

存储器组织形式

- ARM异常和中断
- 异常处理模式
- ARM的异常中断响应过程
- ARM的存储数据类型
- 存储器组织
- ARM存储器管理单元MMU
- ARM微处理器的应用选型

■ ARM异常和中断

- 异常(exception)由内部或外部源产生以引起处理器处理一个事件。例如，外部中断或试图执行未定义指令都会引起异常。在处理异常之前，处理器状态必须保留，以便在异常处理程序完成后，原来的程序能够重新执行。同一时刻可能出现多个异常。
- ARM支持7种类型的异常。表列出了异常的类型以及处理这些异常的处理器模式。异常出现后，强制从异常类型对应的固定存储器地址开始执行程序。这些固定的地址称为异常向量(Exception Vectors)。

■ 异常处理模式

异常类型	模式	正常地址	高端向量地址
复位	管理	0x00000000	0xFFFF0000
未定义指令	未定义	0x00000004	0xFFFF0004
软件中断（SWI）	管理	0x00000008	0xFFFF0008
预取中止（取指令存储器中止）	中止	0x0000000C	0xFFFF000C
数据中止（数据访问存储器中止）	中止	0x00000010	0xFFFF0010
IRQ （中断）	IRQ	0x00000018	0xFFFF0018
FIQ （快速中断）	FIQ	0x0000001C	0xFFFF001C

•ARM的异常中断响应过程

当异常出现时，异常模式分组的R14和SPSR用于保存状态，即

```
R14_<exception mode> = return link
SPSR_<exception_mode> = CPSR
CPSR[4:0] = exception mode number
CPSR[5] = 0 /*在ARM状态执行*/
if <exception mode> = Reset or FIQ then
    CPSR[6] = 1 /*禁止快速中断*/
/*否则CPSR[6]不变*/
CPSR[7] = 1 /*禁止正常中断*/
PC = exception vector address
```

当处理异常返回时，将SPSR传送到CPSR，R14传送到PC。这可用两种方法自动完成，即使用带S位的数据处理指令，将PC作为目的寄存器；使用带恢复CPSR的多加载指令。

■ 异常优先级

如果同时发生两个或更多异常，一个固定的优先级系统决定它们被处理的顺序：

异常类型	优先级	 优先级降低
复位	1 （最高优先级）	
数据中止	2	
FIQ	3	
IRQ	4	
预取中止	5	
未定义指令	6	
SWI	7 （最低优先级）	

■ ARM的存储数据类型

- ARM指令系统属于RISC指令系统。标准的ARM指令每条都是32位长，有些ARM核还可以执行Thumb指令集，该指令集是ARM指令集的子集，每条指令只有16位。

ARM处理器一般支持下列6种数据类型：

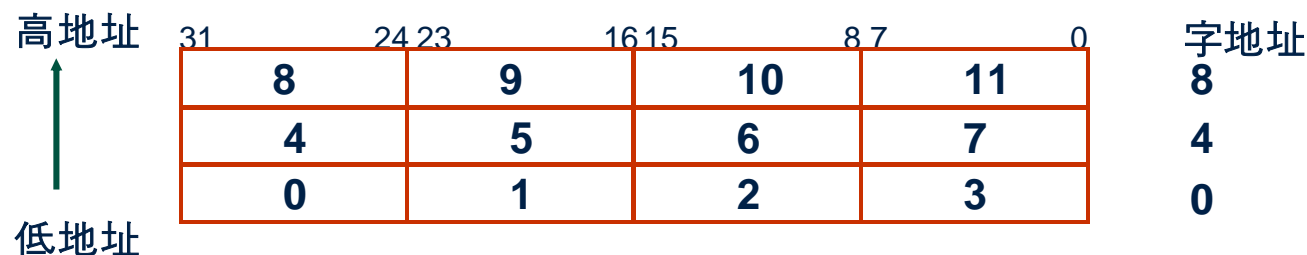
- (1) 8位有符号字节类型数据；
- (2) 8位无符号字节类型数据；
- (3) 16位有符号半字类型数据；
- (4) 16位无符号半字类型数据；
- (5) 32位有符号字类型数据；
- (6) 32位无符号字类型数据；

