第3章CPU内核结构

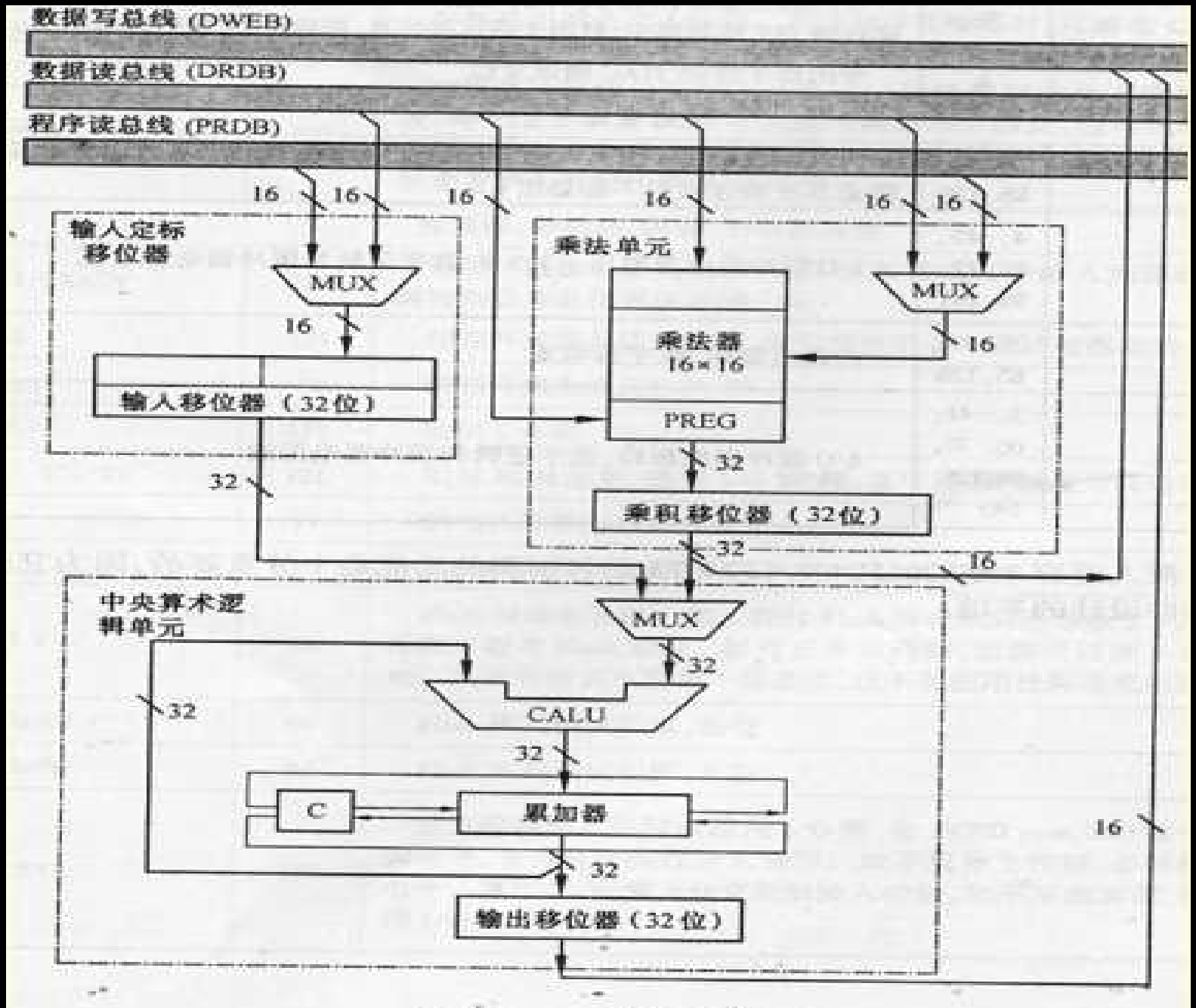

图 3.1　CPU 模块的功能结构

输入定标移位器在算术定标及逻辑操作设置时非常有用。输入定标移位器对输入数据进行0-15位左移。左移时，输出的最低有效位（LSB）为0，最高有效位（MSB）根据状态寄存器ST1 （见3.1.6）的SXM位（符号扩展方式）的值来决定是否进行符号扩展。

