

ARM

- Advanced RISC Machines Ltd.
- Compañía inglesa Dedicada al desarrollo de procesadores RISC, software y tecnologías relacionadas, fundada en 1990 por Apple Computer, Acorn Computer Group y VLSI Technology.
- Cubre aprox. el 75 % del mercado mundial en microcontroladores RISC
- Costos actuales (a igualdad de prestaciones) inferior a 8051.

ARM

- En la actualidad, ARM Ltd no hace procesadores, solo los diseña y licencia sus diseños a fabricantes (P. ej: Analog Devices, Atmel, Cirrus Logic, Hyundai, Intel, Oki, Philips, Samsung, Sharp, Lucent, 3Comp, HP, IBM, Sony, etc.).

Características

- Computadora de 3 direcciones (registros) de 32 bits
- Single clock machines cycles
- Extensión Thumb
- Excepciones vectorizadas
- Número de transistores: > 74,209 implica bajo consumo.
- Frecuencias de operación: 45 – 133 MHz.
- Bus de 32 bits para datos e instrucciones.
- Elevado rendimiento: hasta 120 MIPS.
- Elevada densidad de código (Diseñado para trabajar en C)

TDII - Microcontroladores - ARM

3

Características

- Se basa en Arquitectura RISC.
- 37 registros de 32 bits (16 disponibles).
- Registros 0 a 7 disponibles en todo momento
- Memoria caché (dependiendo de la aplicación)
- Estructura del bus tipo Von Neuman (ARM7), tipo Harvard (ARM9)

TDII - Microcontroladores - ARM

4

ARM

- Tipos de datos de 8/16/32 bits
- 6 modos de operación: usr y sys, fiq, irq, svc, abt, sys, und.
- Todas las familias de procesadores ARM comparten el mismo conjunto de instrucciones

TDII - Microcontroladores - ARM

5

Risc

- Instrucciones de Procesamiento de datos
- Instrucciones de Transferencia de Datos
- Instrucciones de Control de Flujo

TDII - Microcontroladores - ARM

6

RISC

- Instrucciones conceptualmente simples.
- Transferencias Memoria/Registros exclusivamente LOAD/STORES.
- Las operaciones aritméticas son entre registros.
- Tamaño de instrucciones uniformes.
- Pocos formatos para las instrucciones.
- Conjunto de instrucciones ortogonal: poco o ningún traslape en la funcionalidad de las instrucciones.
- Pocos modos de direccionamiento.

TDII - Microcontroladores - ARM

7

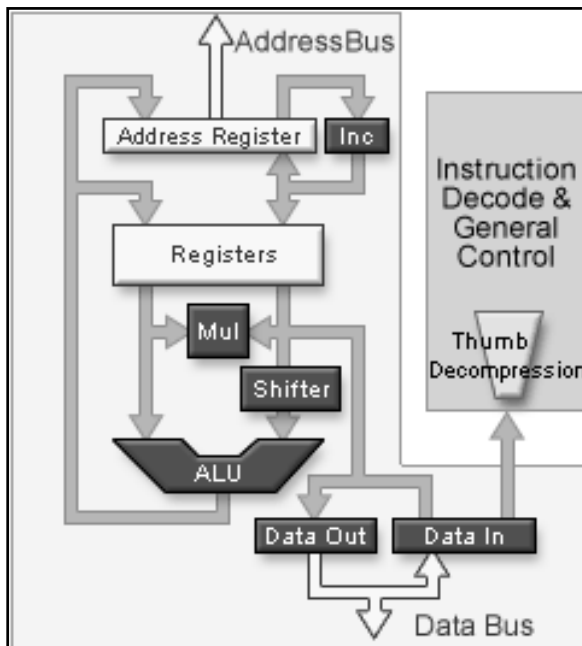
Risc

- Casi todas las instrucciones se ejecutan en un ciclo de reloj.
- Tendencia a tener un gran número de registros.
- Arquitectura RISC predomina en los procesadores de elevado rendimiento.

TDII - Microcontroladores - ARM

8

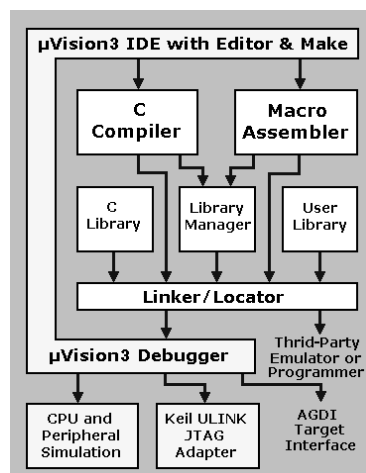
Arquitectura



TDII - Microcontroladores - ARM

9

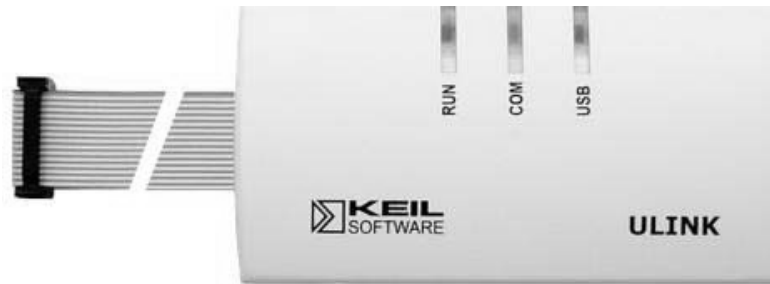
Ciclo de desarrollo



TDII - Microcontroladores - ARM

10

Herramientas - Jtag



TDII - Microcontroladores - ARM

11

ARM7

- Alimentación: 3.3 V y 5 V.
- Bajo consumo de potencia: 80 mW.
- Tecnología CMOS.
- Extensiones: Thumb, Jazelle.
- Los miembros de ARM7 tienen un coprocesador de interfaz que permite la conexión hasta con 16 coprocesadores más.

TDII - Microcontroladores - ARM

12

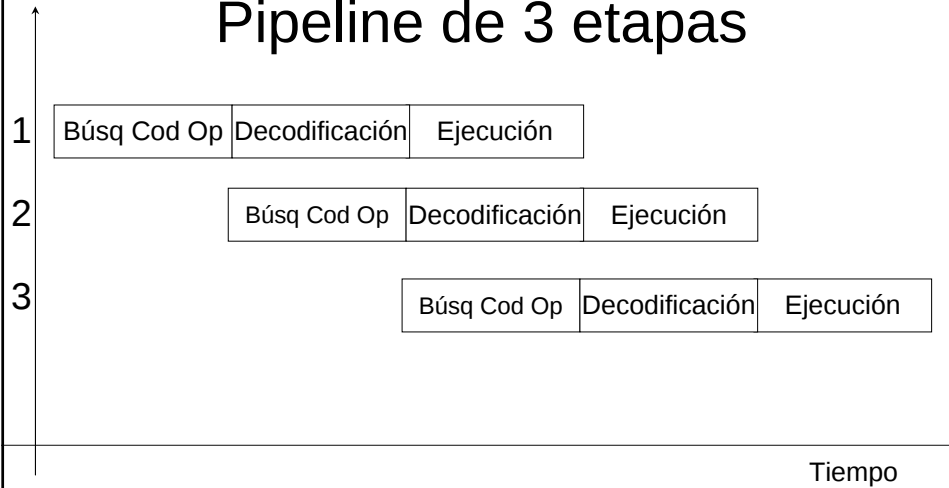
ARM7TDMI

- Es la versión mas utilizada de ARM7.
- ¿ TDMI ?
- T: “Thumb”, soporta esta extensión.
- D: “Debug-interface”.
- M: “Multiplier”, hardware multiplicador.
- I: “Interrupt”, interrupciones veloces.

