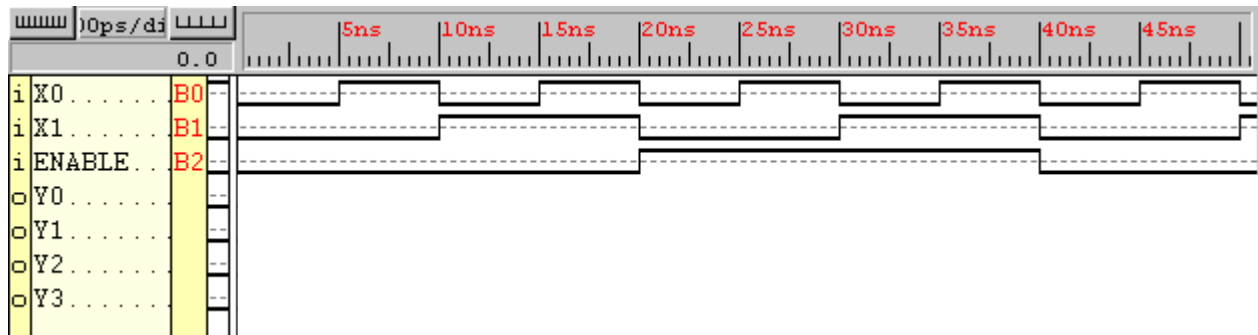
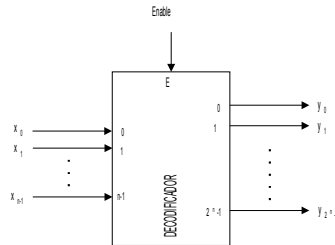


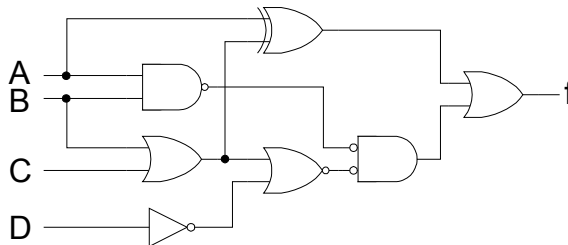


HOJA DE PROBLEMAS 6: MÓDULOS COMBINACIONALES BÁSICOS

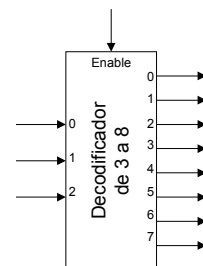
1. Dado el módulo combinacional de la figura se pide dibujar las formas de onda de las señales de salida y escribir sus ecuaciones de conmutación.



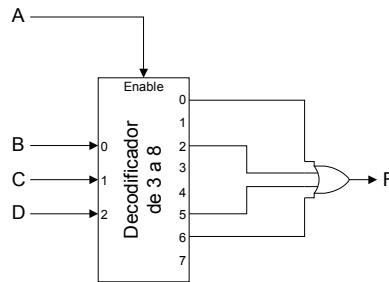
2. Dado el circuito combinacional de la figura:
- Analizarlo e indicar cuál es la función lógica que implementa, simplificándola al máximo.
 - Materializar un circuito equivalente al anterior mediante un decodificador de 2 a 4 y las puertas lógicas necesarias.



3. Construir un decodificador de 4 a 16 utilizando dos decodificadores de 3 a 8 como el mostrado en la figura más las puertas lógicas que se consideren necesarias. Al igual que los decodificadores de 3 a 8, el decodificador de 4 a 16 resultante contará con entrada de habilitación activa por nivel alto.



4. Para el sistema combinacional de 4 entradas $F(A,B,C,D)$ de la figura, se pide responder a lo siguiente:



- Analizar el esquema, generando la tabla de verdad del sistema.
- Realizar la simulación del esquema, generando la forma de onda de la salida F sobre la figura situada debajo.

