

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.
2. ¿Por qué se dice que el set de instrucciones Thumb permite mayor densidad de código? Explique
3. ¿Qué entiende por arquitectura load-store? ¿Qué tipo de instrucciones no posee este tipo de arquitectura?
4. ¿Cómo es el mapa de memoria de la familia?
5. ¿Qué ventajas presenta el uso de los "shadowed pointers" del PSP y el MSP?
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.
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.