- 有些ARM处理器不支持半字和有符号字节数据类型。在ARM内部，所有指令都是32操作数据。短的数据数据类型只有在数据传送类指令中才被支持当1个字节数据取出后，被扩展到32位，在内部数据处理时，作为32位的值进行处理，并且ARM指令以字为边界。
- 所有Thumb指令都是16位指令时，并且以2个字节为边界。
- ARM协处理器可以支持另外的数据类型，包括一套浮点数据类型，ARM的核并没有明确的支持。

• 存储器组织

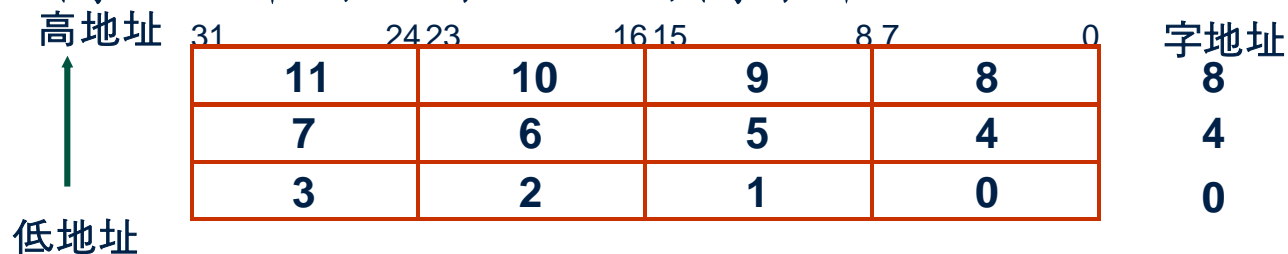
■ 大端模式

- ◆ 字数据的高位字节存储在低地址中
- ◆ 字数据的低字节则存放在高地址中



■ 小端模式

- ◆ 低地址中存放字数据的低字节
- ◆ 高地址中存放字数据的高字节



示例

A. 小端存储法(0x01234567)

地址	0x8000	0x8001	0x8002	0x8003
数据（16进制表示）	0x67	0x45	0x23	0x01
数据（二进制表示）	01100111	01000101	00100011	00000001

B. 大端存储法

地址	0x8000	0x8001	0x8002	0x8003
数据（16进制表示）	0x01	0x23	0x45	0x67
数据（二进制表示）	00000001	00100011	01000101	01100111

■ ARM的存储器层次

- 寄存器组
- 片上RAM
- 片上Cache
- 主存储器
- 硬盘

■ ARM存储器管理单元MMU

- 实现虚拟地址空间到物理存储空间的映射;
- 存储器访问权限的控制;
- 设置虚拟存储空间的缓冲特性。

1. ARM微处理器内核的选择

用户如果希望使用WinCE或标准Linux等操作系统以减少软件开发时间，就需要选择ARM720T以上带有MMU（Memory Management Unit）功能的ARM芯片，ARM720T、ARM920T、ARM922T、ARM946T、Strong-ARM都带有MMU功能。

而ARM7TDMI则没有MMU，不支持Windows CE和标准Linux，但目前有uCLinux等不需要MMU支持的操作系统可运行于ARM7TDMI硬件平台之上。

2. 系统的工作频率

系统的工作频率在很大程度上决定了ARM微处理器的处理能力。ARM7系列微处理器的典型处理速度为0.9MIPS/MHz，常见的ARM7芯片系统主时钟为20MHz-133MHz，ARM9系列微处理器的典型处理速度为1.1MIPS/MHz，常见的ARM9的系统主时钟频率为100MHz-233MHz，ARM10最高可以达到700MHz。

不同芯片对时钟的处理不同，有的芯片只需要一个主时钟频率，有的芯片内部时钟控制器可以分别为ARM核和USB、UART、DSP、音频等功能部件提供不同频率的时钟。

3. 芯片内存储器的容量

大多数的ARM微处理器片内存储器的容量都不太大，需要用户在设计系统时外扩存储器，但也有部分芯片具有相对较大的片内存储空间，如ATMEL的AT91F40162就具有高达2MB的片内程序存储空间，用户在设计时可考虑选用这种类型，以简化系统的设计。

4. 片内外围电路的选择

除ARM微处理器核以外，几乎所有的ARM芯片均根据各自不同的应用领域，扩展了相关功能模块，并集成在芯片之中，我们称之为片内外围电路，如USB接口、IIS接口、LCD控制器、键盘接口、RTC、ADC和DAC、DSP协处理器等，设计者应分析系统的需求，尽可能采用片内外围电路完成所需的功能，这样既可简化系统的设计，同时提高系统的可靠性。