5. Diseñar un circuito combinacional que reciba como entrada un número de cuatro bits en binario puro $B = b_3b_2b_1b_0$, y que devuelva dos salidas Y y Z :
- $Y=1$ cuando B es múltiplo de 3, $Y=0$ en caso contrario.
 - $Z=1$ cuando B es múltiplo de 5, $Z=0$ en caso contrario.

Para el diseño se permitirá utilizar únicamente un decodificador de 4 entradas de datos activas por nivel alto y 16 salidas activas por nivel bajo y señal de habilitación activa por nivel bajo, más dos puertas NAND con tantas entradas cada una de ellas como sea preciso.

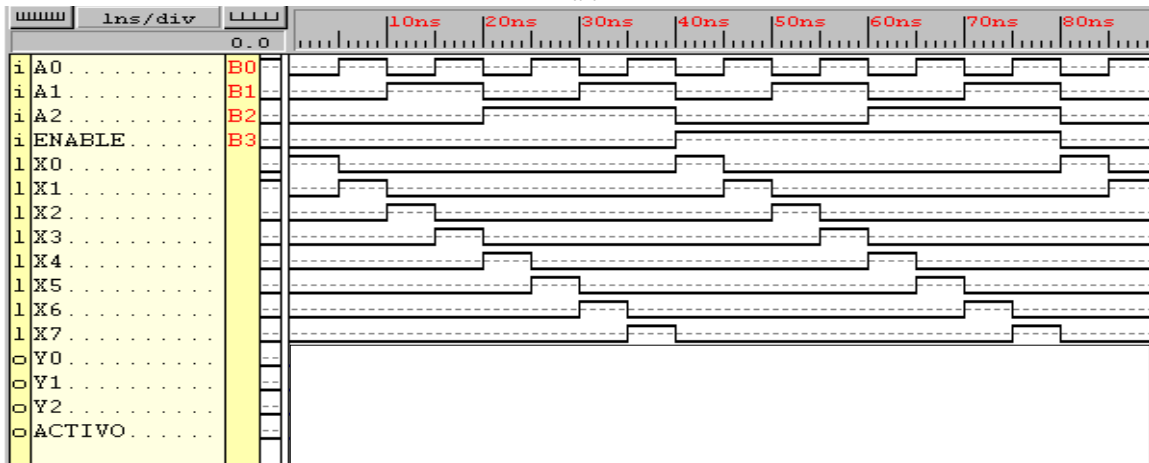
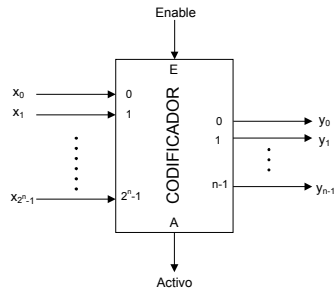
6. Se pretende diseñar un circuito que tome como entrada un número representado en binario natural de 4 bits y genere a la salida su representación mediante dos dígitos BCD de 4 bits.
- Entradas: $A(a_3, a_2, a_1, a_0)$ codificada en binario natural de 4 bits.
 - Salidas: $X(x_3, x_2, x_1, x_0)$ e $Y(y_3, y_2, y_1, y_0)$ codificadas en BCD natural de 4 bits, siendo X el primer dígito BCD e Y el segundo.

Se pide:

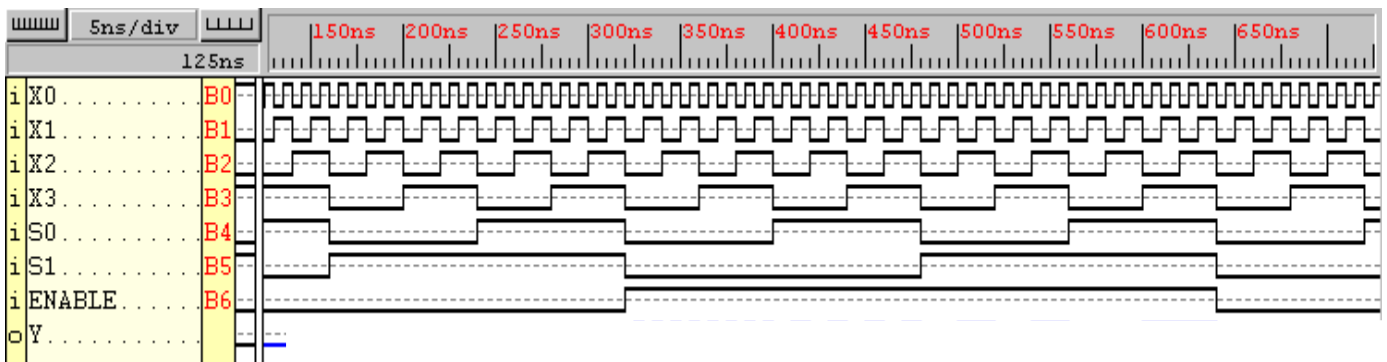
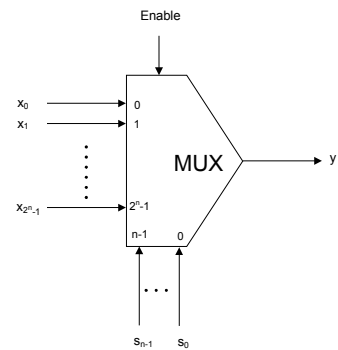
- Hallar la tabla de verdad de las funciones.
- Expresar las funciones en forma de suma de minterms.

- c) Simplificar al máximo las funciones correspondientes a la salida Y dejándolas en forma de suma de productos con variables simples.
- d) Materializar las funciones correspondientes a la salida Y empleando decodificadores de 3 a 8 y el menor número de puertas lógicas.

7. Dado el módulo combinacional de la figura se pide dibujar las formas de onda de las señales de salida.

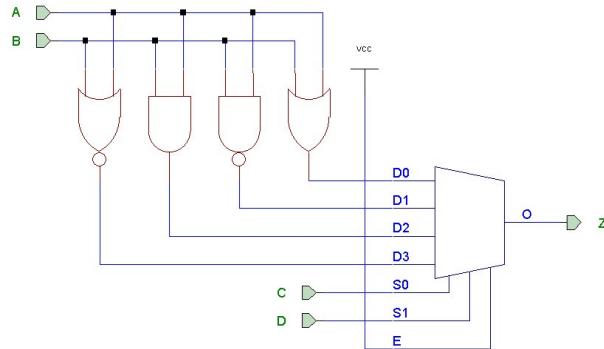


8. Dado el módulo combinacional de la figura se pide dibujar las formas de onda de las señales de salida.

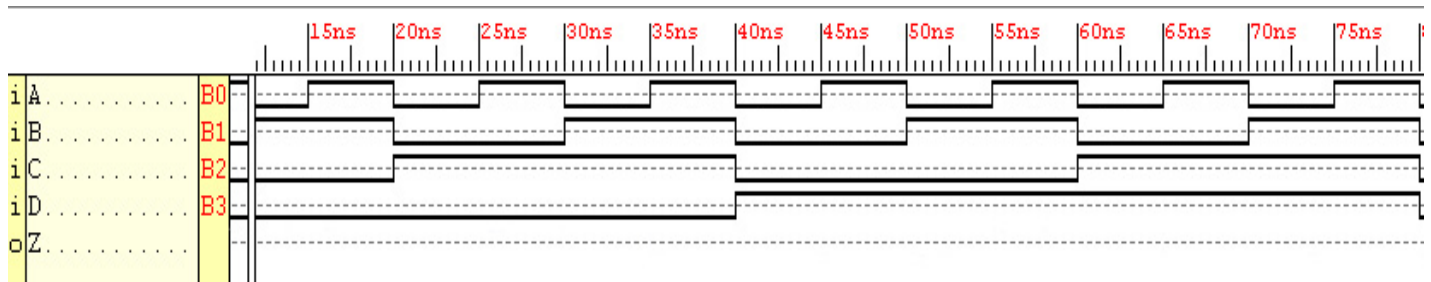


9. Construir un multiplexor de 8 entradas de datos a partir de dos multiplexores de 4 entradas de datos cada uno y las puertas lógicas necesarias. Todos los multiplexores contarán con entrada de activación activa por nivel alto.

