

ARM7TDMI结构简介

◆北京微芯力科技有限公司

主要内容

- ❖ ARM7TDMI介绍
- ❖ ARM7寄存器
- ❖ ARM7指令系统
- ❖ MMU与协处理器
- ❖ 调试接口

ARM7TDMI介绍

❖ 指令流水线：ARM7TDMI使用3级流水线：

❖ 取指

❖ 译码

❖ 执行

❖ 存储器访问：

ARM7TDMI核是冯·诺依曼体系结构，使用单一32位数据总线传送指令和数据。

ARM7TDMI介绍

❖ 存储器接口:

接口被设计成在使用存储器最少的情况下实现其潜能。

❖ 嵌入式ICE-RT逻辑:

为ARM7TDMI核提供了集成的在片调试支持。

ARM7TDMI 体系结构

ARM7TDMI有如下两个指令集：

- 32位ARM指令集
- 16位Thumb指令集

1 指令集压缩

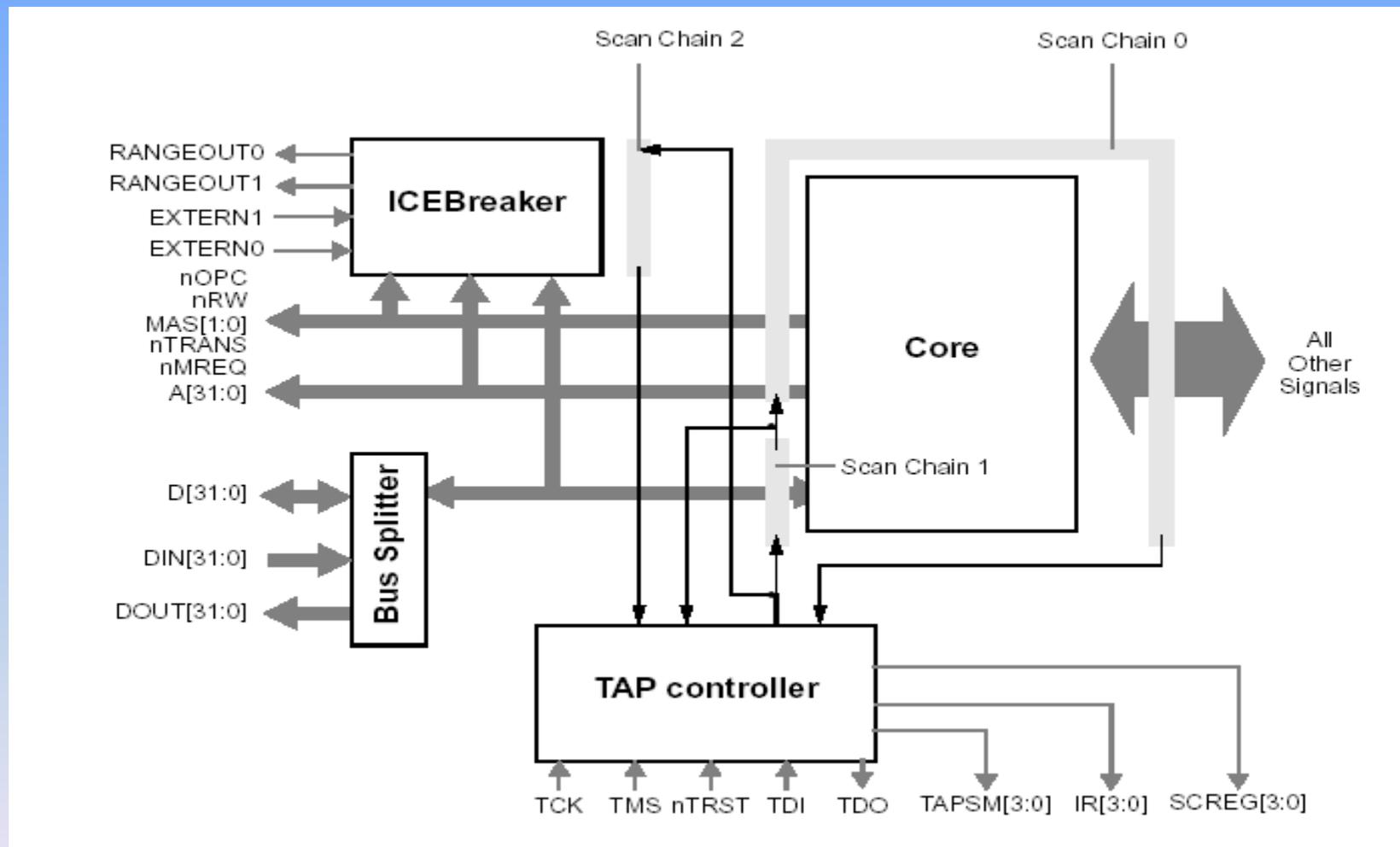
- 比16位体系结构更高的性能；
- 比32位体系结构更高的代码密度。

ARM7TDMI 体系结构

