

ESTRUCTURA DE COMPUTADORES

AUTOEVALUACIÓN

Tema 0

1. ¿Cuál es el elemento dominante de una operación AND? ¿Y de una O?
2. Para la representación de números binarios con signo ¿qué ventajas tiene el complemento a dos que hace que sea el más utilizado?
3. ¿Qué es la representación BCD?
4. La notación científica permite representar de manera aproximada números muy grandes o muy pequeños con un número de dígitos fijos ¿cómo se llama la representación en base dos para computadoras?
5. ¿Qué es el bit de paridad?
6. ¿Qué es la distancia Hamming de un código? ¿Cuál es la distancia necesaria para detectar y corregir errores de dos bits?
7. ¿Qué es el código Hamming?
8. ¿Cuántos bits de redundancia/paridad son necesarios para 5 bits de datos? ¿En qué posiciones se colocan dichos bits?
9. En recepción se calculan los bits de control, si todos son cero es que no ha habido error; pero en caso de que haya un error, ¿cómo sabemos en qué bit?
10. Una vez obtenida la señal que indica si un bit es o no erróneo, ¿qué operación se realiza para corregirlo (invertirlo)?

Tema 1

1. ¿Qué es la UAL (Unidad Aritmético Lógica)?
2. ¿De qué se compone la CPU (Unidad Central de Proceso)?
3. ¿Qué es la Ruta de Datos?
4. ¿Qué es la palabra de control?
5. La UC (Unidad de Control) genera la palabra de control ¿a partir de qué información?
6. ¿Cómo genera la palabra de control la UC cableada?
7. La UC microprogramada tiene las palabras de control almacenadas en la memoria de microprograma, ¿cómo se direcciona esta memoria?
8. En caso de tratarse de una UC de ciclo múltiple la ejecución de una instrucción necesitará, por lo general, de la ejecución de un microprograma (secuencia de microinstrucciones o palabras de control). La secuenciación de dichas microinstrucciones puede ser implícita o explícita. ¿Cómo funciona y qué supone cada uno de estos mecanismos de secuenciación?
9. ¿Qué es la ejecución en canalización o pipeline?
10. ¿Qué es el riesgo de datos? ¿Cómo evitarlo?

Tema 2

1. El formato de instrucción es una caja rectangular que representa los bits del código binario de la instrucción. Los campos y número de bits de estos dependen del formato de instrucción. Un campo que siempre aparece es el *código de operación* o *instrucción*. ¿Qué representa ese conjunto de bits?
2. Un campo que puede aparecer o no es el campo de dirección, de hecho puede no haber ninguno, haber uno, dos o incluso tres campos de dirección. ¿Qué diferencia habrá en un programa que realiza una operación si se utilizan instrucciones con más o menos campos de dirección?
3. ¿Qué es la dirección efectiva?
4. ¿Qué es el campo de *modo* y cómo se relaciona con el campo *dirección*?
5. Si el direccionamiento es directo, el campo dirección contiene la dirección efectiva, ¿y si el direccionamiento es indirecto?
6. En el microprocesador 8085 un par de registros (generalmente la pareja HL) puede utilizarse como puntero a memoria ¿qué modo de direccionamiento se está utilizando en este caso?
7. Los tipos de instrucciones se clasifican en tres: las de transferencia de datos (mueven datos entre registros, de memoria a registro o viceversa), de procesamiento de datos (realizan operaciones aritméticas, lógicas o de desplazamiento) y de control de secuencia. ¿Para qué sirven estas últimas?
8. Las instrucciones lógicas permiten modificar los bits deseados (ponerlos a '1' o a '0') manteniendo el resto como estaban. Para ello es necesario una palabra con una combinación concreta de ceros y unos. ¿Cómo se denomina dicha palabra?
9. Las instrucciones PUSH y POP introducen y extraen respectivamente un elemento de la pila, pero ¿qué es la pila? ¿Y el SP (puntero de pila)?
10. ¿Qué es el diagrama de flujo?