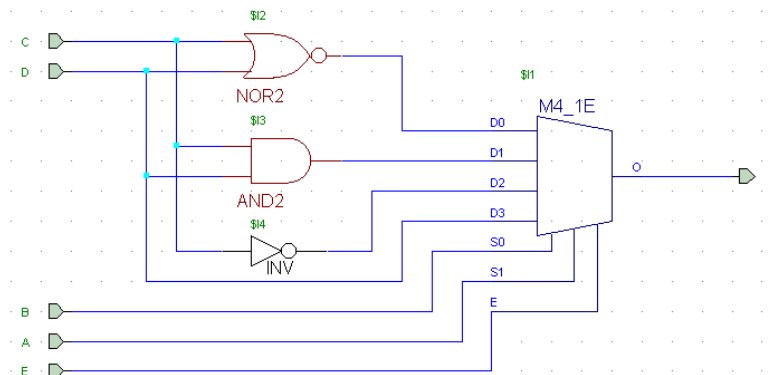
10. Para el sistema combinacional de 4 entradas F(A,B,C,D) de la figura, se pide responder a lo siguiente:



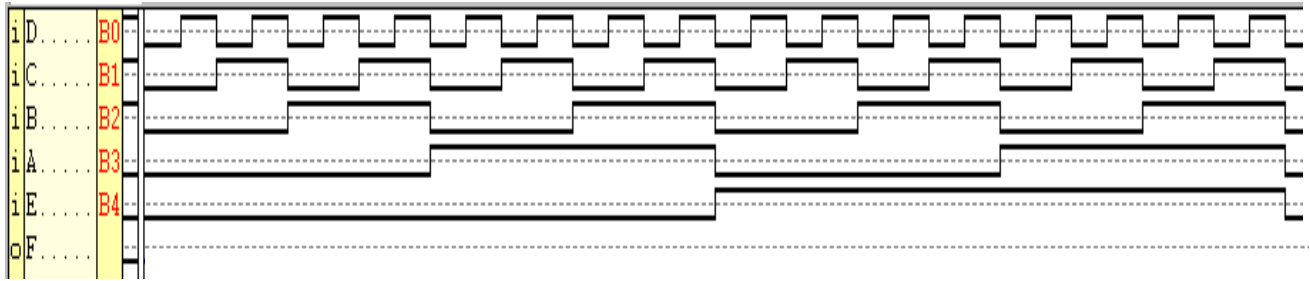
- Analizar el esquema, generando la tabla de verdad del sistema.
- Realizar la simulación del esquema, generando la forma de onda de la salida F sobre la figura situada debajo.



11. Para el sistema combinacional de 5 entradas F(A,B,C,D,E) de la figura, se pide responder a lo siguiente:



- Analizar el esquema, generando la tabla de verdad del sistema.
- Realizar la simulación del esquema, generando la forma de onda de la salida F sobre la figura situada debajo.



12. Dado el circuito de la figura se pide:

- Indicar la función lógica realizada por el circuito.
- Construir la tabla de verdad de la función.
- Simplificar la función por el método de Karnaugh.
- Dibujar un circuito que realice la función anterior, empleando únicamente puertas NAND de dos entradas.