当SXM＝1时，则高位进行符号扩展；

当SXM＝0时，则高位填0。

移位的次数由包含在指令中的常量或临时寄存器（TREG）中的值来指定。

3.1.2 乘法器

16×16位的硬件乘法器，单个机器周期内产生一个32位的有符号或无符号乘积。

除了执行无符号乘法指令（MPYU）外，所有的乘法指令均执行有符号的乘法操作，即相乘的两个数都作为二进制的补码数，而运算结果为一个32位的二进制的补码数。乘法器接收的两个乘数，一个来自16位的临时寄存器（TREG），另一个通过数据读总线（DRDB）取自数据存储器，或通过程序读总线（PRDB）取自程序存储器。

两个输入值相乘后，32位的乘积结果保存在32位的乘积寄存器（PREG）中。

PREG的输出连接到乘积定标移位器，通过乘积定标移位器，乘积结果可以从PREG传到CALU或数据存储器。

乘积定标移位器对乘积采用4种乘积移位方式，如表3.1所示。

移位方式由状态寄存器ST1的乘积移位方式位（PM）指定，对于执行乘法/累加操作、进行小数运算或者进行小数乘积的调整都很有用。

表3.1　乘积定标移位器的乘积移位方式

PM	移位	作用和意义
00	无移位	乘积送CALU或数据写总线，不移位
01	左移1位	移去二进制补码乘法产生的额外符号位，产生Q31格式的乘积
10	左移4位	当与一个13位的常数相乘时，移去在16×13位（常数）二进制补码产生的额外的4位符号位，产生Q31格式的乘积
11 12	右移6位	对乘积结果定标，以使得运行128次的乘积累加而累加器不会溢出

3.1.3 中央算术逻辑单元

中央算术逻辑单元（CALU）实现大部分算术和逻辑运算功能，大多数功能只需一个时钟周期，这些运算功能包括：16位加、16位减、布尔运算、位测试以及移位和循环功能。

由于CALU可以执行布尔运算，因此使得控制器具有位操作功能。CALU的移位和循环在累加器中完成。

CALU是一个独立的算术单元，它和后面介绍的辅助寄存器算术单元（ARAU）在程序执行时，是完全不同的两个模块。

一旦操作在CALU中被执行，运算结果会被传送到累加器中，在累加器中再实现如移位等附加操作。

CALU有两个输入，一个由累加器提供，另一个由乘积寄存器（PREG）或数据定标移位器的输出提供。当CALU执行完一次操作后将结果送至32位累加器，由累加器对其结果进行移位。累加器的输出送到32位输出数据定标移位器。

经过输出数据定标移位器，累加器的高、低16位字可 分别被移位或存入数据寄器。

CALU的溢出饱和方式可以由状态寄存器ST0（见3.1.6 小节的介绍）的溢出模式（OVM）位来使能或禁止。

根据CALU和累加器的状态，CALU可执行各种分支指令。这些指令可以根据这些状态位有意义的结合，有条件地执行。

这些条件包括OV(根据溢出跳转)和EQ(根据累加器是否为0跳转)等。

另外，BACC（跳到累加器的地址）指令可以跳转到由累加器所指定的地址；不影响累加器的位测试指令（BIT和BITT）允许对数据存储器中的一个指定位进行测试。

对绝大多数的指令，状态寄存器ST1的第10位符号扩展位（SXM）决定了在CALU计算时是否使用符号扩展。

SXM=0，符号扩展无效，
SXM=1，符号扩展有效。

3.1.4 累加器（ACC)