2 Thumb指令集

- 32位寻址空间；
- 32位寄存器；
- 32位移位器和算术逻辑单元ALU；
- 32位存储器传送。

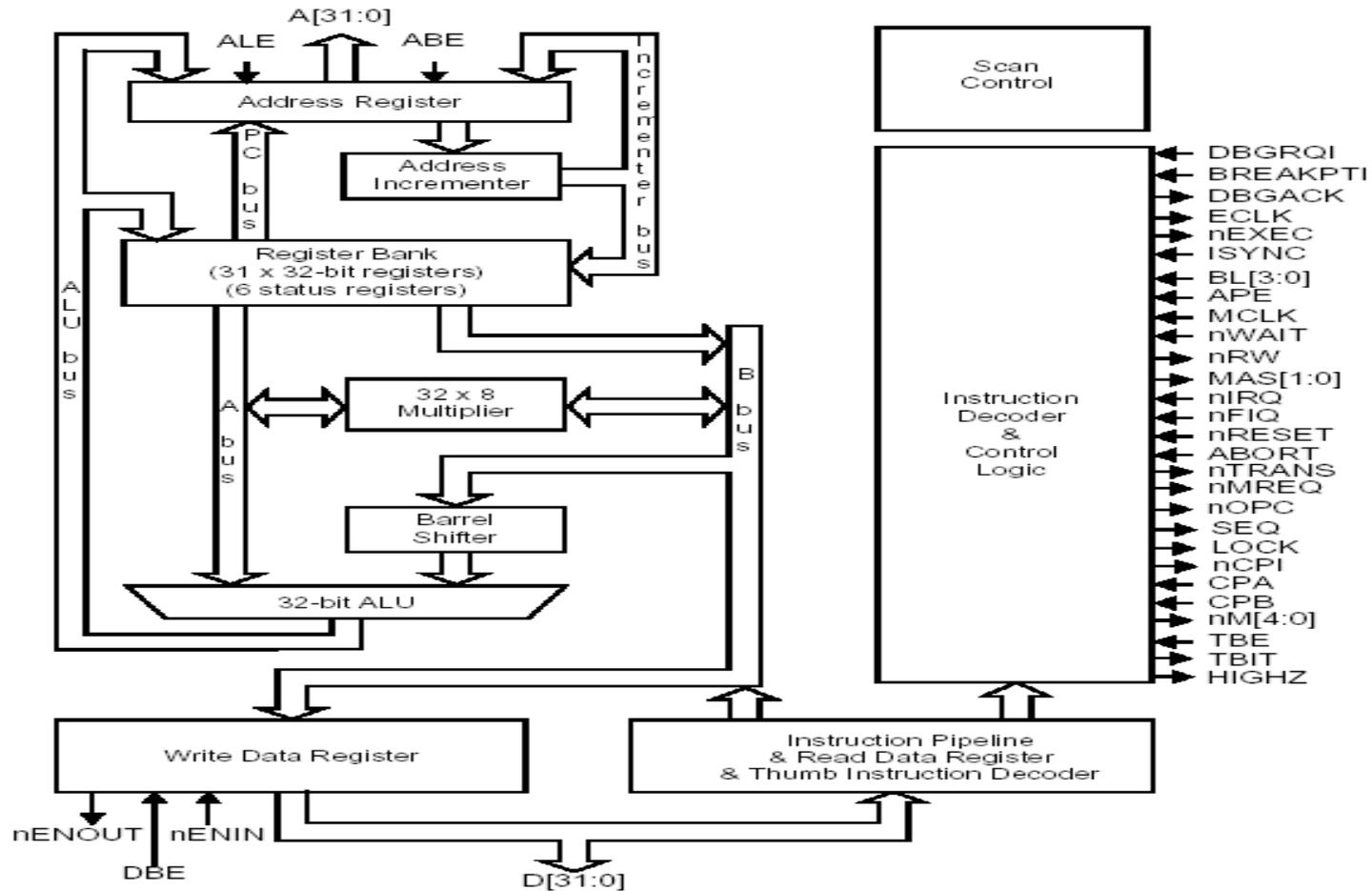
ARM7框图



ARM7核心框图

ARM7TDMI Core Diagram

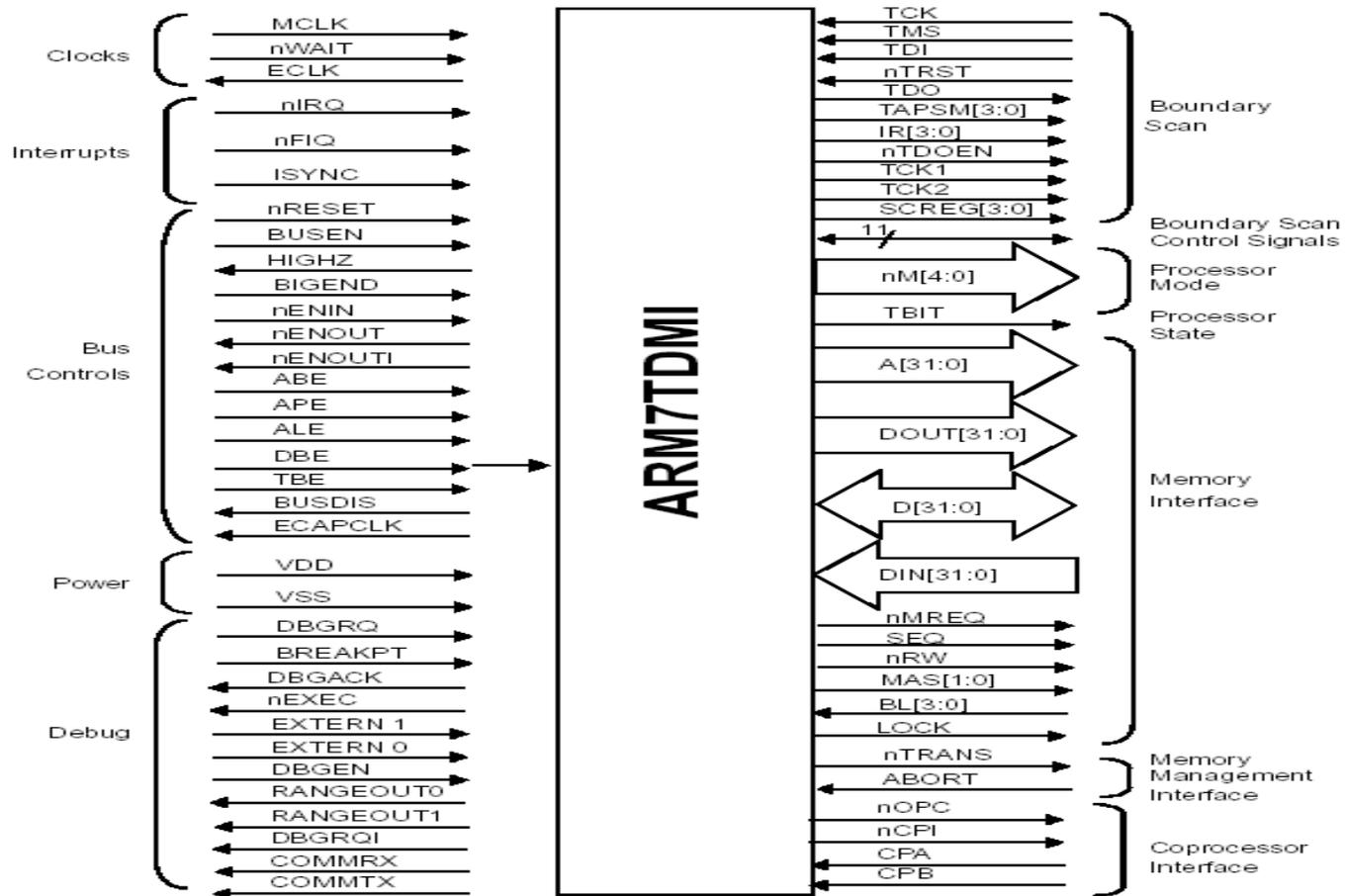
Figure 2. ARM7TDMI Core



ARM7功能框图

ARM7TDMI Functional Diagram

Figure 3. ARM7TDMI Functional Diagram



ARM7寄存器

Figure 6. Register Organization in ARM State

ARM State General Registers and Program Counter

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