Tema 3

1. ¿Qué son las subrutinas?
2. Si durante la ejecución secuencial de un programa se encuentra una instrucción de salto, la ejecución del programa seguirá a partir de la dirección que en ella se indique. ¿Qué diferencia hay entonces entre una instrucción de salto y una llamada a subrutina?
3. La instrucción RET no toma parámetros, por lo que la dirección de retorno ha de guardarse en un lugar conocido; pero además sabemos que una llamada a subrutina puede darse durante la ejecución de otra subrutina. ¿Cuál es la solución para almacenar la dirección de retorno permitiendo además el anidamiento de subrutinas?
4. ¿Qué es el bloque de activación?
5. ¿Por qué es necesario el puntero al bloque de activación para acceder a los parámetros, espacio de resultados, etc. mediante un direccionamiento indexado?
6. ¿Qué diferencia hay entre una llamada a subrutina y una interrupción (hardware)?

7. Un *trap* es un tipo de interrupción generada por la CPU ante una condición de error, ¿por qué esta es síncrona mientras que la interrupción hardware es asíncrona?
8. Cuando se produce una llamada a subrutina es necesario guardar la dirección de retorno (el valor del PC); ante una interrupción hay que guardar además la información del bloque de registros o al menos el acumulador y los flags de estado, ¿a qué se debe esta diferencia?
9. ¿Qué tres formas existen para identificar la causa de la interrupción?
10. Una interrupción puede indicar la dirección de su rutina de servicio a la interrupción (RSI) o simplemente un vector que la identifique. En el segundo caso ¿dónde se encuentra la RSI?

Tema 4

1. Define los métodos de acceso SAM, RAM y CAM.
2. ¿Cuál de estos tiempos es mayor, el tiempo de acceso (T_A) o el tiempo de ciclo de memoria (T_C)?
3. La gran mayoría de las memorias principales son memorias de semiconductor de acceso aleatorio. Las memorias secundarias pueden ser magnéticas, ópticas o magneto-ópticas. Los discos duros (HDD) magnéticos están siendo sustituidos por discos de estado sólido (SSD) en los portátiles, ¿qué tipo de memoria es esta? ¿Qué ventajas/inconvenientes presentan frente a los HDD?
4. Las memorias de solo lectura (ROM) tienen un contenido inalterable. Las memorias PROM son una variante que permiten una única escritura por parte del usuario. Compuestas por una matriz fija de puertas AND y una matriz programable de puertas OR ¿para qué se utiliza este tipo de memorias?
5. Define los términos, dinámico/estático y volátil/no volátil en referencia a las características de una memoria.
6. ¿Cómo se interpreta la información del bus de direcciones para una organización 2D (o lineal) y para una organización 3D (o por coincidencia)?
7. Explica el concepto *jerarquía de memorias*.
8. ¿Qué es el principio de localidad?
9. ¿Qué es la memoria caché?
10. ¿Cómo se define la tasa de acierto (en relación con la memoria caché)?
11. Si la correspondencia es directa, a cada bloque de memoria le corresponde una línea de caché, lo que hace más fácil buscarla en la caché, pero también impone una clara limitación ¿cuál?
12. La dirección total de un dato en memoria principal se divide, en el caso de que la correspondencia sea asociativa por conjuntos, en: *etiqueta*, *línea* y *palabra* (o *byte*). El último localiza el dato dentro del bloque y la *línea* indentifica una de las posibles líneas de la caché; ¿para qué es necesaria la *etiqueta*?
13. Si el método de correspondencia es totalmente asociativo, cualquier bloque de memoria principal puede alojarse en cualquier línea de caché; en caso de que la caché esté llena, hay que decidir qué bloque se desaloja para meter en nuevo. Para ello

