S3C44B0X 中文数据手册

2．ARM 处理器工作模式

S3C44B0X 采用了非常先进的 ARM7TDMI 内核，它是由 ARM（Advanced RISC Machines）公司研制的。

2．1 处理器工作状态

从程序员的角度上看，ARM7TDMI 内核可以工作在下面两种工作状态的一种下：

- ARM 状态： 此时执行 32 位字对齐的 ARM 指令。
- THUMB 状态：此时执行 16 位半字对齐的 THUMB 指令。在这种状态下，PC 采用第 1 位来选择一个字中的哪个半字。

注意：这两种状态的转换不影响处理器状态和寄存器的内容。

2．2 状态切换

进入 Thumb 状态

进入 Thumb 状态，可以通过执行 BX 指令，同时操作数寄存器的状态位（0 位）置 1 来实现。

当从异常（IRQ, FIQ, UNDEF, ABORT, SWI 等）返回时，也会自动进入 Thumb 状态，只要进入异常处理前处理器处于 Thumb 状态。

进入 ARM 状态

- 进入 ARM 状态，可以通过执行 BX 指令，并且操作数寄存器的状态位（0 位）清零来实现。
- 当处理器进入异常（IRQ, FIQ, RESET, UNDEF, ABORT, SWI 等）。这时，PC 的值保存在异常模式下的 link 寄存器重，并从异常向量地址处开始执行异常处理程序。

2．2 数据类型

ARM7TDMI 支持字节（byte，8－bit），半字（16－bit）和字（32－bit）数据类型。字必须按照 4 字节排列，半字必须按照 2 字节排列。

2．3 存储空间的格式

ARM7TDMI 将存器空间视为一个从 0 开始由字节组成的线性集合，字节 0 到 3 中保存了第一个字，字节 4 到 7 保存第二个字，依此类推。ARM7TDMI 对存储的字，可以按照小端（little endian）或大端（big endian）的方式对待：

