



DNI:	Apellidos:	Nombre:
------	------------	---------

Cuadrícula de respuestas:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
C✓	C✓	C✓	A✓	C✓	B✓	A✓	C✓	B✓	C✓	B✓	A✓	C✓	B✓	A✓	B✓	C✓	A✓	C✓	C✓

Bien	Mal	NC

Instrucciones del examen

1. **Duración:** 1 hora y media.
2. **Normas en el aula:** Queda estrictamente prohibido abandonar el aula sin entregar el examen, así como el uso de cualquier tipo de documentación, calculadoras o dispositivos de comunicación.
3. **Al acabar el examen:** Deberás entregar tu hoja de respuestas y devolver el material de apoyo proporcionado por el profesorado. (Puedes quedarte con tu copia del examen.)
4. **Criterios de calificación:** El examen se corregirá **únicamente** a partir de las respuestas anotadas en la cuadrícula.
 - Respuesta correcta: +0,5 puntos.
 - Respuesta errónea: $-0,167$ puntos (resta $\frac{0,5}{3}$).
 - Pregunta no contestada: no puntúa.





1. ¿Por qué el lenguaje ensamblador proporciona pseudoinstrucciones si estas no existen físicamente en el repertorio de la arquitectura?
 - a) Para engañar al compilador de alto nivel y reducir el tiempo de compilación, ocultándole las instrucciones máquina reales que se generan para cada operación.
 - b) Para permitir que el procesador ejecute operaciones que sus circuitos no soportan físicamente, ampliando el repertorio real de instrucciones sin necesidad de rediseñar el hardware.
 - c) Para facilitar la labor del programador, permitiendo que el ensamblador genere secuencias de instrucciones complejas automáticamente.
 - d) Para que el código fuente ensamblador sea portable entre arquitecturas de distintos fabricantes (como Intel x86), evitando tener que reescribirlo al cambiar de procesador.
2. En el programa del Apéndice A, la subrutina «suba» preserva r4 y lr con «push»/«pop», pero la función «subb» retorna directamente con «mov pc, lr» sin apilar ningún registro. ¿Cuál es la razón correcta para cada caso?
 - a) «suba» apila r4 para pasarle el argumento adicional a «subb» a través de la pila, puesto que la instrucción «bl» no puede transmitir más de un parámetro mediante registros según el convenio de llamadas ARM.
 - b) «suba» apila registros porque contiene varias instrucciones que alteran el flujo de control; «subb» no los apila porque, al tener una sola instrucción de cómputo, el procesador gestiona automáticamente la preservación de registros sin necesidad de apilarlos.
 - c) «suba» apila lr porque llama a «subb» con «bl» (lo que sobrescribiría lr), y apila r4 porque lo usa como registro temporal, debiendo restituir su valor al llamante. «subb» no modifica ningún registro fuera del r0 al r3, ni realiza llamadas, por lo que no necesita apilar nada.
 - d) Ambas funciones deberían apilar lr al inicio y recuperarlo con «pop» al final; «subb» omite el «push» porque el ensamblador detecta en tiempo de compilación que lr no se modifica y elimina las instrucciones de apilado redundantes.
3. ¿Por qué surgió históricamente la necesidad de los lenguajes de alto nivel frente al uso exclusivo del lenguaje ensamblador?
 - a) Porque el lenguaje ensamblador no permitía realizar operaciones matemáticas complejas ni expresar estructuras de datos de alto nivel como registros, vectores o punteros.
 - b) Para aumentar la velocidad de ejecución de los microprocesadores de primera generación, acercándola a la de las memorias y unidades de E/S disponibles en aquel momento.
 - c) Para permitir que un mismo programa pudiera ejecutarse en distintas arquitecturas sin ser reescrito totalmente.



- d) Para reducir el número de transistores necesarios en la Unidad de Control.
4. La pila en ARM crece hacia direcciones bajas de memoria. Si el puntero de pila **SP** vale **0x2007 06FC** y se ejecuta la instrucción «**push** {r4, r5, lr}», ¿cuál será el nuevo valor del **SP**?
- a) **0x2007 06F0**
 - b) **0x2007 06F4**
 - c) **0x2007 0708**
 - d) **0x2007 06FC**
5. Si un fabricante decide añadir una memoria caché para solucionar el «cuello de botella» entre procesador y memoria, ¿qué está modificando en el computador?
- a) La arquitectura del conjunto de instrucciones (ISA).
 - b) El mapa de memoria lógico visto por el programador.
 - c) La organización del sistema y su jerarquía de memoria.
 - d) Los modos de direccionamiento de carga y almacenamiento.
6. En una organización *Big Endian*, si leemos un byte de la dirección **0x2007 0002** donde se ha guardado previamente la palabra **0xAABB CCDD** (en la dirección **0x2007 0000**), ¿qué valor obtendremos?
- a) **0xBB.**
 - b) **0xCC.**
 - c) **0xAA.**
 - d) **0xDD.**
7. En las arquitecturas RISC como ARM, ¿cuál es la característica predominante en el uso de los operandos en las instrucciones de transformación de datos?
- a) Suelen tener tres operandos ubicados siempre en registros o en la instrucción (excepto en instrucciones de carga/almacenamiento).
 - b) Los operandos siempre deben residir en la memoria principal.
 - c) Utilizan exclusivamente el acumulador como destino.
 - d) El formato de instrucción es de tamaño variable según el modo de direccionamiento, lo que aumenta la densidad de código a costa de una mayor complejidad en la etapa de decodificación.
8. ¿Qué restricción de alineamiento impone ARM para el acceso a una *media palabra* (half-word) en memoria?
- a) Debe estar en una dirección múltiplo de 4.



