

| | |
|----------------|---------------|
| Nombre: | |
| DNI: | Grupo: |

Test de Teoría (3.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 3/30 si es correcta, 0 si está en blanco o claramente tachada, -1/30 si es errónea.

Anotar las respuestas (a, b, c o d) en la siguiente tabla.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

- En una máquina little-endian con memoria de bytes y representación en complemento a dos que permite accesos a memoria de tamaño byte (1B), media palabra (2B) y palabra (4B), si se almacena en la posición 0xBABC una palabra de valor -2, ¿qué se obtendría al consultar la media palabra en la posición 0xBABE?
 - 0
 - 1
 - 1
 - 2
- Según la clasificación m/n, las máquinas de acumulador son de tipo
 - 0/0
 - 1/1
 - 1/2
 - 2/2 ó 2/3
- Según la clasificación m/n, las máquinas con arquitectura R/R son de tipo
 - 0/0
 - 0/x con x=2,3
 - x/0 con x=2,3
 - x/x con x=2,3
- El bus del sistema es
 - el que conecta las distintas partes del sistema: UC, ALU, E/S, M
 - en un sistema con bus único, todo el bus salvo la parte relacionada con E/S (SATA, GPU, USB, Ethernet, etc)
 - en un sistema con buses separados, el que conecta el sistema E/S con el resto
 - el que conecta CPU-M, ya sea un sistema con bus único o con múltiples buses
- Un sistema con direcciones de 8bits utiliza una puerta NAND conectada a las líneas A7...A5 para atacar la entrada CS# (activa baja) de un módulo de memoria. En el mapa de memoria las siguientes posiciones corresponderán a dicho módulo
 - 0x00 a 0x0f y 0x80 a 0x8f
 - 0x00 a 0x1f
 - 0x70 a 0x7f y 0xf0 a 0xff
 - 0xe0 a 0xff
- La ecuación básica de rendimiento calcula
 - cuánto tiempo tarda en ejecutarse un programa concreto conociendo su número de instrucciones y el número de etapas (promedio) y la frecuencia del procesador
 - cómo de mejor es un procesador frente a otro, conociendo las prestaciones de las respectivas UC, ALU, E/S y M
 - el promedio de las ganancias obtenidas con una serie de programas de punto entero

- d. la media geométrica de los cocientes entre los tiempos de ejecución de una serie de programas predeterminados
-
7. Por x86-64 se entiende la misma arquitectura de repertorio (ISA) que
- x86
 - IA-32
 - IA-64
 - AMD64
-
8. ¿Qué combinación de *flags* aritmético-lógicos corresponde al código de condición b (*below*)?
- CF
 - OF
 - CF xor OF
 - OF xor SF
-
9. En los modos de direccionamiento del tipo *Desplazamiento(Base,Indice,Factor Escala)*, puede usarse como
- desplazamiento, cualquier constante de 1, 2 o 4 bytes (incluso el nombre de una variable, por su dirección)
 - base, cualquiera de los 8 registros enteros salvo %esp
 - índice, también cualquiera salvo %ebp
 - factor de escala, cualquier constante de 1, 2, 4 u 8 bytes
-
10. La diferencia entre las instrucciones **test** y **cmp** consiste en que
- test realiza una operación *and* lógico, mientras que cmp realiza una resta
 - test modifica sólo los *flags* lógicos (ZF,SF) mientras que cmp modifica los aritmético-lógicos (ZF,SF,CF,OF)
 - ambas respuestas son correctas
 - ambas respuestas son incorrectas
-
11. Para traducir una construcción **if-then-else** de lenguaje C a lenguaje ensamblador, gcc utiliza generalmente
- un salto condicional, según la condición expresada en el código C
 - un salto condicional, según la condición opuesta a la del código C, y otro salto incondicional
 - dos saltos condicionales (uno para la parte if y otro para la parte else)
 - dos saltos condicionales y dos saltos incondicionales
-
12. Para traducir una construcción **do-while** de lenguaje C a lenguaje ensamblador, gcc utiliza generalmente
- un salto condicional hacia adelante, según la misma condición que en lenguaje C
 - un salto condicional hacia atrás, según la misma condición que en lenguaje C
 - un salto condicional hacia adelante, según la condición opuesta a la de lenguaje C
 - un salto condicional hacia atrás, según la condición opuesta a la de lenguaje C
-
13. alguna de las siguientes señales **no** sirve de entrada a la unidad de control. ¿Cuál?
- señal de reloj (CLK)
 - estado de la unidad de proceso (*flags* Z, S, C, O...)
 - instrucción actual (bits del registro IR)
 - contador de programa (bits del registro PC)
-
14. alguna de las siguientes señales **no** es salida de la unidad de control. ¿Cuál?
- señales de carga, habilitación y/o desplazamiento de registros (*Load, Enable, ShiftL, ShiftR*)
 - códigos de selección en multiplexores, decodificadores, ALU, etc (00, 01, 10, 11...)
 - señales de lectura y escritura en memoria (RD, WR)
 - dirección de la siguiente microinstrucción (bits del campo DIR o Memoria B de Wilkes)
-
15. alguna de las siguientes **no** es una operación básica de la unidad de control
- transferir un registro a otro

