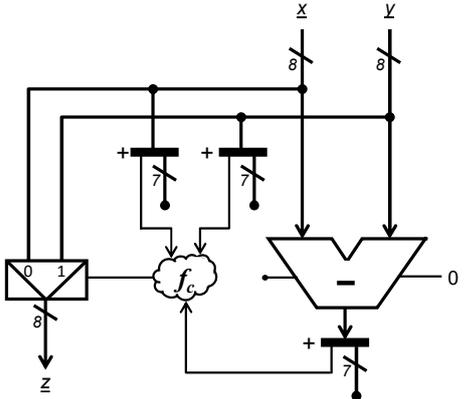
14. a)



b)



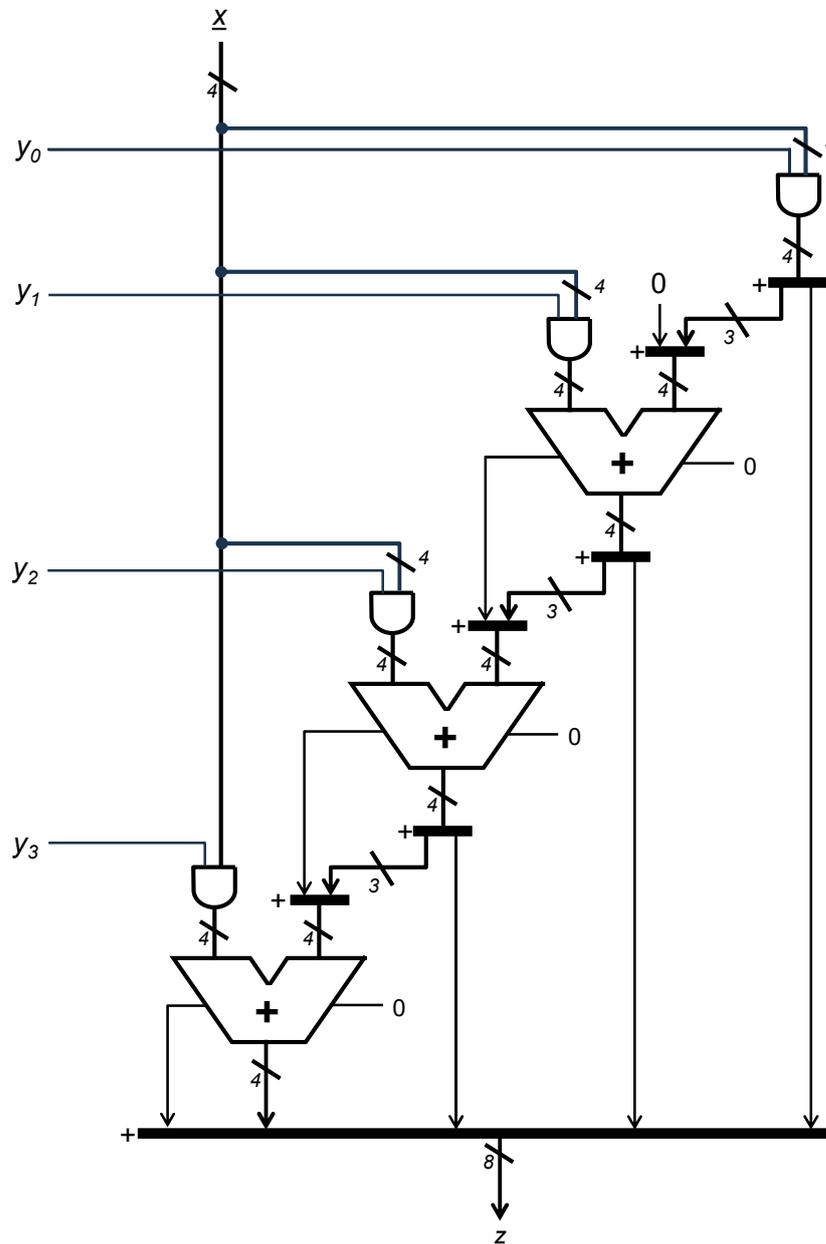
c)



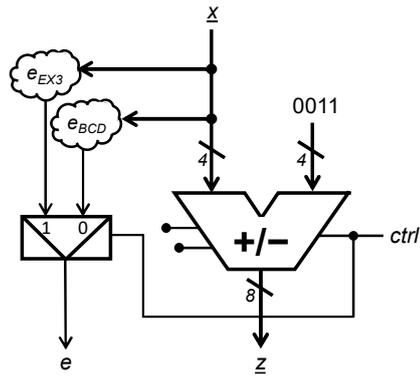
$$f_b(x_7, y_7, b) = \sum m(1,4,5,7) = x_7b + x_7\bar{y}_7 + y_7b$$

$$f_c(x_7, y_7, r_7) = \sum m(0,2,3,6) = y_7\bar{r}_7 + \bar{x}_7y_7 + \bar{x}_7\bar{r}_7$$

15.



