



Microprocesadores ARM

- Introducción a los microprocesadores ARM
- Comparación con procesadores Intel

Historia de ARM

1990



ARM

2004



ARM

2018



arm



ARM

- ARM = Advanced RISC Machine (antes Acorn Risc Machine)
- En 1985 comienzan con el primer micro (ARM1)
- Se crea entre tres empresas: Acorn, Apple y VLSI
- No fabrica procesadores sino que realiza las especificaciones
- Comercializa licencias (IP License (Intellectual Property))
- Tiene procesadores de 32 y 64 bits.
- Muy útil para dispositivos móviles (Alta relacion: MIPS / watt)
- Extensiones: Thumb, Jazelle, SIMD (Neon), VFP



Licencias de ARM

- Samsung
- AMD
- Broadcom
- ST-Ericsson
- Toshiba
- NVIDIA
- Texas Instruments
- Philips

Diseño de procesadores

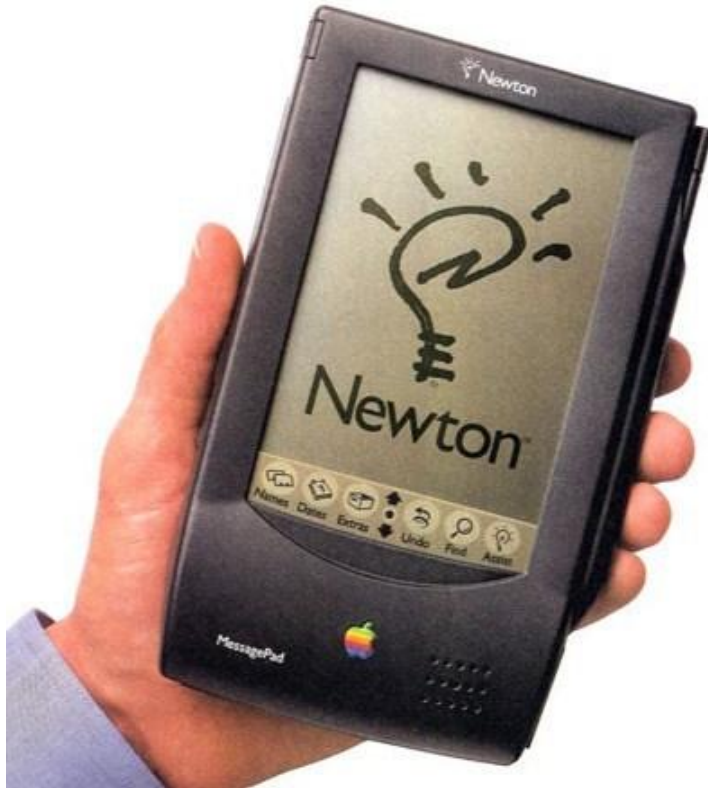
Se utiliza software para simular el microcódigo que realizan las compuertas de silicio

- Simulación lógica
- Simulación eléctrica
- Simulación térmica

Ejemplo de lenguaje : VERILOG

Lenguaje de tipo HDL (Hardware Description Language)

Apple Newton



- Se lanza en 1993
- ARM610 RISC
- Muchos bugs, caro, fracasó.

Nokia 6110



- Texas Instrument fabrica un ARM y lo prove a Nokia.
- Se lanza en 1997
- Arquitectura ARM 7
- Primer telefono GSM (Global System for Mobile communications) con ARM.
- Exito total.

Ejemplos de productos



- iPod de Apple.
- Contiene un ARM7TDMI de 90 Mhz
- 32 Mb de DRAM.
- Manejan discos de 20 y 40 GB.

Crecimiento (2002-2005)

- El procesador ARM se vuelve mucho más pequeño.
- Las empresas no tienen personal para diseñar su propio procesador o las herramientas para usarlo.
- Se orienta al SoC (System on a Chip)
- Se lanza el **ARM926EJ-S** soporta Linux, Windows CE y Symbian. Soporte DSP y aceleración Java. Cinco niveles de Pipeline