ARM7TDMI

- Arquitectura de bus unificada.
- Lógica de depuración EmbeddedICE-RT.
- Interface ETM (Embedded Trace Macrocell).

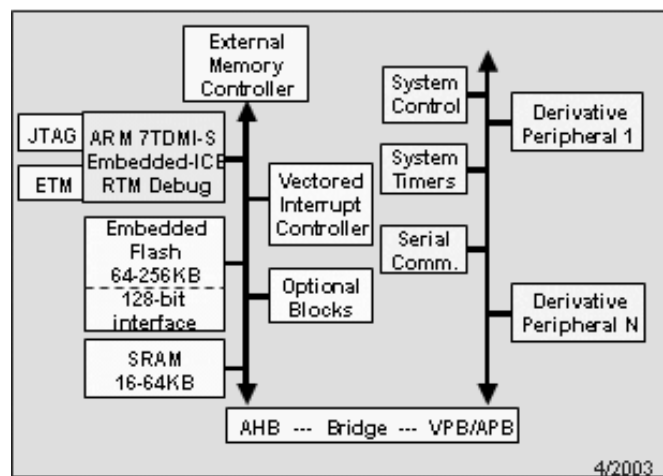
Pipeline de 3 etapas



TDII - Microcontroladores - ARM

15

Arquitectura ARM 7



TDII - Microcontroladores - ARM

16

Philips

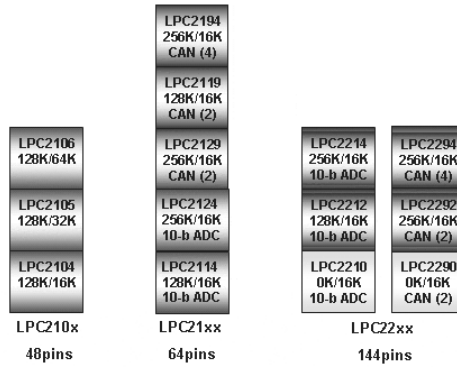
PHILIPS

Product Line Microcontrollers

LPC2000 overview

• released

• Q2/04

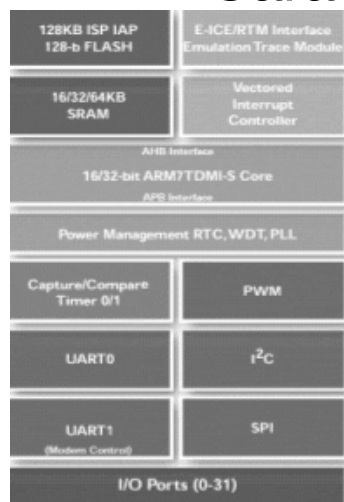


Semiconductors

TDII - Microcontroladores - ARM

17

Características



CAN
ADC
Ethernet
USB

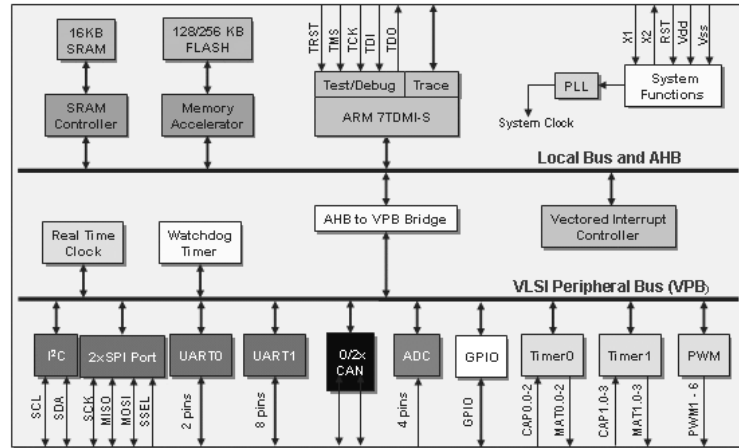
Soporte de depuración
Keil
ICE
JTAG

TDII - Microcontroladores - ARM

18

Philips

LPC2114/24/19/29 Block Diagram

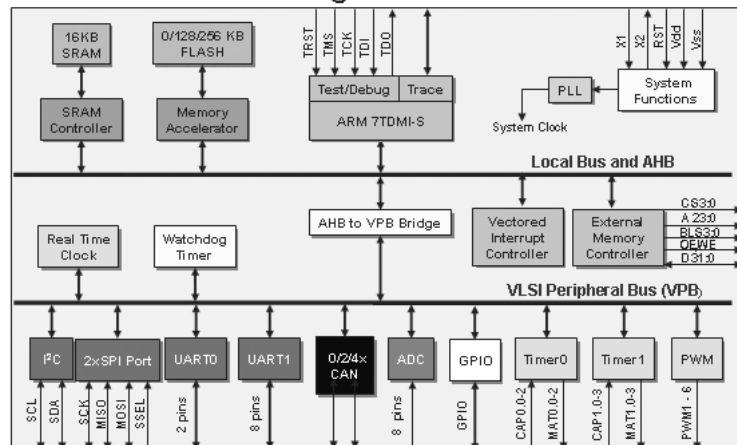


TDII - Microcontroladores - ARM

19

Philips

LPC22xx Block Diagram



TDII - Microcontroladores - ARM

20

LPC22xx

- Núcleo a 1,8 V
- E/S a 3,3 V compatible TTL
- Icc = 35 mA
- Idle = 20 mA
- Sleep = 25 µA

TDII - Microcontroladores - ARM

21



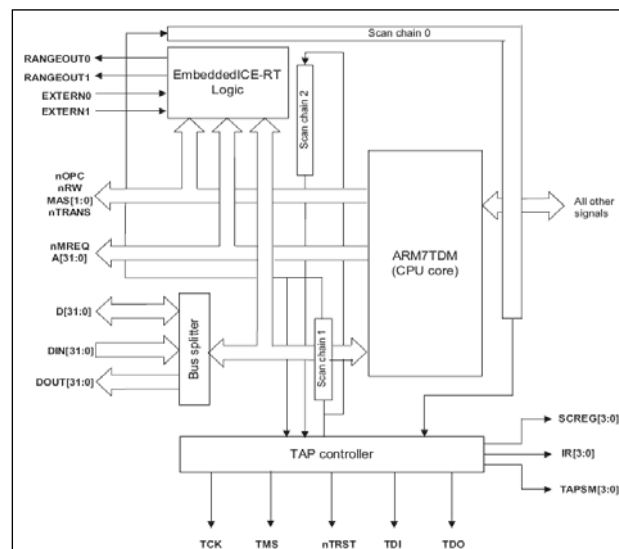
Dispositivos con ARM



TDII - Microcontroladores - ARM

23

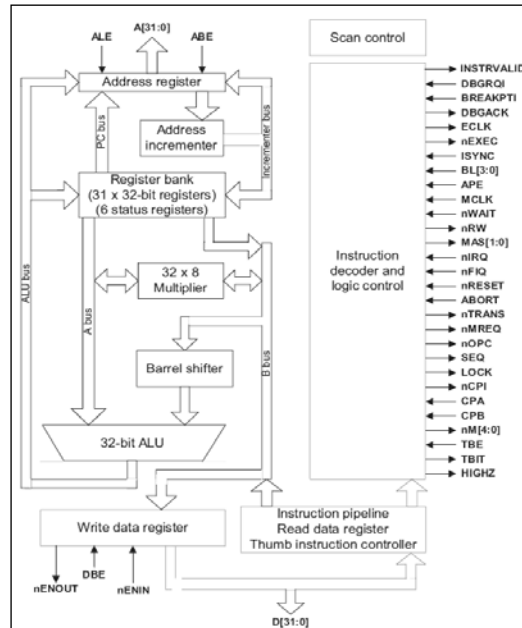
Diagrama en Bloques ARM7TDMI



TDII - Microcontroladores - ARM

24

Núcleo del ARM7TDM



TDII - Microcontroladores - ARM

25

Características

- 3 tipos de Interrupciones
 - FIQ
 - Vectorizadas
 - Interrupciones Generales
 - 32 entradas de interrupción
- PLL
 - Frec in = 10 – 25 MHz
 - Trabaja hasta 60 MHz