- b. (leer o escribir) un registro (de / a) memoria
 - c. (guardar o recuperar) un registro (en / de) la pila
 - d. realizar operación ALU y guardar resultado en registro
-

16. Tipos de riesgos que hemos estudiado en cauces segmentados (señalar la opción **incorrecta**)

- a. riesgos estructurales
 - b. riesgos de (dependencia de) datos
 - c. riesgos de control
 - d. riesgos de transferencia
-

17. ¿Cuál de los siguientes modos de direccionamiento es **menos** preferible para un procesador con segmentación de cauce?

- a. registro
 - b. directo
 - c. indirecto a través de registro
 - d. indexado (o relativo a base, o base+índice)
-

18. La ganancia en velocidad ideal de un cauce de K etapas de igual duración T ejecutando un programa de N instrucciones es

- a. $S = KN/(K-N+1)$
 - b. $S = NKT/(N-K+1)T$
 - c. $S = KN/(K+N-1)$
 - d. $S = NT/(N+K-1)T$
-

19. Respecto al salto retardado y al salto anulante, ¿cuál permite que se ejecute la siguiente instrucción, y cuál no?

- a. el retardado ejecuta la siguiente instrucción (con el correspondiente retraso), el anulante no la ejecuta (de hecho la anula)
- b. el retardado la ejecuta sólo si se cumple la condición de salto, el anulante sólo si no se cumple
- c. el retardado la ejecuta sólo si no se cumple la condición de salto, el anulante no la ejecuta nunca

- d. el retardado la ejecuta siempre, el anulante la ejecuta sólo si se cumple la condición de salto
-

20. Respecto a las técnicas de direccionamiento por selección lineal, decodificación centralizada y distribuida

- a. todas ellas impiden que haya cortocircuito en el bus de datos
 - b. todas ellas impiden que haya cortocircuito en el bus de direcciones
 - c. la selección lineal permitiría escribir un mismo dato a varios puertos E/S
 - d. usando decodificación centralizada es más fácil realizar expansiones al sistema de E/S
-

21. Ventajas de la E/S independiente (separada, aislada) (señalar la opción **incorrecta**)

- a. diseño del procesador más sencillo (E/S mapeada añade complejidad al diseño)
 - b. protección de E/S más fácil (E/S mapeada añade dificultad a la protección de E/S)
 - c. mayor aprovechamiento del espacio de memoria (E/S mapeada resta espacio a la memoria)
 - d. decodificación de memoria más elegante, limpia, sencilla (E/S mapeada añade complejidad a la decodificación)
-

22. Parecidos y diferencias entre los métodos de E/S (señalar la opción **incorrecta**)

- a. la consulta del estado del dispositivo por parte de la CPU se suele/puede hacer con E/S programada y con E/S por IRQ
 - b. se suele avisar a la CPU (con una IRQ) de que debe realizar alguna tarea, tanto en E/S por IRQ como en E/S por DMA
 - c. sólo E/S por DMA libera a la CPU de realizar la consulta de estado del dispositivo
 - d. sólo E/S por DMA libera a la CPU de realizar la transferencia de los datos de E/S
-