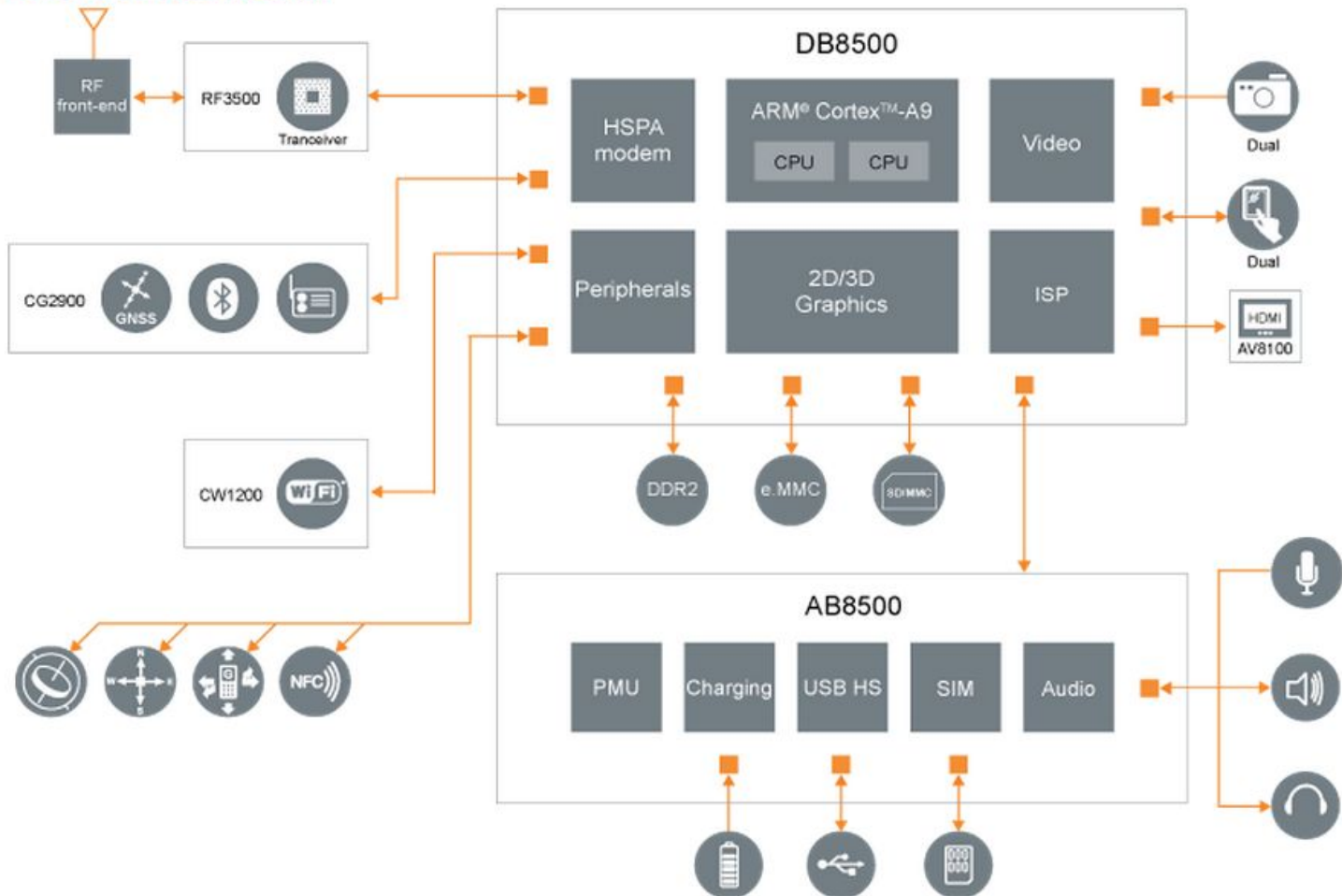
Systems on Chip (SoC)

- Un SoC está integrado por:
 - Procesador
 - Memorias (ROM, RAM, Flash)
 - Osciladores
 - Conversores A/D y D/A
 - Interfaces (USB, Ethernet, USART)

Systems on Chip (SoC)