13. A los lados de un río hay un hombre (H), un lobo (L), una oveja (V) y una col (C). El hombre no está hambriento, luego no tiene la menor intención de comer nada, pero además tampoco permite que ninguno de los demás coma. El lobo y la oveja sí que están hambrientos, pero el lobo (exclusivamente carnívoro) no podrá comerse a la oveja

si el hombre está en su misma orilla, y lo mismo le sucederá a la oveja (exclusivamente vegetariana) con la col. Se pide:

- Hallar la tabla de verdad de la función $F_c(H,L,V,C)$ sabiendo que debe valer 1 si alguien ha comido a alguien o a algo y 0 en caso contrario. Sugerencia: codificar las variables con 1 para la orilla izquierda y 0 para la derecha.
- Expresar la función en forma de producto de maxterms.
- Simplificar la función dejándola en forma de suma de productos con variables simples, utilizando el método más conveniente.
- Materializar la función mediante un multiplexor de 8 entradas de datos, una salida y 3 señales de selección.
- Materializar la función mediante un decodificador de 4 a 16 y puertas lógicas.

A	B	C	D	$f(A,B,C,D)$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	X
0	0	1	1	X
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	X
1	0	1	1	X
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	X

14. Sea una función $f(A,B,C,D)$ con la siguiente tabla de verdad:

- Escribir la representación de f en forma de minterms y en forma de maxterms.
- Simplificar f en forma de suma de productos.
- Materializar la función mediante dos multiplexores de 4 a 1 y el mínimo número de puertas lógicas.
- Materializar la función mediante dos decodificadores de 3 a 8 y el mínimo número de puertas lógicas.

15. Diseñar un desplazador combinacional con 4 entradas de datos, 2 entradas de selección y 4 salidas que realice las siguientes operaciones:

S1	S0	Operación
0	0	Desplazamiento a la derecha de longitud 1
0	1	Desplazamiento a la derecha de longitud 2
1	0	Rotación a la derecha de longitud 1
1	1	Rotación a la derecha de longitud 2

Por los bits sobrantes entran ceros. Se podrán utilizar en el diseño las puertas lógicas y los módulos estándares (multiplexores, etc.) que se considere necesario. Se valorará la sencillez del circuito obtenido.

16. Utilizando los circuitos combinacionales estándares y las puertas lógicas que sean necesarias, diseñar un circuito combinacional que genere 4 bits de salida ($S_0 - S_3$), a partir de 4 bits de entrada ($E_0 - E_3$), desplazándolos según los valores que toman las entradas de control M_0 y M_1 :

M1	M0	Operación
0	0	Desplazamiento lógico a la derecha
0	1	Desplazamiento lógico-aritmético a la izquierda
1	0	Desplazamiento aritmético a la derecha
1	1	Rotación a la derecha

NOTA: Se recuerda que en la rotación el bit que se introduce es el mismo que rebosa al rotar: Por otra parte, en el desplazamiento lógico a la derecha se introducen ceros por delante, mientras que en el aritmético se introduce el bit de signo. En los desplazamientos a la izquierda, tanto en el lógico como en el aritmético, se introducen ceros.

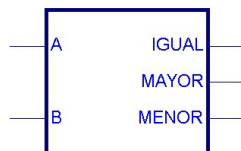
17. Dada la función lógica siguiente:

$$f(A,B,C,D) = \bar{A} \cdot B \cdot C + A \cdot (\bar{B} \oplus D) + B \cdot \bar{C} \cdot D + A \cdot B \cdot \bar{C} \cdot \bar{D}$$

Se pide:

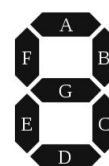
- Hallar la tabla de verdad de la función.
- Expresar la función en forma de suma de minterms.
- Simplificar la función dejándola en forma de suma de productos con variables simples, utilizando el método más conveniente.
- Materializar la función empleando únicamente puertas NAND (de dos o más entradas) e inversores.
- Materializar la función mediante un decodificador de 4 a 16 y el mínimo número de puertas lógicas.
- Materializar la función mediante un multiplexor de 8 a 1 y el mínimo número de puertas lógicas.
- Materializar la función mediante un multiplexor de 4 a 1 y el mínimo número de puertas lógicas.
- Materializar la función mediante una PLA, seleccionando adecuadamente el número de productos, entradas y salidas necesarios.