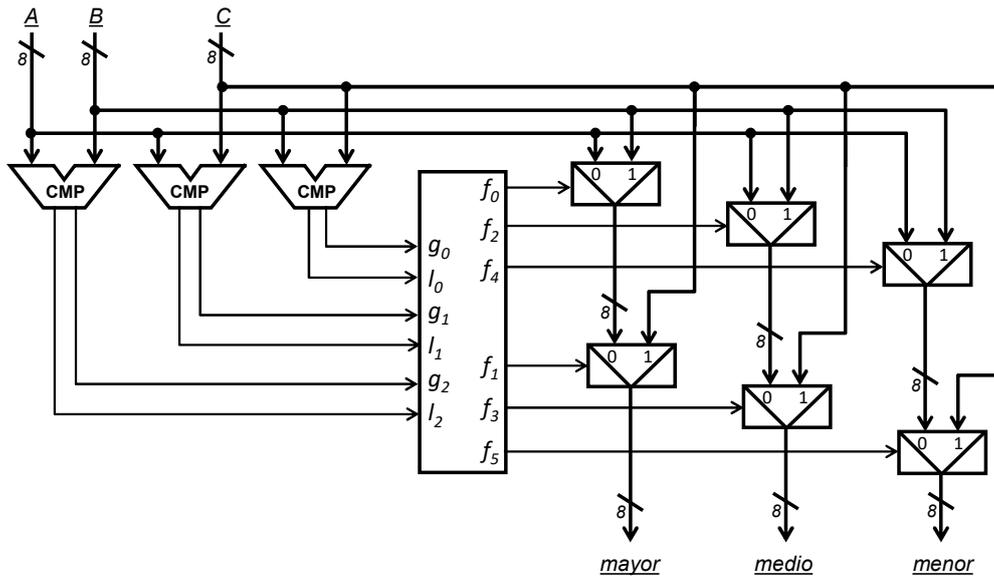
16.



$$e_{EX3}(x) = \sum m(0,1,2,13,14,15)$$

$$e_{BCD}(x) = \sum m(10,11,12,13,14,15)$$

17.



$$f_0(g_2, l_2, g_1, l_1, g_0, l_0) = \sum m(5,18,20,21,22,26,37) + t^* = l_2 + l_1$$

$$f_1(g_2, l_2, g_1, l_1, g_0, l_0) = \sum m(5,21,37) + t^* = l_1 l_0$$

$$f_2(g_2, l_2, g_1, l_1, g_0, l_0) = \sum m(0,10,20,21,22,33,40,41,42) + t^* = \overline{l_2} \oplus l_1$$

$$f_3(g_2, l_2, g_1, l_1, g_0, l_0) = \sum m(20,22,33,41) + t^* = l_1 \oplus l_0$$

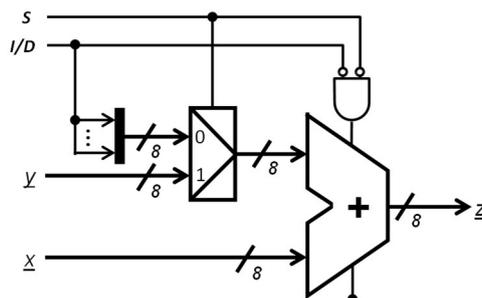
$$f_4(g_2, l_2, g_1, l_1, g_0, l_0) = \sum m(0,5,10,18,26,33,37,40,41,42) + t^* = \overline{l_2} + \overline{l_1}$$

$$f_5(g_2, l_2, g_1, l_1, g_0, l_0) = \sum m(0,10,18,26,40,42) + t^* = \overline{l_1} \overline{l_0}$$

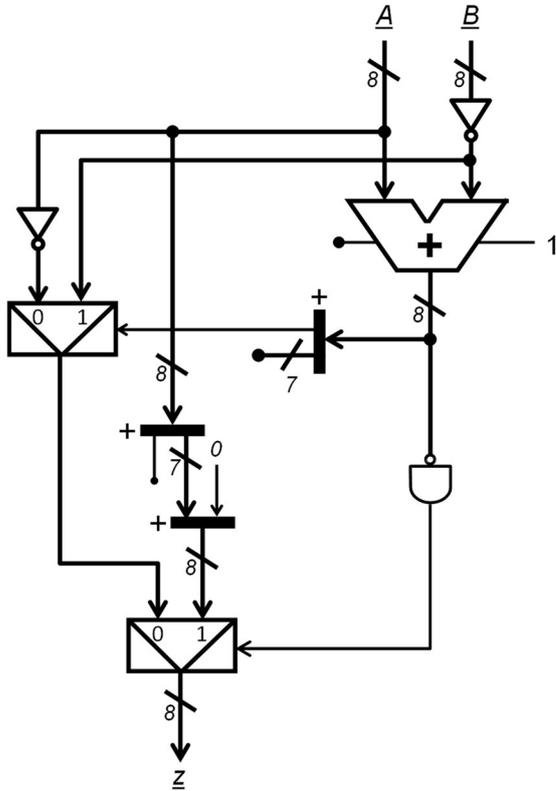
$$t^* = \sum d(1 \dots 4, 6 \dots 9, 11 \dots 17, 19, 23 \dots 25, 27 \dots 32, 34 \dots 36, 38, 39, 43 \dots 63)$$

18. 4 bits:	coste = 199.07 μm^2	tp = 1641 ps	tc = 237 ps
8 bits:	coste = 398.13 μm^2	tp = 3045 ps	tc = 237 ps
n bits:	coste = (49.76·n) μm^2	tp = (588+351·(n-1)) ps	tc = 237 ps

19.



20.



21.

