



Solución Interrogación 1

Pregunta 1

- a) Se tienen dos números de punto flotante de precisión simple en formato IEEE754: $A = 0x3E200000$ y $B = 0x00000000$. ¿Cuál es el resultado, en formato IEEE754, de $A : B$? **(0,75 ptos.)**

Solución: $A > 0$ y $B = 0$, luego, dado que la división entre un número positivo y cero, en IEEE754, es $+\infty$, el resultado en correcto es $0x7F800000 = 01111111100000000000000000000000$.

- b) ¿Cuántos relés se necesitan para representar cada una de las siguientes compuertas: AND, OR, NOT y XOR? **(0,75 ptos.)**

Solución: AND = 2, OR = 2, NOT = 1. Para el XOR hay dos opciones 1. $(\neg p \wedge q) \vee (p \wedge \neg q) = 8$, 2. $\neg(p \wedge q) \wedge (p \vee q) = 7$. Cualquier resultado que indique una cantidad diferente de componentes debe ir acompañado con un diagrama o una explicación.

- c) Dados $A = 45$ y $B = 57$, ¿cuál es el resultado, en binario, de la operación $A - B$? **(0,75 ptos.)**

Solución: En complemento de 2 de 7 bits, $A = 0101101$, $B = 0111001$, luego, $A - B = -12 = 1110100$.

- d) ¿Cuál es la diferencia entre un latch y un flip-flop? **(0,75 ptos.)**

Solución: Un latch puede realizar cambios de estado mientras se encuentra con la señal de control en 1, mientras que el flip-flop sólo lo hace cuando está en uno de los dos flancos, ya sea de subida o de bajada.

- e) Un programador desarrolla una aplicación bancaria de manejo de créditos. ¿Que consideraciones debe tomar en relación a los tipos de dato a usar? **(0,75 ptos.)**

Solución: El programador debería utilizar tipos de datos numéricos que usen base decimal y no binario, como el BigDecimal de Java.

- f) Al leer 2 palabras contiguas de 1 byte de una memoria en orden big endian, se obtiene el número 272. ¿Qué número se obtendría al leer estas palabras en orden little endian? **(0,75 ptos.)**

Solución: $272 = 0x0110$, luego, en little endian, el valor obtenido es $4097 = 0x1001$.

- g) ¿Cuántos bytes puede almacenar una memoria con palabras de 40 bits y bus de direcciones de 12 bits? **(0,75 ptos.)**

Solución: $\frac{2^{12} \times 40}{8} = 20480$.

- h) Dada una representación de punto fijo, con 1 bit para el signo, 3 para la parte entera y 4 para la fraccional, y una de punto flotante con 1 bit para el signo del significante, 3 para el significante, 1 para el signo del exponente y 3 para el exponente. ¿Cuál de estas representaciones recomendaría para almacenar el número 0,10111101b y por qué? **(0,75 ptos.)**

Solución: Es mejor utilizar la representación de punto fijo, ya que pierde menos bits de la parte fraccional del número.

Pregunta 2

- a) Describa una representación y/o mecanismo para obtener el inverso aditivo de un número en representación hexadecimal. Este proceso no debe incluir conversiones a otras representaciones numéricas, como binario o decimal. **(3 ptos.)**

Solución: Para obtener el inverso aditivo de un número hexadecimal se puede utilizar el siguiente proceso, llamado complemento a 16:

- a) Complementar cada dígito del número con respecto a F . Ej.: $C(0x43) = 0xBC$.
- b) Sumar uno al complemento

Para demostrar que este procedimiento es correcto, basta realizar la suma entre un número y su inverso: $0x43 + C16(0x43) = 0x43 + 0xBD = 0$.

- b) Usando el proceso descrito anteriormente, calcule el resultado de $0x43 - 0x64$. Indique claramente cada una de las etapas del proceso. **(1,5 ptos.)**

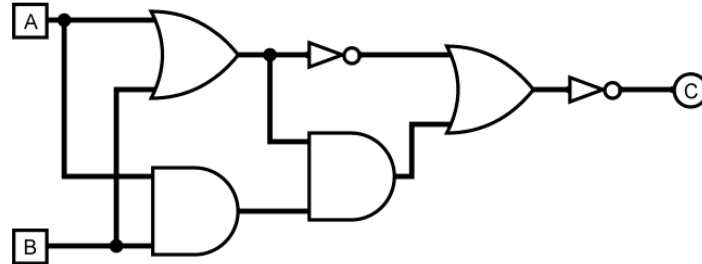
Solución:

- a) $0x43 - 0x64 = 0x43 + -0x64$
 - b) Obtener el complemento a 16 de $0x64$: $C16(0x64) = 0x9C$
 - c) Realizar la suma: $0x43 + 0x9C = 0xDF$
 - d) Dado que $C16(0xDF) = 0x21 = 33$, el resultado es correcto.
- c) ¿Usando este esquema, cuál es el número hexadecimal con mayor valor absoluto que puede representarse usando sólo 2 dígitos?, ¿Por qué? **(1,5 ptos.)**

Solución: La cantidad de número hexadecimales representables con dos dígitos son 256 (8 bits). Luego, el número con mayor absoluto que puede representarse es $-128 = 0x80$.

Pregunta 3

- a) Obtenga una fórmula de lógica booleana que represente el circuito de la figura. Utilice sólo los conectivos \wedge , \vee y \neg . Deje por escrito el desarrollo que justifique su respuesta. (2 ptos.)