当CALU中的运算完成后，其结果就被送至累加器，并在累加器中执行单一的移位或循环操作。

累加器的高位和低位字中的任意一个可以被送至输出数据定标移位器，在此定标移位后，再保存于数据存储器。与累加器有关的状态位和转移指令：

1．进位标志位C

ST1的第9位。下述情况之一将影响进位标志位C。

(1)加到累加器或从累加器减

- 当C=0，减结果产生借位时或加结果未产生进位时。
- 当C=1，加结果产生进位时或减结果未产生借位时。

(2)将累加器数值移1位或循环移1位

在左环移或循环左移的过程中，累加器的最高有效位被送至C位。在右环移或循环右移的过程中，累加器的最低有效位被送至C位。

2．溢出方式标志位OVM

ST0的第11位。OVM位决定ACC如何反映算术运算的溢出。

OVM＝1，ACC运算溢出，累加器被设定为下列两个特定值之一：

- 若正溢出，ACC中填最大正数:7FFF FFFFh
- 若负溢出，ACC中填最大负数:8000 0000h

OVM＝0，ACC中的结果正常溢出。

3．溢出标志位OV

ST0的第12位，

C＝0，累加器未溢出；

C＝1，累加器溢出，且被锁存。

4．测试/控制标志位TC

ST1的第11位，根据被测位的值置1或清0。

与累加器有关的转移指令大都取决于C、OV、TC的状态和累加器的值。

3.1.5 输出数据定标移位器

它存储指令中指定的位数，将累加器输出的内容左移0~7位，然后将移位器的高位字或低位字存到数据存储器中（用SACH或SACL指令）。在此过程中，累加器的内容保持不变。

3.1.6 状态寄存器ST0和ST1

ST0和ST1包含了DSP运行时的各种状态和控制位。内容可被读出并保存到数据存储器（用SST指令），或从数据存储器读出加载到ST0和ST1（用LST指令）,用来在子程序调用或进入中断时实现CPU各种状态的保存。