TDII - Microcontroladores - ARM

26

Comunicación Serie

- UART
 - 16550 compatible (con FIFO y flag de fifo llena).
 - Velocidad hasta 1/16 del clock
- SPI
 - Hasta 2 canales que admiten master-slave
 - Veloc = 1/8 clock

TDII - Microcontroladores - ARM

27

Comunicación Serie

- I2C
 - Hasta 750 kHz con 7 bits de direccionamiento
 - Bidireccional
 - Sin Maestro (multimaster)
- CAN

TDII - Microcontroladores - ARM

28

Timers

- De 32 bits con 4 registros de captura
- De 32 bits con 4 registros de coincidencia
- Watchdog que para debug no resetea al micro sino que genera excepciones

E/S

- 10 bits – 0 a 3 V y 400 kmuestras/s
- 4 – 8 canales
- PWM de 32 bits con timer específico
- RTC
- 32 bits de E/S

ARM7

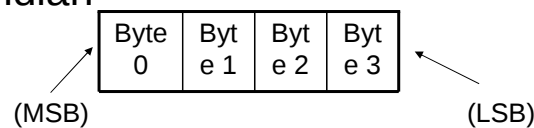
- Para sistemas que requieren manejo completo de memoria virtual y espacios de ejecución protegidos.
- Memoria caché de 8K
- MMU: unidad controladora de memoria.
- Para aplicaciones de plataforma abierta como Windows CE, Linux, Palm OS y Symbian OS.

TDII - Microcontroladores - ARM

31

Endian-ness

- Orden de los bytes en Little endian – Big endian

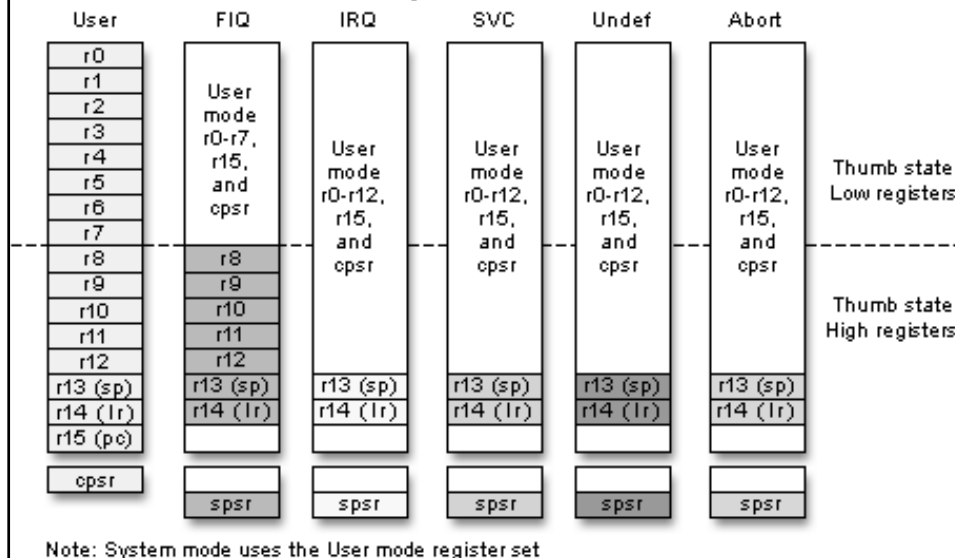


Dirección de Memoria	+0	+1	+2	+3	
Big Endian	Byte 0	Byte 1	Byte 2	Byte 3	MSB en la dirección más baja
Little Endian	Byte 3	Byte 2	Byte 1	Byte 0	LSB en la dirección más baja

TDII - Microcontroladores - ARM

32

Arquitectura



ARM7 Bancos de Registros

Modo Usuario						
	Modos Privilegiados					
		Modos de excepción				
User	System	Supervisor	Abort	Undefined	IRQ	FIQ
R0-R7	R0-R7	R0-R7	R0-R7	R0-R7	R0-R7	R0-R7
R8-R12	R8-R12	R8-R12	R8-R12	R8-R12	R8-R12	R8-R12
R13-R14	R13-R14	R13-R14	R13-R14	R13-R14	R13-R14	R13-R14
PC	PC	PC	PC	PC	PC	PC
CPSR	CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
		SPSR	SPSR	SPSR	SPSR	SPSR

ARM vs. Thumb

- **ARM**

- Instrucciones fijas de 32bit
- Instrucciones simples pueden realizar más funciones que una THUMB.
- Identical execution speed compared to THUMB from Flash/EE (CD > 0) and SRAM (CD >= 0)
- La tabla de vectores en ARM

- **THUMB**

- Instrucciones fijas de 16bits
 - Aumentan la densidad de código
 - Aumentan la velocidad de ejecución
- Repertorio de instrucciones sencillo
- Conmuta a modo ARM en cada excepción
- Acceso limitado al banco de registros.

ARM7 – Modos de operación

- User - *Modo de ejecución normal del programa*
- System - *Para ejecutar tareas de sistema operativo privilegiadas*
- Supervisor - *Modo protegido para el sistema operativo.*
- Abort
 - *Utilizado para implementar la protección de memoria y/o procesador*
 - *Dos clases de Abort. Abort de prefetch o de datos*
- Undefined - *Soporta emulación de software de instrucciones no soportadas o coprocesadores no implementados.*
- FIQ - *Manejo de la Fast interrupt*
- IRQ - *Manejo de interrupciones de propósito general*

ARM7 – Modelo de programación

- R15-R0 — *Dieciseis registros de propósito general*
- Funciones especiales
 - R15 es el Contador de Programa (PC)
 - *Si R15 es el operando destino, algunas instrucciones se comportarán particularmente en los cambios de modo.*
 - R14 es el Registro de vinculación (LR)
 - *Para llamadas a subrutina, interrupciones/excepciones, la direcciones de retorno es almacenada en LR. Debe salvarse antes de hacer llamadas en las subrutinas.*
 - R13 es utilizado como puntero a la pila (SP)

TDII - Microcontroladores - ARM

37

ARM7 – Modelo de programación

- Current Process Status Register (CPSR)
 - Flags del código de condición (N, Z, C, V)
 - Bits de deshabilitación de interrupción (I, F)
 - Habilidad del modo Thumb (T)
 - ¡Nunca debe cambiarse directamente!
 - Selección de Modo
 - No pueden cambiarse en modo usuario
 - Bits Reservados
 - No alterar para mantener compatibilidad con futuros productos

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
N	Z	C	V	reserved															I	F	T	mode									