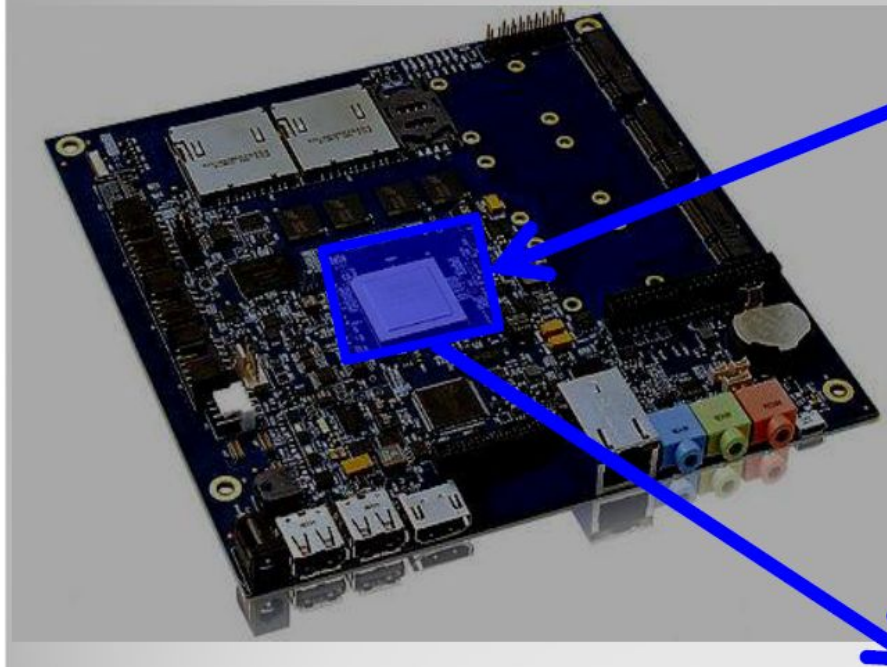
- Ejemplo: ST-Ericcson
- Novathor U8500

U8500 BLOCK DIAGRAM



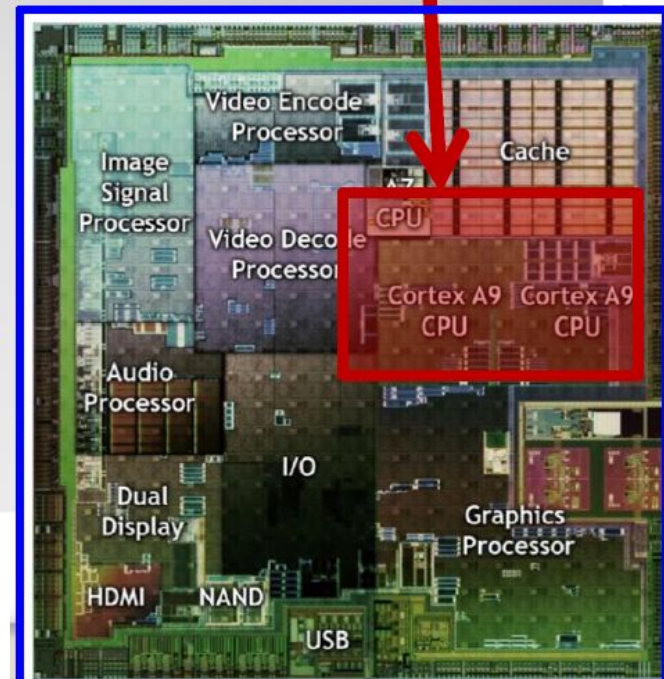
Systems on Chip (SoC)

ARM Embedded Processor



**Main CPU Chip
(SoC)**


**ARM Embedded
Processor**



Systems on Chip (SoC)

- Apple M1
 - Lanzado en 2020
 - Primer SoC que usa MaC
 - Basado en ARM
 - 8 Cores de CPU
 - Incluye GPU
 - Procesador de señales (DSP)
 - Cifrado por Hardware (AES)





Era Cortex (2005-2012)

- Smartphones !
- ARM responde con Cortex A9
- Hoy ARM tiene el 96% del Mercado móvil.



ARM Cortex A9

- Arquitectura ARMv7-A
- Simple ó Multi Procesador (hasta 4)
- Varios niveles de control de consumo
- Basado en RISC
- Instrucciones ARM y Thumb
- 37 registros de 32 bits

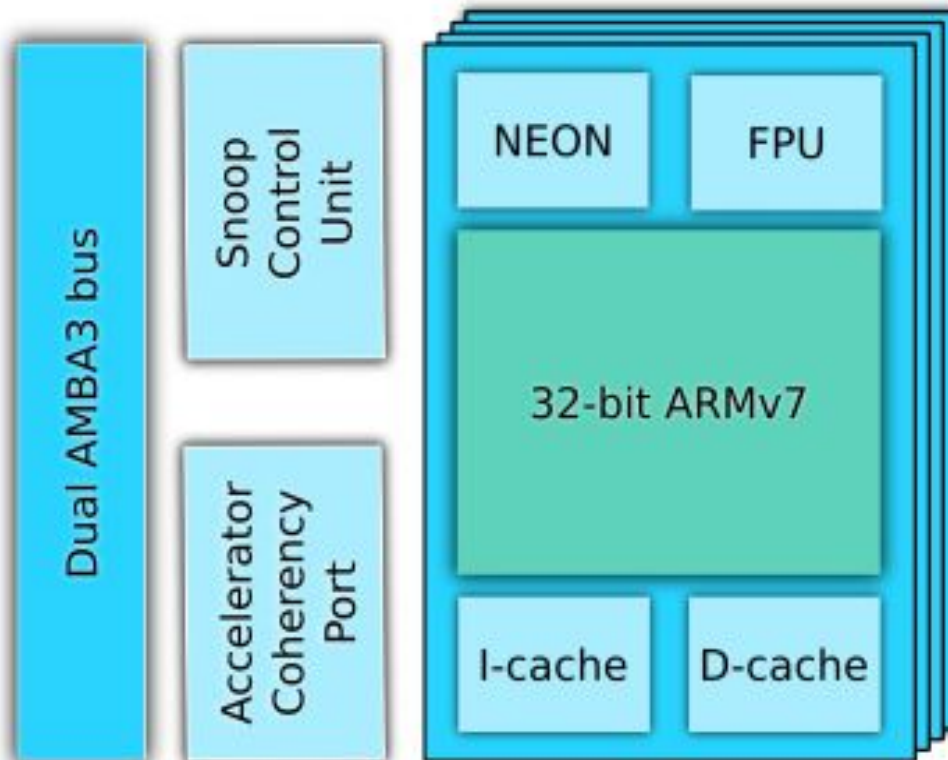


ARM Cortex A9

- Velocidad de clock entre 800MHz y 2 GHz
- Floating Point
- Controlador de caché L2
- Jazelle (Optimizador Java)
- DSP (Digital Signal Processing)

ARM Cortex A9

- Caché de Instrucciones/Datos de 16,32 o 64 KB
- Arquitectura Harvard modificada



ARM Cortex A9 SoC que lo usan:

- Apple A5 (Ipad 2 y 3, Iphone 4s, Ipad mini)
 - L1 Cache 32 KB de instrucciones y 32 KB de datos
 - L2 Cache 1 MB
- OMAP4 (Motorola Razr, kindle Fire)
- Exynos 4 (Samsung Galaxy S3, Samsung Note)
 - Quad Core ARM-Cortex-A9
 - CPU 1.4-1.6 GHz

iPhone X



- A11 Bionic chip with 64-bit architecture
- Microarquitectura: ARMv8-A
- Cache L1: 32 KB instruction, 32 KB data