可用指令对ST0和ST1中的各个位单独置1或清0（SETC或CLRC指令）。

ST0各位的含义如下：

ARP（位15-13）：辅助寄存器（AR）间接寻址的指针，选择当前的8个辅助寄存器AR中的一个。AR被装载时，原ARP的值被复制到ARB中。

OV（位12）：溢出标志位。用以指示CALU中是否发生溢出，如溢出则该位为1。

OVM（位11）：溢出方式标志位

＝0，累加器中结果正常溢出。

＝1，根据溢出的情况，累加器被设定为它的最大正值或负值。

INTM（位9）：中断总开关位

＝1，所有可屏蔽中断被禁止

＝0，所有可屏蔽中断有效。

DP（位8-0）：数据存储器页面指针

9位的DP与指令中的7位形成16位的数据存储器的直接地址。

ST1各位的含义如下：

ARB（位15-13）：辅助寄存器指针缓冲器。

当ARP被加载到ST1时，原来的ARP被复制到ARB中，也可将ARB复制到ARP中。

CNF（位12）：片内DARAM配置位

0—片内DARAM映射到数据存储器区；

1—片内DARAM映射到程序存储器区。

TC（位11）：测试/控制标志位。根据被测试位的值，该位被置1或清0。

SXM（位10）:符号扩展方式位，决定在计算时是否使用符号扩展：

1—数据通过定标移位器传送到累加器时将产生符号扩展；

0—不产生符号扩展。

C（位9）：进位标志位

XF（位4）：XF引脚状态位

可用SETC指令置1或清0。

PM（位1-0）：乘积移位方式

00－乘法器的32位乘积不移位，直接入CALU。

01－PREG左移1位后装入CALU，最低位填0；

10－PREG左移4位后装入CALU，低4位填0；

11－PREG输出进行符号位扩展，右移6位。

3.1.7 辅助寄存器算术单元（ARAU）

ARAU完全独立于中央算术逻辑单元,见图3.4。

主要功能：是在CALU操作的同时执行8个辅助寄存器AR7~AR0中的算术运算, AR7~AR0提供了强大而灵活的间接寻址能力。

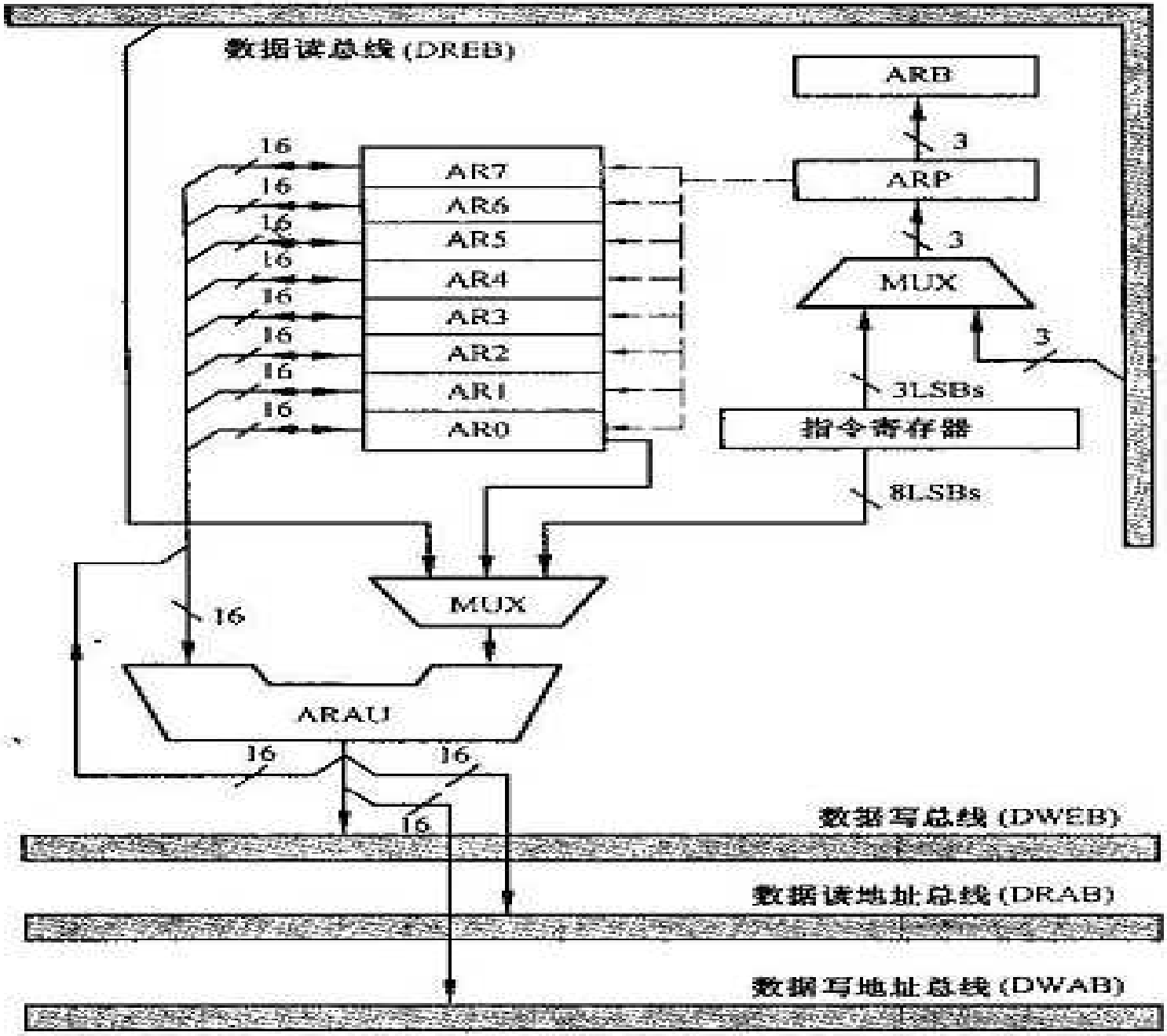

图 3.4　辅助寄存器算术单元(ARAU)

利用ARAU中的16位地址可访问数据存储器64K字空间的任一单元。

如何来选择某一辅助寄存器？可通过指令向ST0的ARP指针装入3位（0~7）数据。

ARAU的8个辅助寄存器提供了强大而灵活的间接寻址能力。

利用辅助寄存器中的16位地址可访问数据存储器64K字空间的任一单元。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/307026131114006054