

Sistemas Empotrados

Tema 1:

Introducción a los sistemas empotrados

Lección 3:

La parte hardware



Jesús González Peñalver

Contenidos

Tema 1: Introducción a los sistemas empotrados

Presentación de la asignatura

Motivación

Descripción de la asignatura

Sistemas empotrados

Utilidad

Caracterización

Clasificaciones

Diseño e implementación

Herramientas de desarrollo

La parte hardware

Procesadores

Co-procesadores y aceleradores

Controladores de sistema

Arquitectura de memoria

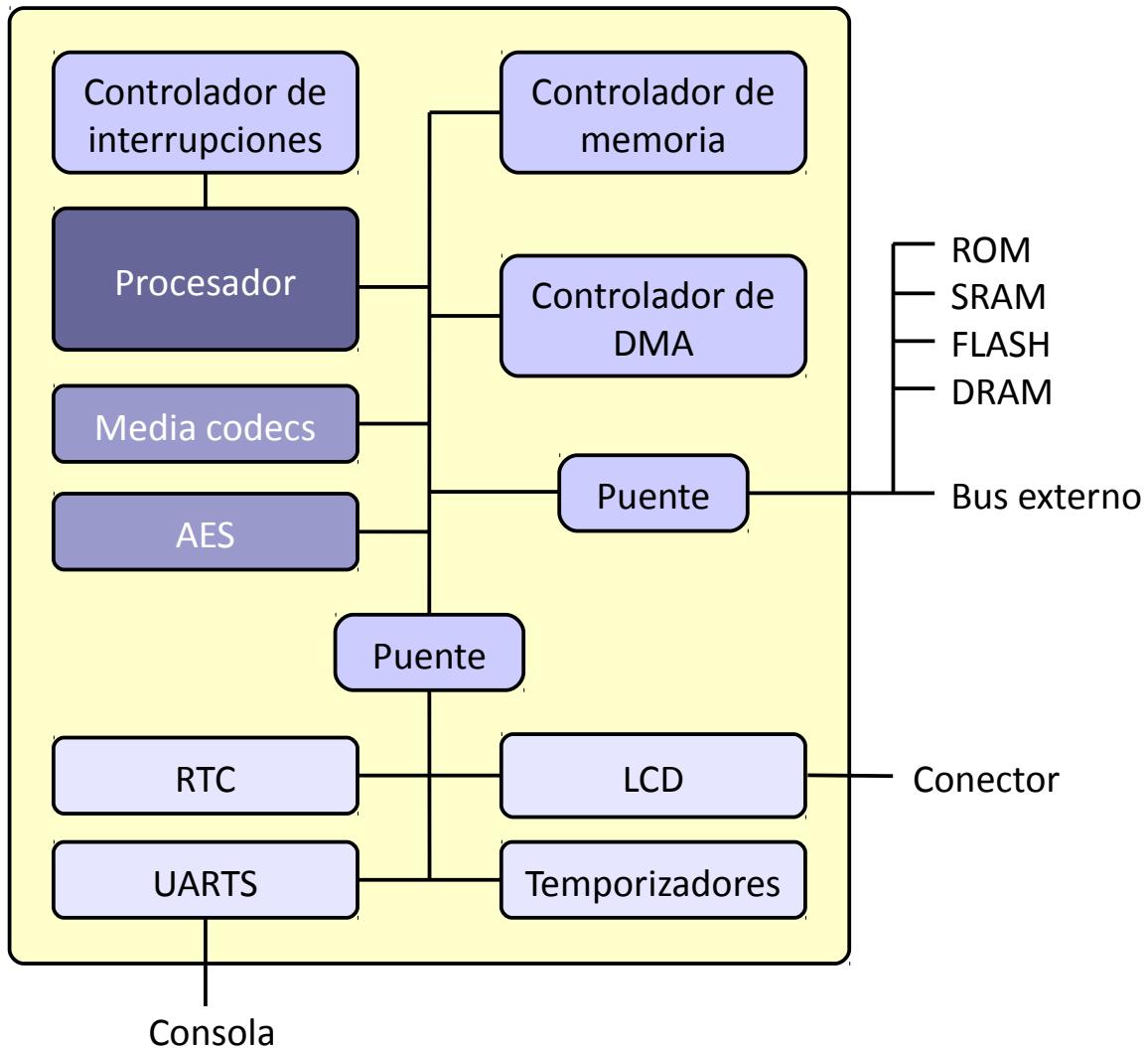
Periféricos

La parte software

Importancia creciente del software empotrado

Componentes del Firmware

Componentes hardware de un sistema empotrado



Procesador

Controla el sistema empotrado y ejecuta las funciones del sistema implementadas en software

Co-procesadores / Aceleradores

Ejecutan funciones complejas mediante hardware específico

Controladores de sistema

Coordinan los componentes importantes del sistema. Los más importantes son el de interrupciones y el de memoria

Periféricos

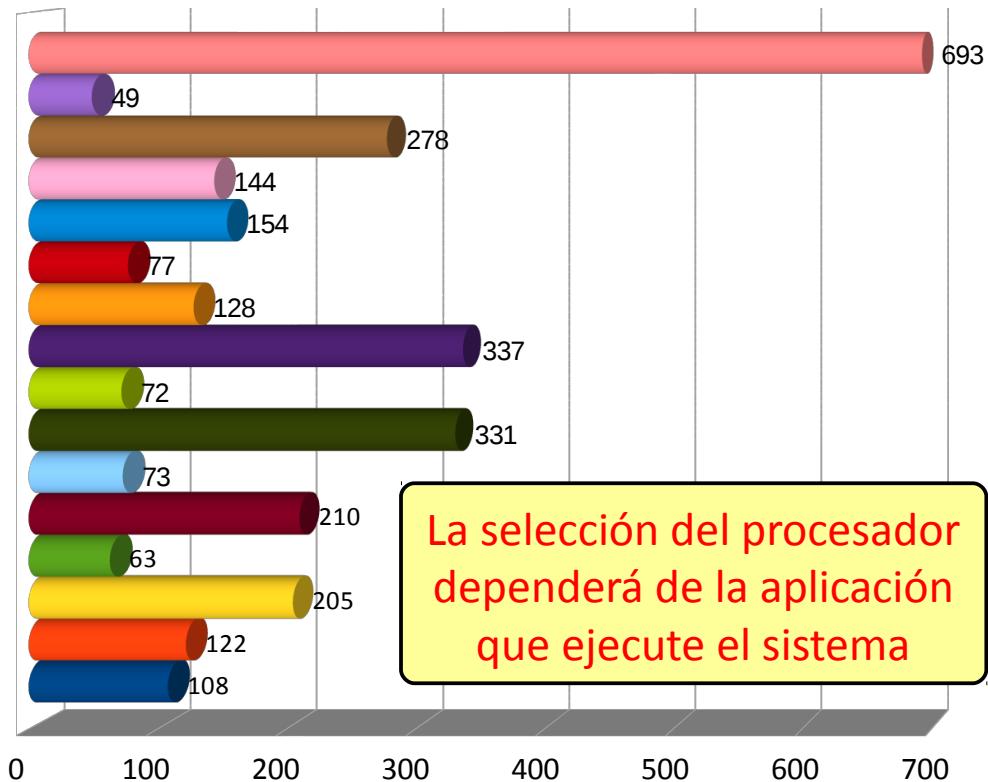
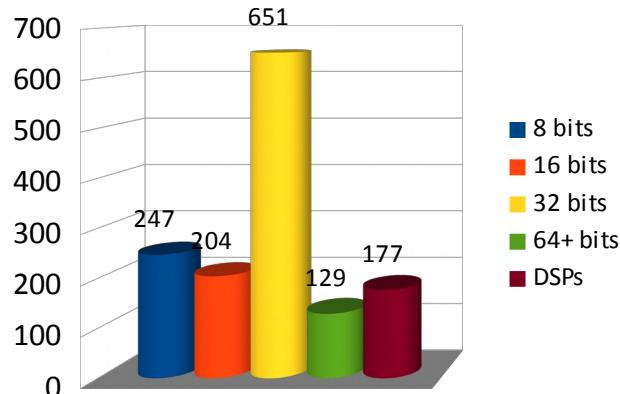
Proporcionan las capacidades de E/S del sistema. Dependen de la aplicación

Buses

Comunican las diferentes partes del sistema

Selección del tipo de procesador

Multitud de arquitecturas y modelos disponibles



La selección del procesador dependerá de la aplicación que ejecute el sistema

Más de 80 empresas diseñan procesadores

Más de 1400 modelos de procesador diferente

Elegir el procesador más adecuado (**relación coste-consumo-prestaciones óptima**) para el sistema no es nada fácil

Audio	Automotive	Communication & Wired	Computers & Peripherals
Consumer	Digital Power	General Purpose	Imaging & Video
Industrial	Medical	Military & Aerospace	Mobile & Wireless
Motor Control	Security	Test & Measurement	Other

Aplicaciones

Control

Complejidad baja, poca potencia de cálculo, poca memoria, imprescindible E/S, periféricos dependientes de la aplicación



Control de iluminación



Electrodomésticos



Mandos universales



Termómetro



Radio despertador



Detector de movimiento



Medidor eléctrico



Juguetes



Pulsioxímetro

Procesador típico

Tamaño de palabra: **8 – 16 bits**

Habitualmente entre **20 – 60 MHz** (~2 – 30 MIPS), aunque hay cores desde 32 kHz hasta 200 MHz

Normalmente no segmentados, sin caches, sin DMA, sin MMU/MPU, memoria direccionable a nivel de bit

Aplicaciones

Propósito general

Complejidad media, necesidad de más memoria y de periféricos más complejos
(conectividad, USB, pantalla táctil, memorias Flash, etc.)



Cámaras



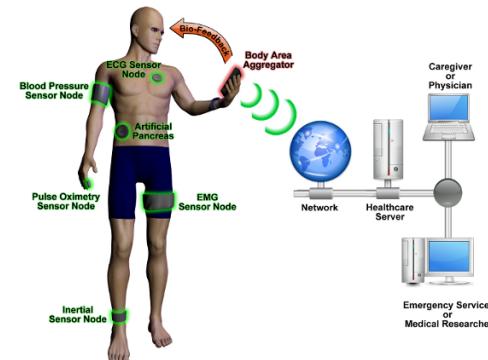
Ofimática doméstica



Scanners



Navegador GPS



e-Salud



Smart meter

Procesador típico

Tamaño de palabra: **32 bits**

Habitualmente entre **60 – 300 MHz (60 – 300 MIPS)**

Normalmente **segmentados, con caches y DMA, sin MMU/MPU**

Aplicaciones

Sistemas críticos y de tiempo real

Altas prestaciones y restricciones de tiempo real



Banda ancha móvil



Traditional Hard disk
mechanical drive



Memory based hard
drive (SSD)



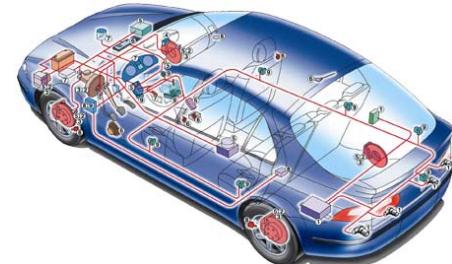
Ofimática profesional



Entretenimiento



Dispositivos médicos



Automoción

Procesador típico

Tamaño de palabra: **32 bits**

Habitualmente entre **200MHz – 1 GHz** (400 – 4600 MIPS)

Segmentados y últimamente superescalares y multicore superescalares. **MPU y ejecución determinista**

Aplicaciones

Aplicaciones de usuario

Altas prestaciones para ejecutar múltiples aplicaciones sobre un sistema operativo de propósito general y conexión a Internet, habitualmente en dispositivos móviles



Media centers



Smart TVs



Tablets



Smart phones



Consolas

Procesador típico

Tamaño de palabra: **32 bits**

Habitualmente entre **800MHz – 2 GHz** (900 – 5000 MIPS)

Superescalares y multicore superescalares. **MMU y conectividad**

Aplicaciones

Procesamiento de señales

Cálculo intensivo, tiempo real, patrones de acceso a datos predecibles, gran ancho de banda de E/S, uso extensivo de productos escalares, fiabilidad de los resultados, programas basados en bucles cortos.



Reconocimiento dactilar



Multi-metro digital



Cámaras de vídeo



Vídeo-conferencia

Convencionales (ultra bajo consumo)

Procesador típico: DSP (16 - 32 bits)

Convencionales

50 – 150 MHz (50 – 150 MMACs). Altas prestaciones minimizando el consumo, el HW del DSP y el tamaño del código. **Repertorios de instrucciones muy irregulares y complejos.** Necesidad de programar en ensamblador

VLIW

150 MHz– 1.2 GHz (300 – 40000 MMACs). Repertorio de instrucciones ortogonal e incremento del consumo y del tamaño del código

VLIW (altas prestaciones)

Estimación de las prestaciones

Dhrystone

- Programa escrito en C en 1984. Sus instrucciones se escogieron de forma arbitraria para cumplir los porcentajes de operaciones de diferentes tipos que solía haber en las aplicaciones de la época
- Genera unas 2.000 líneas de ensamblador y no realiza ninguna llamada al sistema operativo

DMIPS

- Estimación de las MIPS usando Dhrystone
- Se usa ampliamente por los fabricantes de procesadores empotrados

Inconvenientes

- Las MIPS dependen de la microarquitectura del procesador (CISC, RISC, VLIW)
- *Dhrystone* tiene secuencias de instrucciones conocidas cuyo resultado puede implementarse con menos instrucciones
- Los fabricantes de procesadores compilan *Dhrystone* usando *flags* de optimización exclusivos para este *benchmark* que detectan estas secuencias y realizan optimizaciones agresivas
- Estas optimizaciones no son aplicables al resto de programas, por lo que las medidas de prestaciones obtenidas no son muy fiables

Estimación de las prestaciones teniendo en cuenta la aplicación

Motivación

La mayoría de los sistemas empotrados sólo ejecutan una aplicación

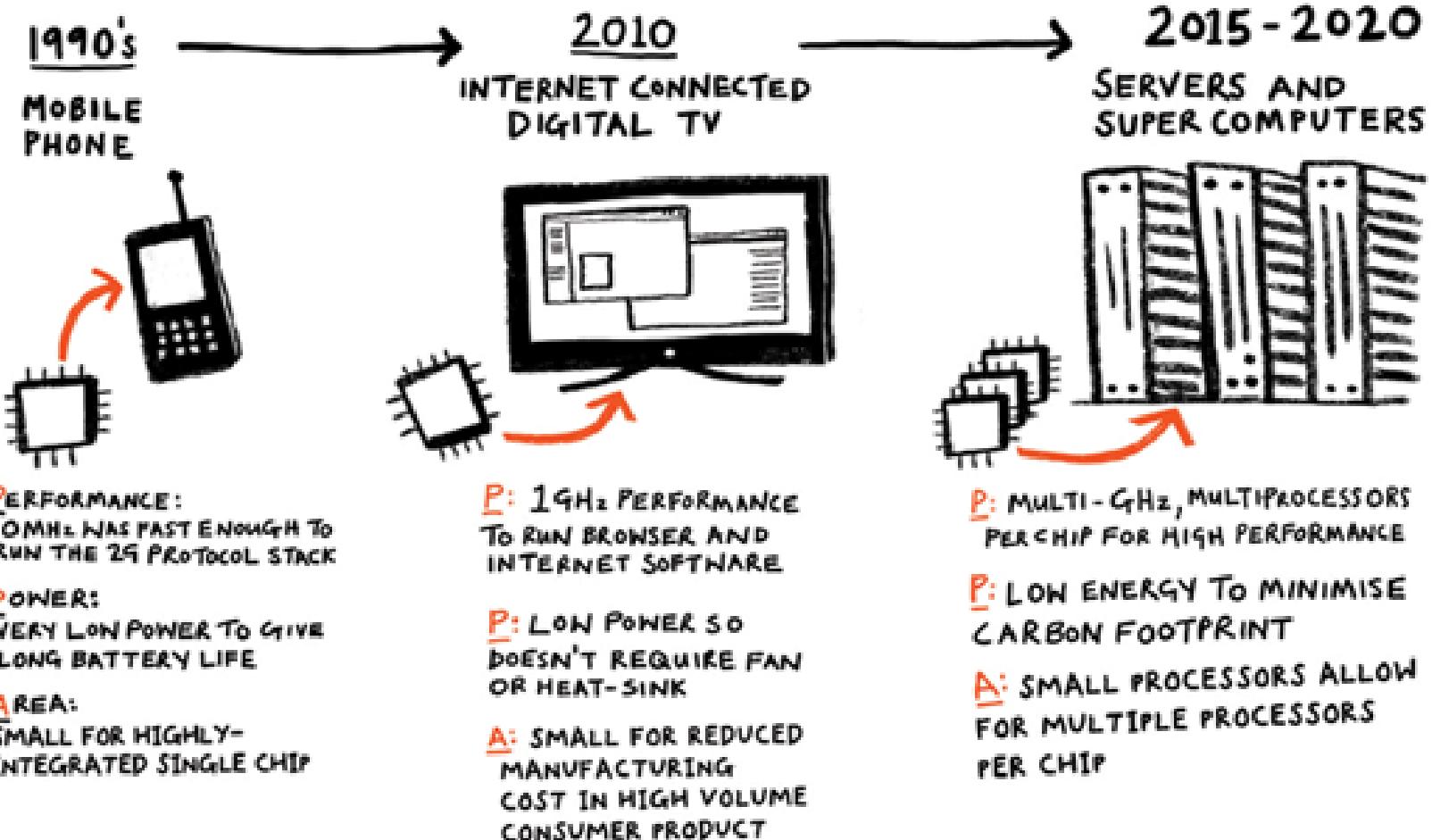
Los programas de test deberían asemejarse a la aplicación final del sistema para que los resultados sean más fiables