TDII - Microcontroladores - ARM

38

ARM7 – Modelo de programación

- **Suspended Process Status Register (SPSR)**
 - SPSR esta sólo presente cuando el ARM esta trabajando en algunos de los modos de excepción
 - *Cada modo de excepción tiene su propio SPSR, ya que los exception handlers pueden causar otras excepciones.*
 - SPSR es una copia del CPSR inmediatamente antes de entrar al modo de excepción.
 - *Al retornar de la excepción, el valor en el SPSR es utilizado para restaurar el CPSR al valor adecuado para el proceso interrumpido por la excepción.*

TDII - Microcontroladores - ARM

39

ARM7 Direcciones reservadas

0x00000000	Reset
0x00000004	Undefined instruction exception
0x00000008	Software interrupt
0x0000000C	Prefetch abort exception
0x00000010	Data abort exception
0x00000014	Reserved
0x00000018	Interrupt request (IRQ)
0x0000001C	Fast interrupt request (FIQ)

TDII - Microcontroladores - ARM

40

ARM7 Reset

- CPSR
 - Modo supervisor
 - I & F set (interrupciones deshabilitadas)
 - T = 0 (ARM mode)
- PC se resetea a 0x00000000 y se busca una instrucción
 - Esta es la instrucción de *bootstrap*
- Todos los registros salvo PC y CPSR estan en un valor indeterminado.

TDII - Microcontroladores - ARM

41

ARM7 Coprocesadores

- La arquitectura ARM fue diseñada pensando en disponer de coprocesadores fuertemente asociados en el hardware (internos o externos)
 - Procesadores de control del sistema (CP15)
 - Coprocesadores de punto flotante
- Se utilizan instrucciones especiales para la lectura/escritura de los registros del coprocesador
 - Si el coprocsador no se halla presente, se genra una excepci3n que permite la emulaci3n por software de las funciones del coprocesador.

TDII - Microcontroladores - ARM

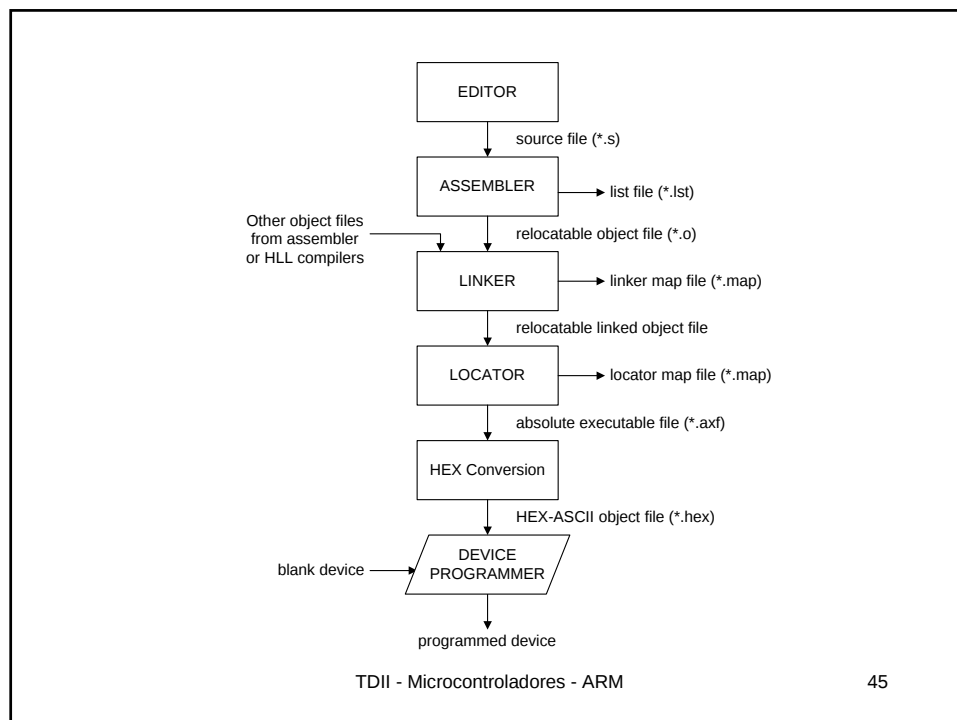
42

ARM7 Códigos de condición

- Cuando utilizamos códigos de condición:
 - Aritmética signada y no signada, requieren diferentes condiciones para obtener el mismo resultado
 - No-signado HS equivale a GE signado
 - La condición AL (always) es la utilizada por omisión cuando no se utilizan nemónicos de condición.
 - El significado de la condición es ***cierto si op1 cond op2***
 - CMP R0, R1
 - BLLT subrutina
 - El salto se producirá si R0 es menor que R1 (signado)

Instrucciones ARM7

- 11 tipos básicos de instrucciones.
- Dos de estos tipos emplean la ALU, el desplazador en barril y el multiplicador para ejecutar operaciones a alta velocidad sobre datos en los registros.
- Ejemplos: AND, EOR, SUB, RSB, ADD, ADC, SBC, RSC, TST, TEQ, CMP, CMN, ORR, MOV, BIC, MVN, (Multiplicaciones) MUL, MLA, MULL, MLAL



45

Formato .HEX

```

:0400000500080000EF
:020000040008F2
:1000000018F09FE518009FE5001090E50114A0E1AD
:100010000110E0E1001080E50000A0E1F8FFFFEA38
:0800200004000800001000803C
:00000001FF
  
```

0x00080000 E59FF018

Record Mark	Record Length	Load Dirección or	Record Type	Data	Checksum	Record Description
:	04	0000	05	00080000	EF	Start linear Dirección record
:	02	0000	04	0008	F2	Extended linear Dirección record
:	10	0000	00	18F09FE518009FE5001090E50114A0E1	AD	Data record
:	10	0010	00	0110E0E1001080E50000A0E1F8FFFFEA	38	Data record
:	08	0020	00	0400080000100080	3C	Data record
:	00	0000	01		FF	End-of-file record

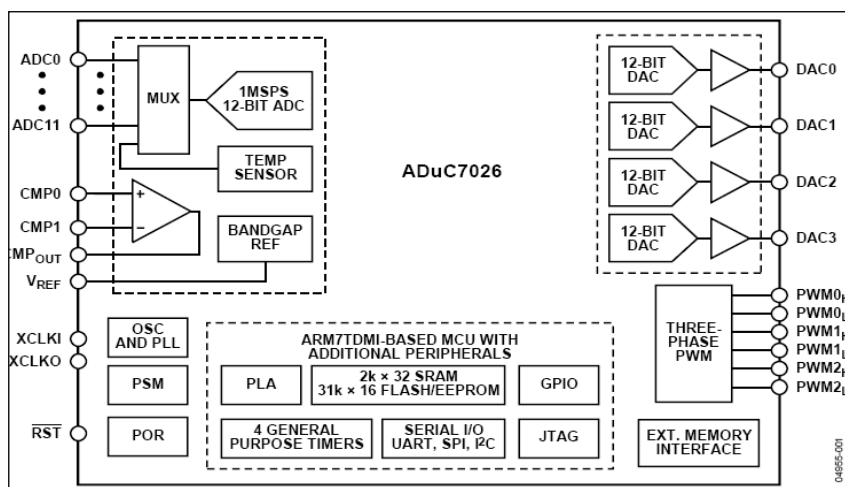
Analog Devices ADuC7026

- Núcleo ARM7TDMI
 - 62kB flash, 8kB SRAM
 - Programación en circuito, Depuración con JTAG
 - 41.78MHz PLL con divisor programable
 - Little-endian
- Múltiples periféricos digitales
 - GPIO
 - Timers (4 de propósito general y watchdog/wake-up)
 - Interfaces serie UART(16550)/I²C/SPI
 - PWM de tres fases
 - Interfaz a memoria externa (Multiplexada a 16-bits)
- Entradas/Salidas Analógicas
 - 12 entradas, 4 salidas
 - Referencia de tensión y medición de temperatura

