

Preguntas orientadoras

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

Cortex A:

Son procesadores de alto rendimiento optimizados para aplicaciones que emplean un sistema operativo de propósito general en sistemas embebidos de alta performance.

Su denominación “A” proviene de *Application*, se pueden encontrar en dispositivos como celulares o tables.

Se destaca la optimización para ejecutar diversas aplicaciones al mismo tiempo a costa de una disminución del tiempo de respuesta de las mismas, aspecto secundario en dispositivo de usuario.

Cortex R:

Son procesadores orientados a sistemas de tiempo real donde prima la necesidad de implementar soluciones con requerimientos temporales estrictos.

En los sistemas de tiempo real la respuesta a eventos o estímulos debe ser en un tiempo acotado y preestablecido comportándose de forma determinística.

Su denominación “R” proviene de *Real Time*, se pueden encontrar en sistemas críticos, como dispositivos médicos o sistemas de operación automovilísticos.

Cortex M:

Son procesadores orientados a dispositivos de consumo masivo y sistemas embebidos compactos. Son procesadores de uso general diseñados para alta densidad de código y con gran cantidad de periféricos. Su denominación “M” proviene de *Microcontrollers*, se pueden encontrar en dispositivos como celulares o tables.

Cortex M

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

El set de instrucciones Thumb está compuesto por instrucciones de 16 bits con funciones específicas. Se dice que permite mayor densidad de código en comparación de un set de instrucciones de 32 bits exclusivamente. En muchas operaciones simples no resulta necesario emplear muchas instrucciones, un set de 16 bits es suficiente y permite optimizar la cantidad de código que se debe almacenar.

En la familia Cortex-M3/4 incorpora el set de instrucciones Thumb-2 que permite instrucciones de 16 y 32 bits mejorando la optimización desde el punto de vista de la densidad del código, eficiencia y performance.

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

Una arquitectura *load-store* significa que para realizar una modificación de un dato almacenado debe cargarse previamente en un registro, procesarse y luego volver a escribirse en memoria utilizando una serie de operaciones separadas.

Por ejemplo, para incrementar un valor almacenado en una memoria SRAM, el procesador necesita usar una instrucción para leer el dato desde la SRAM y colocarlo en un registro interno, una segunda instrucción para incrementar el valor del registro y finalmente, una tercera instrucción para escribir el valor modificado en la posición de memoria donde se encontraba.

Dentro del set de instrucciones del dispositivo no existen instrucciones que permitan directamente modificar un valor del mapa de memoria.

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

En la familia de procesadores Cortex-M el mapa de memoria está compuesto por un ancho de palabra de 32 bits y puede contener hasta 4 Gbytes de espacio, límite impuesto por la cantidad de direcciones posibles que se pueden acceder con un 32 bits de direccionamiento ($2^{32} = 4.29\text{Gbits}$).

El mapa de memoria se encuentra particionado en diferentes secciones donde se alojan ("*mapean*") todos los componentes del sistema, como memoria flash, memoria SRAM, periféricos, etc. Exceptuando restricciones específicas sobre algunas zonas de memoria es posible acceder a todos los registros mediante un mismo bus de datos y set de instrucciones, característica que permite acceder fácilmente al contenido o registro del dispositivo empleando punteros en lenguaje C.

La partición del mapa de memoria está determinada por el fabricante de cada microcontrolador que contiene un procesador ARM y debe consultarse en la hoja de datos del dispositivo.

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

La familia de procesadores Cortex-M3/4 poseen dos modos de operación, modo privilegiado y no privilegiado, también llamado modo usuario.

El modo de funcionamiento privilegiado se caracteriza porque permite el acceso a todo el mapa de memoria por parte de la/s aplicación/es del usuario. En contraposición, en el modo de operación de usuario existen áreas determinadas de memoria protegidas que no son posibles acceder durante la ejecución del programa.

Una aplicación típica del uso de los modos de usuario, se da al emplear un sistema operativo. Es deseable que las tareas o aplicación del usuario no accedan a los componentes del sistema operativo ubicandolo en un área protegida de memoria solo accesible en el modo de privilegiado.