EEMBC BENCHMARKS	
SYSTEM	PROCESSOR
Android, Smartphone Browsers and Java AndEBench™ - Provides a standardized, industry-accepted method of evaluating Android platform performance. BrowsingBench™ (Version 2.0 in development) - Evaluate Web browser performance on Smartphones and other systems. GrinderBench™ - Test a system's performance running Java applications.	Microprocessor Benchmark Suites AutoBench™ (Version 2.0 in development) - Automotive, industrial, and general-purpose applications. ConsumerBench™ - Digital cameras, printers, and other embedded systems doing digital imaging tasks. CoreMark™ - Single number score for quick comparison of processor and microcontroller core functionality. DENBench™ - Digital entertainment products such as smartphones, MP3 players, digital cameras, TV set-top boxes, and in-car entertainment systems. EnergyBench™ - Power and energy performance with insights to power budget costs. FPPBench™ (in development) - Floating-point performance in graphics, audio, motor control, and other high-end processing tasks. MultiBench™ - Multicore architectures, memory bottlenecks, OS thread scheduling, and efficiency of synchronization. Networking - Moving and analyzing packets in networking applications. OABench™ - Office Automation tasks in printers, plotters, and other systems that handle text and image processing tasks. TeleBench™ - Telecommunications processors in modem, xDSL, and related fixed-telecom applications.
Telecom/Networking System Benchmarks ETCPBench™ - TCP/IP performance and conformance for platforms from micro-controllers to high-end processors. DPIBench™ (in development) - Evaluate throughput and latency to highlight the strengths and weaknesses of DPI systems, processors, and middleware.	
Benchmark Testing Services EEMBC Technology Center (ETC) - Services include porting and benchmark execution, performance analysis, and preparation of platforms for benchmark score certification. These services allow any company to make EEMBC benchmark testing an integral part of its product development and release process - without tying up internal engineering resources.	

Equilibrio coste-consumo-prestaciones

Objetivo

Maximizar las prestaciones con un consumo y un área (coste) mínimos



Fuente: ARM Annual Report and Accounts 2010

<http://www.arm.com/annualreport10/20-20-vision/best-technology.html>

Comparativa entre diferentes procesadores

■ Intel Core i7 950 (4 cores superescalares, 8 hebras)



Servidor

■ Intel Atom N280 (1 core superescalar, 2 hebras)



Netbook

■ ARM Cortex A9 (2 cores superescalares, 2 hebras)



Smartphone

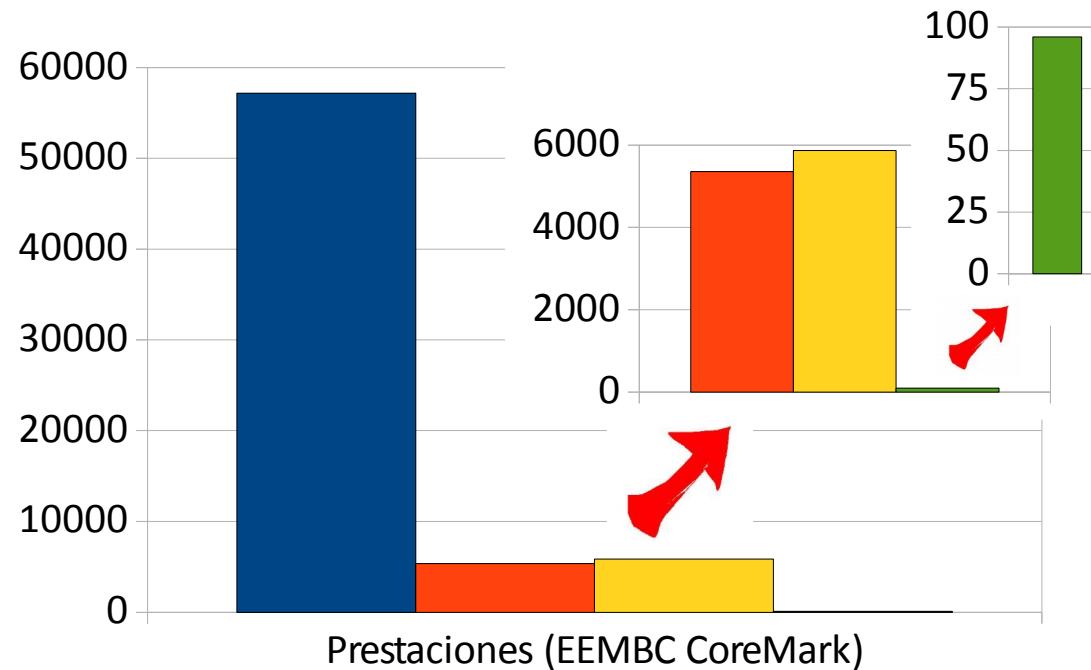
■ ARM Cortex M3 (1 core segmentado, 1 hebra)



Internet of things

Comparativa entre diferentes procesadores

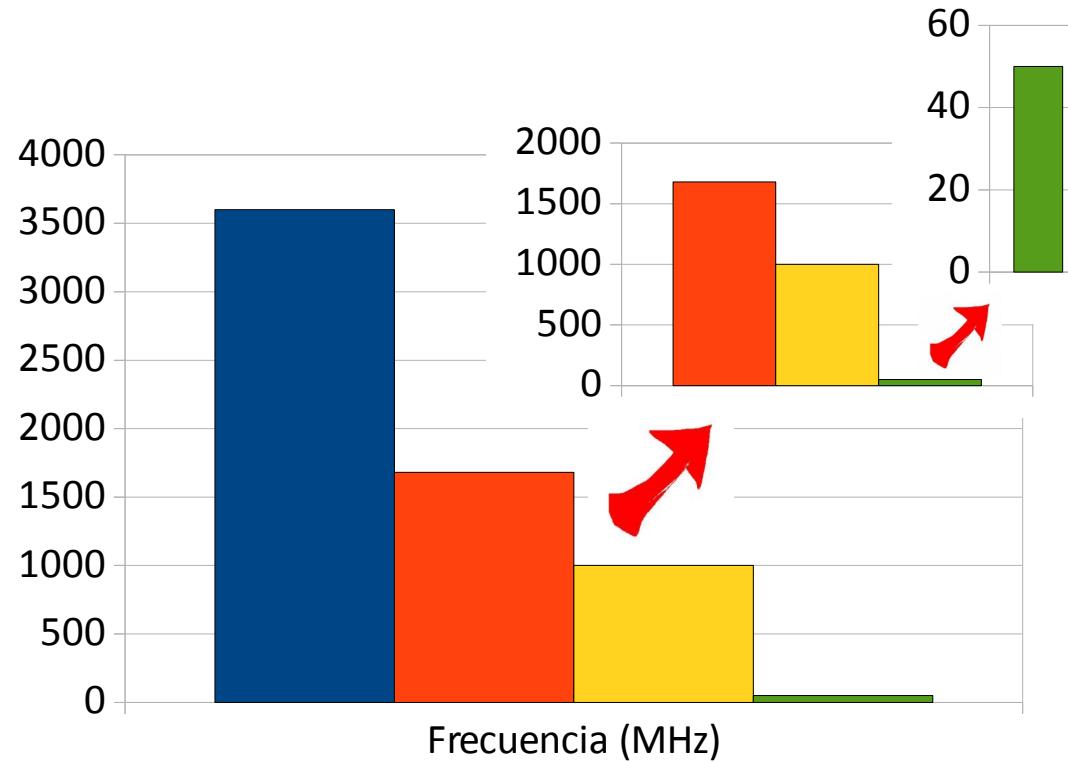
- Intel Core i7 950 (4 cores superescalares, 8 hebras)
- Intel Atom N280 (1 core superescalar, 2 hebras)
- ARM Cortex A9 (2 cores superescalares, 2 hebras)
- ARM Cortex M3 (1 core segmentado, 1 hebra)



En prestaciones absolutas gana claramente Intel, aunque si nos centramos en procesadores para dispositivos móviles, el ARM Cortex A9 supera al Intel Atom

Comparativa entre diferentes procesadores

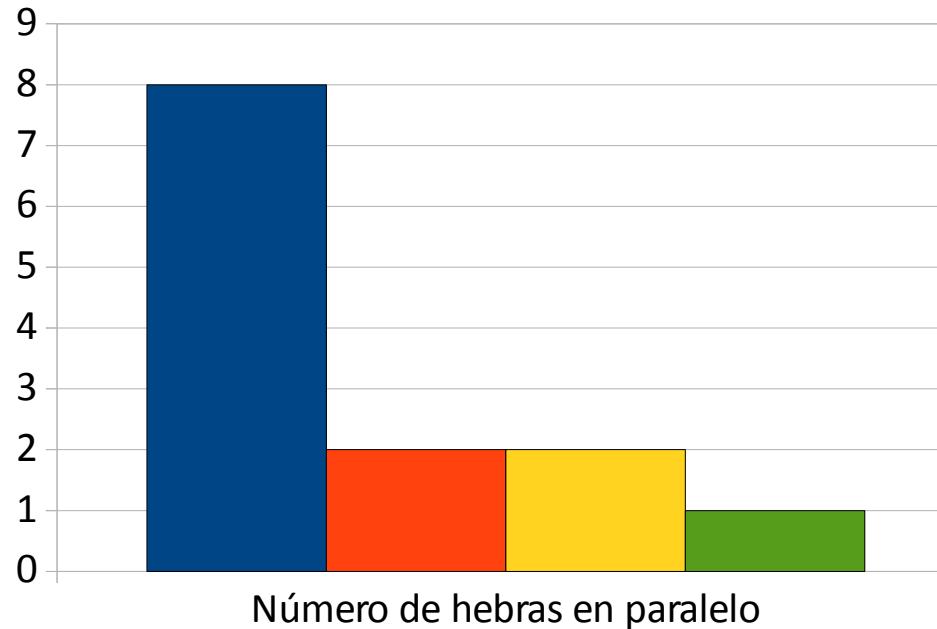
- Intel Core i7 950 (4 cores superescalares, 8 hebras)
- Intel Atom N280 (1 core superescalar, 2 hebras)
- ARM Cortex A9 (2 cores superescalares, 2 hebras)
- ARM Cortex M3 (1 core segmentado, 1 hebra)



Aunque no todos los procesadores funcionan a la misma frecuencia

Comparativa entre diferentes procesadores

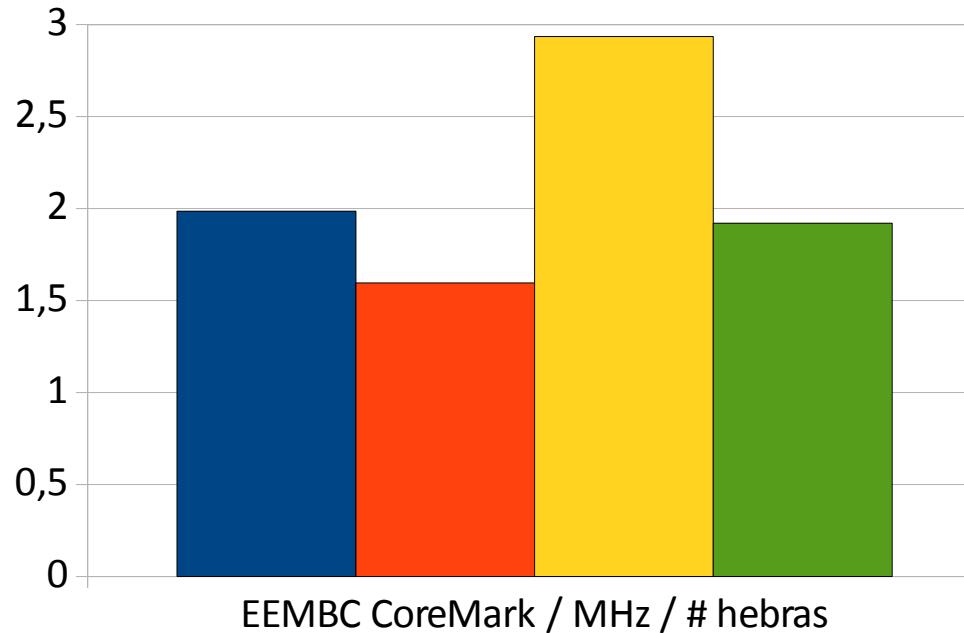
- Intel Core i7 950 (4 cores superescalares, 8 hebras)
- Intel Atom N280 (1 core superescalar, 2 hebras)
- ARM Cortex A9 (2 cores superescalares, 2 hebras)
- ARM Cortex M3 (1 core segmentado, 1 hebra)