Arquitecturas y familias

Architecture	Family
ARMv1	ARM1
ARMv2	ARM2, ARM3
ARMv3	ARM6, ARM7
ARMv4	StrongARM, ARM7TDMI, ARM9TDMI
ARMv5	ARM7EJ, ARM9E, ARM10E, Xscale
ARMv6	ARM11, ARM Cortex-M
ARMv7	ARM Cortex-A, ARM Cortex-M, ARM Cortex-R
ARMv8	ARM Cortex-A57, ARM Cortex-A53

Arquitecturas y familias

Processor Modes

- The ARM has seven basic operating modes:
 - Each mode has access to own stack and a different subset of registers
 - Some operations can only be carried out in a privileged mode

Exception modes	Mode	Description	
	Supervisor (SVC)	Entered on reset and when a Software Interrupt instruction (SWI) is executed	Privileged modes
	FIQ	Entered when a high priority (fast) interrupt is raised	
	IRQ	Entered when a low priority (normal) interrupt is raised	
	Abort	Used to handle memory access violations	
	Undef	Used to handle undefined instructions	
	System	Privileged mode using the same registers as User mode	Unprivileged mode
	User	Mode under which most Applications / OS tasks run	

Set de Registros

Current Visible Registers

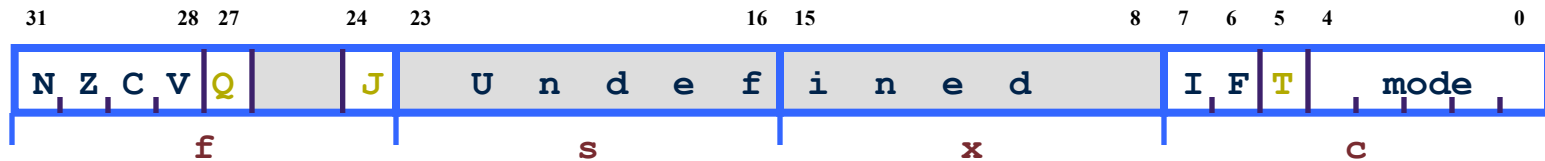
Abort Mode

r0
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13(sp)
r14(lr)
r15(pc)
cpsr
spsr

Banked out Registers

User	FIQ	IRQ	SVC	Undef
	r8			
	r9			
	r10			
	r11			
	r12			
r13(sp)	r13(sp)	r13(sp)	r13(sp)	r13(sp)
r14(lr)	r14(lr)	r14(lr)	r14(lr)	r14(lr)
	spsr	spsr	spsr	spsr

Registros - Flags



■ Condition code flags

- N = **N**egative result from ALU
- Z = **Z**ero result from ALU
- C = ALU operation **C**arried out
- V = ALU operation o**V**erflowed

■ Sticky Overflow flag - Q flag

- Architecture 5TE/J only
- Indicates if saturation has occurred

■ J bit

- Architecture 5TEJ only
- J = 1: Processor in Jazelle state

■ Interrupt Disable bits.

- I = 1: Disables the IRQ.
- F = 1: Disables the FIQ. (Fast IRQ)

■ T Bit

- Architecture xT only
- T = 0: Processor in ARM state
- T = 1: Processor in Thumb state

■ Mode bits

- Specify the processor mode



Características generales

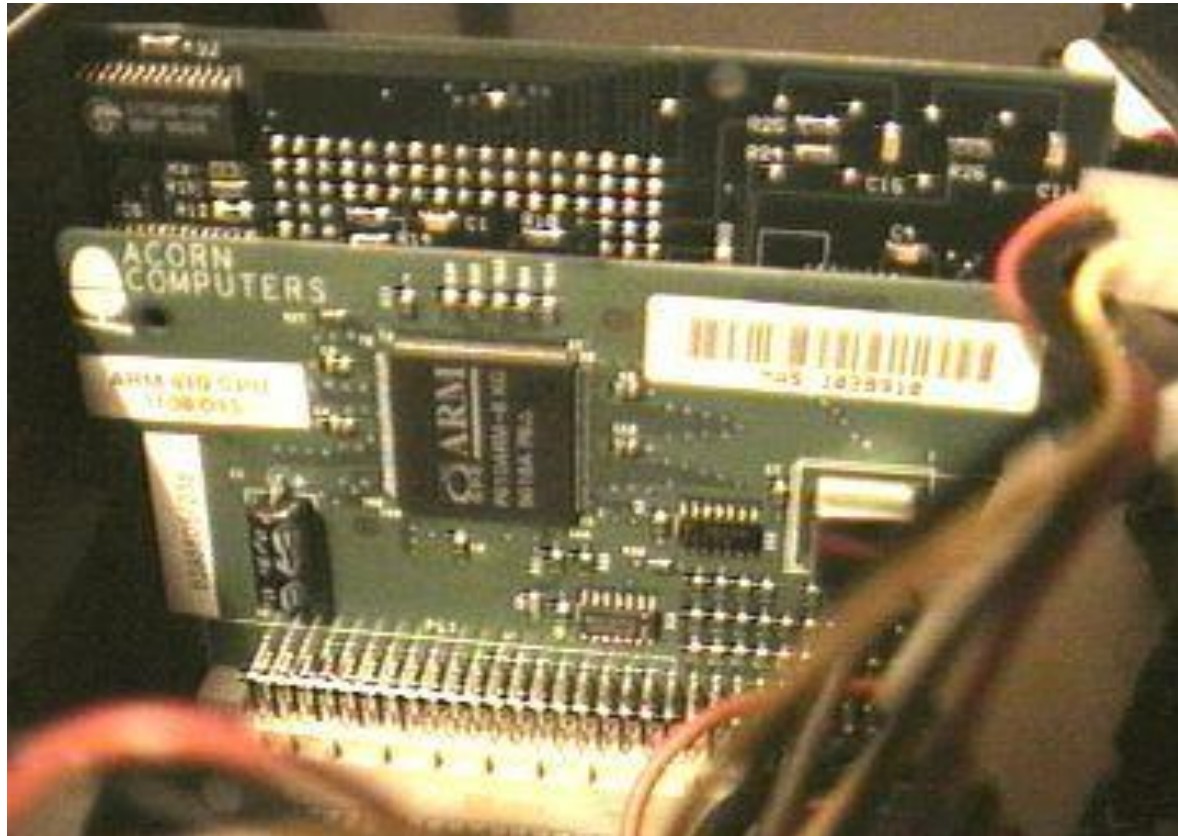
- Casi todas las instrucciones se ejecutan en un ciclo de clock y tienen tamaño fijo.
- Todas las familias de procesadores comparten el mismo conjunto de instrucciones
- Tipo de datos de 8/16/32 bits.
- Pocos modos de direccionamiento
- No se crea fragmentación de memoria