大端格式：

在这种格式中，字数据的高字节存储在低地址中，而字数据的低字节则存放在高地址中，如图3－4所示：

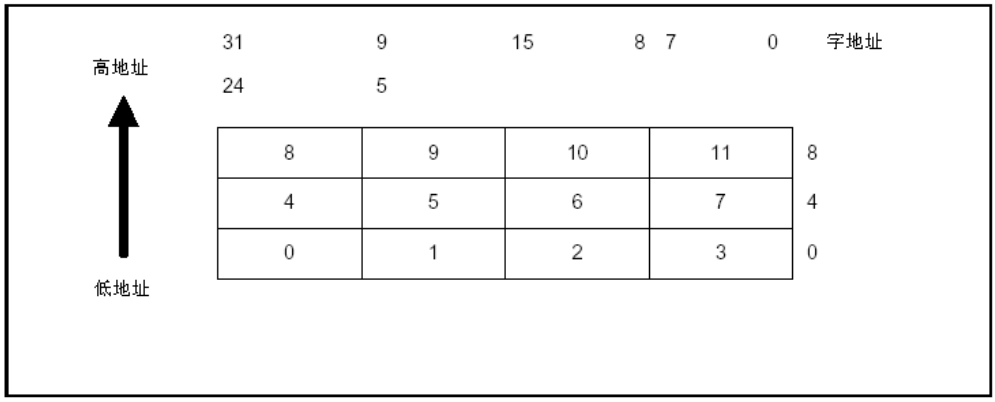

图 3－4　以大端格式存储字数据

小端格式：

与大端存储格式相反，在小端存储格式中，低地址中存放的是字数据的低字节，高地址存放的是字数据的高字节。如图3－5所示：

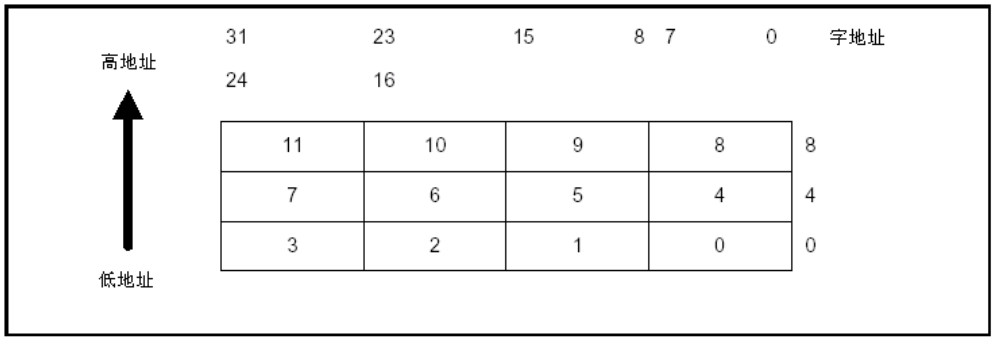

图 3－5 以小端格式存储字数据

2．4 操作模式

ARM7TDMI 内核支持 7 种操作模式：

- 用户模式（usr 模式），运行应用程序的普通模式。
- 超级用户模式（SVC 模式），主要用于 SWI（软件中断）和 OS (操作系统)。这个模式有额外的特权，允许你进一步控制处理器。
- 中断模式（IRQ 模式），用作通用中断处理。这个模式也是有特权的。导致 IRQ 的设备有键盘、定时器、串行口、硬盘、软盘、等等……
- 快速中断模式(FIQ 模式)，用来处理外设发起的快速中断。用于支持特殊的数据传送与通道处理。这个模式是有特权的。
- 未定义模式（und 模式）：当执行了未定义指令时进入该模式。

IRQ 和 FIQ 之间的区别是对于 FIQ 你必须尽快处理事情并离开这个模式。IRQ 可以被 FIQ 所中断但 IRQ 不能中断 FIQ。为了使 FIQ 更快，所以有更多的影子寄存器。FIQ 不能调用 SWI。FIQ 还必须禁用中断。如果一个 FIQ 例程必须重新启用中断，则它太慢了，应该是 IRQ 而不是 FIQ。

可用软件控制操作模式的切换，同时外部的中断和异常处理也会导致操作模式的切换。

绝大多数的用户应用程序运行在用户模式。

当系统响应中断或异常、或访问受保护的系统资源时，处理器会进入特权模式（除用户模式以外的所有模式）。

2．5 寄存器

ARMTIDMI 共具有 37 个 32 位的寄存器：31 个通用寄存器，6 个状态寄存器，但并不是所有的寄存器都能总是被访问到。在某一时刻寄存器能否访问，由处理器的当前工作状态和操作模式决定。

根据微处理器内核的当前工作状态，可分别访问 ARM 状态寄存器集和 Thumb 状态寄存器集。ARM 状态寄存器集包含 16 个可以直接访问的寄存器：R0～R15。除 R15 以外，其余的寄存器为通用寄存器，可用于存放地址或数据值。R16 寄存器是当前程序状态寄存器 CPSR，用于保存状态信息。详细的说明如下：

- 寄存器 0 到寄存器 7 是通用寄存器并可以用做任何目的。
- 寄存器 8 到 12 是通用寄存器，但是在切换到 FIQ 模式的时候，使用它们的影子(bank)寄存器。
- 寄存器 13 典型的用做 OS 栈指针，但可被用做一个通用寄存器。这是一个操作系统问题，不是一个处理器问题，所以如果你不使用栈，只要你以后恢复它，你可以在你的代码中自由的使用它。每个处理器模式都有这个寄存器的影子寄存器。
- 寄存器 14 专职持有返回点的地址，在系统执行一条“跳转并连接”（BL）指令的时候，R14 将接收到一个 R15 的拷贝。其它的时候，它可以用作一个通用寄存器。相应的在其它模式下的影子寄存器，也同样用来保存在中断或异常发生时，或是在中断和异常中的 BL 指令执行时，R15 的返回值。
- 寄存器 15 是程序计数器（PC）。在 ARM 状态下，R15 的 bits[1:0]为 0，bits[31:2]保存了 PC 的值。在 Thumb 状态下，bits[31:1]保存了 PC 值。
- 寄存器 16 是 CPSR（当前程序状态寄存器），用来保存当前代码标志和当前处理器模式位。