Ni ejecutan el mismo número de hebras

Comparativa entre diferentes procesadores

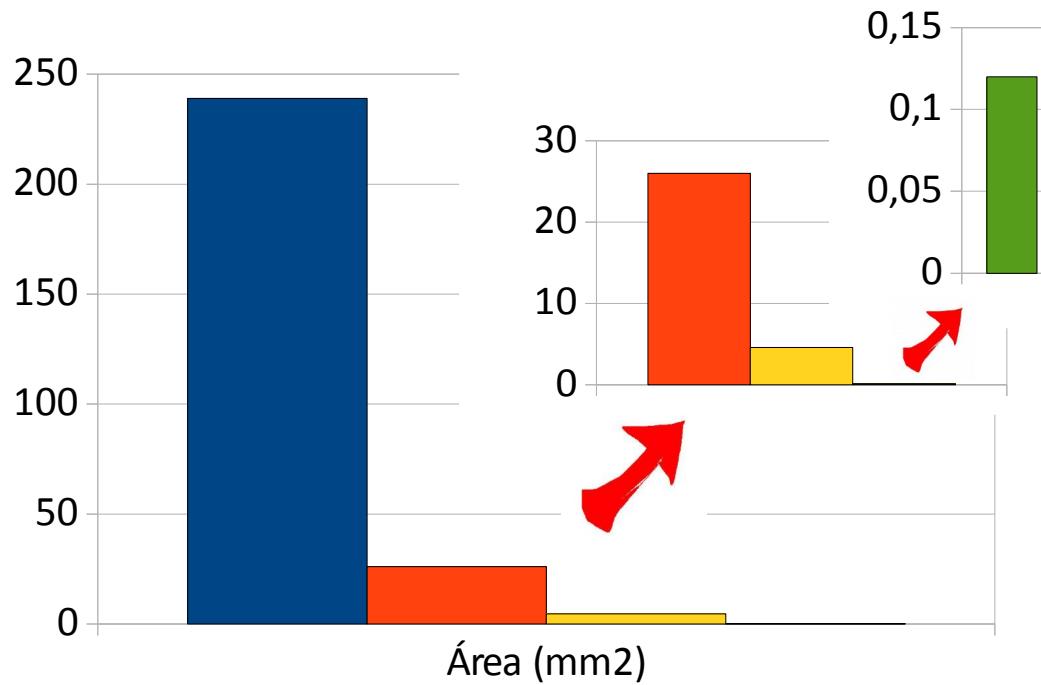
- Intel Core i7 950 (4 cores superescalares, 8 hebras)
- Intel Atom N280 (1 core superescalar, 2 hebras)
- ARM Cortex A9 (2 cores superescalares, 2 hebras)
- ARM Cortex M3 (1 core segmentado, 1 hebra)



Si normalizamos las prestaciones entre la frecuencia y el número de hebras, vemos que los diseños de ARM son más eficaces

Comparativa entre diferentes procesadores

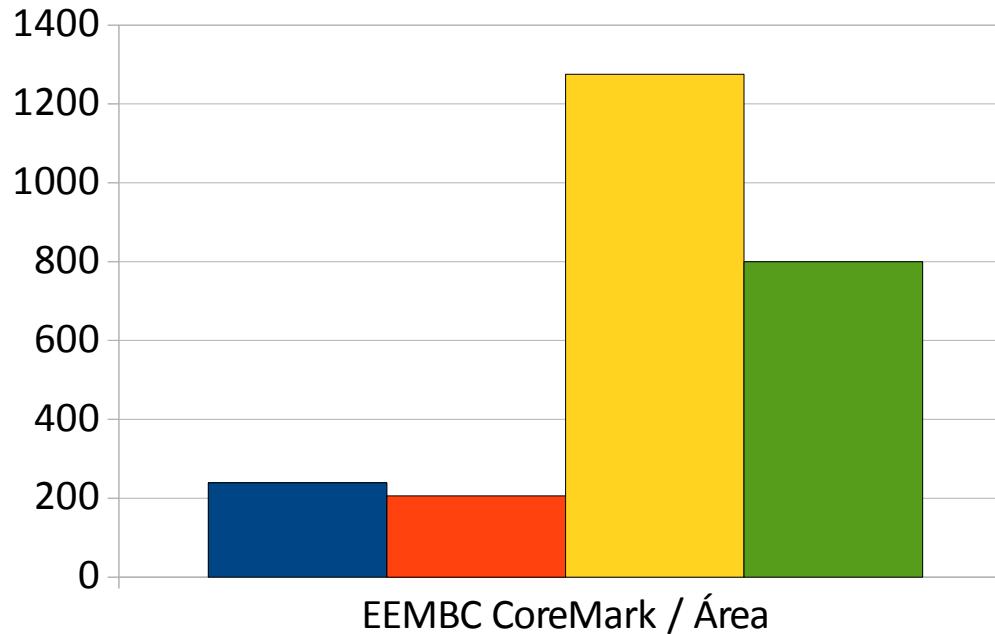
- Intel Core i7 950 (4 cores superescalares, 8 hebras)
- Intel Atom N280 (1 core superescalar, 2 hebras)
- ARM Cortex A9 (2 cores superescalares, 2 hebras)
- ARM Cortex M3 (1 core segmentado, 1 hebra)



Los procesadores de ARM son mucho más baratos, ya que tienen un número menor de transistores

Comparativa entre diferentes procesadores

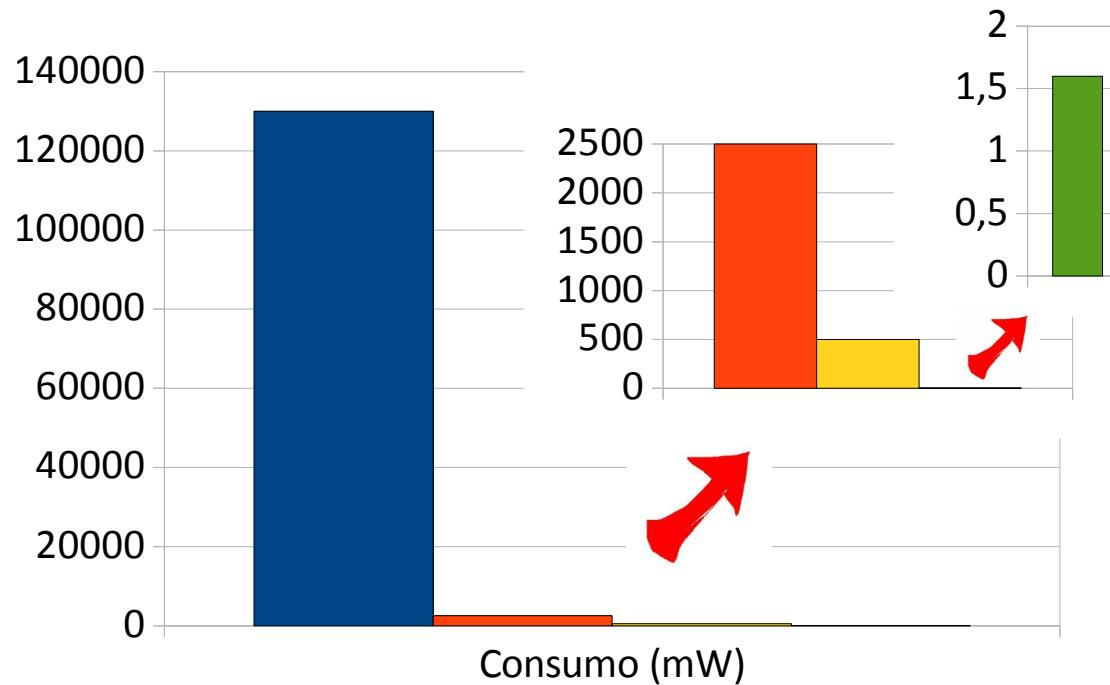
- Intel Core i7 950 (4 cores superescalares, 8 hebras)
- Intel Atom N280 (1 core superescalar, 2 hebras)
- ARM Cortex A9 (2 cores superescalares, 2 hebras)
- ARM Cortex M3 (1 core segmentado, 1 hebra)



Si normalizamos las prestaciones entre el área, los diseños de ARM siguen siendo más eficaces

Comparativa entre diferentes procesadores

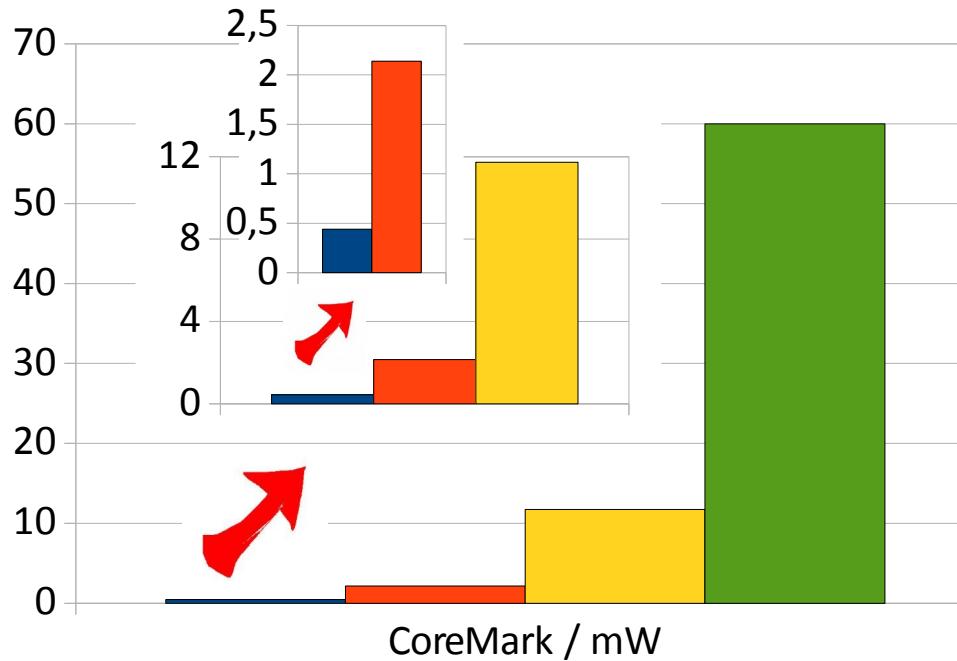
- Intel Core i7 950 (4 cores superescalares, 8 hebras)
- Intel Atom N280 (1 core superescalar, 2 hebras)
- ARM Cortex A9 (2 cores superescalares, 2 hebras)
- ARM Cortex M3 (1 core segmentado, 1 hebra)



Los procesadores de ARM consumen mucha menos energía

Comparativa entre diferentes procesadores

- Intel Core i7 950 (4 cores superescalares, 8 hebras)
- Intel Atom N280 (1 core superescalar, 2 hebras)
- ARM Cortex A9 (2 cores superescalares, 2 hebras)
- ARM Cortex M3 (1 core segmentado, 1 hebra)

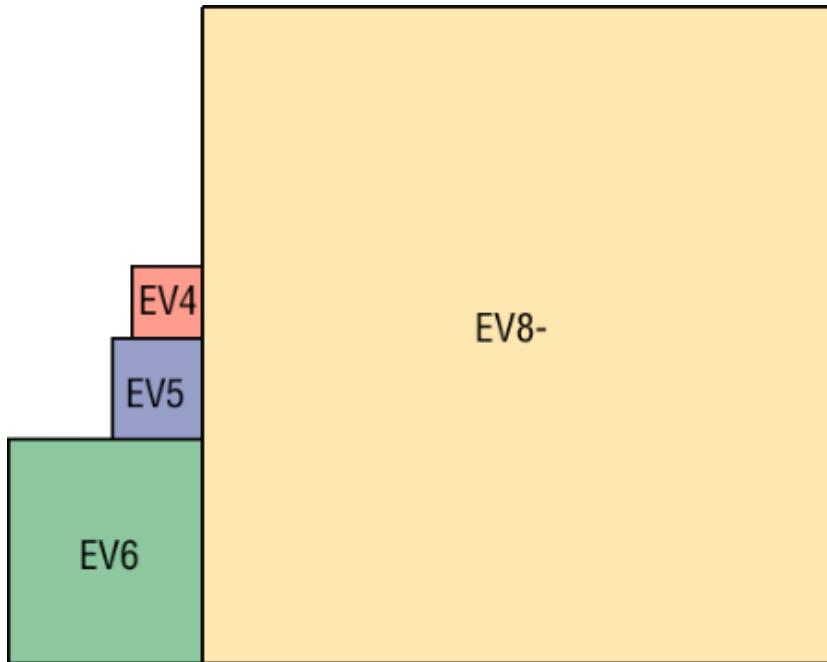


Si normalizamos las prestaciones entre el consumo, los diseños de ARM siguen siendo más eficaces

Necesidad de arquitecturas multiprocesador

Tamaño relativo de los procesadores de Alpha.

El EV8 es 80 veces más grande que el EV4



Potencia disipada y prestaciones de cuatro generaciones de microprocesadores Alpha

Core	Peak power (Watts)	Average power (Watts)	Performance (norm. IPC)
EV4	4.97	3.73	1.00
EV5	9.83	6.88	1.30
EV6	17.8	10.68	1.87
EV8	92.88	46.44	2.14

Comparación entre el EV8 y el EV4

Tamaño: 80 veces mayor

Consumo medio: 12.45 veces mayor

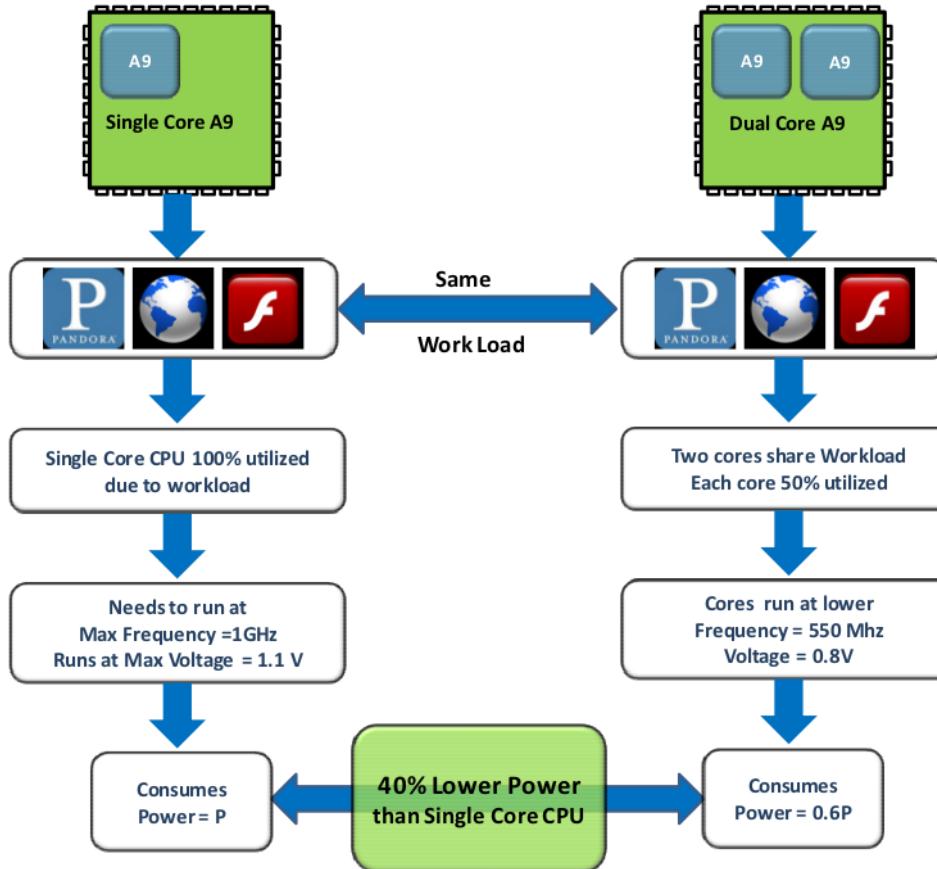
Prestaciones: 2.14 veces mayor

Coste y consumo no escalan linealmente con el incremento en prestaciones de un procesador

Es mejor dividir la aplicación en varios procesos que se ejecuten en procesadores más sencillos

Se mejora el coste y el consumo energético del sistema

Necesidad de arquitecturas multiprocesador



Se reduce notablemente el consumo energético
para alcanzar las mismas prestaciones

