

## **Preguntas orientadoras**

1. Describa brevemente los diferentes perfiles de familias de microprocesadores / microcontroladores de ARM. Explique alguna de sus diferencias características.

La arquitectura ARM cuenta con tres familias: Cortex A, Cortex R y Cortex M. La familia Cortex A (Application) está orientada a la implementación de sistemas operativos en sistemas embebidos de alta performance. Un ejemplo de estos podrían ser teléfonos celulares.

La familia Cortex R (Real time) está orientada a la implementación de sistemas operativos real time, donde es preciso mantener baja latencia en los procesos. Ejemplos de uso podrían ser todo tipo de procesos críticos, como computadoras automotrices, sistemas médicos, etc.

La familia Cortex M (Microcontrollers) engloba a los microcontroladores que abarcan una gama amplia de rendimiento. Incluye desde los de bajo consumo y menor poder de procesamiento (M0), hasta los de mayores prestaciones (unidad de punto flotante o unidad de protección de memoria, por ejemplo). Su uso se puede ver en todo tipo de sistemas embebidos.

### **Cortex M**

1. Describa brevemente las diferencias entre las familias de procesadores Cortex M0, M3 y M4.

La principal diferencia es que el Cortex M0 tiene arquitectura Von Neumann (ARM V6) mientras que M3 y M4 tienen arquitectura Harvard (ARM V7).

Muchas de las características son opcionales por lo que dependen del fabricante, pero en general las diferencias más salientes son:

SysTickTimer - lo tienen todos los M3 y M4, pero es opcional para M0.

Memory protection unit - Es opcional para M3 y M4, pero no se incluye en M0 (si es opcional para M0+).

Las extensiones de división por hardware y matemática saturada no se encuentran disponibles para M0 y sí para M3 y M4. Y las de DSP y FPU se incluyen en M4 (FPU opcional).

2. ¿Por qué se dice que el set de instrucciones Thumb permite mayor densidad de código? Explique

El set de instrucciones Thumb permite realizar las mismas operaciones que la instrucción original (en las instrucciones soportadas) pero con formato de 16 bits. Al hacerlo por hardware, no afecta la performance, pero consume menos memoria. A esto se refieren como mayor densidad de código.

3. ¿Qué entiende por arquitectura load-store? ¿Qué tipo de instrucciones no posee este tipo de arquitectura?

Se le dice load-store a las arquitecturas que realizan las instrucciones lógicas y aritméticas solo sobre registros como operandos, tanto de entrada como de salida, obligando al programador a transferir los datos de memoria a los registros de entrada (load) y el registro de salida a memoria (store). No posee instrucciones que operen desde o hacia memoria directamente.

4. ¿Cómo es el mapa de memoria de la familia?

La familia Cortex tiene un mapa de memoria plano direccionable de 4 GB. Está dividido en 6 zonas. La zona de direcciones más baja es la zona de código, de 500 MB; seguida de la zona de RAM estática, también de 500 MB ; le sigue la zona de RAM de 1 GB; a continuación una zona de 1 GB para dispositivos y los 500 MB más altos son para sistema o específicos para el fabricante. En esta última zona es donde se encuentran las facilidades para debug, NVIC, etc.

5. ¿Qué ventajas presenta el uso de los “shadowed pointers” del PSP y el MSP?

En la implementación de OSs (o RTOSs) mejora la robustez, ya que si una aplicación tiene problemas de pila, el kernel y las otras aplicaciones no se ven afectadas.

Un OS puede usar la protección de memoria (MPU) para definir un área de memoria para la pila de una aplicación. En caso de salirse de este área (overflow) se producirá una excepción de manejo de memoria y evitará la sobrescritura de la memoria.

6. Describa los diferentes modos de privilegio y operación del Cortex M, sus relaciones y como se conmuta de uno al otro. Describa un ejemplo en el que se pasa del modo privilegiado a no privilegiado y nuevamente a privilegiado.

Cortex M tiene dos estados de operación:

- Estado Debug: cuando esta detenido (halt) y tiene un debugger conectado.
- Estado Thumb: cuando está ejecutando código Thumb (normal). En este, hay dos modos de operación:
  - Modo Handler: cuando se está ejecutando un handler de excepción, o una rutina de atención de interrupción. En este modo el procesador tiene nivel privilegiado de acceso.
  - Modo Thread: Cuando se ejecuta código normal de aplicaciones. Puede o no tener nivel de acceso privilegiado, según se configure en el registro de control.

Un ejemplo podría ser:

el scheduler de un sistema operativo no cooperativo (preemptive) se ejecuta en modo privilegiado y le cede el procesador a una tarea común (no

privilegiada). Al vencer su tiempo, se produce una interrupción que en su rutina de atención le devuelve el control al scheduler.

7. ¿Qué se entiende por modelo de registros ortogonal? Dé un ejemplo
8. ¿Qué ventajas presenta el uso de instrucciones de ejecución condicional (IT)? Dé un ejemplo
9. Describa brevemente las excepciones más prioritarias (reset, NMI, Hardfault).
10. Describa las funciones principales de la pila. ¿Cómo resuelve la arquitectura el llamado a funciones y su retorno?
11. Describa la secuencia de reset del microprocesador.
12. ¿Qué entiende por “core peripherals”? ¿Qué diferencia existe entre estos y el resto de los periféricos?
13. ¿Cómo se implementan las prioridades de las interrupciones? Dé un ejemplo
14. ¿Qué es el CMSIS? ¿Qué función cumple? ¿Quién lo provee? ¿Qué ventajas aporta?
15. Cuando ocurre una interrupción, asumiendo que está habilitada ¿Cómo opera el microprocesador para atender a la subrutina correspondiente? Explique con un ejemplo
17. ¿Cómo cambia la operación de stacking al utilizar la unidad de punto flotante?
16. Explique las características avanzadas de atención a interrupciones: tail chaining y late arrival.
17. ¿Qué es el systick? ¿Por qué puede afirmarse que su implementación favorece la portabilidad de los sistemas operativos embebidos?
18. ¿Qué funciones cumple la unidad de protección de memoria (MPU)?
19. ¿Cuántas regiones pueden configurarse como máximo? ¿Qué ocurre en caso de haber solapamientos de las regiones? ¿Qué ocurre con las zonas de memoria no cubiertas por las regiones definidas?
20. ¿Para qué se suele utilizar la excepción PendSV? ¿Cómo se relaciona su uso con el resto de las excepciones? Dé un ejemplo
21. ¿Para qué se suele utilizar la excepción SVC? Explíquelo dentro de un marco de un sistema operativo embebido.

## **ISA**

1. ¿Qué son los sufijos y para qué se los utiliza? Dé un ejemplo
2. ¿Para qué se utiliza el sufijo ‘s’? Dé un ejemplo
3. ¿Qué utilidad tiene la implementación de instrucciones de aritmética saturada? Dé un ejemplo con operaciones con datos de 8 bits.
4. Describa brevemente la interfaz entre assembler y C ¿Cómo se reciben los argumentos de las funciones? ¿Cómo se devuelve el resultado? ¿Qué registros deben guardarse en la pila antes de ser modificados?
5. ¿Qué es una instrucción SIMD? ¿En qué se aplican y que ventajas reporta su uso? Dé un ejemplo.