Thumb 状态寄存器集是 ARM 状态寄存器集的一个子集。可以访问的寄存器有：8 个通用寄存器 R0～R7，程序计数器 PC、堆栈指针寄存器 SP、连接寄存器 LR 和当前程序状态寄存器 CPSR。在每一种特权模式下，都有对应的分组堆栈指针寄存器 SP、连接寄存器 LR 和备份的程序状态寄存器 SPSR。

Thumb 状态寄存器集与 ARM 状态寄存器集的对应关系如下：

－ Thumb 状态下 R0～R7 寄存器与 ARM 状态下 R0～R7 寄存器是相同的。
－ Thumb 状态下的 CPSR 和 SPSRs 与 ARM 状态下的 CPSR 和 SPSRs 是相同的。
－ Thumb 状态下的 SP、LR 和 PC 直接对应 ARM 状态寄存器 R13、R14 和 R15。

在 Thumb 状态下，寄存器 R8～R15 不属于标准寄存器集的一部分，但在必要的情况下，用户可以通过汇编语言程序访问他们，用作快速的临时存储将单元。

2．6 程序状态寄存器

ARM7TDMI包含一个当前程序状态寄存器（CPSR）和五个备份的程序状态寄存器（SPSRs）。备份的程序状态寄存器用来进行异常处理，这些寄存器的功能包括：

- 保存ALU当前操作的有关信息
- 控制中断的允许和禁止
- 设置处理器的运行模式

程序状态寄存器组成如图3－6所示：

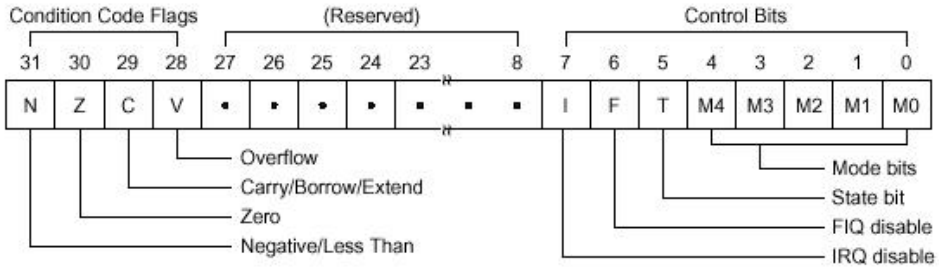

图 3－6 程序状态寄存器格式

条件码标志（Condition Code Flags）

N、Z、C、V均为条件码标志位。它们的内容根据算术或逻辑运算的结果所改变，并且用来作为一些指令是否的运行的检测条件。

在ARM状态下，绝大多数的指令都是有条件执行的。

在Thumb状态下，仅有分支指令是有条件执行的。

条件码标志各位的具体含义下表所示：

标志位	含 义
N	当用两个补码表示的带符号数进行运算时，N=1 表示运算的结果为负数；N=0 表示运算的结果为正数或零；
Z	Z=1 表示运算的结果为零；Z=0 表示运算的结果为非零；
C	可以有 4 种方法设置 C 的值： — 加法运算（包括比较指令 CMN）：当运算结果产生了进位时（无符号数溢出），C=1，否则 C=0。 — 减法运算（包括比较指令 CMP）：当运算时产生了借位（无符号数溢出），C=0，否则 C=1。 — 对于包含移位操作的非加/减运算指令，C 为移出值的最后一位。 — 对于其他的非加/减运算指令，C 的值通常不改变。

V	可以有 2 种方法设置 V 的值： — 对于加/减法运算指令，当操作数和运算结果为二进制的补码表示的带符号数时，V=1 表示符号位溢出。 — 对于其他的非加/减运算指令，C 的值通常不改变。

控制位（Control Bits）

PSR的低8位（包括I、F、T和M[4：0]）称为控制位，当发生异常时这些位可以被改变。如果处理器运行特权模式，这些位也可以由程序修改。

○ 中断禁止位I、F：

I=1 禁止IRQ中断;