Ejemplo de arquitectura multiprocesador

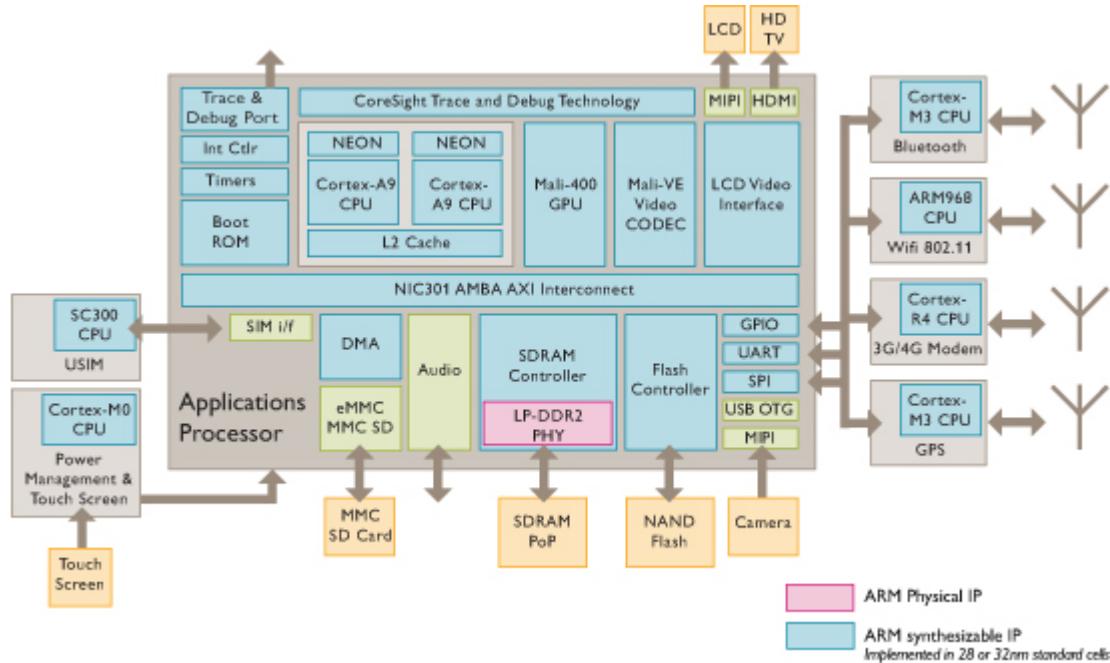


Diagrama de bloques de un smartphone

2 cores para el SO y aplicaciones

1 procesador para gestión de energía

1 procesador para la gestión de la SIM

1 procesador para gestionar las comunicaciones bluetooth

1 procesador para gestionar las comunicaciones Wi-Fi

1 procesador para gestionar las comunicaciones 3G/4G

1 procesador para gestionar el GPS

8 procesadores en total

Contenidos

Tema 1: Introducción a los sistemas empotrados

Presentación de la asignatura

Motivación

Descripción de la asignatura

Sistemas empotrados

Utilidad

Caracterización

Clasificaciones

Diseño e implementación

Herramientas de desarrollo

La parte hardware

Procesadores

Co-procesadores y aceleradores

Controladores de sistema

Arquitectura de memoria

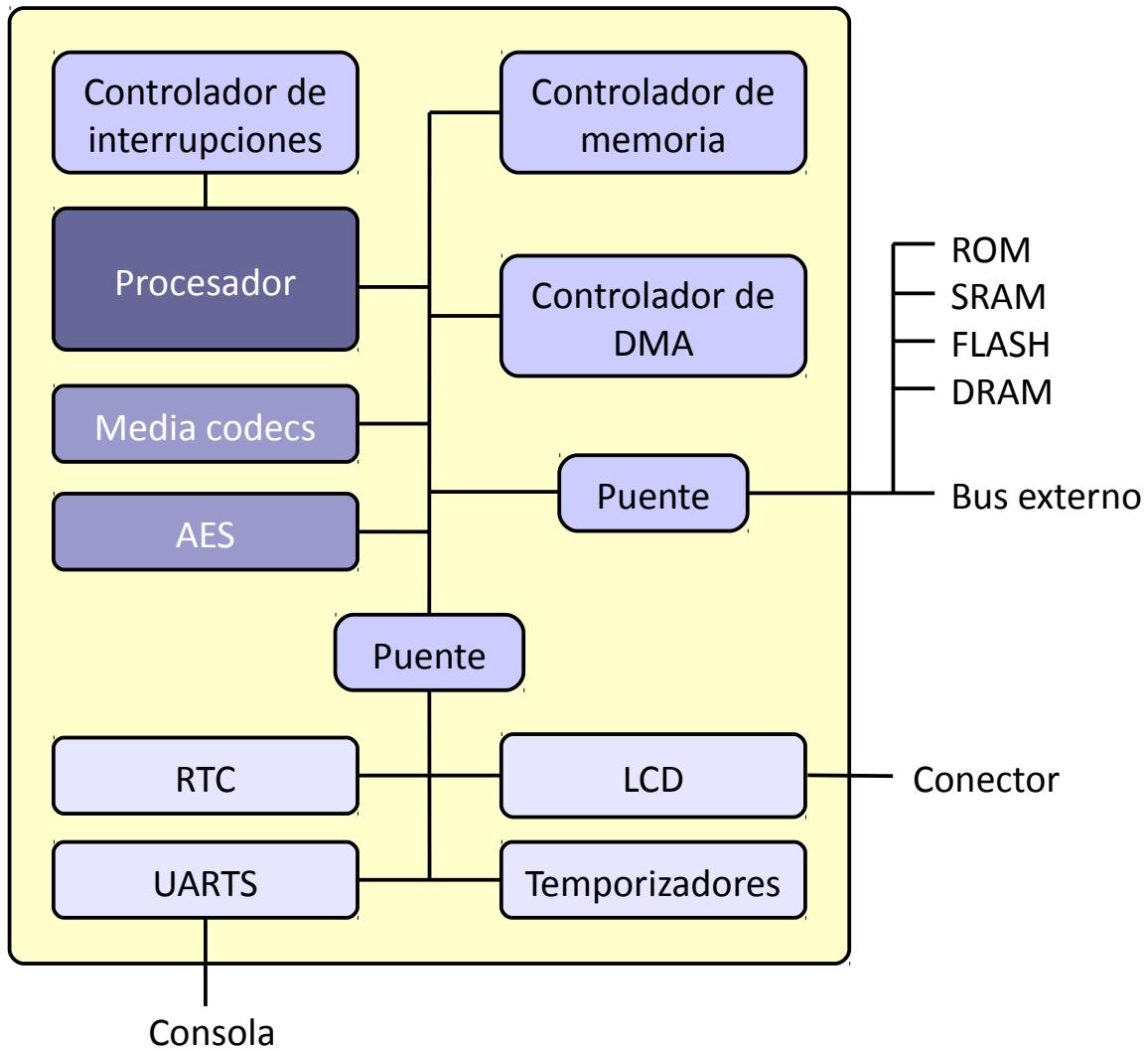
Periféricos

La parte software

Importancia creciente del software empotrado

Componentes del Firmware

Componentes hardware de un sistema empotrado



Procesador

Controla el sistema empotrado y ejecuta las funciones del sistema implementadas en software

Co-procesadores / Aceleradores

Ejecutan funciones complejas mediante hardware específico

Controladores de sistema

Coordinan los componentes importantes del sistema. Los más importantes son el de interrupciones y el de memoria

Periféricos

Proporcionan las capacidades de E/S del sistema. Dependen de la aplicación

Buses

Comunican las diferentes partes del sistema

Co-procesadores y aceleradores

Aceleran determinadas funciones del sistema

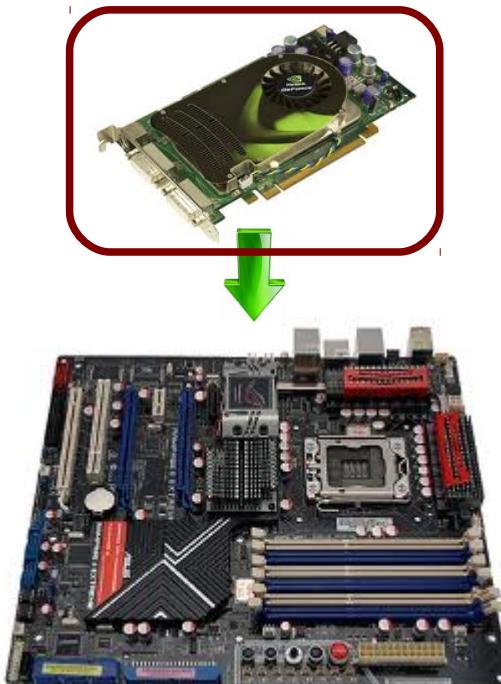
Tareas computacionalmente intensas dependientes de cada aplicación

Mejoran las prestaciones y el consumo energético del sistema

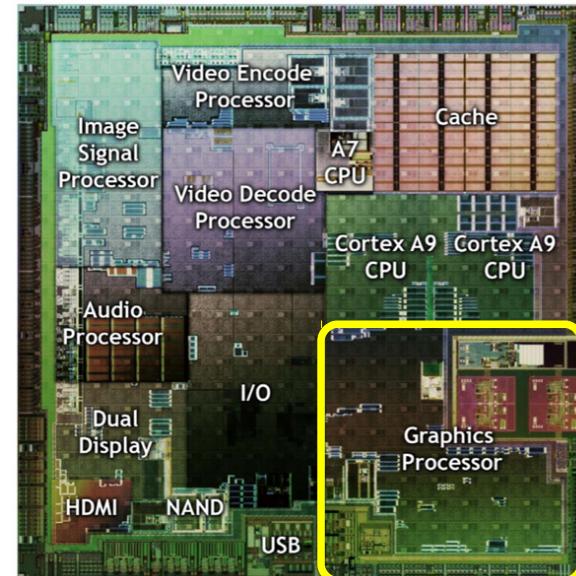
El datapath se diseña a medida. Pueden ser programables o no programables

Fuertemente acoplados al procesador

La comunicación con el procesador no debe causar cuellos de botella



GPU en un PC



GPU en un sistema empotrado

Co-procesadores

Se gestionan mediante instrucciones de la CPU

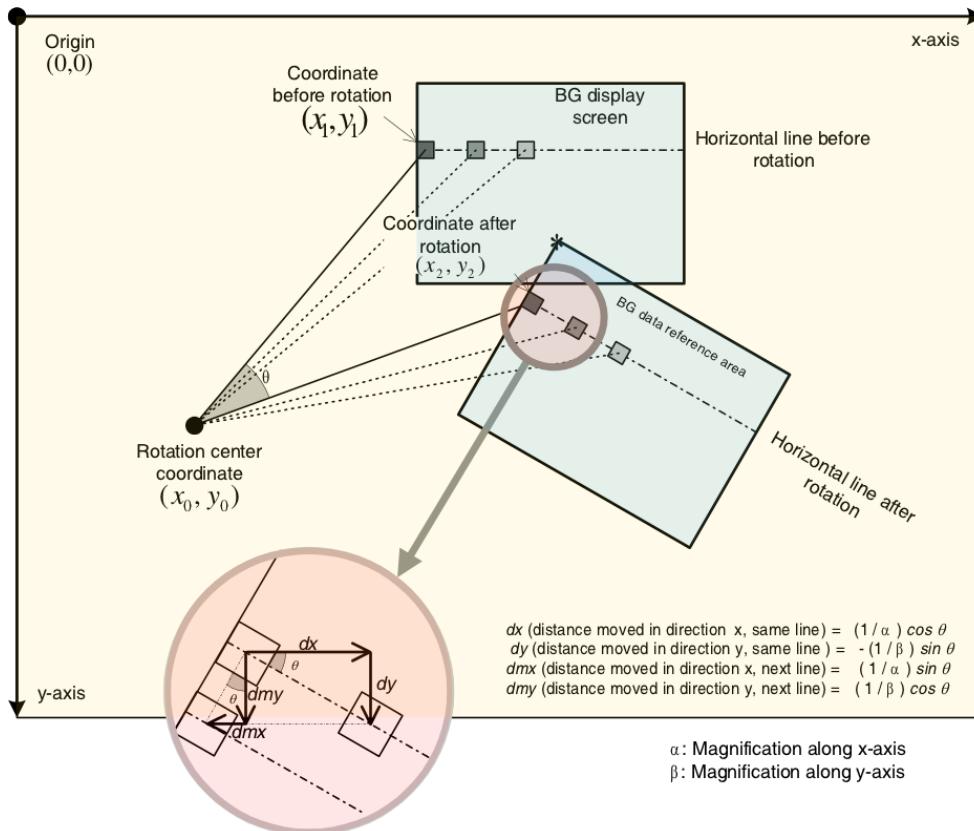
```
@ Macro to obtain a column of the result
.macro mul_col_f32 res_q, col0_d, col1_d
    vmul.f32    \res_q, q8, \col0_d[0]          @ multiply col element 0 by matrix col 0
    vmla.f32    \res_q, q9, \col0_d[1]          @ multiply-acc col element 1 by matrix col 1
    vmla.f32    \res_q, q10, \col1_d[0]         @ multiply-acc col element 2 by matrix col 2
    vmla.f32    \res_q, q11, \col1_d[1]         @ multiply-acc col element 3 by matrix col 3
.endm

@ Four-by-four floating point matrix multiply
vld1.32    {d16-d19}, [r1]!                  @ load first eight elements of matrix 0
vld1.32    {d20-d23}, [r1]!                  @ load second eight elements of matrix 0
vld1.32    {d0-d3}, [r2]!                   @ load first eight elements of matrix 1
vld1.32    {d4-d7}, [r2]!                   @ load second eight elements of matrix 1
mul_col_f32 q12, d0, d1                     @ matrix 0 * matrix 1 col 0
mul_col_f32 q13, d2, d3                     @ matrix 0 * matrix 1 col 1
mul_col_f32 q14, d4, d5                     @ matrix 0 * matrix 1 col 2
mul_col_f32 q15, d6, d7                     @ matrix 0 * matrix 1 col 3
vst1.32    {d24-d27}, [r0]!                  @ store first eight elements of result
vst1.32    {d28-d31}, [r0]!                  @ store second eight elements of result
```

Uso del co-procesador NEON de ARM para mejorar las prestaciones en la multiplicación de matrices FP 4 x 4

Aceleradores

Se gestionan mediante registros de control/estado y memoria compartida



$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

$$A = \frac{1}{\alpha} \cos \theta, \quad B = \frac{1}{\alpha} \sin \theta, \quad C = -\frac{1}{\beta} \sin \theta, \quad D = \frac{1}{\beta} \cos \theta$$

$$x_2 = A(x_1 - x_0) + B(y_1 - y_0) + x_0 \\ y_2 = C(x_1 - x_0) + D(y_1 - y_0) + y_0$$



Rotación del fondo en la consola Gameboy Advance

Registers for Setting the Starting Point of BG Data

Address	Register	15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00	Attributes	Initial Value
028h	BG2X_L	X-coordinate of reference starting point (rotation/scaling results)	W	0000h
038h	BG3X_L			
02Ah	BG2X_H	X-coordinate of reference starting point (rotation/scaling results)	W	0000h
03Ah	BG3X_H			
02Ch	BG2Y_L	Y-coordinate of reference starting point (rotation/scaling results)	W	0000h
03Ch	BG3Y_L			
02Eh	BG2Y_H	Y-coordinate of reference starting point (rotation/scaling results)	W	0000h
03Eh	BG3Y_H			

Registers for Setting the Direction Parameters of BG Data

Address	Register	15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00	Attributes	Initial Value
020h	BG2PA	dx: distance of movement in x direction along same line	W	0100h
030h	BG3PA			
022h	BG2PB	dmx: distance of movement in x direction along next line	W	0000h
032h	BG3PB			
024h	BG2PC	dy: distance of movement in y direction along same line	W	0000h
034h	BG3PC			
026h	BG2PD	dmy: distance of movement in y direction along next line	W	0100h
036h	BG3PD			

Fuente:
 Nintendo of America Inc.
 AGB Programming Manual,
 version 1.22, 1999 – 2001.