MIPS

Millones de instrucciones por segundo

Procesador	Clock	MIPS
80486	100 Mhz	70
Pentium I	60 Mhz	100
ARM720T	60 Mhz	60
ARM9TDMI	180 Mhz	200

Diferencias de tamaños



ARM610 (33MHz) y 486SXL-40 (33MHz)

Diferencias de tamaños



ARM610 (33MHz) y 486SXL-40 (33MHz)

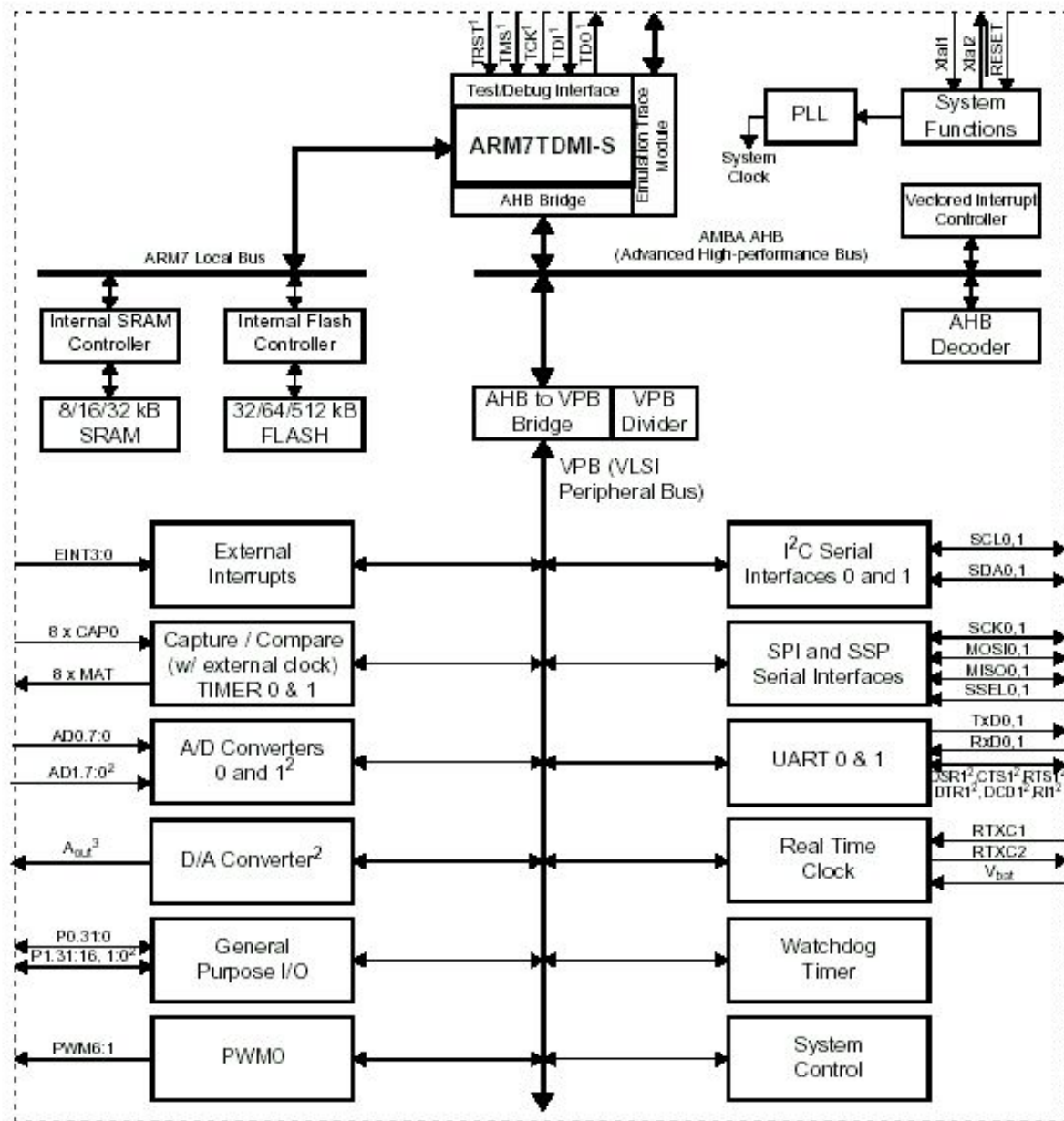
Familia LPC213x

- Microcontrolador 16/32-bit ARM7TDMI-S
- RAM de 8/16/32 kB en chip.
- 32/64/512 kB de memoria Flash.
- 1 ó 2 Conversores A/D con 10 bits de resolución.
- Conversor D/A con 10 bits de resolución.
- 2 contadores/timer de 32 bits.
- Real Time Clock con batería independiente
- Interfaces serie (UART, I2C, SPI y SPP)