F=1 禁止FIQ中断。

○ T标志位：该位反映处理器的运行状态。

对于ARM体系结构v5及以上的版本的T系列处理器，当该位为1时，程序运行于Thumb状态，否则运行于ARM状态。

对于ARM体系结构v5及以上的版本的非T系列处理器，当该位为1时，执行下一条指令以引起为定义的指令异常；当该位为0时，表示运行于ARM状态。

○ 运行模式位M[4：0]：M0、M1、M2、M3、M4是模式位。这些位决定了处理器的运行模式。具体含义如下表所示：

M[4：0]	处理器模式	可访问的寄存器
0b10000	用户模式	PC，CPSR,R0-R14
0b10001	FIQ 模式	PC，CPSR, SPSR_fiq，R14_fiq-R8_fiq, R7～R0
0b10010	IRQ 模式	PC，CPSR, SPSR_irq，R14_irq,R13_irq,R12～R0
0b10011	管理模式	PC，CPSR, SPSR_svc，R14_svc,R13_svc,,R12～R0,
0b10111	中止模式	PC，CPSR, SPSR_abt，R14_abt,R13_abt, R12～R0,
0b11011	未定义模式	PC，CPSR, SPSR_und，R14_und,R13_und, R12～R0,
0b11111	系统模式	PC，CPSR（ARM v4 及以上版本）, R14～R0

由上表可知，并不是所有的运行模式位的组合都是有效地，其他的组合结果会导致处理器进入一个不可恢复的状态。

保留位（reserved）

PSR中的其余位为保留位，当改变PSR中的条件码标志位或者控制位时，保留位不要被改变，在程序中也不要使用保留位来存储数据。保留位将用于ARM版本的扩展。

2．7 异常

当正常的程序执行流程被中断时，称为产生了异常。例如程序执行转向响应一个外设的中断请求。在优先处理异常时，处理器的当前状态必须保留，以便在异常处理完成之后程序流程能正常返回。并且，多个异常可能会同时发生，这时，它们将按照固定的顺序依次处理。当进入异常状态时，内核应该进行以下动作：

1. 将原来执行的程序的下一条指令地址保存到 lr 中；
2. 拷贝 CPSR 到 SPSR;
3. 根据发生的异常类型改变 CPSR 的模式位的值；
4. 令 PC 的值指向异常处理向量所指的下一条指令。

这时也可能设置中断禁能标志，以防止不可估计的异常嵌套发生。

当处理器处于 Thumb 状态时发生了异常，当 PC 载入异常矢量所在地址时，它将自动地切换到 ARM 状态。

当完成异常处理，将推出该状态时，处理器应该进行如下动作：

1. 移出 lr，减去相应的偏移量，赋给 PC(偏移量的值依据于发生的异常的类型)。
2. 将 SPSR 拷贝到 CPSR；
3. 清除中断禁止标志（如果开始时置位了的话)。

下表列出推荐用来完成第 1 步的指令：

	返回指令	之前的状态		注意
		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

注意 1：这里 PC 所赋的是 BL/SWI 等指令所取得的地址，它们在预取指的阶段就被中断了；

2：这里 PC 所赋的是由于 FIQ 或 IRQ 取得了优先权，而没有来得及得到执行的指令地址。

3：这里 PC 所赋的地址是 Load 或 Store 指令的地址，它们在执行时产生了数据的异常中断。

FIQ

FIQ（快速中断请求）异常通常是用来支持数据传输和通道操作的，在 ARM 状态下，它具有充分的私有(影子)寄存器，用来减少对寄存器存取的需要**(从而减少进入中断前的“上下文切换”的工作，即中断前保存寄存器的值，完成后又恢复寄存器)。**

FIQ 中断是由外部设备通过拉低 nFIQ 引脚触发的。通过对 ISYNC 输入引脚的控制，nFIQ 可以排除同步或者异步的传输情况。当 ISYNC 为低电平,nFIQ 和 nIRQ 将被异步地考虑。

无论 FIQ 异常是发生在 ARM 或者 Thumb 状态，FIQ 处理程序在离开中断处理时，可以通过执行以下的语句完成：

```
SUBS    PC, R14_fiq, #4
```