Ejemplo

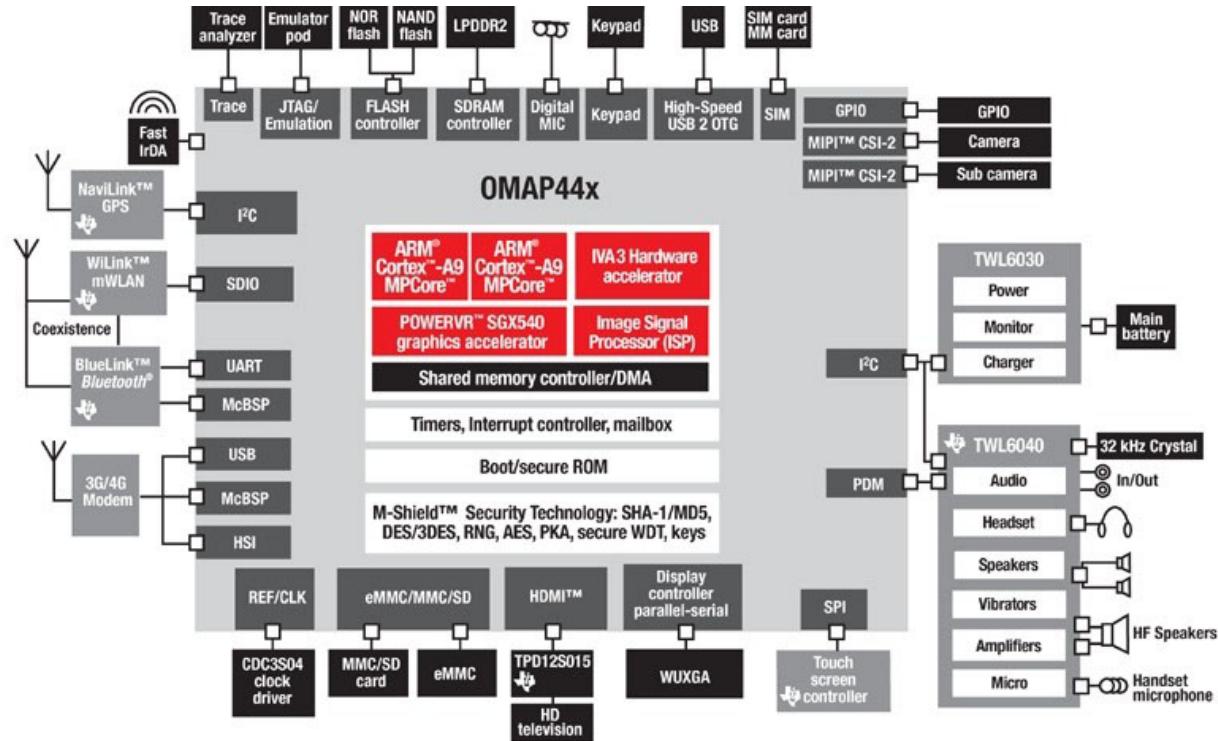


Diagrama de bloques de la plataforma OMAP4 para smartphones y tablets

2 co-procesadores SIMD NEON (dentro de los Cortex-A9)

1 GPU PowerVR SGX540 para aceleración de gráficos 3D

1 media codec IVA3 full HD para la aceleración de la codificación/decodificación multimedia

1 procesador de imágenes estáticas para la mejora de las imágenes capturadas por las cámaras

5 co-procesadores / aceleradores en total

Fuente: Texas Instruments. OMAP™ Mobile Processors : OMAP™ 4 Platform

<http://www.ti.com> > Wireless Handset Solutions > OMAP™ Mobile Processors > OMAP™ 4 Platform

Contenidos

Tema 1: Introducción a los sistemas empotrados

Presentación de la asignatura

Motivación

Descripción de la asignatura

Sistemas empotrados

Utilidad

Caracterización

Clasificaciones

Diseño e implementación

Herramientas de desarrollo

La parte hardware

Procesadores

Co-procesadores y aceleradores

Controladores de sistema

Arquitectura de memoria

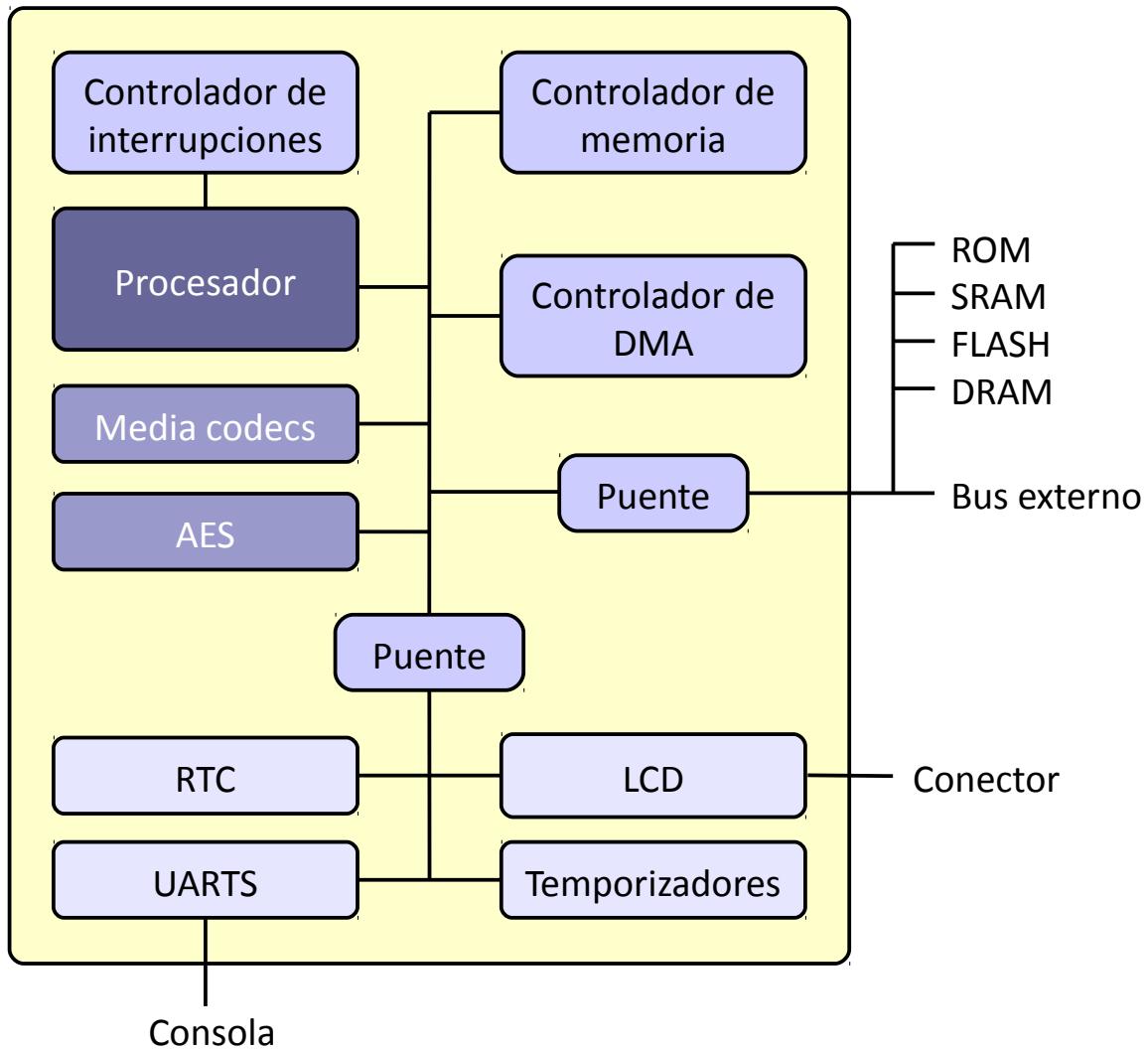
Periféricos

La parte software

Importancia creciente del software empotrado

Componentes del Firmware

Componentes hardware de un sistema empotrado



Procesador

Controla el sistema empotrado y ejecuta las funciones del sistema implementadas en software

Co-procesadores / Aceleradores

Ejecutan funciones complejas mediante hardware específico

Controladores de sistema

Coordinan los componentes importantes del sistema. Los más importantes son el de interrupciones y el de memoria

Periféricos

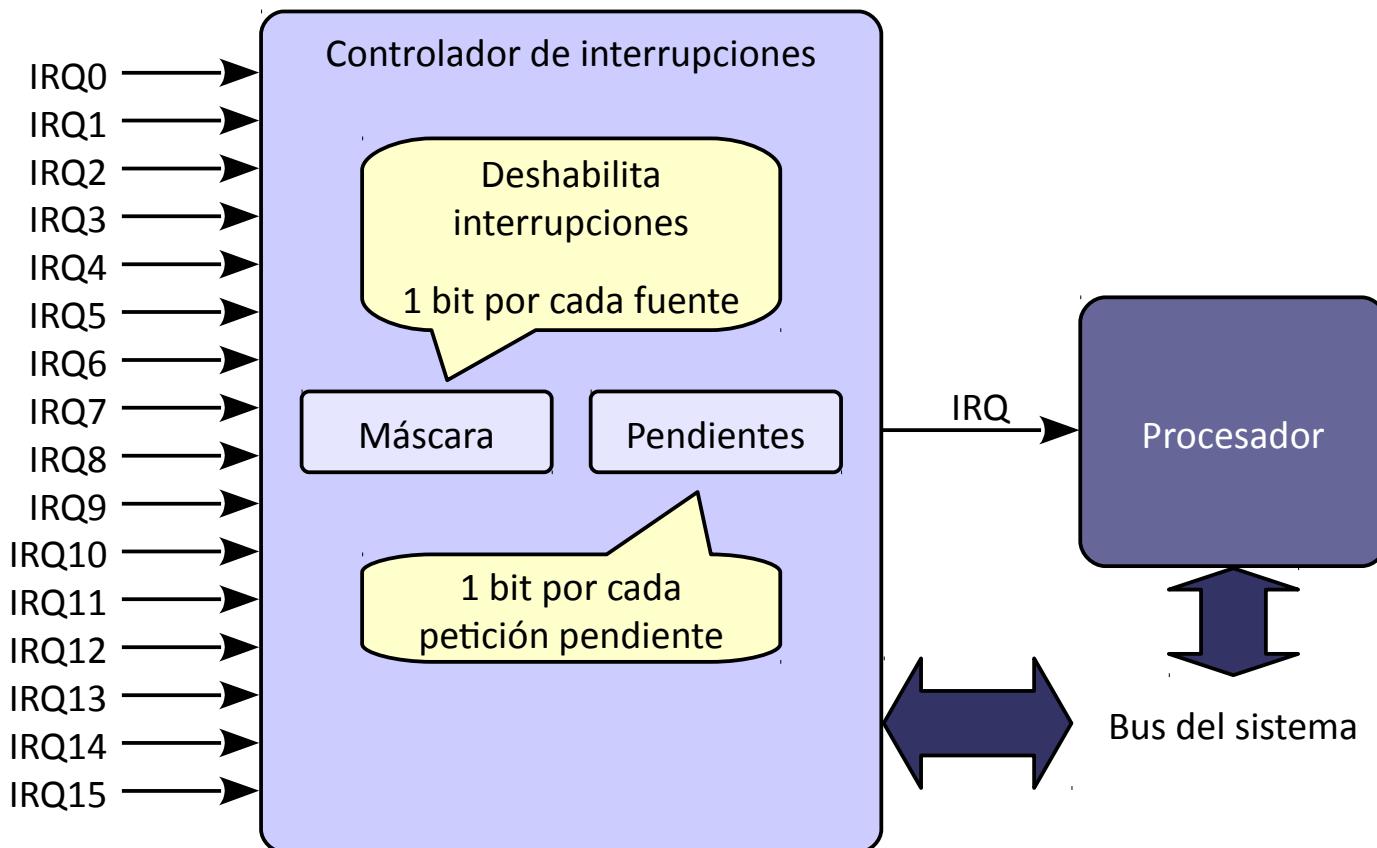
Proporcionan las capacidades de E/S del sistema. Dependen de la aplicación

Buses

Comunican las diferentes partes del sistema

Controlador de interrupciones básico

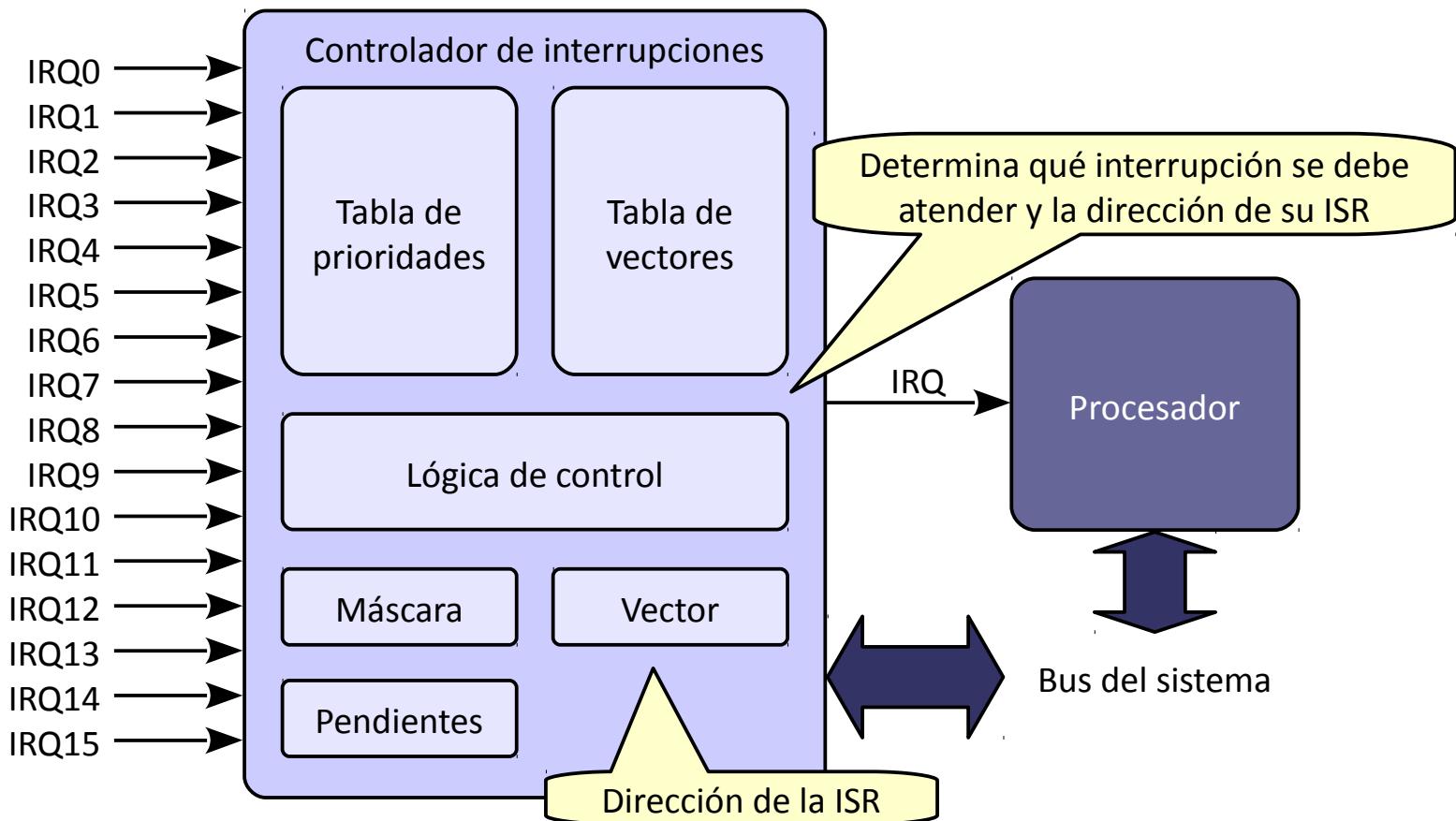
Permite manejar múltiples fuentes de interrupción



- El manejador de interrupciones debe determinar la fuente y llamar a la ISR adecuada
- La latencia de las ISR es alta