18. Materializar un circuito que reciba un número binario de tres bits ($x_2 x_1 x_0$) y calcule el cuadrado de dicho número (6 bits de salida: $y_5 y_4 y_3 y_2 y_1 y_0$).
- Escribir la tabla de verdad del sistema.
 - Expresar cada variable de salida en forma de suma de productos.
 - Simplificar las funciones correspondientes a y_5 e y_4 .
 - Materializar las funciones correspondientes a y_5 e y_4 mediante puertas NAND únicamente. (No se permite el uso de inversores).
 - Materializar las funciones correspondientes a y_5 e y_4 mediante un decodificador de 4 entradas de datos y el mínimo número de puertas lógicas.
 - Materializar las funciones correspondientes a y_5 e y_4 mediante una PAL, seleccionando adecuadamente y razonando el número de productos, entradas y salidas necesarios.
19. Se desea diseñar un circuito comparador para palabras de entrada de dos bits, de forma que la salida sea 1 si y sólo si $X \geq Y$. Se pide:
- Construir la tabla de verdad de las tres funciones de salida.
 - Simplificar las funciones por el método de Karnaugh..
 - Dibujar un circuito que realice las tres funciones anteriores, empleando únicamente puertas NOR de dos entradas.
 - Materializar las tres funciones mediante un decodificador de 4 a 16 y el mínimo número de puertas lógicas.
 - Materializar las tres funciones mediante una PLA, seleccionando adecuadamente el número de productos, entradas y salidas necesarios.
20. Utilizando los bloques combinacionales estándares y las puertas lógicas necesarias, diseñar un circuito que, dados dos números de cuatro bits A y B en binario sin signo, proporcione a la salida el máximo de ambos.
21. Basándonos en un comparador de número de cuatro bits en binario puro, diseñar un nuevo comparador para números de 4 bits en complemento a 2. Para ello, además del comparador de cuatro bits en binario puro indicado en la figura, se podrán utilizar otros módulos combinacionales básicos y/o puertas lógicas.



Comprobar la corrección del diseño calculando la salida del sistema con estos tres ejemplos y **razonar** su validez:

- $A = 0110_{Ca2}$; $B = 1101_{Ca2}$
- $A = 1101_{Ca2}$; $B = 0101_{Ca2}$
- $A = 1100_{Ca2}$; $B = 1001_{Ca2}$

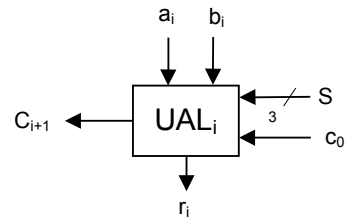


22. Sean dos dígitos decimales A y B representados en BCD. Se pretende diseñar un circuito que muestre el mayor de ellos en un visualizador de 7 segmentos, y si son iguales, no visualice nada. Diseñar el circuito correspondiente, utilizando para ello los módulos combinacionales y las puertas lógicas necesarias.
23. Diseñar un circuito lógico que obtenga el valor absoluto de un número de cuatro bits codificado en complemento a 2.
24. Utilizando puertas lógicas y sumadores de 8 bits, materializar un circuito que convierta un número de 8 bits dado en módulo y signo a su representación en complemento a 2. ¿Es necesario contemplar una posible situación de desbordamiento? En caso afirmativo, incluir la circuitería necesaria.
25. Utilizando puertas lógicas y sumadores de 8 bits, materializar un circuito que convierta un número de 8 bits dado en complemento a 2 a su representación en módulo y signo. ¿Es necesario contemplar una posible situación de desbordamiento? En caso afirmativo, incluir la circuitería necesaria. Reflexionar sobre la relación de este circuito con el del ejercicio anterior.
26. Materializar un circuito para cambiar el signo de un número de 8 bits dado en complemento a 2, incluyendo una señal de desbordamiento. Analizar cómo se comporta el circuito para la entrada 10000000.
27. Materializar un circuito para cambiar el signo de un número de 8 bits dado en complemento a 2 que utilice el método de copiar de derecha a izquierda todos los bits hasta encontrar un 1, y a partir del siguiente bit los invierta.
28. Materializar sendos circuitos para realizar la extensión de signo de 4 a 8 bits para números en binario puro, complemento a 1, complemento a 2 y módulo y signo. La extensión de signo consiste en aumentar el tamaño del número en cuestión sin modificar su valor.
29. Se desea diseñar una unidad aritmético-lógica de 4 bits para datos en binario puro y complemento a 2 que realice las siguientes operaciones, reguladas por las señales de selección S_0 , S_1 y S_2 (las operaciones que incluyen signo son en complemento a 2):