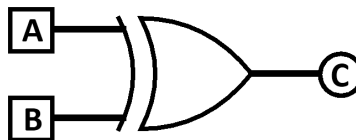
Solución: La tabla de verdad del circuito es la siguiente:

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Esta tabla corresponde a la del xor, luego, una fórmula de lógica booleana que satisface lo requerido es la siguiente: $(\neg A \wedge B) \vee (A \wedge \neg B)$.

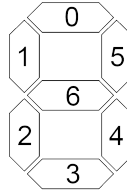
- b) ¿Es posible construir un circuito equivalente al anterior, usando menos compuertas y no necesariamente las mismas? De ser así, diga cuantas son necesarias y dibuje el circuito. Justifique su respuesta en cualquiera de los dos casos. (2 ptos.)

Solución: Dado que el circuito es equivalente a una compuerta xor, el diagrama de éste es:

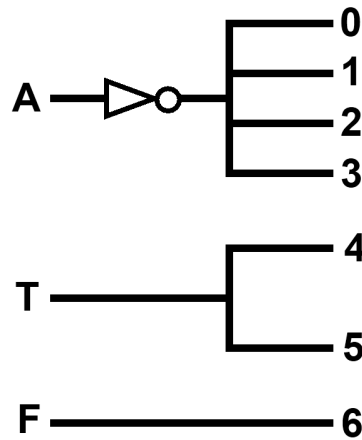


- c) Construya con compuertas lógicas el controlador de un display de 7 segmentos. Este controlador recibe como input sólo un bit, i.e., el display de 7 segmentos sólo mostrará 0 ó 1. **(2 ptos.)**

Solución: Dado la siguiente enumeración de los segmentos del display:



Un circuito controlador del display puede ser el siguiente:



Donde T corresponde a una señal constante de 1 lógico (true) y F a una señal constante de 0 lógico (false).

Pregunta 4

Una imagen en escala de grises se define por su tamaño en pixeles y la profundidad de cada uno de estos, i.e., la cantidad de valores distintos que puede tomar un pixel. En base a estos, conteste las siguientes preguntas:

1. Dada una memoria con bus de direccionamiento y palabras de 1 byte, ¿cómo almacenaría en ella una imagen en escala de grises de 5×5 pixeles y 16 bits de profundidad? **(0.75 ptos.)**

Solución: La imagen se puede representar correctamente mediante una matriz, que se almacena usando la secuencia de filas, donde cada uno de los elementos ocupa 2 palabras en la memoria.

2. Obtenga una expresión para la dirección de memoria del elemento (i, j) de la imagen del ejercicio anterior, donde $(1, 1)$ corresponde a las coordenadas del pixel superior izquierdo. Asumiendo que la imagen está almacenada en la memoria a partir de la dirección $0x6E$, calcule la dirección de memoria del pixel ubicado en $(3, 4)$. **(0.75 ptos.)**

Solución: La expresión para la dirección de memoria de un elemento de una matriz está dada por:

$$dir(matriz[i, j]) = dir(matriz) + i \times sizeof(matriz[i, j]) \times cols + j \times sizeof(matriz[i, j])$$

Dado que la imagen está con un desfase de un pixel por dimensión (comienza en $(1, 1)$) con respecto a una matriz, sus dimensiones son 5×5 y cada elemento tiene un tamaño de dos palabras, la expresión anterior adaptada a la image queda de la siguiente manera:

$$dir(img[i, j]) = dir(img) + (i - 1) \times 2 \times 5 + (j - 1) \times 2$$

Evaluando esta expresión en el pixel $(3, 4)$ y en la dirección de inicio de la image, $0x6E$, obtenemos la siguiente dirección de memoria:

$$dir(img[3, 4]) = 0x6E + 2 \times 2 \times 5 + 3 \times 2 = 0x6E + 20 + 6 = 0x88 = 136$$

3. Una imagen en colores se diferencia de una imagen en escala de grises en que cada uno de los pixeles posee ahora 3 valores de intensidad distintos, uno para cada color (RGB). Esto es análogo a tener 3 imágenes de escala de grises distintas, donde cada una describe la distribución de intensidad para un color.
¿Como almacenaría una imagen en colores de 7×7 pixeles y 1 byte de profundidad por color en una memoria igual a la descrita anteriormente? **(1.5 ptos.)**

Solución: Para almacenar una imagen en colores, basta con almacenar los 3 valores de color de cada pixel uno detrás de otro. De esta manera, almacenar esta imagen en específico es análogo a almacenar una matriz en orden de filas, donde cada uno de los elementos tiene 3 palabras de largo.

4. ¿Porqué no sirve esta misma memoria para almacenar una imagen en color de 7×7 pixeles y 2 bytes de profundidad por color? Indique el problema y una posible solución. **(1 pto.)**

Solución: Una imagen en colores de 7×7 pixeles y bytes bytes de profundidad por color, pesa $7 \times 7 \times 3 \times 2 = 294$ bytes. Dado que la memoria que se tiene es de sólo $2^8 \times 1$ bytes, la imagen no alcanza. La solución es utilizar una memoria de mayor tamaño, i.e., que tenga una mayor cantidad de palabras disponibles.

5. Obtenga una expresión para la dirección de memoria del elemento (i, j, k) de una imagen en colores, donde i, j es la ubicación del pixel y k es uno de los tres canales de color. **(2 ptos.)**

Solución: Para obtener una expresión, basta sólo con adaptar la expresión obtenida anteriormente:

$$\begin{aligned} \text{dir}(\text{img}[i, j, k]) &= \text{dir}(\text{img}) + i \times \text{sizeof}(\text{img}[i, j, k]) \times 3 \times \text{cols} + \\ &\quad j \times \text{sizeof}(\text{img}[i, j, k]) \times 3 + k \times \text{sizeof}(\text{img}[i, j, k]) \end{aligned}$$