- Controlador de interrupciones



Familia LPC213x

- Oscilador en chip. Entre 1 y 30 MHz
- Interrupciones manejadas con vector.
- Hasta 9 interrupciones de hardware
- Modo de conservación de energía.
- Encendido por interrupción de hardware.



¹Shared with GPIO.

²LPC2138 only.

³LPC2132/2138 only.



Memorias en LPC213x

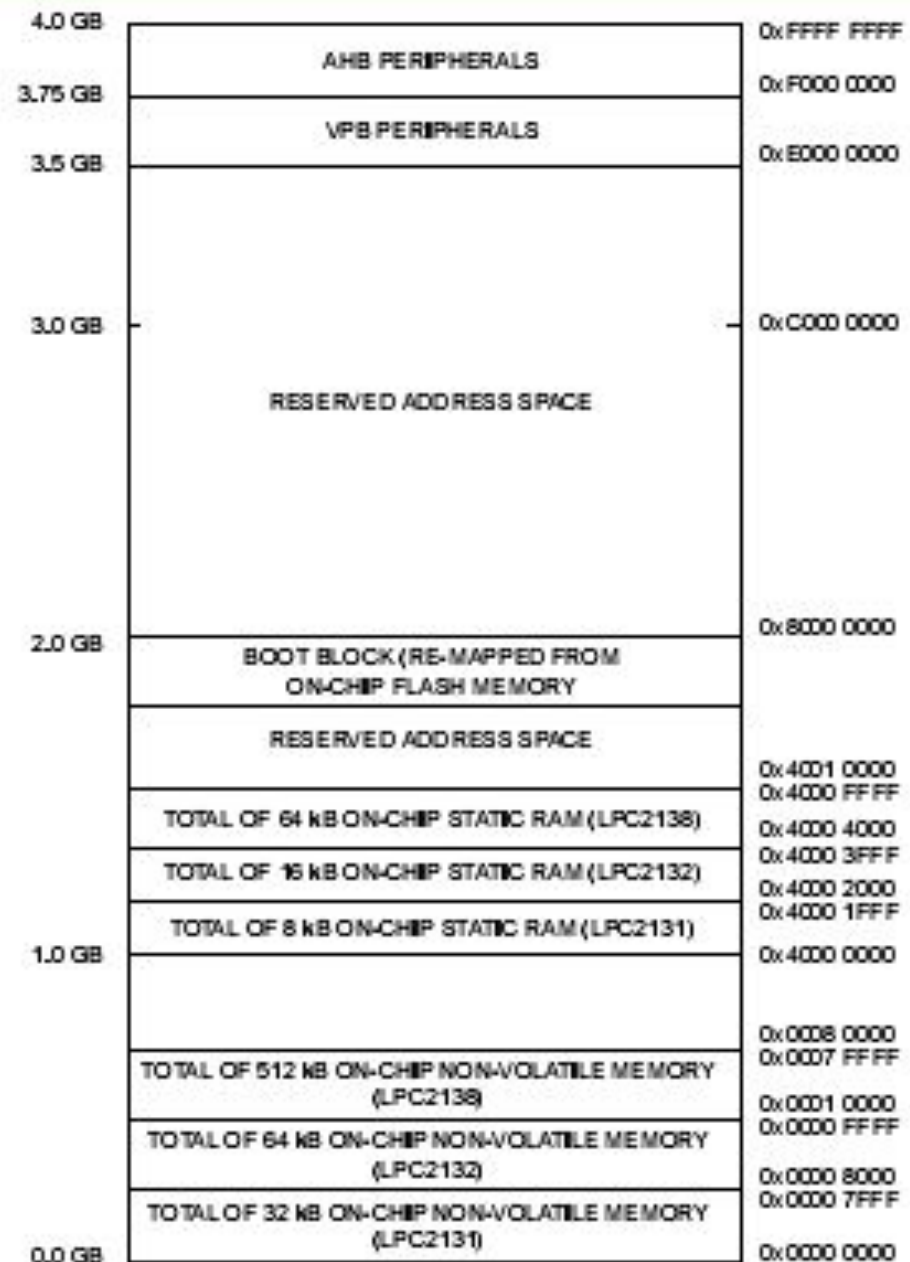
Memoria Flash On-Chip

- En ella se puede almacenar código y dato.
- Se puede programar a través del puerto serie
- Se puede programar o borrar mientras está corriendo el código
- (muy util para upgrade firmware)
- Soporta 10.000 borrados y/o escrituras
- 10 años puede retener la información guardada.