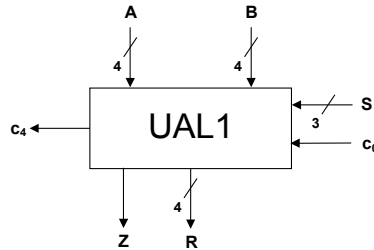
S_2	S_1	S_0	Operación
0	0	0	A ADD B (suma aritmética)
0	0	1	A ADD 1 (suma aritmética)
0	1	0	A SUB 1 (resta aritmética)
0	1	1	-B (cambio de signo)
1	0	0	A OR B
1	0	1	A AND B
1	1	0	A XOR B
1	1	1	A' (complementación lógica)

a) Utilizando multiplexores, sumadores binarios elementales y/o puertas lógicas, materializar una celda elemental (que llamaremos UAL_i) de dicha UAL. La celda tendrá las siguientes entradas y salidas:

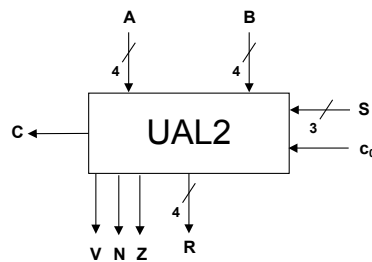
- Entradas de operandos: a_i (primer operando) y b_i (segundo operando).
- Entradas de selección de la operación: S_2, S_1 y S_0 .
- Entrada de acarreo: c_i .
- Salida de resultado: r_i .
- Salida de acarreo: c_{i+1} .



- b) Utilizando celdas elementales como las diseñadas en el apartado a, materializar la UAL de 4 bits (que llamaremos UAL1), que generará indicadores de resultado nulo Z y de acarreo superior c_3 .



- c) Crear una nueva UAL de 4 bits (que llamaremos UAL2) añadiendo a UAL1 lo



necesario para obtener los indicadores de resultado C, N, Z y V.

- d) Con las UAL de 4 bits diseñadas en los apartados b y c, construir una UAL de 8 bits con generación de C, V, N y Z.
- 30.** Analizar el circuito de la figura, e indicar qué operación realiza, siendo A (a_3, a_2, a_1, a_0) y B (b_3, b_2, b_1, b_0) dos números de cuatro bits y C un dato de 1 bit. La puerta lógica XOR de la figura realmente son cuatro puertas XOR que realizan la operación indicada entre C y cada uno de los bits de B, es decir:

$$D(d_3, d_2, d_1, d_0) = (C \oplus b_3, C \oplus b_2, C \oplus b_1, C \oplus b_0)$$

- 31.** Analizar el circuito de la figura, e indicar qué resultado sale por la señal R en función de los valores de las entradas A y B. Es preciso tener en cuenta que:

- La puerta XOR de la izquierda representa 4 puertas XOR, cada una de las cuales realiza un XOR entre la señal $A > B$ y cada uno de los bits de A, es decir:
 $((A > B) \oplus a_3, (A > B) \oplus a_2, (A > B) \oplus a_1, (A > B) \oplus a_0)$
- La puerta XOR de la derecha representa 4 puertas XOR, cada una de las cuales realiza un XOR entre la señal $A < B$ y cada uno de los bits de B, es decir:
 $((A < B) \oplus b_3, (A < B) \oplus b_2, (A < B) \oplus b_1, (A < B) \oplus b_0)$