Controlador de interrupciones vectorizado

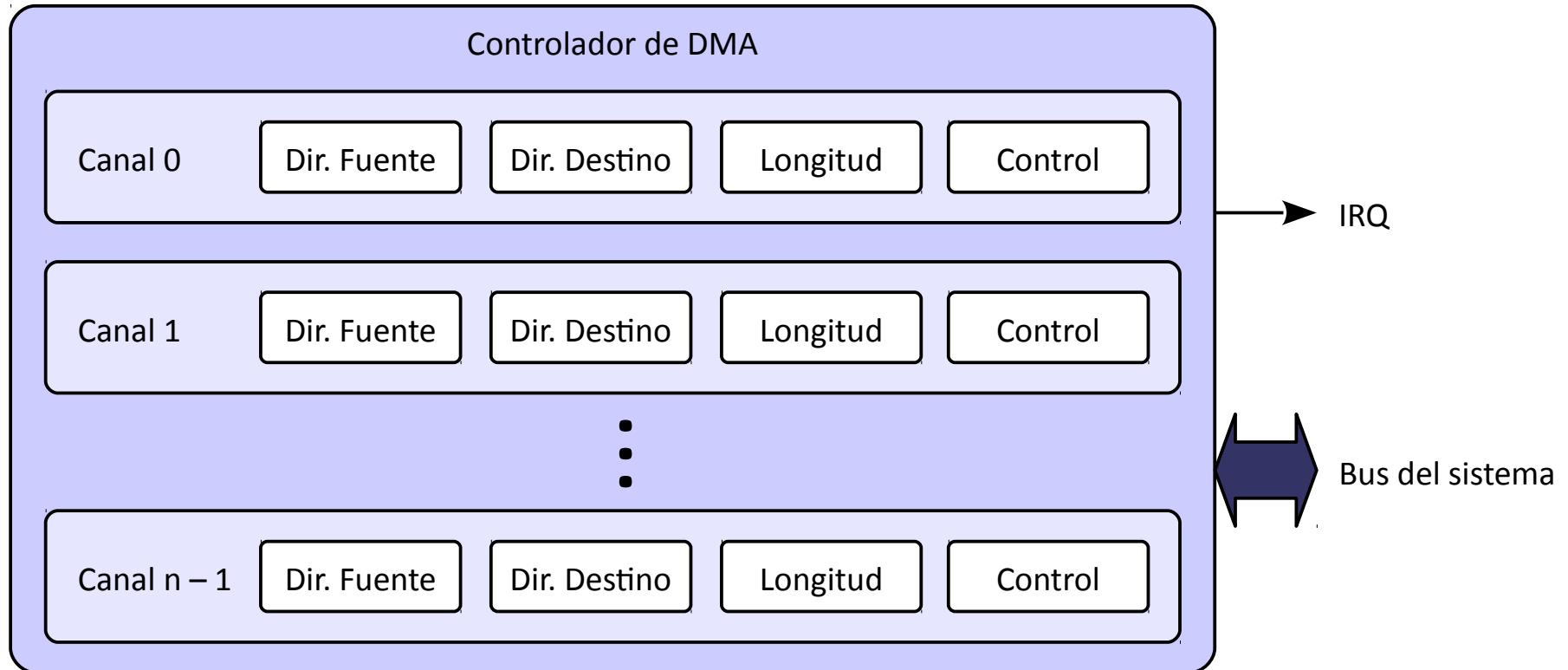
Aumenta el número de fuentes de interrupción y minimiza la latencia de la ISR



- **Priorización:** Decide qué interrupciones tienen prioridad sobre el resto
- **Vectorización:** Proporciona el puntero a la ISR de la interrupción

Controlador de DMA

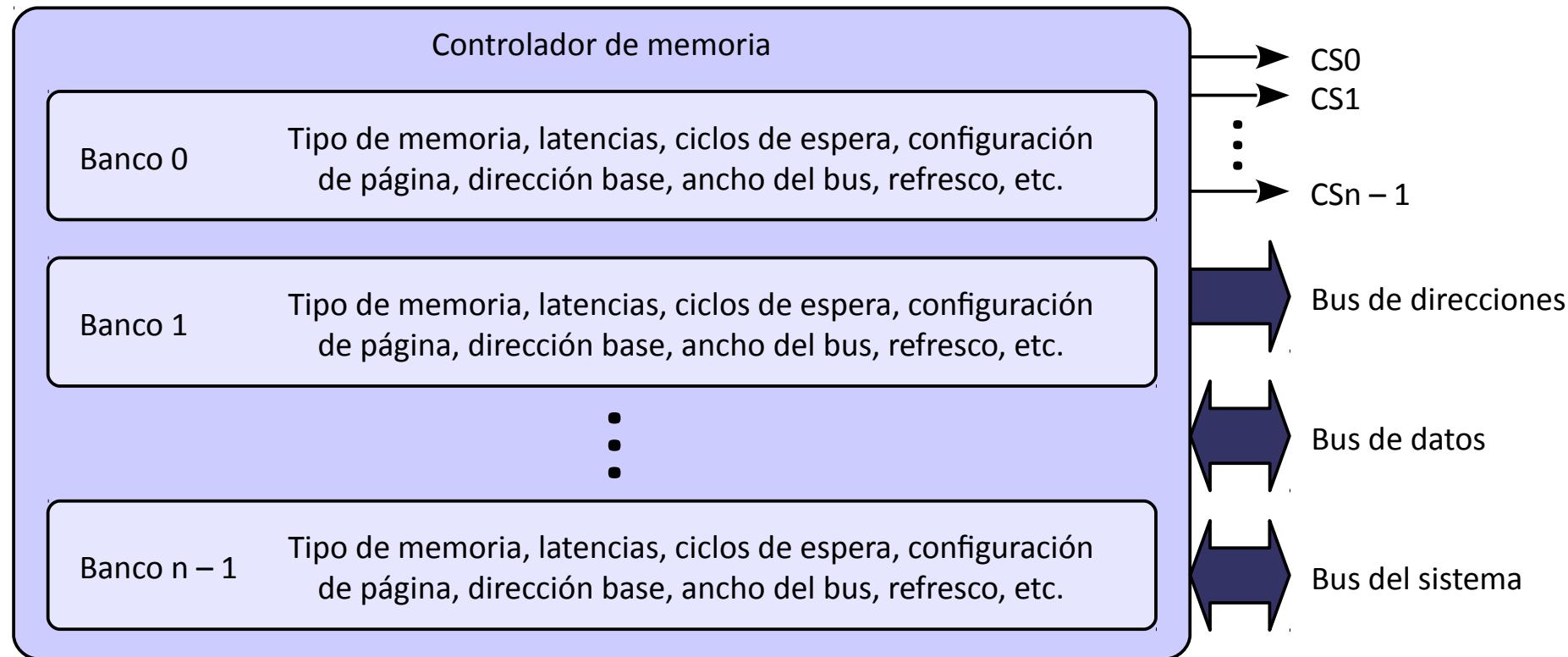
Transfiere grandes volúmenes de datos



Las transferencias se hacen mientras el procesador realiza otras tareas

Controlador de memoria

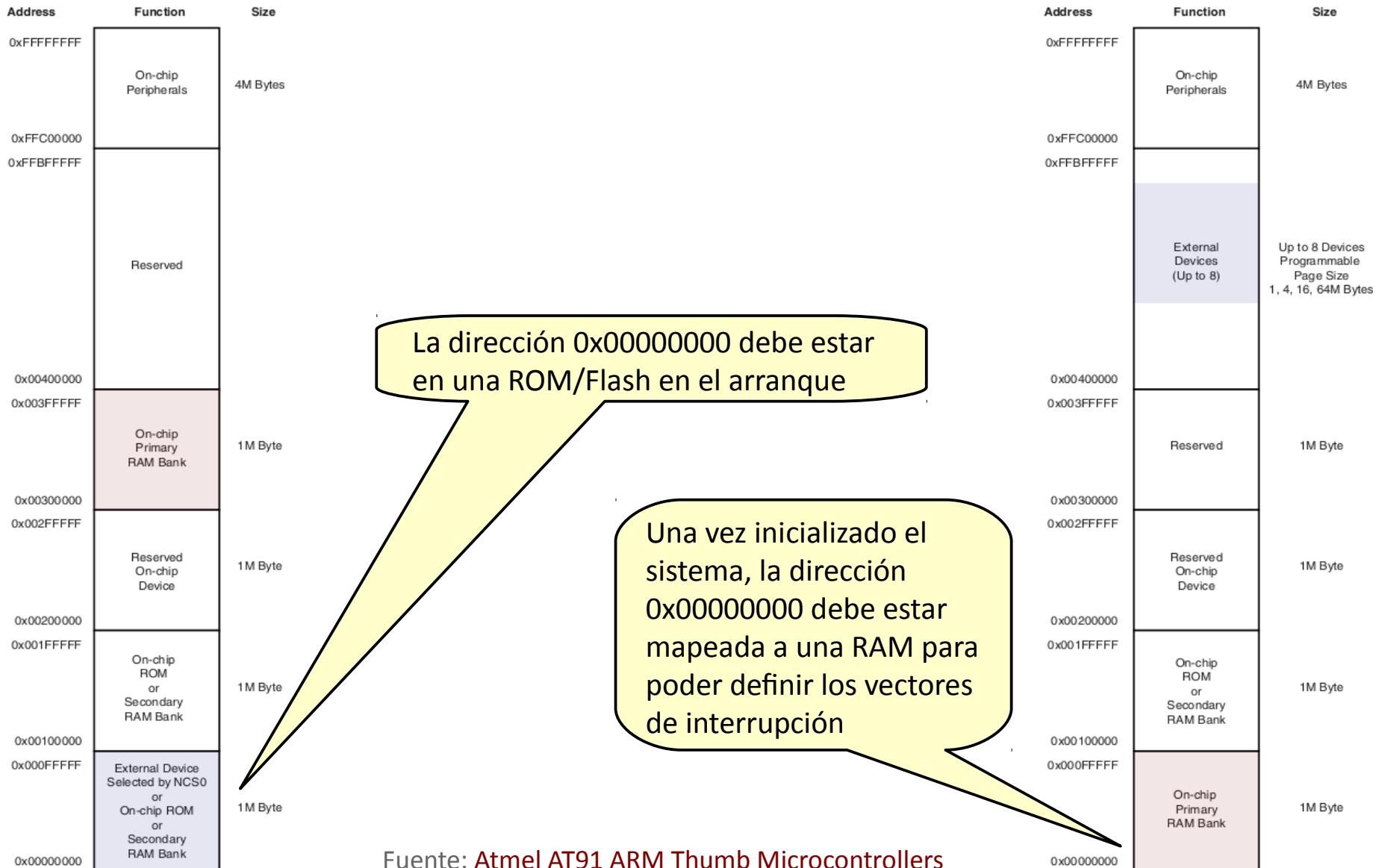
Gestiona las diferentes memorias del sistema



- Permite leer y escribir en las diferentes memorias conectadas al sistema. Refresca las SDRAM
- En sistemas sin MMU permite cambiar el mapa de memoria (REMAP) en tiempo de ejecución

Ejemplo de remapeo de memoria

Mapa de memoria del AT91R40008 antes y después del remapeo



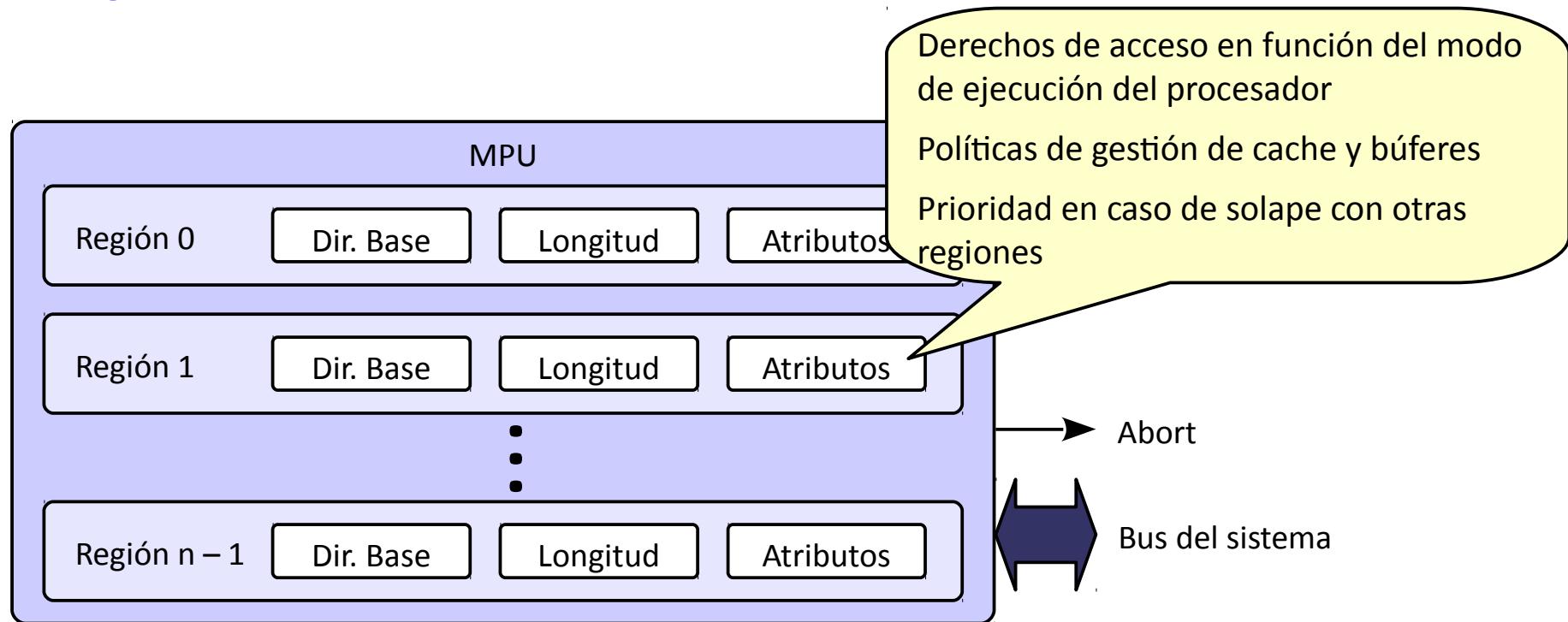
Antes del remapeo

Fuente: Atmel AT91 ARM Thumb Microcontrollers
www.atmel.com/Images/doc1354.pdf

Después del remapeo

MPU

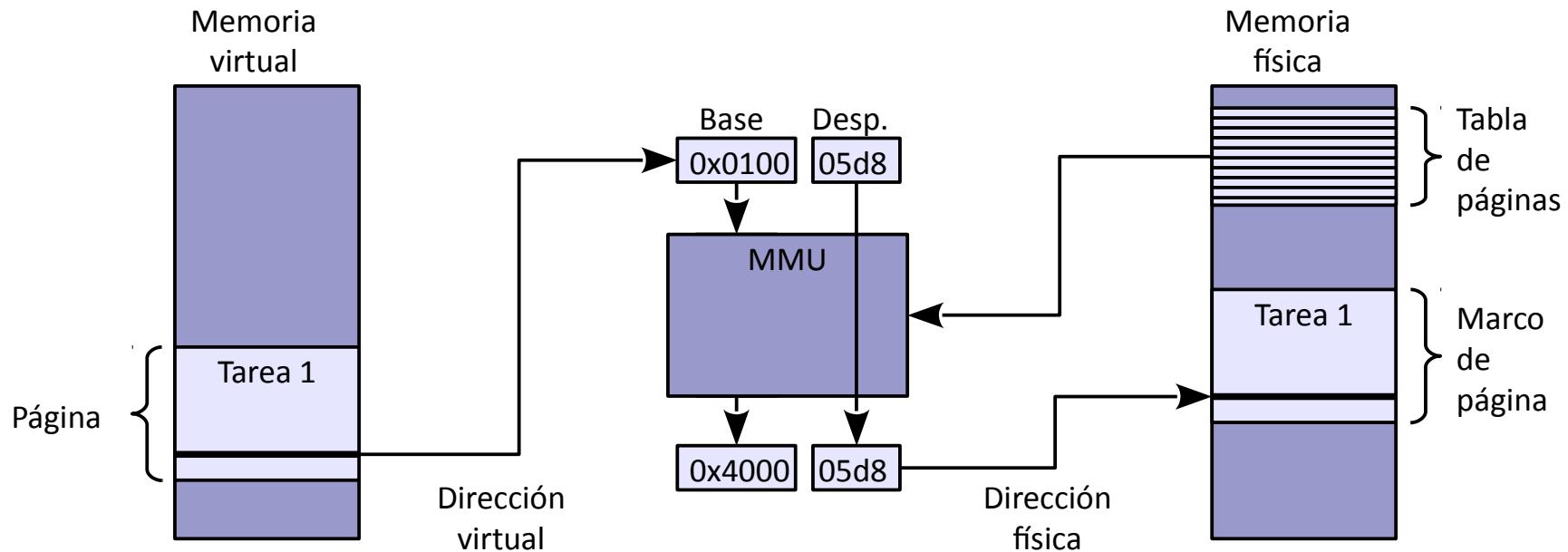
Protege la memoria de accesos no deseados