RAM

- Se puede utilizar para código y datos.

Mapa de memoria

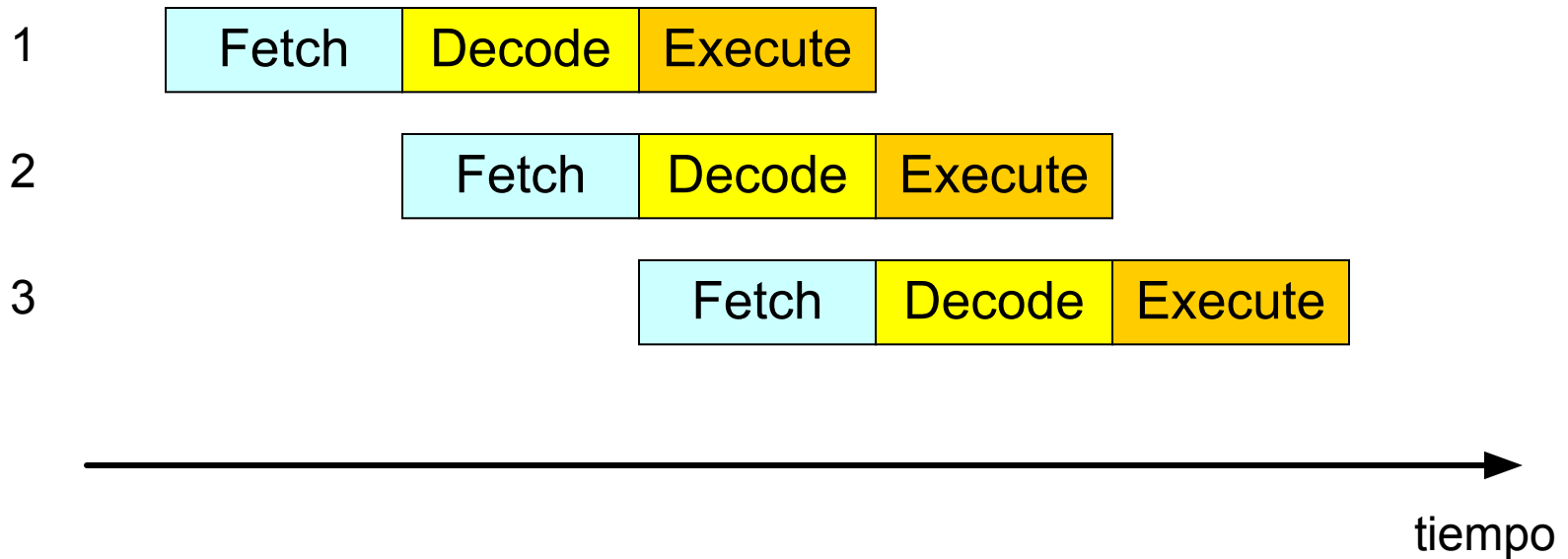


002aab 089

Fig 3. LPC2131/2132/2138 memory map.

Pipeline en ARM7

Posee un pipeline de 3 etapas

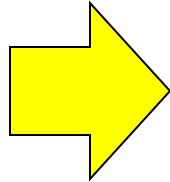


Una instrucción de salto, provoca que se vacíe el pipeline

Instrucciones en ARM

Instrucciones condicionales

```
If ( a==b)  
{ a++; } else { a--; }
```



```
Cmp R1,R2 /* se asume R1 vale a y R2 b */  
Addeq R1,#1 /* suma 1 a R1 */  
Subne R1,#1 /* resta si no es igual */
```

Vemos que con las instrucciones condicionales se evitan los saltos en el código que demoran la ejecución del programa

Instrucciones en ARM

Instrucciones condicionales

31

28

Condición (4bits)

(Otros campos de la instrucción)

Antes de la ejecución de cada instrucción se chequea los bits de condición (31:28) para determinar si se debe ejecutar o no.

Por ejemplo:

La condición '0000' significa EQUAL. Por lo tanto la instrucción sólo podrá ser ejecutada si el flag Z (zero) está activo.

Condiciones

- The possible condition codes are listed below:
 - Note AL is the default and does not need to be specified

Suffix	Description	Flags tested
EQ	Equal	Z=1
NE	Not equal	Z=0
CS/HS	Unsigned higher or same	C=1
CC/LO	Unsigned lower	C=0
MI	Minus	N=1
PL	Positive or Zero	N=0
VS	Overflow	V=1
VC	No overflow	V=0
HI	Unsigned higher	C=1 & Z=0
LS	Unsigned lower or same	C=0 or Z=1
GE	Greater or equal	N=V
LT	Less than	N!=V
GT	Greater than	Z=0 & N=V
LE	Less than or equal	Z=1 or N!=V
AL	Always	



TDMI

T : THUMB (set de instrucciones)

Set de instrucciones de 16 bits que se suman a las estándar de 32 bits.

D : Debug Interface (JTAG)

(Permite realizar un HALT del micro para debug)

M : Multiplicador (en hardware, de alta resolución)

Componente interno del micro, logra resultados de 64 bits.