- b) No tiene restricciones; puede estar en cualquier dirección de memoria.
 - c) Debe estar en una dirección múltiplo de 2.
 - d) Debe estar siempre en direcciones impares.
9. Un controlador de E/S dispone de varios tipos de registros. Si el procesador necesita saber si un dispositivo ha terminado una operación o si ha ocurrido algún error, ¿qué registro debe consultar?
- a) Registro de Control.
 - b) Registro de Estado.
 - c) Registro de Datos de Entrada.
 - d) Registro de Propósito General.
10. ¿Cuál es la diferencia de comportamiento entre las directivas «**.balign 4**» y «**.space 4**»?
- a) Son equivalentes; ambas reservan una palabra de 32 bits e inicializan su contenido a cero en la sección de datos activa en el momento de su uso.
 - b) «**.balign 4**» reserva exactamente 4 bytes consecutivos en la sección actual, mientras que «**.space 4**» avanza el contador de localización hasta el siguiente múltiplo de 4.
 - c) «**.balign 4**» asegura que el siguiente dato esté en una dirección múltiplo de 4, mientras que «**.space 4**» reserva 4 bytes inicializados por defecto a cero.
 - d) «**.balign 4**» solo puede emplearse en la sección «**.text**» para alinear instrucciones, mientras que «**.space 4**» está restringida a la sección «**.data**» con el objetivo de reservar espacio para variables.
11. Al caracterizar el rendimiento de un sistema de E/S, ¿en qué se diferencia la *productividad máxima* de la *productividad de una transacción*?
- a) La productividad máxima incluye en su cómputo el tiempo de latencia inicial de cada operación, mientras que la productividad de una transacción lo descuenta para reflejar únicamente el rendimiento sostenido.
 - b) La máxima se calcula considerando el mejor caso posible y excluyendo la latencia, mientras que la transaccional incluye todo el tiempo desde el inicio hasta el fin de la operación.
 - c) La productividad máxima caracteriza solo dispositivos de entrada (teclado, escáner), mientras que la productividad de una transacción es aplicable únicamente a dispositivos de salida o bidireccionales.
 - d) No hay diferencia matemática entre ambas métricas; la distinción es meramente comercial y varía según el fabricante del dispositivo.
12. El ciclo de instrucción se compone de varias fases. ¿En qué subetapa de la fase de ejecución se determina físicamente la transformación de los datos, como por ejemplo una operación lógica bit a bit en la ALU?



- a) Ejecución.
 - b) Lectura de los operandos.
 - c) Lectura de la instrucción.
 - d) Escritura de resultados.
13. Considerando el programa del Apéndice A, ¿qué valor contiene el registro `r0` justo antes de ejecutar la instrucción «`wfi`»?
- a) 3.
 - b) 6.
 - c) 11.
 - d) 16.
14. ¿Para qué se utiliza un registro de control en un dispositivo de E/S?
- a) Indicar al procesador si el dispositivo ha completado satisfactoriamente la operación en curso.
 - b) Configurar el modo de funcionamiento del dispositivo (velocidad, paridad, dirección de transferencia, etc.).
 - c) Almacenar temporalmente los datos en tránsito entre el dispositivo y la memoria principal, actuando como búfer para absorber la diferencia de velocidad entre ambos extremos de la transferencia.
 - d) Contener la dirección base de memoria destino utilizada por el controlador DMA, de modo que pueda escribir datos directamente en RAM sin intervención del procesador principal.
15. En el contexto de la gestión de sistemas, ¿por qué la disparidad de velocidades entre el procesador y los periféricos se considera un reto de diseño crítico?
- a) Porque puede provocar que el procesador quede bloqueado inútilmente esperando respuestas de dispositivos mucho más lentos, degradando el rendimiento general.
 - b) Porque obliga a que todos los dispositivos de E/S funcionen de forma síncrona con el reloj del sistema, lo que exige circuitos de sincronización costosos en cada controlador.
 - c) Porque el procesador solo puede ejecutar instrucciones si los datos de E/S están disponibles en la memoria caché, lo que obliga a mantener coherencia permanente entre caché y periféricos.
 - d) Porque en la arquitectura Harvard el bus de instrucciones y el de datos son independientes, lo que impide acceder a dispositivos de E/S mientras se busca la siguiente instrucción.
16. ¿Qué diferencia técnica existe entre los registros denominados registros bajos y los registros altos en el juego de instrucciones Thumb?



