

## Preguntas orientadoras

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

**Los diferentes perfiles de familias, se basan principalmente en sus distintas funcionalidades, capacidades y prestaciones.**

Las familias Cortex-A esta diseñadas principalmente para aplicaciones de alto rendimiento utilizadas en sistemas operativos para embebidos.

Los Cortex-R se utilizan en sistemas de tiempo real cuando se necesita sistemas baja latencia y alto procesamiento.

Los Cortex-M se utilizan para sistemas embebidos compactos que pueden correr grandes códigos y soporta programación en C.

Los Cortex-M0 son los que tienen menos funcionalidades, menos memorias, etc son mas baratos y se usan para aplicaciones donde requieran pocas funcionalidades, bajo consumo, recurso del procesador.

Los Cortex M3 al Cortex-M7 son los mas caros, son los que tienen mas funcionalidades, mas performance, mas memoria, se utilizan en microcontroladores, etc.

## 1.Cortex M

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

Las diferencias de los CORTEX-M:

## Cortex M : Comparación arquitecturas

ARM Cortex-M	SysTick Timer	Bit-banding	Memory Protection Unit (MPU)	Tightly-Coupled Memory (TCM)	CPU cache	Memory architecture	ARM architecture
Cortex-M0 <sup>[1]</sup>	Optional*	Optional <sup>[9]</sup>	No	No	No <sup>[10]</sup>	Von Neumann	ARMv6-M
Cortex-M0+ <sup>[2]</sup>	Optional*	Optional <sup>[9]</sup>	Optional (8)	No	No	Von Neumann	ARMv6-M

Cortex-M1 <sup>[3]</sup>	Optional	Optional	No	Optional	No	Von Neumann	ARMv6-M
--------------------------	----------	----------	----	----------	----	-------------	---------

Cortex-M3 <sup>[4]</sup>	Yes	Optional*	Optional (8)	No	No	Harvard	ARMv7-M
Cortex-M4 <sup>[5]</sup>	Yes	Optional*	Optional (8)	No	Possible <sup>[11]</sup>	Harvard	ARMv7E-M
Cortex-M7	Yes	No	Optional (8 or 16)	Optional	Optional	Harvard	ARMv7E-M

Observando la columna de SysTickTimer, vemos que en los CORTEX.-M3-M4 lo tienen incorporados:

Esta opcion lo que permite, es brindar interrupciones cada un tiempo seteado, lo que nos da la posibilidad de generar interrupción para algun codigo que lo requiera.

La columna de Memory Protection Unit: Sirve para proteger la unidad de memoria. Lo que obliga pedir permisos para poder tener acceso a en ella. Tambien dar prioridades.

La columna de Memory architecture vemos la Von Neumann y Harvard

Von Neumann: esta tiene un bus para comunicar con la memoria(datos) y usa el mismo bus para acceder a las instrucciones del código.

Harvard: esta tiene 2 bus, uno para la memoria(datos) y otro bus para las instrucciones.

Conclusion la arquitectura Harvard es mas rápida, consume más y es más cara que la Von Neumann.

Ultima columna es se puede observar cuales CORTEX usan ARMv6-M y cuales ARMv7-M

ARMv6-M usa las instrucciones THUMB( instrucciones de 16 bit) y algunas de THUMB-2(incorpora una instruccion condicional y son de 32bit.)

ARMv7-M usa las instrucciones THUMB-2. Los ARMv7E-M usan las instrucciones THUMB-2 y ademas esta preparado para procesamiento de señales digitales(DSP=permite ejecutar instrucciones en menos ciclos de reloj).

Los CORTEX M0 y M0+ tienen instrucciones con resolución de 32 bit. Pero M3 y M4 tiene mas instrucciones de multiplicación en hardware con resolución de 64 bit.

Los CORTEX M0 y M0+ no tienen matematica saturada y los CORTEX-M3 y M4 tiene. El resultado de la operación satura sobre una variable cuando sobrepasa el maximo valor.

Los CORTEX-M4 tienen unidad de punto flotante en su hardware, esto hace que las operaciones sean mucho mas rapidas.

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

El set de instrucciones de Thumb, son de 16 bit y son un subconjunto de las instrucciones de 32 bit de ARM. Con la introducción del set de instrucciones de Thumb 2, ahora es posible manejar el procesamiento en un solo estado de operación (no es necesario alternar entre los dos estados).

Instrucciones Thumb 2 (una de las características más importantes), utiliza en forma conjunta instrucciones de 32 y 16 bits logrando alta densidad de código, alta eficiencia y potente además de fácil de usar.

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

En la arquitectura load-store(cargar y almacenar), como lo dice su nombre deben cargarse(load) un dato de la memoria en un registro, luego utilizar las instrucciones que necesita el código para luego almacenar(store) en memoria. Esta arquitectura son ademas, las encargadas de ejecutar las instrucciones de acceso a la memoria RAM, tanto para lectura como escritura.

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

Esta particionada en regiones, su capacidad es de 4gb. Los sectores se ubican: sistema, external device, external RAM, periféricos, SDRAM, código, componentes internos del procesador, etc.

## **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?
- 1Aarquitectura de Microprocesadores  
Carrera de Especialización en Sistemas Embebidos  
Universidad de Buenos Aires
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.