- Muy útil en aplicaciones multi-tarea (**1 sola aplicación con varias tareas**)
- Todas las tareas se ejecutan en un único espacio de direcciones físicas (**deben repartirse el espacio de direcciones**)
- En cada acceso a memoria se comprueba que se cumplen los atributos o se genera una excepción
- Como los periféricos están mapeados a memoria, también se usa para proteger su acceso

MMU

Protección y soporte para direcciones virtuales



- Imprescindible en sistemas multi-tarea (**varias aplicaciones ejecutándose en paralelo. Ej. Linux, Android, ...**)
- Cada proceso se ejecuta en su propio espacio privado de direcciones virtuales
- La MMU traduce las direcciones virtuales a direcciones físicas
- Los permisos de acceso y las políticas de gestión de caches y búferes se aplican a nivel de página
- Necesidad de un kernel de SO para gestionar las tablas de páginas y los cambios de contexto

Contenidos

Tema 1: Introducción a los sistemas empotrados

Presentación de la asignatura

Motivación

Descripción de la asignatura

Sistemas empotrados

Utilidad

Caracterización

Clasificaciones

Diseño e implementación

Herramientas de desarrollo

La parte hardware

Procesadores

Co-procesadores y aceleradores

Controladores de sistema

Arquitectura de memoria

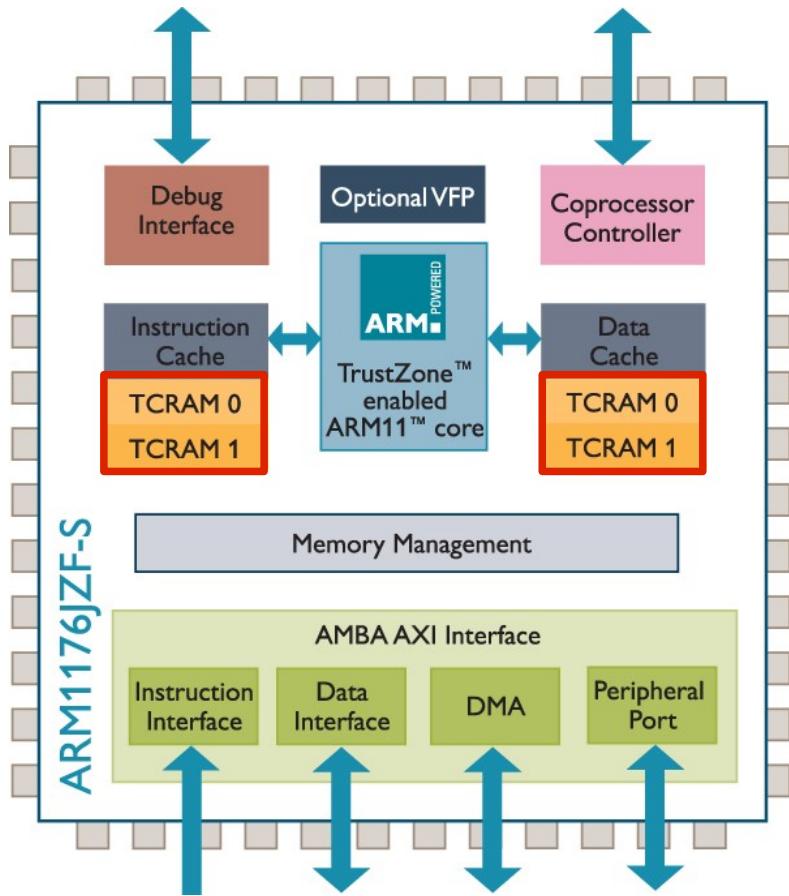
Periféricos

La parte software

Importancia creciente del software empotrado

Componentes del Firmware

Tightly-coupled Memories



Características

Tiempo de acceso a memoria mínimo y determinista

Gestionadas mediante software

Uso

Rutinas críticas (ISR, tiempo real, etc.)

Datos críticos (pilas de interrupciones y datos cuya localidad no se adapta a la política de la cache)

Fundamental en

Procesadores multicore

Aplicaciones de tiempo real

Contenidos

Tema 1: Introducción a los sistemas empotrados

Presentación de la asignatura

Motivación

Descripción de la asignatura

Sistemas empotrados

Utilidad

Caracterización

Clasificaciones

Diseño e implementación

Herramientas de desarrollo

La parte hardware

Procesadores

Co-procesadores y aceleradores

Controladores de sistema

Arquitectura de memoria

Periféricos

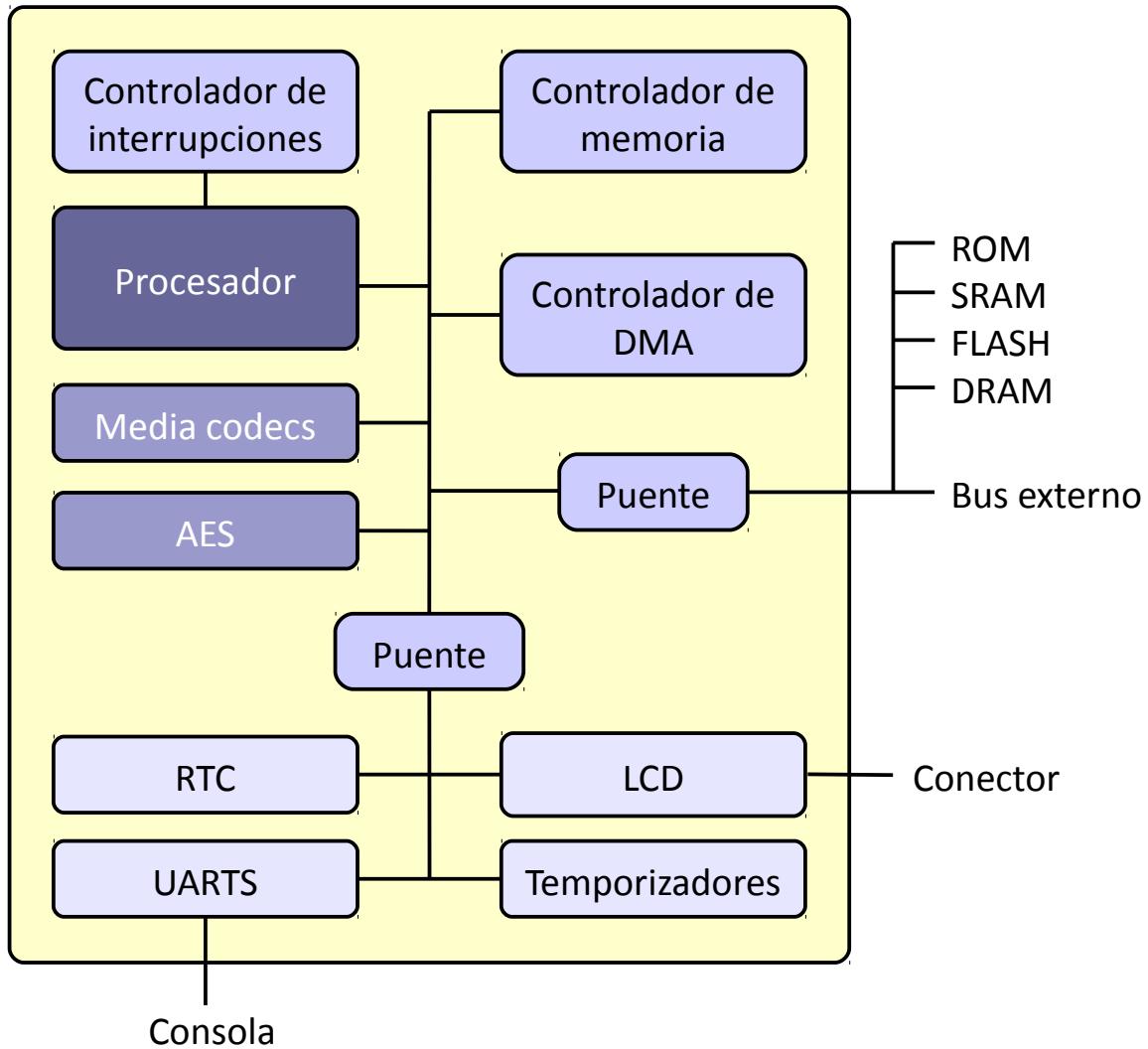
La parte software

Importancia creciente del software empotrado

Desarrollo del Firmware

Componentes del Firmware

Componentes hardware de un sistema empotrado



Procesador

Controla el sistema empotrado y ejecuta las funciones del sistema implementadas en software

Co-procesadores / Aceleradores

Ejecutan funciones complejas mediante hardware específico

Controladores de sistema

Coordinan los componentes importantes del sistema. Los más importantes son el de interrupciones y el de memoria

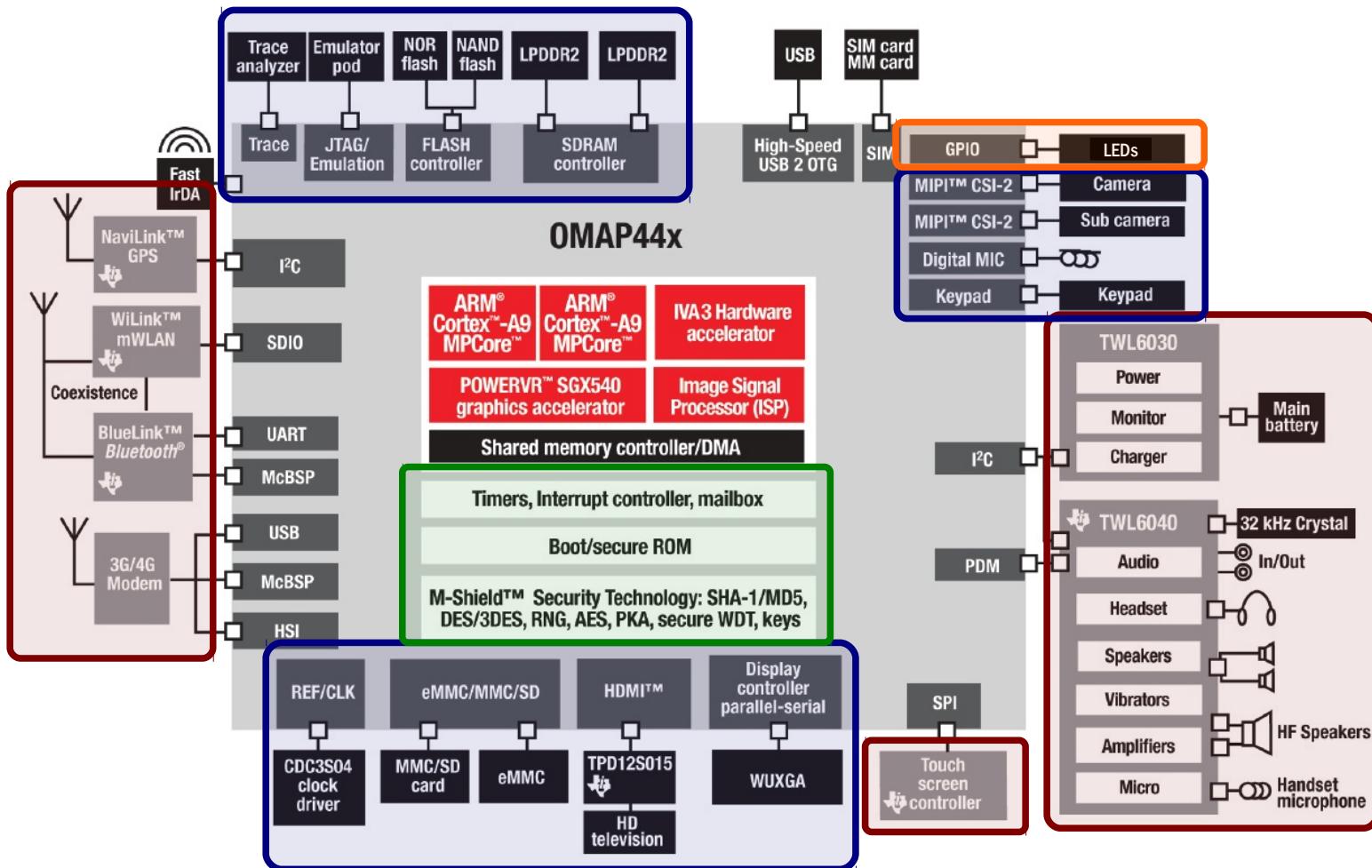
Periféricos

Proporcionan las capacidades de E/S del sistema. Dependen de la aplicación

Buses

Comunican las diferentes partes del sistema

Posibilidades de conexión de los periféricicos



Interno

Controlador interno,
Dispositivo externo

Externo, conectado
a un bus serie

Externo,
conectado a GPIO

Gestión de los periféricos y controladores internos

Normalmente, los registros de control/estado están mapeados en memoria

09FFFFFFh	Game Pak ROM Wait State 0 (32 MB)
08000000h	
070003FFh	OAM (1 Kbyte)
07000000h	
06017FFFh	VRAM (96 Kbytes)
06000000h	
050003FFh	Palette RAM (1 Kbyte)
05000000h	
04000000h	I/O, Registers
03007FFFh	CPU Internal Working RAM (32 Kbytes)
03000000h	
0203FFFFh	CPU External Working RAM (256 Kbytes)
02000000h	
00003FFFh	System ROM (16 Kbytes)
00000000h	

Control Game Pak	04000204h
Interrupciones	04000200h
Puerto serie	04000134h
Teclado	04000130h
Puerto serie	04000120h
Temporizadores	04000100h
DMA	040000B0h
Sonido	04000060h
Vídeo	04000000h

Mapa de memoria de la consola Gameboy Advance

Fuente:

Nintendo of America Inc. AGB Programming Manual, version 1.22, 1999 – 2001.