TDII - Microcontroladores - ARM

47

Analog Devices ADuC7026



TDII - Microcontroladores - ARM

48

Mapa de Memoria

0xFFFFFFF	MMRs
0xFFFF0000	RESERVED
0x40000FFF	EXTERNAL MEMORY REGION 3
0x40000000	RESERVED
0x30000FFF	EXTERNAL MEMORY REGION 2
0x30000000	RESERVED
0x20000FFF	EXTERNAL MEMORY REGION 1
0x20000000	RESERVED
0x10000FFF	EXTERNAL MEMORY REGION 0
0x10000000	RESERVED
0x00080FFF	FLASH/EE
0x00080000	RESERVED
0x00011FFF	SRAM
0x00010000	REMAPPABLE MEMORY SPACE (FLASH/EE OR SRAM)
0x00000FFF	
0x00000000	

TDII - Microcontroladores - ARM

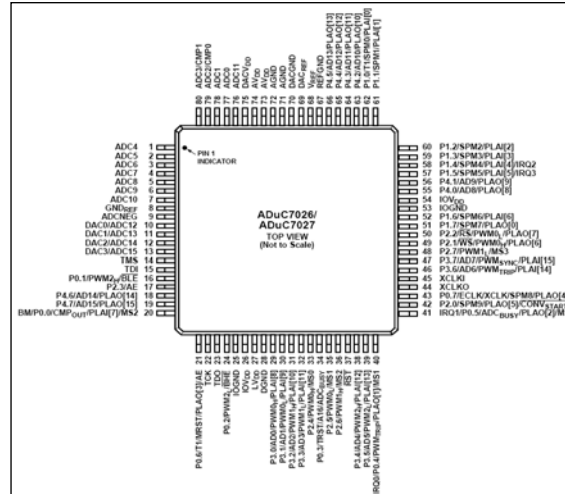
49

Mapa de Memoria

0xFFFFFFF	
0xFFFFC3C	PWM
0xFFFFC00	
0xFFFF820	FLASH CONTROL INTERFACE
0xFFFF800	
0xFFFF40C	GPIO
0xFFFF400	
0xFFFF0E4	PLA
0xFFFF000	
0xFFFF0A14	SPI
0xFFFF0A00	
0xFFFF0940	IC1
0xFFFF0900	
0xFFFF0840	IC0
0xFFFF0800	
0xFFFF0720	UART
0xFFFF0700	
0xFFFF0620	DAC
0xFFFF0600	
0xFFFF0538	ADC
0xFFFF0500	
0xFFFF0420	BAND GAP REFERENCE
0xFFFF040C	
0xFFFF0440	POWER SUPPLY MONITOR
0xFFFF0440	
0xFFFF0420	PLL AND OSCILLATOR CONTROL
0xFFFF0404	
0xFFFF0370	WATCHDOG TIMER
0xFFFF0360	
0xFFFF0350	WAKE UP TIMER
0xFFFF0340	
0xFFFF0234	GENERAL PURPOSE TIMER
0xFFFF0220	
0xFFFF0210	TIMER 0
0xFFFF0200	
0xFFFF0238	REMAP AND SYSTEM CONTROL
0xFFFF0220	
0xFFFF0110	INTERRUPT CONTROLLER
0xFFFF0000	

TDII - Microcontroladores - ARM

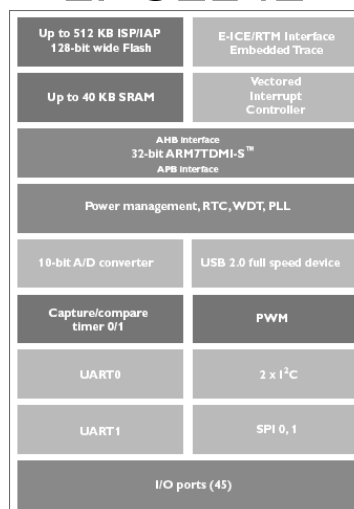
ADuC7026 Pin-Out (LQFP-80)



TDII - Microcontroladores - ARM

51

LPC2142



TDII - Microcontroladores - ARM

52

LPC2142

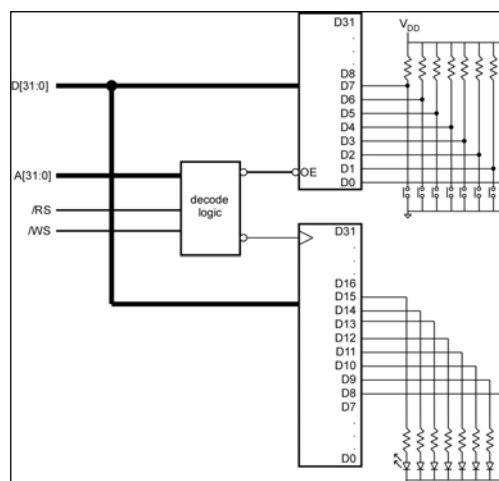
Product	Flash	RAM	ADC	DAC	Packages
LPC2141	32KB	8KB	1 (6 Channels)	-	LPFP64
LPC2142	64KB	16KB	1 (6 Channels)	1	LPFP64
LPC2144	128KB	16KB	2 (14 Channels)	1	LPFP64
LPC2146	256KB	32KB + 8KB ¹	2 (14 Channels)	1	LPFP64
LPC2148	512KB	32KB + 8KB ¹	2 (14 Channels)	1	LPFP64

	LPC2141	✓	Microcontroller with USB 2.0 full-speed device, 32KB ISP/IAP flash, 10-bit ADC
	LPC2142	✓	Microcontroller with USB 2.0 full-speed device, 64KB ISP/IAP flash, 10-bit ADC, DAC
	LPC2144	✓	Microcontroller with USB 2.0 full-speed device, 128KB ISP/IAP flash, 2x10-bit ADC, DAC
	LPC2146	✓	Microcontroller with USB 2.0 full-speed device, 256KB ISP/IAP flash, 2x10-bit ADC, DAC
	LPC2148	✓	Microcontroller with USB 2.0 full-speed device, 512KB ISP/IAP flash, 2x10-bit ADC, DAC

TDII - Microcontroladores - ARM

53

Ejemplo



TDII - Microcontroladores - ARM

54

Fuente en assembler

```
; ;** NO ** usar como base de un programa!

EXPORT Reset_Handler ;Hacerlo visible al linker

IOPORT_Dirección EQU 0x80001000

AREA Reset, CODE, READONLY
ARM

Reset_Handler
LDR PC, aLoop ;jump absoluto
Loop
LDR R0, aPort ;Cargar la dirección de un puerto
LDR R1, [R0] ;Leer el estado de un interruptor
LSL R1, #8 ;Rotar a la izquierda para alinear
MVN R1, R1 ;complementar
STR R1, [R0] ;Escribir LEDs
NOP ;Demora
B Loop ;repetir indefinidamente

aLoop DCD Loop
aPort DCD IOPORT_Dirección

END
```