I: Interrupt (interrupciones rápidas)

THUMB

El microprocesador tiene 2 sets de instrucciones:

El ARM clásico de 32 bits y el Thumb de 16 bits.

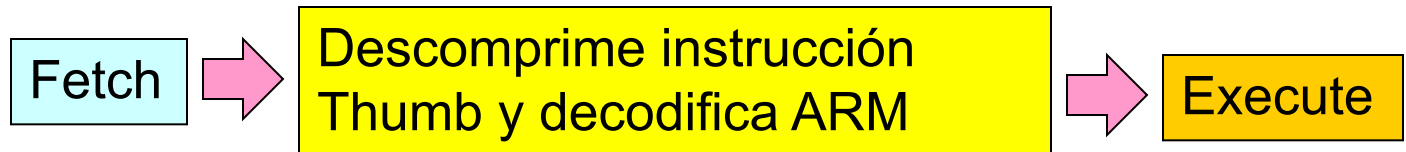
En Thumb:

- Instrucciones de 16 bits.
- Se descomprimen en forma dinámica en el módulo de “decode” del pipeline

**Pipeline
ARM**



**Pipeline
Thumb**





THUMB

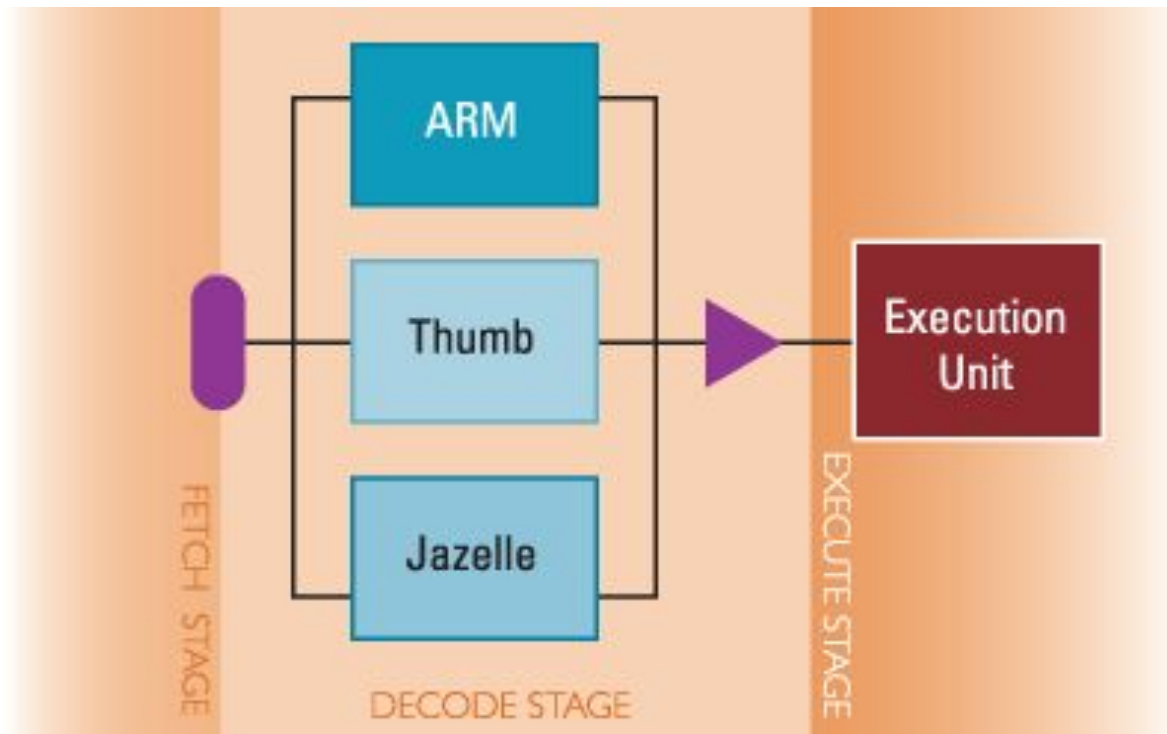
En Thumb:

- Se gana entre un 35% y un 40% de memoria comparado con el set de 32 bits
- Cada instrucción de 16 bits tiene su correspondiente en 32 bits.
- Al usar enteros de 32 bits la ventaja la tiene el set ARM de 32 bits.
- Al funciones, por ejemplo, de manejo de caracteres, es mejor Thumb
- Se puede switchear de Thumb a ARM en forma dinámica, para aplicaciones combinadas (16 y 32 bits) y así no perder performance.
- La mayoría de las instrucciones Thumb no son condicionales.
- La mayoría de las instrucciones utilizan 2 operandos
- En ARM la mayoría de las instrucciones utilizan 3 operandos.
- Debido a la compresión se pierden algunas funcionalidades especiales

Extensión Jazelle

Los procesadores con extension Jazelle, son capaces de ejecutar bytes code de Java directamente en hardware.

Las instrucciones Java que no posee las emula con las instrucciones ARM.





ARM9TDMI

- Es el reemplazo de los ARM7
- Es compatible a nivel binario con ARM7
- Tiene un pipeline de 5 niveles en lugar de 3. Esto permite aumentar la frecuencia del clock.
- Con memoria cache.
- Arquitectura **Harvard Modificada**. Instrucciones y datos separados en cache pero mismo espacio de memoria.