Gestión de los periféricos y controladores internos

<u>04000204h</u>	Control Game Pak
<u>04000200h</u>	Interrupciones
<u>04000134h</u>	Puerto serie
<u>04000130h</u>	Teclado
<u>04000120h</u>	Puerto serie
<u>04000100h</u>	Temporizadores
<u>040000B0h</u>	DMA
<u>04000060h</u>	Sonido
<u>04000000h</u>	Vídeo

<u>040000F2h</u>	DMA3CNT H	DMA3 control
<u>040000F0h</u>	DMA3CNT L	DMA3 word count
<u>040000E6h</u>	DMA3DAD	DMA3 Dest. Address
<u>040000E2h</u>	DMA3SAD	DMA3 Source Address
<u>040000E0h</u>	DMA2CNT H	DMA2 control
<u>040000DEh</u>	DMA2CNT L	DMA2 word count
<u>040000DAh</u>	DMA2DAD	DMA2 Dest. Address
<u>040000C6h</u>	DMA2SAD	DMA2 Source Address
<u>040000C4h</u>	DMA1CNT H	DMA1 control
<u>040000C2h</u>	DMA1CNT L	DMA1 word count
<u>040000BEh</u>	DMA1DAD	DMA1 Dest. Address
<u>040000BAh</u>	DMA1SAD	DMA1 Source Address
<u>040000B8h</u>	DMA0CNT H	DMA0 control
<u>040000B6h</u>	DMA0CNT L	DMA0 word count
<u>040000B4h</u>	DMA0DAD	DMA0 Dest. Address
<u>040000B0h</u>	DMA0SAD	DMA0 Source Address

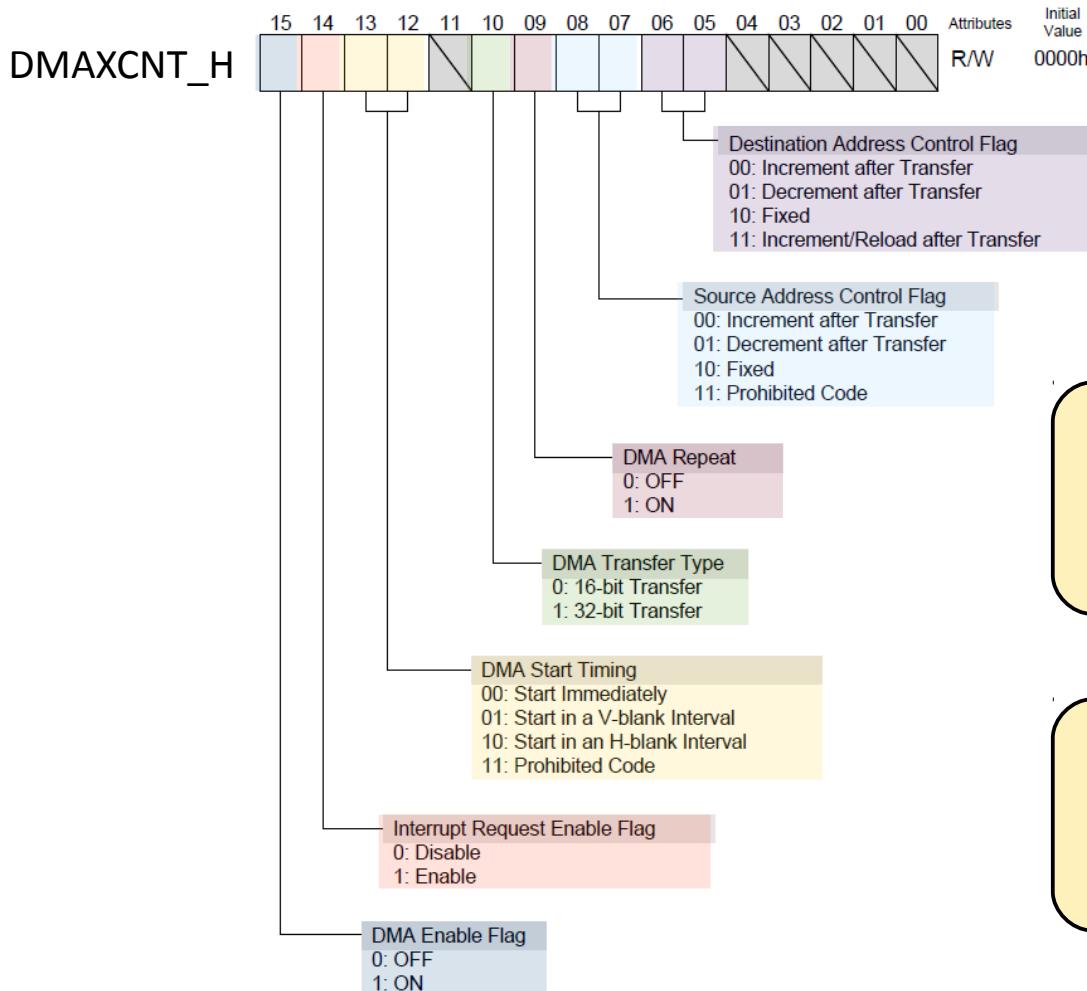
Registros de control/estado del controlador de DMA

Fuente:

Nintendo of America Inc. AGB Programming Manual, version 1.22, 1999 – 2001.



Gestión de los periféricos y controladores internos



El controlador implementa todas las funciones de gestión del dispositivo mediante hardware

El software sólo debe escribir las órdenes adecuadas en los registros de control

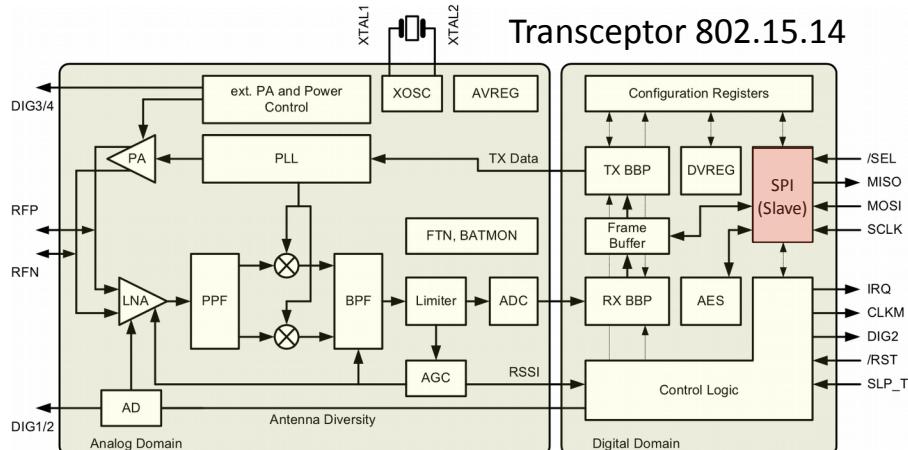
Registro de control de las transferencias

Fuente:

Nintendo of America Inc. AGB Programming Manual, version 1.22, 1999 – 2001.

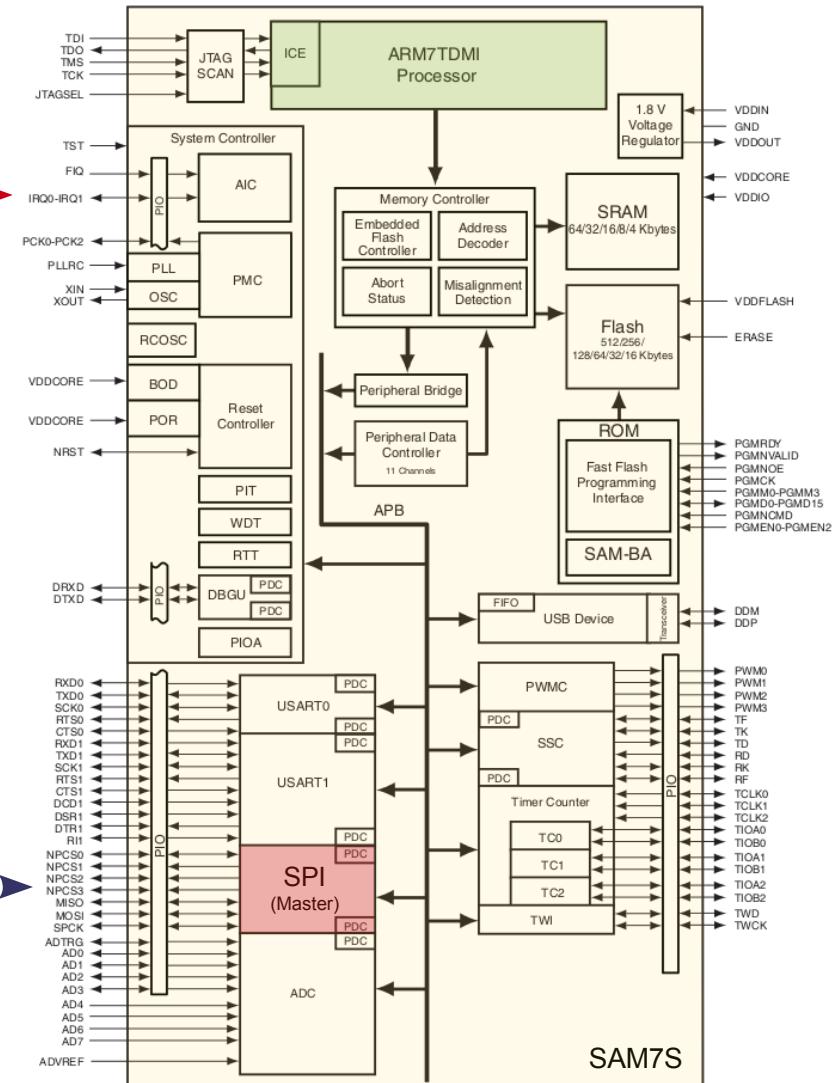


Periféricos externos conectados a un puerto serie



El periférico implementa todas las funciones de gestión del dispositivo mediante hardware

El software sólo mandar las órdenes adecuadas y recibir los datos por el puerto serie

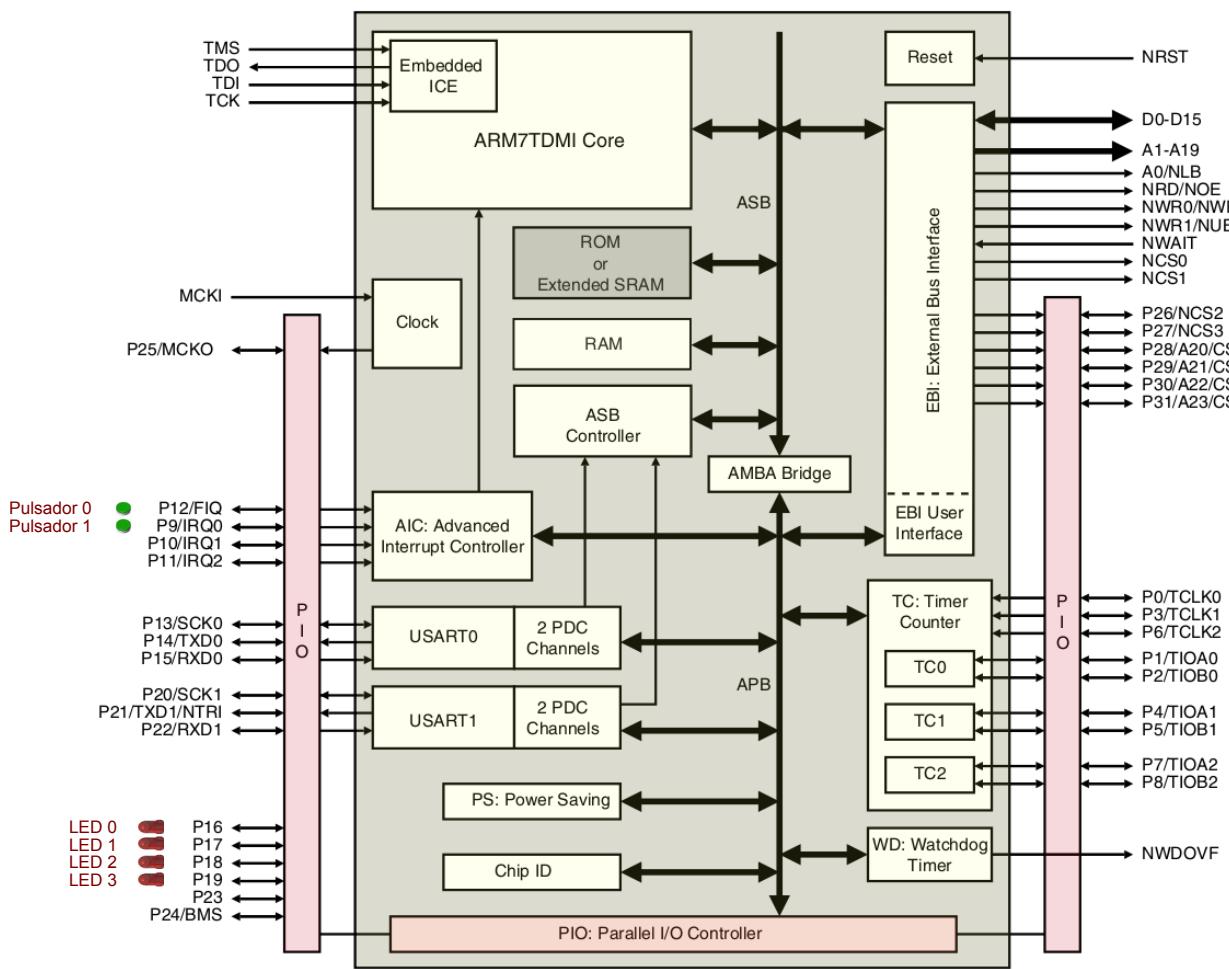


Fuente:

Atmel. AT91SAM ARM-based Flash MCU. SAM7S512, SAM7S256, SAM7S128, SAM7S64, SAM7S321, SAM7S32, SAM7S161, SAM7S16, 2011.

Atmel. AVR Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE, SP100, WirelessHART, and ISM Applications. AT86RF231-ZU, AT86RF231-ZF, 2009

Uso del GPIO (General Purpose Input/Output)



Los pines de E/S están multiplexados

El GPIO configura el uso de los pines

Conectados a un periférico interno o E/S

Si son de entrada o de salida

Generación de interrupciones

La gestión de dispositivos externos es totalmente software

Dispositivos gestionados por el PIO en la placa de desarrollo AT91EB40A de Atmel

Fuente:

Atmel. AT91EB40A Evaluation Board User Guide, 2002.

Atmel. AT91 ARM Thumb Microcontrollers. AT91M40800, AT91R40807, AT91M40807 y AT91R40008, 2002

Lecturas recomendadas

Procesadores empotrados:

A. S. Berger. *Embedded System Design. An introduction to Processes, Tools, & Techniques.* CMP Books, 2002. Capítulo 2

S. Heath. *Embedded System Design*. Newness, 2^a edición, 2003. Capítulo 2

W. Wolf. *Computer as Components*. Morgan Kaufmann, 2^a edición, 2008. Capítulos 2, 3 y 7

F. Vahid y T. Givargis. *Embedded System Design*. John Wiley & Sons, 2002. Capítulos 1.3 y 3

P. Lapsley *et al.* *DSP Processor Fundamentals. Architectures and Features*. John Wiley & Sons, 1997. Capítulo 2

Embedded Insights. *Embedded Processing Directory Index (2010)*.

<http://www.embeddedinsights.com/directory.php>

nNidia. *The Benefits of Multiple CPU Cores in Mobile Devices*.

http://www.nvidia.com/content/PDF/tegra_white_papers/

Lecturas recomendadas

Medida de prestaciones:

J. L. Hennessy y D. A. Patterson. *Computer Architecture. A Quantitative Approach*. Morgan Kaufmann, 5^a edición, 2011. Capítulo 1

D. Mann y P. Cobb. *When Dhrystone Leaves You High and Dry*. EDN, Mayo 1998

EDN Embedded Microprocessor Benchmark Consortium. <http://www.eembc.org>

Berkeley Design Technology, Inc. *The BDTImark2000: A Summary Measure of Signal Processing Speed*. Septiembre 2004

Co-procesadores y Aceleradores:

F. Vahid y T. Givargis. *Embedded System Design*. John Wiley & Sons, 2002. Capítulos 1.3 y 2

W. Wolf. *Computer as Components*. Morgan Kaufmann, 2^a edición, 2008. Capítulo 7

Controladores de sistema:

F. Vahid y T. Givargis. *Embedded System Design*. John Wiley & Sons, 2002. Capítulos 1.3 y 2

A. N. Sloss, D. Symes y C. Wright. *ARM System Developer's Guide. Designing and Optimizing System Software*. Morgan Kaufmann, 2004. Capítulos 13 y 14