TDII - Microcontroladores - ARM

55

ARM7 – Instrucciones

		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Multiply (accumulate)	cond	0	0	0	0	0	0	0	0	A	S	Rd			Rn			Rs			1	0	0	1	Rm									
Multiply (accumulate) long	cond	0	0	0	0	0	1	U	A	S	Rd_MSW			Rd_LSW			Rn			1	0	0	1	Rm										
Salto and exchange	cond	0	0	0	1	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	Rn					
Single data swap	cond	0	0	0	1	0	B	0	0	Rn			Rd			0	0	0	0	1	0	0	1	Rm										
Halfword data transfer, registro de offset	cond	0	0	0	P	U	0	W	L	Rn			Rd			0	0	0	0	1	0	1	1	Rm										
Halfword data transfer, immediate offset	cond	0	0	0	P	U	1	W	L	Rn			Rd			offset			1	0	1	1	offset											
Signed data transfer (byte/halfword)	cond	0	0	0	P	U	B	W	L	Rn			Rd			addr_mode			1	1	H	1	addr_mode											
Data processing and PSR transfer	cond	0	0	1	opcode			S	Rn			Rd			operand2																			
Load/store register/unsigned byte	cond	0	1	I	P	U	B	W	L	Rn			Rd			addr_mode																		
Undefined	cond	0	1	1																									1					
Block data transfer	cond	1	0	0	P	U	0	W	L	Rn			register list																					
Salto	cond	1	0	1	L	offset																												
Coprocessor data transfer	cond	1	1	0	P	U	N	W	L	Rn			CRd			CP#			offset															
Coprocessor data operation	cond	1	1	1	0	CP opcode			CRn			CRd			CP#			CP	0	CRm														
Coprocessor register transfer	cond	1	1	1	0	CP opc			CRn			Rd			CP#			CP	1	CRm														
Software interrupt	cond	1	1	1	1	ignored by processor																												

TDII - Microcontroladores - ARM

56

ARM7 Códigos de condición

Opcode [31:28]	Mnemonic extension	Meaning	Condition flag state
0000	EQ	Equal	Z==1
0001	NE	Not equal	Z==0
0010	CS/HS	Carry set / unsigned higher or same	C==1
0011	CC/LO	Carry clear / unsigned lower	C==0
0100	MI	Minus / negative	N==1
0101	PL	Plus / positive or zero	N==0
0110	VS	Overflow	V==1
0111	VC	No overflow	V==0
1000	HI	Unsigned higher	(C==1) AND (Z==0)
1001	LS	Unsigned lower or same	(C==0) OR (Z==1)
1010	GE	Signed greater than or equal	N == V
1011	LT	Signed less than	N != V
1100	GT	Signed greater than	(Z==0) AND (N!=V)
1101	LE	Signed less than or equal	(Z==1) OR (N!=V)
1110	AL	Always (unconditional)	Not applicable
1111	(NV)	Never	Obsolete, ARM7TDMI unpredictable

Instrucciones de Movimiento de Datos

Instrucción MOV

- **Sintaxis**
 - MOV{<cond>}{S} <Rd>, <shifter_operand>
- **RTL**
 - if (cond es verdad)
 - $Rd \leftarrow \text{shifter_operand}$
 - if((S==1) AND (Rd==R15))
 - CPSR $\leftarrow$ SPSR
- **Flags (Si Rd no es R15)**
 - **N, Z, C** (*C basado en el operando shifteado*)

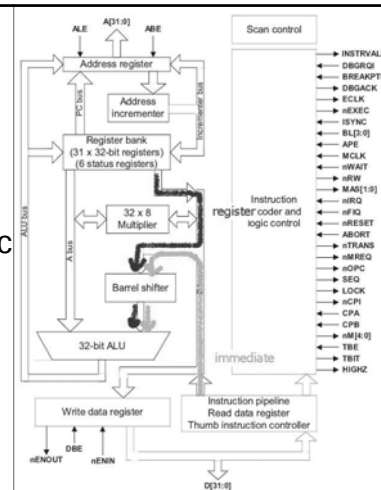
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
cond				0	0	I	1	1	0	1	S	SBZ				Rd				shifter operand																

TDII - Microcontroladores - ARM

59

Operandos desplaz

- **Inmediato (I=1)**
 - Valor de 8-bits, 4-bit cod rotac
 - Dado por #<número>
- **Registro**
 - Sin desplazamiento
- **Registro desplazado**
 - LSL, LSR, ASR, ROR
 - Valor de desplazamiento inmediato o registro
- **Rotación derecha con extensión**
 - Rotación de registro a derecha 1 bit a través del CY



TDII - Microcontroladores - ARM

60

Ejecución condicional y flags

- Si una condición es utilizada, la instrucción se ejecutará solamente si la condición es verdadera.
- Los Flags sólo se actualizan si se usa S (Intel/Motorola)
 - Algunas instrucciones (B, BL, CMP, CMN, TST, TEQ, etc.) no utilizan S
- Utilizando Flags y condiciones
 - MOVS R1, R1
 - MOV R0, #1
 - MOVEQ R0, #0
 - MOVMI R0, #-1

TDII - Microcontroladores - ARM

61

Casos especiales de MOV

- Si R15 es el destino de una instrucción MOV y se agrega S:
 - **MOVS R15, Rxx**
 - En adición al movimiento al PC, el CPSR es cargado del SPSR del modo de excepción actual. Los flags no se ven afectados.
 - Este comportamiento solo será utilizado en el retorno de modos de excepción
 - NO realizar esta acción en los modos usuario o system pues no hay SPSR y los resultados son impredecibles.

TDII - Microcontroladores - ARM

62

Casos especiales de MOV

- MVN – move negado
 - El operando de desplazamiento es complementado (complemento a 1)
- MRS – move CPSR/SPSR a registro GP
- MSR – move registro GP al CPSR/SPSR
- SWP – swap
- SWPB – swap byte
 - La instrucción Swap intercambia valores entre memoria y registros en una operación atómica

TDII - Microcontroladores - ARM

63

Instrucciones de Procesamiento

TDII - Microcontroladores - ARM

64

Instrucciones de procesamiento de datos

- Todas las instrucciones de procesamiento de datos utilizan una estructura similar para los operandos incluyendo el operando desplazador
 - *Nemónico{cond}{S}* Rd, Rn, <shifter_operand>
 - En general, $Rd \leftarrow Rn \text{ operación shifter_operand}$
 - Las instrucciones NO destructivas no utilizarán un registro destino (Rd).
 - Algunas instrucciones invierten el orden de los operandos
- Ya que los desplazamientos pueden ser parte de una instrucción de MOV o de procesamiento de datos, no hay instrucciones dedicadas de desplazamiento (shifts).