（即 PC=R14_fiq－4）

FIQ 异常可以通过设置 CPSR 中的 F 标志位来禁止（但是在用户模式下是不可能的)。当 F 标志位被清除，也就是 FIQ 使能，ARM7TDMI 将在每条指令的最后，检测 FIQ 输入是否为低电平。

IRQ

IRQ(中断请求)是支持普通中断操作的。IRQ 比 FIQ 的优先级低，并且当系统进入 FIQ 处

理中时，IRQ 将被屏蔽。IRQ 也可以通过设置 CPSR 中的 I 标志位来禁止，同样也不能够在用户模式中这样做（只能在特权模式下这样做）。

无论 IRQ 发生在 ARM 或者 Thumb 状态下，退出中断处理时，采用以下语句完成：

```
SUBS    PC,R14_irq,#4
```

Abort

Abort(异常中断)的产生，说明当前的处理器操作不能够完成。该情况，可以通过 ABORT 输入信号来告知处理器。ARM7TDMI 在存储器操作周期中检测该异常是否发生。

有两种类型的 ABORT 异常：

○ 预取指 abort:发生在预取指令时；

○ 数据 abort:发生在数据操作时。

Abort 机制允许要求以页为单位的虚拟存储器系统的实现。在这样一个系统中，处理器可以产生一个任意的地址。当一个地址上的数据不可用，MMU（存储器管理单元）将产生一个 abort 信号。Abort 处理必须辨认出 abort 的原因，使得要求的数据可用，并重新开始运行那条被 abort 的指令。这样，应用程序就不需要了解存储空间的大小，也不需要了解异常中断对它的影响。

在完成了对 abort 异常中断的处理后，通过以下语句退出中断处理：

```
SUBS    PC,R14_abt,#4          ;预取指 abort
SUBS    PC,R14_abt,#8          ;数据 abort
```

通过执行该语句，就恢复了 PC 和 CPSR,并重试被中断的指令。

SWI

SWI（软件中断指令）用来进入超级用户模式，通常是通过产生 SWI 来请求特殊的超级用户功能。SWI 的处理程序通过执行以下语句，退出异常处理：

```
MOV     PC,R14_svc
```

通过执行该语句，就恢复了 PC 和 CPSR,并返回到 SWI 后面的指令上。

注意：上述所称的 nFIQ,nIRQ,ISYNC,LOCK,BIGEND 和 ABORT 信号引脚是存在于 ARMTDMI 内核中。

未定义指令

当 ARM7TDMI 遇到一个它不能执行的指令，立即调用一个未定义指令陷阱处理程序。这个机制是软件仿真器用来扩展 Thumb 或 ARM 指令集用的。

在仿真器指定到失败指令之后，陷阱处理程序应该执行以下的语句：

```
MOVS    PC,R14_und
```

通过执行该语句，恢复了 CPSR,并返回未定义指令的下一条指令的地址。

异常中断向量

异常中断的向量地址如下表所示：

地址	异常中断类型	入口时处理器的操作模式
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

异常中断优先级

当几种异常中断同时发生时，处理器根据一个固定的优先级系统来决定处理它们的顺序。
最高有限级：
1. 复位
2. 数据 abort；
3. FIQ；
4. IRQ；
5. 预取指 abort；

最低优先级：
6. 未定义指令，软件中断。

并非所有的异常中断都可能同时发生

未定义指令和软件中断是相互排斥的，因为它们都对应于对当前指令的特殊的（非重叠的）解码方式。

如果一个数据 abort 和 FIQ 中断同时发生了，并且此时的 FIQ 中断是使能的，ARM7TDMI 进入到数据 abort 处理中，并且立即继续进入 FIQ 向量。从 FIQ 正常的返回后，数据 abort 的处理程序才恢复执行。

复位

当 nRESET 信号为低，ARM7TDMI 放弃指令的执行，并从下一个字地址取指。
当 nRESET 信号为高，ARM7TDMI：
1. 将当前的 PC 值和 CPSR 值写入 R14_svc 和 SPSR_svc；
2. 强制 M[4:0]为 10011（超级用户模式），将 CPSR 中的 I 和 F 位置 1，并将 T 位清零。
3. 强制 PC 从 0X00 地址取得下一条指令；

重新开始在 ARM 状态中执行。