R8	R8_fiq	R8	R8	R8	R8
R9	R9_fiq	R9	R9	R9	R9
R10	R10_fiq	R10	R10	R10	R10
R11	R11_fiq	R11	R11	R11	R11
R12	R12_fiq	R12	R12	R12	R12
R13	R13_fiq	R13_svc	R13_abt	R13_irq	R13_und
R14	R14_fiq	R14_svc	R14_abt	R14_irq	R14_und
R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)	R15 (PC)

ARM State Program Status Registers

CPSR	CPSR SPSR_fiq	CPSR SPSR_svc	CPSR SPSR_abt	CPSR SPSR_irq	CPSR SPSR_und
------	------------------	------------------	------------------	------------------	------------------

ARM7寄存器 (Thumb)

Figure 7. Register Organization in Thumb State

THUMB State General Registers and Program Counter

System & User	FIQ	Supervisor	Abort	IRQ	Undefined
R0	R0	R0	R0	R0	R0
R1	R1	R1	R1	R1	R1
R2	R2	R2	R2	R2	R2
R3	R3	R3	R3	R3	R3
R4	R4	R4	R4	R4	R4
R5	R5	R5	R5	R5	R5
R6	R6	R6	R6	R6	R6
R7	R7	R7	R7	R7	R7
SP	SP_fiq	SP_svc	SP_abt	SP_irq	SP_und
LR	LR_fiq	LR_svc	LR_abt	LR_irq	LR_und
PC	PC	PC	PC	PC	PC