TDII - Microcontroladores - ARM

65

Procesamiento de datos - Lógicas

- AND – bit a bit AND lógica
- BIC – bit clear (AND con complemento del operando de desplazamiento)
- EOR – bit a bit OR exclusiva
- ORR – bit a bit OR inclusiva
- TEQ – test equivalence (XOR no destructiva)
- TST – test (AND no destructiva)

TDII - Microcontroladores - ARM

66

Procesamiento de datos - Aritméticas

- ADC – add with carry
- ADD – add
- CMN – comparación negativa
- CMP – comparación
- RSB – Resta inversa (reverse subtract)
- RSC – Resta inversa con carry
- SBC – Resta con carry
- SUB - Resta

TDII - Microcontroladores - ARM

67

Multiplicar y Acumular

- Multiplicar 32x32 – Resultado 32-bit
 - MLA – multiplicar-acumular
 - MUL – multiplicar
- Multiplicar 32x32 – Resultado 64-bit
 - SMLAL – Multiplicar y Acumular signada larga
 - SMULL – Multiplicación signada larga
 - UMLAL – Multiplicar y Acumular NO signada larga
 - UMULL – Multiplicación NO signada larga

TDII - Microcontroladores - ARM

68

Modelos de direccionamiento

Modelos de direccionamiento a memoria

- Direccionamiento de memoria Lineal
 - La instrucción especifica la Dirección completo.
- Direccionamiento de memoria segmentado
 - Las instrucciones no tienen la totalidad de la dirección. Solo una parte (el *offset*)
 - El resto de la dirección esta dada por un **registro de página** o un **registro de segmento**.
 - Pueden haber múltiples registros de segmento
 - La **dirección física** esta formada por la combinación del registro de segmento/página y el offset de la instrucción
 - Ventajas / Desventajas

Modos de direccionamiento de memoria

- **Direccionamiento Directo**
 - La dirección del operando esta contenida en la instrucción.
 - En las instrucciones de longitud variable, suele codificarse la totalidad de la dirección física.
 - En las instrucciones de longitud fija, solo suele codificarse la parte menos significativa de la dirección.
 - El resto de la dirección puede ponerse en 0 (direccionamiento a página base)
 - El resto de la dirección puede obtenerse de un registro de página o de segmento.

TDII - Microcontroladores - ARM

71

Modos de direccionamiento de memoria

- **Direccionamiento Indirecto por Registro**
 - La instrucción especifica un registro que contiene la dirección de memoria para obtener la dirección del operando.
 - Puede contener la actualización del registro (auto-incremento, auto-decremento, etc.)
- **Direccionamiento indirecto a memoria**
 - Una posición de memoria (codificada dentro de la instrucción) contiene la dirección del operando

TDII - Microcontroladores - ARM

72

Modos de direccionamiento de memoria

- Direccionamiento indexado
 - La dirección física se calcula con una dirección de inicio fija y el contenido de un registro.
 - Típicamente se emplea para acceder datos en arrays
 - Dirección de Base = Dirección de inicio del array
 - Registro contiene (índice al elemento × tamaño del elemento)
 - Si es un arreglo de bytes, tamaño del elemento = 1
 - Si es un arreglo de media palabra, tamaño del elemento = 2
 - Si es un arreglo de palabras, tamaño del elemento = 4

TDII - Microcontroladores - ARM

73

Modos de direccionamiento de memoria

- Direccionamiento Basado
 - La dirección física se calcula con una dirección base contenida en un registro más un offset constante codificado dentro de la Instrucción
 - Se utiliza habitualmente para acceder a estructura de datos.
 - Un registro contiene la dirección de inicio de la estructura.
 - El Offset es la distancia desde el inicio de la estructura al elemento deseado de la estructura
 - El programa puede acceder a cualquier instancia de la estructura solo cambiando el contenido del registro que lo apunta.

TDII - Microcontroladores - ARM

74

Modos de direccionamiento de memoria

- Direccionamiento relativo al PC
 - La dirección se calcula sumando un offset codificado dentro de la instrucción al valor actual del PC.
 - En muchos microprocesadores, el PC no forma parte del modelo de programación, por lo que el direccionamiento relativo al PC se considera distinto al direccionamiento basado o indexado.
 - ARM utilizará el direccionamiento relativo al PC para implementar un direccionamiento similar al direccionamiento directo.
 - La distancia de la instrucción a la etiqueta debe ser conocida en el momento de ensamblado.

TDII - Microcontroladores - ARM

75

Resumen Modos de direccionamiento de memoria

- TODOS los modos de direccionamiento de memoria del ARM7 utilizan un registro base
 - También pueden utilizar un offset constante u otro registro como offset
 - El segundo registro puede ser desplazado (Desplazado)
- El equivalente del direccionamiento directo se obtiene utilizando el PC

TDII - Microcontroladores - ARM

76

Instrucciones Load/Store

- LDR/STR
 - Carga y almacena registros de 32-bits
 - Dirección alineada por palabra
- LDRB/STRB
 - Carga y almacena un byte no signado
 - En la carga el valor esta relleno con ceros hasta los 32-bits

TDII - Microcontroladores - ARM

77

Load/Store - Direccionamiento

- Registro Base +/- offset inmediato
 - Dirección = (Rn) +/- offset_12
 - Sintaxis: [Rn, +/- #<offset>]
- Registro base +/- registro de offset
 - Dirección = (Rn) +/- (Rm)
 - Sintaxis: [Rn, +/- Rm]
- Registro base +/- registro de offset Desplazado
 - Dirección = (Rn) +/- (Desplazado Rm)
 - Modos de desplazamiento con cuenta inmediata
 - LSL, LSR, ASR, ROR, RRX
 - Sintaxis: [Rn, +/- Rm, shift_mode #count]
- Rn no es afectado por estos modos de direccionamiento

TDII - Microcontroladores - ARM

78

ARM7 Load/Store - Direcccionamiento

- Pre-indexado
 - Rn es actualizado con la dirección calculada
 - Sintaxis: [Rn, +/-#<offset>]!
 - Sintaxis: [Rn, +/-Rm]!
 - Sintaxis: [Rn, +/-Rm, shift_mode #count]!
- Post-indexado
 - Rn es usado como la dirección de la transferencia. Luego, Rn es actualizado como la dirección calculada
 - Sintaxis: [Rn], +/-#<offset>
 - Sintaxis: [Rn], +/-Rm
 - Sintaxis: [Rn], +/-Rm, shift_mode #count

TDII - Microcontroladores - ARM

79

Instrucciones Load/Store

- LDRSB/LDRSH
 - Carga un byte/mediapalabra signado de memoria
 - Byte/mediapalabra es extendido a 32 bits (con signo)
- LDRH/STRH
 - Carga y almacena una media palabra NO signada
 - En la carga el valor es rellenado con ceros para llegar a los 32 bits
- Los modos de direccionamiento son similares a LDR/STR, pero mas restringidos
 - Base +/- offset_8
 - Base +/- registro
 - Pre-indexado and post-indexado

TDII - Microcontroladores - ARM

80

Múltiples Load/Store

- LDM/STM carga y almacena múltiples registros en una sola instrucción
 - Sintaxis: LDM{<cond>}<Modo de direccionamiento>, <Rn>{!}, <registros>
 - Modos de direccionamiento
 - IA – Incrementa en 4 Luego de la transferencia
 - IB – Incrementa en 4 Antes de la transferencia
 - DA – Decrementa en 4 Luego de la transferencia
 - DB – Decrementa en 4 Antes de la transferencia
 - Write-back de registros controlado por “!”
 - Registros son siempre escritos/leídos de memoria con el registro de menor número en la dirección más baja

TDII - Microcontroladores - ARM

81

ARM7 Pseudo-Instrucciones de memoria

- Direccionamiento Directo (i.e. LDR R0, my_label)
 - Codificada como LDR R0, [PC, #offset]
- ADR (i.e. ADR R0, my_label)
 - Codificada como ADD/SUB R0, PC, #number
- LDR – Load register
 - LDR R0, =(expresión)
 - Si la expresión es legal inmediata, se codifica como MOV/MVN
 - De otra manera, emplea una palabra para almacenar la expresión y carga la palabra utilizando MOV con direccionamiento relativo al PC.
 - LDR R0, =(label)
 - Emplea una palabra para almacenar la dirección de la etiqueta, emplea una palabra para almacenar la etiqueta y carga la palabra utilizando MOV con direccionamiento relativo al PC.