- a) Los registros altos (r8–r15) solo pueden usarse para almacenar datos constantes, ya que carecen de puertos de escritura en la ruta de datos de la ALU.
 - b) La mayoría de instrucciones Thumb de 16 bits solo pueden operar con los registros bajos (r0–r7).
 - c) Los registros bajos (r0–r7) son de 16 bits y los altos (r8–r15) de 32 bits, por lo que las operaciones sobre datos de 32 bits requieren combinar pares de registros bajos.
 - d) Los registros altos (r8–r15) están reservados para uso interno del procesador y del sistema operativo, y no son directamente accesibles al programador en modo Thumb.
17. En la instrucción «**ldr** r3, [r0, #4]», identifica los modos de direccionamiento utilizados para el operando destino y el operando fuente, respectivamente.
- a) Directo a registro para el destino (r3), e Inmediato para la fuente, puesto que el desplazamiento #4 se codifica como constante dentro de la instrucción.
 - b) Indirecto con registro para el destino y Relativo al PC para la fuente, dado que la dirección efectiva se calcularía sumando r0 al valor del contador de programa.
 - c) Directo a registro e Indirecto con desplazamiento.
 - d) Absoluto para el destino, pues r3 referencia una posición fija de memoria, y Base más índice para la fuente, con r0 como base y #4 como índice numérico.
18. Si un controlador SATA en un PC expone registros tanto en el mapa de E/S independiente como en el mapa de memoria del sistema, esto demuestra que:
- a) Ambas estrategias de acceso no son excluyentes y pueden coexistir en una misma arquitectura si el procesador lo soporta.
 - b) El procesador del PC es exclusivamente RISC y, por ello, requiere que todos sus periféricos expongan registros tanto en el espacio de E/S como en el espacio de memoria.
 - c) El sistema está duplicando la información en ambos espacios de direccionamiento para proporcionar redundancia y tolerancia a fallos en caso de error de acceso al bus del sistema.
 - d) El dispositivo utiliza el controlador DMA para copiar automáticamente sus registros internos al mapa de memoria del sistema, eliminando la necesidad de instrucciones especiales de E/S.
19. La jerarquía de memoria busca un compromiso entre velocidad, coste y tamaño. ¿Cuál es el principio de diseño que permite que la velocidad global del sistema sea comparable a la del elemento más rápido?
- a) Incrementar el tamaño de la memoria principal (RAM) hasta igualar al del almacenamiento secundario.
 - b) Usar solo memoria ROM para evitar la volatilidad, de modo que los datos persistan entre reinicios sin necesidad de cargarlos desde el almacenamiento secundario.



- c) Mantener copias de los datos más frecuentes en la memoria caché, que funciona a la velocidad del procesador.
 - d) Eliminar el bus de datos compartido e integrar la memoria en el mismo chip que el procesador, usando únicamente operandos inmediatos en las instrucciones.
20. La conexión de red (Ethernet o Wi-Fi) se clasifica habitualmente como un dispositivo con interlocutor “Máquina” porque:
- a) Se conecta físicamente a un router o punto de acceso inalámbrico, lo que hace que toda la información transite por un nodo de red antes de llegar al destinatario final.
 - b) Los datos que transmite son siempre binarios, al igual que los del resto de dispositivos digitales, lo que la equipara en naturaleza a un disco duro o una memoria flash.
 - c) El ente final que recibe o genera los datos es otro servidor o computador, aunque el uso final sea para humanos.
 - d) Funciona a velocidades del orden de gigabits por segundo, lo que la equipara en prestaciones a otros periféricos de alta velocidad y justifica su clasificación especial.

A. Programa ARM Thumb M0

```
1      .data
2  a:    .word 3
3  b:    .word 5
4
5      .text
6
7      ldr r0, =a
8      ldr r0, [r0]
9      ldr r1, =b
10     ldr r1, [r1]
11     bl suba
12     wfi
13
14  suba: push {r4, lr}
15         mov r4, r1
16         bl subb
17         add r0, r0, r4
18         pop {r4, pc}
19
20  subb: lsl r0, r0, #1
21         mov pc, lr
```