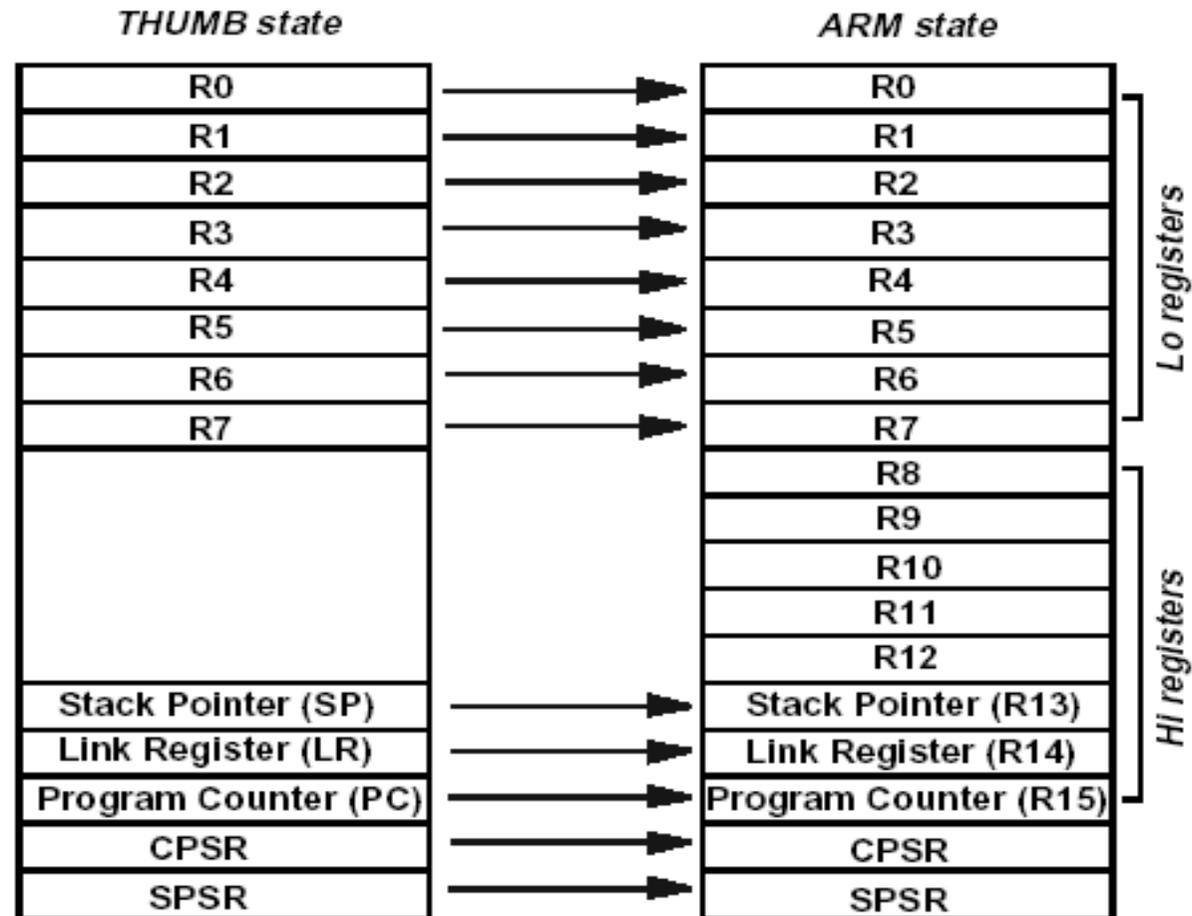
THUMB State Program Status Registers

CPSR	CPSR	CPSR	CPSR	CPSR	CPSR
	SPSR_fiq	SPSR_svc	SPSR_abt	SPSR_irq	SPSR_und

 = banked register

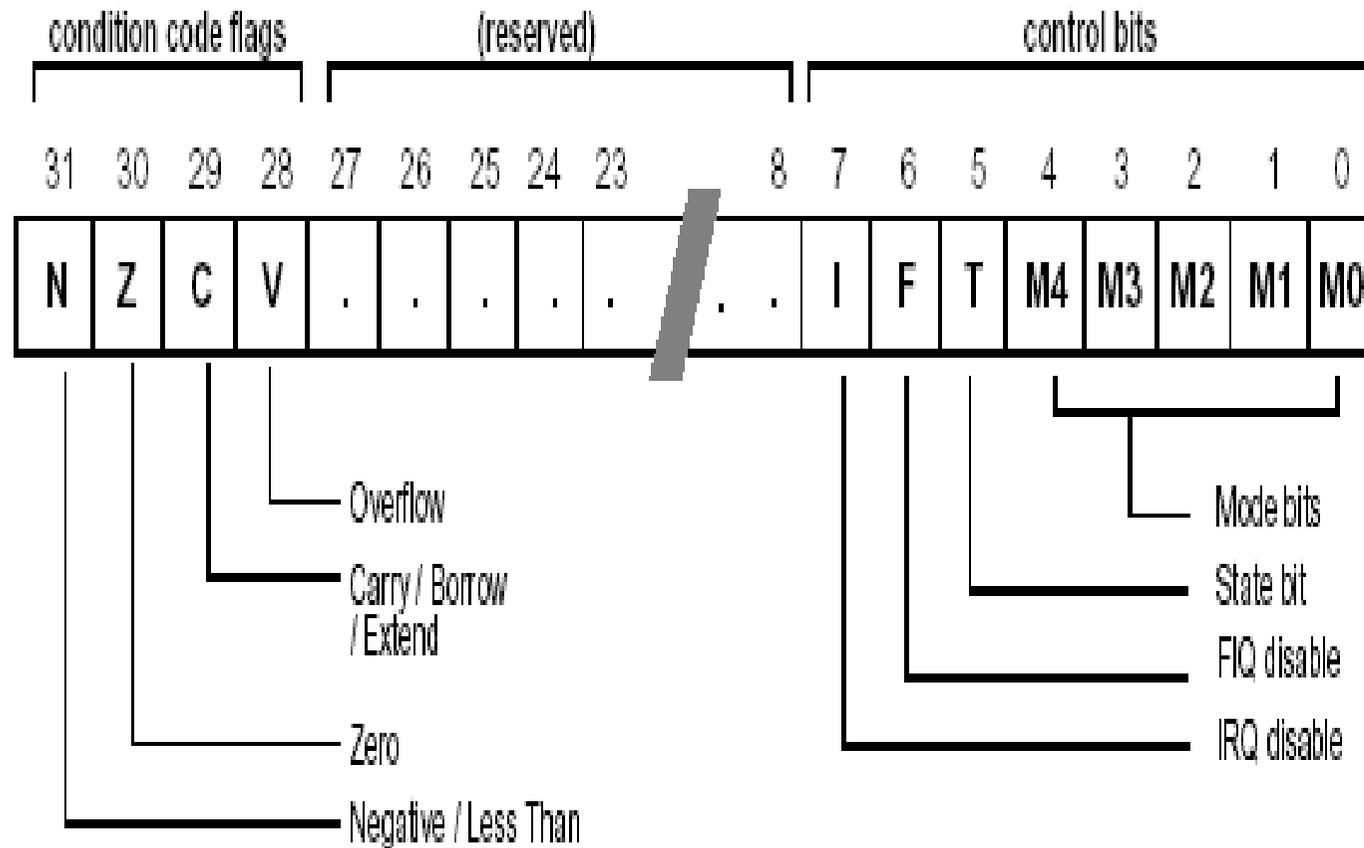
ARM7寄存器映射

Figure 8. Mapping of THUMB State Registers onto ARM State Registers



ARM7程序状态寄存器

Figure 9. Program Status Register Format



ARM7程序状态寄存器模式

Table 2. PSR Mode Bit Values

M[4:0]	Mode	Visible THUMB state registers	Visible ARM state registers
10000	User	R7..R0, LR, SP PC, CPSR	R14..R0, PC, CPSR
10001	FIQ	R7..R0, LR_fiq, SP_fiq PC, CPSR, SPSR_fiq	R7..R0, R14_fiq..R8_fiq, PC, CPSR, SPSR_fiq
10010	IRQ	R7..R0, LR_irq, SP_irq PC, CPSR, SPSR_irq	R12..R0, R14_irq..R13_irq, PC, CPSR, SPSR_irq