- pueden seguirse diferentes algoritmos de sustitución, define los siguientes: FIFO, LRU y LFU.
14. Si un dato es modificado en la caché, ha de actualizarse en memoria principal. A este respecto la estrategia elegida puede ser la escritura inmediata (*write through*) o la post-escritura (*write back*). Descríbelas indicando sus ventajas e inconvenientes.
 15. ¿Qué es la memoria virtual?
 16. El mapeado es el mecanismo por el que las direcciones virtuales se traducen a direcciones físicas. ¿Qué campos componen estas direcciones y en qué consiste este proceso de mapeado?
 17. Familia de memorias RAM.
 18. Familia de memorias ROM.
 19. Se pretende diseñar una memoria de 4kB con circuitos integrados de 2kB, ¿cuántos C.I.s hacen falta? ¿Cuántos chip select (CS) hay que generar? ¿Qué bits del bus de direcciones utilizarías para ello?
 20. Se pretende diseñar una memoria de 4kB con circuitos integrados de 2kx4, ¿cuántos C.I.s hacen falta? ¿Cuántos chip select (CS) hay que generar? ¿Qué bits del bus de direcciones utilizarías para ello?

Tema 5

1. Pon algún ejemplo de periféricos de entrada, de salida y de entrada salida.
2. ¿Por qué es necesario el controlador de periférico?
3. Describe los métodos de sincronización *strobing* y *handshaking*.
4. Para la transferencia de datos entre los dispositivos de E/S y memoria puede utilizarse la E/S programada (sin interrupciones) o la E/S mediante interrupciones. ¿Qué ventaja supone esta segunda opción?
5. Describe el denominado Acceso Directo a Memoria (DMA).
6. ¿Cómo sabe la CPU si la transferencia de datos mediante DMA se ha completado correctamente?
7. Describe el funcionamiento de un monitor de cristal líquido (LCD).
8. Un disco duro contiene varios discos o platos, que se dividen en pistas y sectores. ¿Cómo se llaman los tiempos medios para acceder a cada uno de ellos?
9. ¿Cómo se determina qué tecla se ha pulsado en un teclado?
10. El puerto paralelo tiene un conector de 25 pines (DB25) por el que se transmiten datos en paralelo, ¿para la conexión de qué dispositivo era mayormente utilizado (aunque hoy en día está siendo sustituido por el USB)?

Tema 6

1. Los buses son líneas que permiten la transferencia de información. Las prestaciones de estos pueden disminuir si se le conectan muchos dispositivos, ¿cuál es la solución a este problema?

2. ¿Qué diferencia hay entre una arquitectura tradicional de bus y una arquitectura de altas prestaciones?
3. ¿Por qué es necesario el arbitraje entre dispositivos que quieren acceder a un bus?
4. ¿En qué se diferencian el arbitraje centralizado y el distribuido?
5. ¿Qué diferencia hay entre un bus dedicado y uno multiplexado?
6. ¿Y entre uno síncrono y uno asíncrono?
7. ¿Y entre uno serie y uno paralelo?
8. El método de arbitraje puede ser centralizado o distribuido, y para cada uno de ellos, la estrategia puede ser *Daisy-Chain*, *encuesta* o *petición independiente*. Describe brevemente estas tres estrategias de arbitraje.
9. Si en un bus síncrono se conecta un dispositivo más lento que los demás, una solución es añadir una señal (WAIT) que permita solicitar otro ciclo de reloj cuando sea necesario, convirtiendo así el bus en un bus
10. El protocolo PCI define un bus AD para direcciones (Address) y datos (Data), es por lo tanto un bus
11. El arbitraje en el bus PCI es
12. ¿Cuáles son las ideas iniciales de diseño del bus USB?
13. ¿El bus USB permite una comunicación serie o paralela?
14. El bus USB se configura con una topología de árbol, ya que a cada concentrador pueden conectarse varios dispositivos de E/S u otros concentradores. ¿Cómo direcciona los datos de la CPU a un dispositivo en concreto?
15. ¿Cómo puede un dispositivo final comunicarse con otro?