TDII - Microcontroladores - ARM

82

Asignación de Memoria

TDII - Microcontroladores - ARM

83

Asignación de Memoria

- Operandos en memoria
 - Almacenados en formato little-endian
- Directivas de asignación de datos
 - DCB, DCW, DCD, SPACE, ALIGN
 - Identificadores e inicializadores
 - Constantes vs. variables
 - Arrays y strings
- Inicializando el área de datos
 - AREAs de Read-write están ubicadas en SRAM

TDII - Microcontroladores - ARM

84

Saltos

- Los saltos redirigen el flujo de programa recargando el PC
 - Un Salto sin retorno es un *salto ó jump*
- ARM7TDMI Instrucciones de Salto
 - **B destino**
 - Destino es una etiqueta
 - La instrucción es codificada por medio de un offset de 24-bit signado
 - **Bx <Rm>**
 - Destino es un registro
 - Puede ser usado para saltar a código Thumb
- Tablas de saltos
 - ¡ Comprobar que el índice esta verificado en su rango!

TDII - Microcontroladores - ARM

Saltos Condicionales y Looping

- Los Saltos Condicionales permiten al programa hacer lógica en respuesta a condiciones
 - La acción esta basada en el estado actual de los flags
 - Previamente se deben utilizar instrucciones para adecuar los flags
- Pueden emplearse saltos hacia atras para crear lazos que pueden terminarse-
 - Por una condición
 - Luego de un determiando número de iteraciones
 - Combinación de ambos
- Ejemplos de lazos
 - Incrementantes
 - Decrementantes

TDII - Microcontroladores - ARM

86

Tablas de Saltos

```
start
    MOV R0, ???      ; Índice en R0
    MOV R1, jmpTbl ; Obtener la dirección base
    LDR R0, [R1,R0,LSL #2] ; look-up
    BX  R0           ; Salto al destino
;
task0
    NOP
    B   start
;
task1
    NOP
    B   start
;
jmpTbl DCD task0, task1; ...
```

TDII - Microcontroladores - ARM

87

Construcciones de Programación estructurada

- Conceptos básicos
 - Un punto de entrada y uno de salida por cada subrutina
 - Basada en tres estructuras básicas de control
 - Secuencia
 - Selección
 - If, If/Else, Switch/Case
 - Repetición
 - While
 - Do-While

TDII - Microcontroladores - ARM

88

Pila

- Pila = LIFO
- Dos operaciones básicas
 - PUSH
 - POP (PULL)
- Pilas en Hardware vs. pilas en memoria

TDII - Microcontroladores - ARM

89

ARM7TDMI Operación de pila

- Por convención el Puntero a la pila es R13
- ARM terminología de la pila
 - Empty versus full
 - Ascendente versus descendente
- Asignando espacio de pila
 - Inicialización del SP
- Operaciones de pila
 - LDR/STR
 - LDM/STM
 - PUSH/POP

TDII - Microcontroladores - ARM

90

B

- Sintaxis
 - $B\{<cond>\} <target_address>$
- RTL
 - if (cond es cierta)
 - $PC \leftarrow PC + offset$
- No se afectan los flags

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cond				1	0	1	0	Inmediato_signado_24																							



BL

- Sintaxis
 - $BL\{<cond>\} <target_address>$
- RTL
 - if (cond es cierta)
 - $R14 \leftarrow$ Dirección de la próxima instrucción
 - $PC \leftarrow PC + offset$
- No afecta Flags

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cond				1	0	1	1	Inmediato_signado_24																							



BX

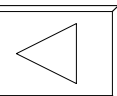
- Sintaxis
 - $BX\{<cond>\} <Rm>$
- RTL
 - if (cond es cierta)
 - $T\text{ flag} \leftarrow Rm[0]$
 - $PC \leftarrow Rm \& 0xFFFFFFFF$
- No afecta los flags
- En ARM v5 y posteriores existe una instrucción BLX.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cond				0	0	0	1	0	0	1	0	SBO			SBO			SBO			0	0	0	1	Rm						



Fragmento desensamblado

```
57:      ADR    R2, sqrt_LUT
0x00080104 E24F2024 SUB    R2,PC,#0x00000024
58:      LDR    R1, sqrt_MASK
0x00080108 E51F1020 LDR    R1,[PC,#-0x0020]
```



Excepciones

- Dir Prox Instrucción → LR
 - Si la excepción ocurre en estado ARM, PC+4 ó PC+8
 - Si la excepción ocurre en Thumb PC+2 ó PC+4
- Copia CPSR al correspondiente SPSR
- Fuerza los bits de modo del CPSR (según la excepción)
- Fuerza a buscar con el PC la instrucción según el vector

TDII - Microcontroladores - ARM

95

Retorno de excepciones

• PC ← LR

Exception	Saved LR value		Recommended return instruction	Return point
	ARM	Thumb		
Reset	-	-	-	After Reset, r14_svc value is Unpredictable.
Data Abort	PC + 8	PC + 8	SUBS PC, R14_abt, #8	Returns to aborted instruction.
FIQ	PC + 4	PC + 4	SUBS PC, R14_fiq, #4	Returns to interrupted instruction.
IRQ	PC + 4	PC + 4	SUBS PC, R14_irq, #4	Returns to interrupted instruction.
Prefetch Abort	PC + 4	PC + 4	SUBS PC, R14_abt, #4	Returns to aborted instruction.
Undefined instruction	PC + 4	PC + 2	MOVS PC, R14_und	Returns to instruction after Undefined instruction.
SWI instruction	PC + 4	PC + 2	MOVS PC, R14_svc	Returns to instruction after SWI instruction.

- CPSR ← SPSR
- Limpia Interrupt Disable

TDII - Microcontroladores - ARM

96

Vectores de excepción

Address	Exception	Mode on entry	I state on entry	F state on entry
0x00000000	Reset	Supervisor	Set	Set
0x00000004	Undefined Instruction	Undefined	Set	Unchanged
0x00000008	SWI	Supervisor	Set	Unchanged
0x0000000C	Prefetch Abort	Abort	Set	Unchanged
0x00000010	Data Abort	Abort	Set	Unchanged
0x00000014	Reserved	Reserved	-	-
0x00000018	IRQ	IRQ	Set	Unchanged
0x0000001C	FIQ	FIQ	Set	Set

TDII - Microcontroladores - ARM

97

Prioridades de las excepciones

Priority	Exception
Highest	Reset
	Data Abort
	FIQ
	IRQ
	Prefetch Abort
Lowest	Undefined Instruction and SWI

TDII - Microcontroladores - ARM

98

Resumen instrucciones

Ejemplos

TDII - Microcontroladores - ARM

99

Repertorio de instrucciones

- Comparación
 - CMP r1,r2
 - CMN r1,r2
 - TST r1,r2
 - TEQ r1,r2

TDII - Microcontroladores - ARM

100

Instrucciones de Movimiento

- MOV r0,r2
- MVN r0,r2 ; r0:= not r2
- LDR r0,[r1]
- STR r0,[r1]
- LDR r0,[r1], #4
- LDR r0,[r1,#4]!

TDII - Microcontroladores - ARM

101

Aritméticas

- ADD r1, r2, r3 ; r1 = r2 + r3
- ADC r1, r2, r3 ; r1 = r2 + r3 + C
- SUB r1, r2, r3 ; r1 = r2 - r3
- SUBC r1, r2, r3 ; r1 = r2 - r3 + C - 1
- RSB r1, r2, r3 ; r1 = r3 - r2
- RSC r1, r2, r3 ; r1 = r3 - r2 + C - 1

TDII - Microcontroladores - ARM

102