



<b>DNI:</b> elprofe	<b>Apellidos:</b>	<b>Nombre:</b>
------------------------	-------------------	----------------

**Cuadrícula de respuestas:**

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

<b>Bien</b>	<b>Mal</b>	<b>NC</b>
<b>20</b>	<b>0</b>	<b>0</b>

**Normas**

1. La duración del examen será de 1 hora y media.
2. No está permitido:
  - a) Abandonar el aula sin entregar el examen.
  - b) Utilizar cualquier tipo de documentación.
  - c) Utilizar calculadora.
3. El examen se calificará teniendo en cuenta únicamente las respuestas anotadas en la cuadrícula de respuestas. Cada respuesta correcta sumará 0,5 puntos. Cada respuesta errónea restará 0,5/3 puntos. Las preguntas no contestadas no se tendrán en cuenta.



1. ¿Cómo se denomina el mecanismo mediante el cual un procesador ejecuta las instrucciones?
  - a) Ciclo de instrucción.
  - b) Pipeline.
  - c) Loop de ejecución.
  - d) Secuencia de control.
  
2. ¿Cuál de las siguientes afirmaciones es cierta?
  - a) El lenguaje ensamblador es más abstracto que el lenguaje de alto nivel.
  - b) El lenguaje máquina es directamente entendido por el hardware sin necesidad de traducción.
  - c) Los lenguajes de alto nivel son menos eficientes y no se utilizan en la programación moderna.
  - d) El lenguaje ensamblador y el lenguaje de máquina son lo mismo.
  
3. ¿Qué describe mejor la ISA de un computador?
  - a) El sistema operativo utilizado en el computador.
  - b) El conjunto de instrucciones que el procesador puede ejecutar.
  - c) El diseño físico del chip del procesador.
  - d) La velocidad del reloj del procesador.
  
4. ¿Cuál es la principal diferencia entre la arquitectura y la organización de un computador?
  - a) La arquitectura se refiere al software, mientras que la organización se refiere al hardware.
  - b) La arquitectura es el diseño conceptual y la estructura lógica, mientras que la organización es la implementación física.
  - c) La arquitectura y la organización son términos intercambiables en informática.
  - d) La organización define el conjunto de instrucciones, mientras que la arquitectura define la implementación física.
  
5. ¿Cuál de las siguientes es una característica que diferencia a las memorias en la jerarquía de memorias?
  - a) Todas las memorias tienen la misma velocidad y capacidad.
  - b) Las memorias secundarias son generalmente más rápidas que las memorias primarias.
  - c) Las memorias de caché son más rápidas pero tienen menos capacidad que la memoria principal (RAM).
  - d) La memoria virtual es físicamente más grande que la memoria caché.



6. ¿Cuál de las siguientes opciones describe correctamente el juego de instrucciones Thumb de ARM M0?
  - a) Un conjunto de instrucciones de 32 bits que permite la compatibilidad con instrucciones de 16 bits.
  - b) Un conjunto de instrucciones de 16 bits con alguna instrucción de 32 bits.
  - c) Un conjunto de instrucciones de 8 bits optimizado para aplicaciones empotradas.
  - d) Un conjunto de instrucciones exclusivo de la arquitectura ARM64.
7. ¿Cuál de las siguientes es una seudoinstrucción en la arquitectura ARM?
  - a) «**add** r0, r1, r2».
  - b) «**ldr** r0, =0x2000».
  - c) «**b** 0x100».
  - d) «**str** r0, [r1]».
8. ¿Para qué sirve la directiva «**.data**» en el ensamblador de GCC para ARM?
  - a) Para definir una sección de código.
  - b) Para reservar espacio en memoria para almacenar datos.
  - c) Para indicar el final de un bloque de instrucciones.
  - d) Para establecer una etiqueta.
9. ¿Qué hace la instrucción «**bne** etiqueta» en ARM Thumb?
  - a) Salta a la etiqueta si el último resultado es no cero.
  - b) Realiza una operación bitwise NOT y luego salta a la etiqueta.
  - c) Salta siempre a la etiqueta.
  - d) Salta a la etiqueta si el último resultado es cero.
10. ¿Cuál es el par de instrucciones correcto para apilar y luego desapilar el registro r0 en ARM Thumb?
  - a) «**push** {r0}» seguido de «**pop** {r0}».
  - b) «**store** {r0}» seguido de «**load** {r0}».
  - c) «**push** {r0}» seguido de «**pull** {r0}».
  - d) «**set** {r0}» seguido de «**get** {r0}».
11. ¿Cuál es el procedimiento correcto para realizar una llamada a subrutina en ARM Thumb?
  - a) Usar la instrucción **call** seguida del nombre de la subrutina.