10011	Supervisor	R7..R0, LR_svc, SP_svc, PC, CPSR, SPSR_svc	R12..R0, R14_svc..R13_svc, PC, CPSR, SPSR_svc
10111	Abort	R7..R0, LR_abt, SP_abt, PC, CPSR, SPSR_abt	R12..R0, R14_abt..R13_abt, PC, CPSR, SPSR_abt
11011	Undefined	R7..R0 LR_und, SP_und, PC, CPSR, SPSR_und	R12..R0, R14_und..R13_und, PC, CPSR
11111	System	R7..R0, LR, SP PC, CPSR	R14..R0, PC, CPSR

ARM7异常进入/退出

Table 3. Exception Entry/Exit

	Return Instruction	Previous State		Notes
		ARM R14_x	THUMB R14_x	
BL	MOV PC, R14	PC + 4	PC + 2	1
SWI	MOVS PC, R14_svc	PC + 4	PC + 2	1
UDEF	MOVS PC, R14_und	PC + 4	PC + 2	1
FIQ	SUBS PC, R14_fiq, #4	PC + 4	PC + 4	2
IRQ	SUBS PC, R14_irq, #4	PC + 4	PC + 4	2
PABT	SUBS PC, R14_abt, #4	PC + 4	PC + 4	1
DABT	SUBS PC, R14_abt, #8	PC + 8	PC + 8	3
RESET	NA	-	-	4

