

La ruta de datos

1. Diseñe un operador inversor condicional para operandos de 8 bits.
2. ¿Qué posibilidades tenemos a la hora de sintetizar un operador de desplazamiento utilizando álgebra de Boole? Realice un análisis comparativo en términos de coste, rendimiento y viabilidad de la implementación.
3. Un detector de cero es un operador que toma como entrada una ristra de n bits y devuelve un '1' si todos los bits son '0' o devuelve un '0' en caso contrario. ¿Qué posibilidades tenemos para sintetizarlo usando álgebra de Boole? Presente varias opciones y comente sus ventajas e inconvenientes.
4. ¿Cómo se cambia el signo de un operando representado en exceso a 2^{n-1} ? Demuestre formalmente su respuesta.
5. Sintetice un operador de cambio de signo para operandos representados en exceso a 2^{n-1} . Evalúe la complejidad *hardware* de su diseño.
6. El operador $CMPL2(x_{2,n})$ es, por definición, $BIN(2^n - x, n)$ donde $x_{2,n}$ es la representación del valor $x \in \mathbb{N}$ sobre n bits. Demuestre que $CMPL2(x_{2,n}) = \bar{x}_{2,n} + 1$.
7. El operador $CMPL1(x_{2,n})$ es, por definición, $BIN(2^n - 1 - x, n)$ donde $x_{2,n}$ es la representación del valor $x \in \mathbb{N}$ sobre n bits. Demuestre que $CMPL1(x_{2,n}) = \bar{x}_{2,n}$.
8. Dada una representación $x_{2,n}$ de $x \in \mathbb{N}$ sobre n bits, demuestre que hacer el complemento a dos del complemento a uno de $x_{2,n}$ es tanto como sumar 1 a dicha representación. ¿Qué sucede si cambiamos el orden? Demuestre también este caso.
9. Especifique qué conversión ha de realizar un operador de cambio de sistema de representación para pasar de complemento a dos sobre 8 bits a exceso a 2^{n-1} también sobre 8 bits. Demuestre su propuesta.
10. Se desea construir un operador *hardware* de 16 bits que convierta números representados en complemento a dos a números representados en signo-magnitud. Especifique qué transformaciones hay que realizar y proponga una implementación suponiendo que dispone de todos los operadores y puertas lógicas que necesite.
11. Complete el operador del problema anterior con los elementos necesarios para realizar la conversión en sentido contrario.
12. La unidad de coma flotante de un procesador solamente trabaja en doble precisión. Cuando se le pasan datos en simple precisión realiza automáticamente las extensiones de signo de exponente y mantisa antes de operar. Explique cómo se hacen dichas extensiones sabiendo que los exponentes se codifican en exceso a 2^{n-1} y las mantisas en signo-magnitud. El formato de simple precisión asigna 8 bits al exponente y 24 a la mantisa mientras que el formato de doble precisión asigna 11 y 53 respectivamente.
13. Diseñe un operador de extensión de signo en complemento a dos que extienda representaciones de 8 bits a 16 bits.
14. Diseñe un sumador de 6 bits basado en mediosumadores. Indique la complejidad $\mathcal{O}(g(x))$ del operador tanto en área como en tiempo. Suponiendo que las ecuaciones lógicas del semisumador son $s_i = x_i \oplus y_i$ y $c_i = x_i \cdot y_i$ indique el retardo temporal medido en niveles lógicos.

15. Podemos realizar un sumador binario de n bits basado en mediosumadores (HA) o en sumadores completos (FA). Comente el modo de funcionamiento de cada uno. Discuta sus ventajas e inconvenientes y sus posibles áreas de aplicación.
16. Usando un restador en complemento a dos y las puertas oportunas, realice un circuito comparador de 2 enteros de 4 bits en complemento a 2 ($x_{C2,4}$ e $y_{C2,4}$) que proporcione las señales IGUAL y MAYORQUE de manera que IGUAL='1' si $x = y$ y MAYORQUE='1' si $x > y$.
17. Diseñe un comparador para números enteros representados en signo-magnitud que proporcione las señales IGUAL y MAYORQUE.
18. Se desea realizar un sumador entero de n bits basado en el selector de acarreo. Los bloques propagadores son todos del mismo tamaño k . ¿Qué fórmula nos da el tiempo de retardo en función de n y k medido en niveles lógicos? Determine el valor de k que minimiza el retardo para un tamaño n dado.
19. Se desea diseñar un sumador con selección de acarreo de 16 bits. Podemos usar bloques propagadores de 2, 4 y 8 bits. Construya el mejor sumador posible suponiendo que los conmutadores tienen un retardo equivalente a $2r_g$ (retardo de una puerta lógica genérica).
20. Sean los siguientes factores representados en binario puro: multiplicando $111010_{2,6}$ y multiplicador $001010_{2,6}$. Realice el producto mediante el algoritmo de sumas-desplazamientos.
21. Sean los siguientes factores representados en complemento a dos: multiplicando $001010_{C2,6}$ y multiplicador $111010_{C2,6}$. Realice el producto mediante el algoritmo de sumas-desplazamientos.
22. Sean los siguientes factores representados en complemento a dos: multiplicando $111010_{C2,6}$ y multiplicador $001010_{C2,6}$. Realice el producto mediante el algoritmo de sumas-desplazamientos.
23. Sean los siguientes factores representados en complemento a uno: multiplicando $010010_{C1,6}$ y multiplicador $100010_{C1,6}$. Realice el producto mediante el algoritmo de sumas-desplazamientos.
24. Sean los siguientes factores representados en binario puro: multiplicando $001010_{2,6}$ y multiplicador $111010_{2,6}$. Realice el producto mediante el algoritmo de sumas-restas.
25. Sean los siguientes factores representados en complemento a dos: multiplicando $001010_{C2,6}$ y multiplicador $111010_{C2,6}$. Realice el producto mediante el algoritmo de sumas-restas.