- b) Usar la instrucción `jmp` seguida del nombre de la subrutina.
  - c) Usar la instrucción `bl` seguida del nombre de la subrutina.
  - d) Usar la instrucción `branch` seguida del nombre de la subrutina.
12. Dado el programa del Anexo A, ¿qué valor habrá en el registro `r6` al acabar su ejecución?
- a) `0xFFFF FFFF`.
  - b) `0x0000 000B`.
  - c) `0x0000 000C`.
  - d) `0x8000 0000`.
13. Dado el programa del Anexo A, ¿qué valor habrá en el registro `r2` la última vez que se ejecute la instrucción etiquetada con «`ccl`»?
- a) Un 0.
  - b) El código ASCII correspondiente al carácter «!».
  - c) El código ASCII correspondiente al carácter «d».
  - d) El código ASCII correspondiente al carácter «o».
14. ¿Qué tipo de datos NO es soportado de forma nativa por la arquitectura ARM Thumb M0?
- a) Enteros.
  - b) Punto flotante.
  - c) Cadena de caracteres.
  - d) Booleanos.
15. ¿Cuántos registros de propósito general tiene la arquitectura ARM Thumb M0?
- a) 8.
  - b) 13.
  - c) 16.
  - d) 31.
16. ¿Cuál de las siguientes afirmaciones describe mejor la problemática general de la entrada/salida en sistemas computacionales?
- a) La entrada/salida no presenta desafíos, ya que los dispositivos siempre responden de manera predecible.
  - b) La entrada/salida es generalmente rápida y no afecta el rendimiento general del sistema.
  - c) La entrada/salida introduce desafíos debido a la variabilidad en el tiempo de respuesta de los dispositivos y la necesidad de sincronización.



- d) La entrada/salida solo es relevante en sistemas de baja complejidad.
17. ¿Cuál de las siguientes opciones representa una forma común en la que el procesador puede acceder a dispositivos de entrada/salida?
- a) A través de registros de control únicamente.
  - b) Mediante la ejecución de instrucciones de entrada/salida o de carga/almacenamiento por parte del procesador.
  - c) A través de acceso directo a memoria exclusivamente.
  - d) Utilizando únicamente interrupciones.
18. ¿Qué función cumplen típicamente los registros de estado en los dispositivos de entrada/salida?
- a) Almacenan datos de entrada provenientes de los dispositivos.
  - b) Indican el estado y condiciones del dispositivo, como si está listo para recibir datos o si ha ocurrido un error.
  - c) Contienen direcciones de memoria para acceder a datos de entrada/salida.
  - d) Almacenan instrucciones de control para el procesador.
19. ¿Cuál de las siguientes afirmaciones es verdadera acerca de la consulta de estado (*polling*) como sistema de sincronización de la entrada/salida?
- a) Realiza un uso eficiente de la CPU, pero puede generar una espera activa innecesaria.
  - b) Permite una respuesta inmediata a eventos, pero puede consumir una gran cantidad de recursos de la CPU.
  - c) Es un método sencillo de sincronización, pero puede causar una latencia variable en la atención a los dispositivos.
  - d) Evita la espera activa innecesaria, pero puede generar una respuesta lenta a eventos.
20. ¿Cuál de las siguientes opciones describe correctamente las características de la entrada/salida de propósito general (GPIO)?
- a) Permite la comunicación con dispositivos de almacenamiento secundario, como discos duros.
  - b) Proporciona interfaces estándar para la conexión de dispositivos periféricos, como teclados y ratones.
  - c) Permite la conexión de dispositivos externos a través de pines configurables como entradas o salidas digitales.
  - d) Ofrece una interfaz de comunicación inalámbrica para la conexión de dispositivos móviles.



## A. Código ARM Thumb M0

```
1      .data
2 str1: .asciz "Hola_mundo!"
3 str2: .asciz "Hello_world!"
4
5      .text
6 main: ldr r0, =str1
7       bl cc
8       mov r4, r0
9
10      ldr r0, =str2
11      bl cc
12      mov r5, r0
13
14      sub r6, r4, r5
15
16      wfi
17
18 cc:   push {lr}
19       mov r1, #0
20
21 ccl:  ldrb r2, [r0, r1]
22       cmp r2, #0
23       beq ccd
24       add r1, r1, #1
25       b ccl
26
27 ccd:  mov r0, r1
28       pop {pc}
29
30      .end
```