ARM7异常向量

Table 4. Exception Vectors

Address	Exception	Mode on entry
0x00000000	Reset	Supervisor
0x00000004	Undefined instruction	Undefined
0x00000008	Software interrupt	Supervisor
0x0000000C	Abort (prefetch)	Abort
0x00000010	Abort (data)	Abort
0x00000014	Reserved	Reserved
0x00000018	IRQ	IRQ
0x0000001C	FIQ	FIQ

ARM指令集

- ❖ 位操作
- ❖ 条件执行
- ❖ 32 位 RISC 风格 (Reduced Instruction Set Computer)
- ❖ 数据总线 / 寄存器 均为32位宽
- ❖ 指令执行典型为一个时钟周期执行一条指令
- ❖ 指令发展通过多重Pipeline: 3 for ARM 7, 5 for ARM 9

ARM指令集

RISC 能改进功率消耗

与CISC相比,执行相同的任务占用更少的时钟周期

举例: `for (j=0x3000; j!=0; j--){}`

RISC指令 (假设计数寄存器为 r12)

again:

```
SUBS r12,r12,#1      ;1 cycle
BNE again            ;3 cycles
```

exit_loop:

ARM指令集

CISC 执行在16-位 80186平台上

;assume count value in register CX

```
loop1: DEC CX           ;2 cycles
```

```
        JCXZ exit_loop ;6 cycles
```

```
        JMP loop1      ;15 cycles
```

```
exit_loop
```

ARM指令集

结论:

RISC 每个循环占用4个时钟周期

CISC每个循环占用23个时钟周期

对于同样的执行效果, 假设IDLE时的功率消耗忽略不记,
RISC将比CISC省电1/5, 且执行速度快约5倍!

THUMB指令集

16 位 “Thumb” 指令集的代码量仅有ARM指令集的65%，
而与16位存储器系统连接时，性能相当于ARM的160%

Thumb的局限性:

不支持乘法和累加指令.

条件跳转限制在 256 byte 偏移范围内,无条件跳转限制为
4K 偏移范围内 (而ARM为 32 Mbytes偏移)

很少指令为有条件的 (所有的ARM指令全为有条件的)

执行效率可能稍慢，典型情况下约为 10%

THUMB指令集

对于简单的指令而言, Thumb 更有效:

Thumb 指令举例:

```
ADD R0, #42 ;adds the value 42 to R0 and places the  
;result in R0.
```

ARM 指令举例

```
ADD R0, R0, #42 ;same as above but requires 32 bit  
;instruction.
```

THUMB指令集

从此例得出的结论:

Thumb 模式使用2 bytes 而ARM模式使用 4 bytes,节省一半内存空间.

THUMB指令集

ARM 代码举例

```
ADDEQ R0, R1, R2 LSL #3 ;If the Z flag is set, then do a  
                           ;multiply R2 by 2^3 followed by an  
                           ;add of R1 and store result into R0
```

Thumb 代码举例

```
BNE label ;skip next two instructions if Z flag is 0  
LSL R2, R2, #3 ;multiply R2 by 2^3 and place in R2  
ADD R0, R1, R2 ;add R1+R2 and place in R0  
:label ;continue
```

THUMB指令集

结论

Thumb 使用 6 bytes, ARM 使用 4 bytes

在这种情况下, ARM 更有效率

MMU与协处理器

- ❖ 32位数据总线，32位地址总线
- ❖ 字、半字地址对齐
- ❖ ARM7TDMI核不包含MMU，由处理器外围提供
- ❖ 提供专用的协处理器指令

调试接口

❖ JTAG调试接口

TMS、TCK、TDI、TDO、nTRST
ARM7TDMI没有提供输入信号的上拉

❖ 嵌入式ICE-RT逻辑

支持两个硬件断点，目标状态监控