23. Tipos de interrupción que suelen contemplar las CPUs comerciales actuales (señalar la opción **incorrecta**)
- a. internas (excepciones o *traps*): generadas internamente por la CPU para indicar una condición que requiere atención (división por cero, *codop* inválido, etc)
 - b. externas (IRQs *hardware*): generadas por un dispositivo externo a la CPU, activan la línea INTR# (o equivalente)
 - c. software: generadas al ejecutar la instrucción INT (o equivalente)
 - d. firmware (*faults*): generadas por el microcódigo de la CPU (*segmentation fault*, *page fault*, etc)
-
24. Técnicas que se pueden usar para determinar la causa de una interrupción (señalar la opción **incorrecta**)
- a. múltiples líneas de interrupción INT1#, INT2#...
 - b. línea de reconocimiento INTA#
 - c. consulta de estado, o *polling*
 - d. interrupciones vectorizadas
-
25. Respecto a salvaguardar los registros de la CPU al inicio de una rutina de servicio de interrupción (ISR)
- a. no es necesario salvar ninguno más, si el contador de programa y los *flags* de estado ya los salva la propia CPU como parte del mecanismo de interrupción
 - b. se deben guardar los registros que se modifiquen en la propia ISR. Eso es posible hacerlo porque el propio programador de la ISR conoce qué registros va a modificar
 - c. se deben guardar los registros salva-invocado (p.ej. EBX, ESI, EDI en el caso de una CPU IA-32), los registros salva-invocante ya los guarda el programa interrumpido
 - d. se deben guardar todos los registros, para restaurarlos a la salida y así garantizar que el programa interrumpido no sufre ninguna modificación (salvo el inevitable retraso temporal) debido a la interrupción
-

26. En EC podemos usar la palabra *directo* para referirnos a... (señalar la opción **incorrecta**)
- a. dispositivo de almacenamiento secuencial directo (DASD)
 - b. cache con correspondencia directa
 - c. acceso directo a memoria
 - d. modo de direccionamiento directo
-
27. Variación de los parámetros de los distintos niveles en una jerarquía de memoria (señalar la opción **incorrecta**)
- a. unidad de transferencia: $x_i \geq x_{i+1}$
 - b. ancho de banda: $b_i \geq b_{i+1}$
 - c. tamaño del nivel: $s_i \leq s_{i+1}$
 - d. tiempo de acceso: $t_i \leq t_{i+1}$
-
28. Las técnicas *write-through* y *write-back* están relacionadas con
- a. coherencia de cache
 - b. métodos de E/S
 - c. arbitraje de buses
 - d. etapas de la unidad de control
-
29. Para diseñar una memoria con ancho de palabra $k \cdot m$ (y mismo n° palabras que los módulos) a partir de módulos con ancho de palabra m , se utilizan k módulos
- a. repartiendo las líneas de datos entre los k módulos: el primero se conecta a $D_0 \dots D_{k-1}$, el segundo a $D_k \dots D_{2k-1}$, etc
 - b. repartiendo las líneas de dirección: el 1º se conecta a $A_0 \dots A_{k-1}$, el 2º a $A_k \dots A_{2k-1}$, etc
 - c. repartiendo líneas datos: el 1º se conecta a $D_0 \dots D_{m-1}$, el 2º a $D_m \dots D_{2m-1}$, etc
 - d. repartiendo líneas dirección: el 1º a $A_0 \dots A_{m-1}$, el 2º a $A_m \dots A_{2m-1}$, etc
-
30. En un sistema con direcciones de 32bits, memoria de bytes, cache de 1MB asociativa por conjuntos de 4 vías y líneas de 64B, el campo etiqueta en el formato de dirección cache es de
- a. 16bits
 - b. 14bits
 - c. 12bits
 - d. 10bits
-