计算机组成原理与系统结构

第四章

运算方法与运算器

http://jpkc.hdu.edu.cn/computer/zcyl/dzkjdx/

第 4 章 运算方法与运算器

4.6 浮点运算器举例

一 80X87算术协处理器

二 浮点运算流水线

一、 80X87 算术协处理器

- 算术协处理器是一个特殊用途的微处理器，专门是为有效地执行算术或超越函数的运算而设计的。
- 微处理器截取和执行常规指令系统中的指令，而协处理器只截取和执行协处理器指令。协处理器指令实际上是**换码（ESC）指令**，微处理器使用这些指令为协处理器产生一个内存地址，使得协处理器可以执行协处理器指令

80387 的主要性能和结构

1. 80387 的性能

① 可与配套的 CPU 芯片异步并行工作

② 支持多种数据类型

③ 具有高性能的 80 位字长的体系结构

④ 具有出色的内部出错管理功能，能检测出 6 种错误

⑤ 可在 80386/80486 微机系统的两种工作模式下运行

⑥ 扩展了 80386/80486 的指令系统

80387 支持的 7 种数据类型及其

数据类型	数据格式	说明
单字整数（16 位整数）	S | 15 位	二进制补码
短整数（32 位整数）	S | 31 位	二进制补码
长整数（64 位整数）	S | 63 位	二进制补码
短实数（32 位浮点数）	S | 阶码（8 位） | 尾数（23 位）	单精度浮点数
长实数（64 位浮点数）	S | 阶码（11 位） | 尾数（52 位）	双精度浮点数
临时实数（80 位浮点数）	S | 阶码（15 位） | 尾数（64 位）	临时浮点数

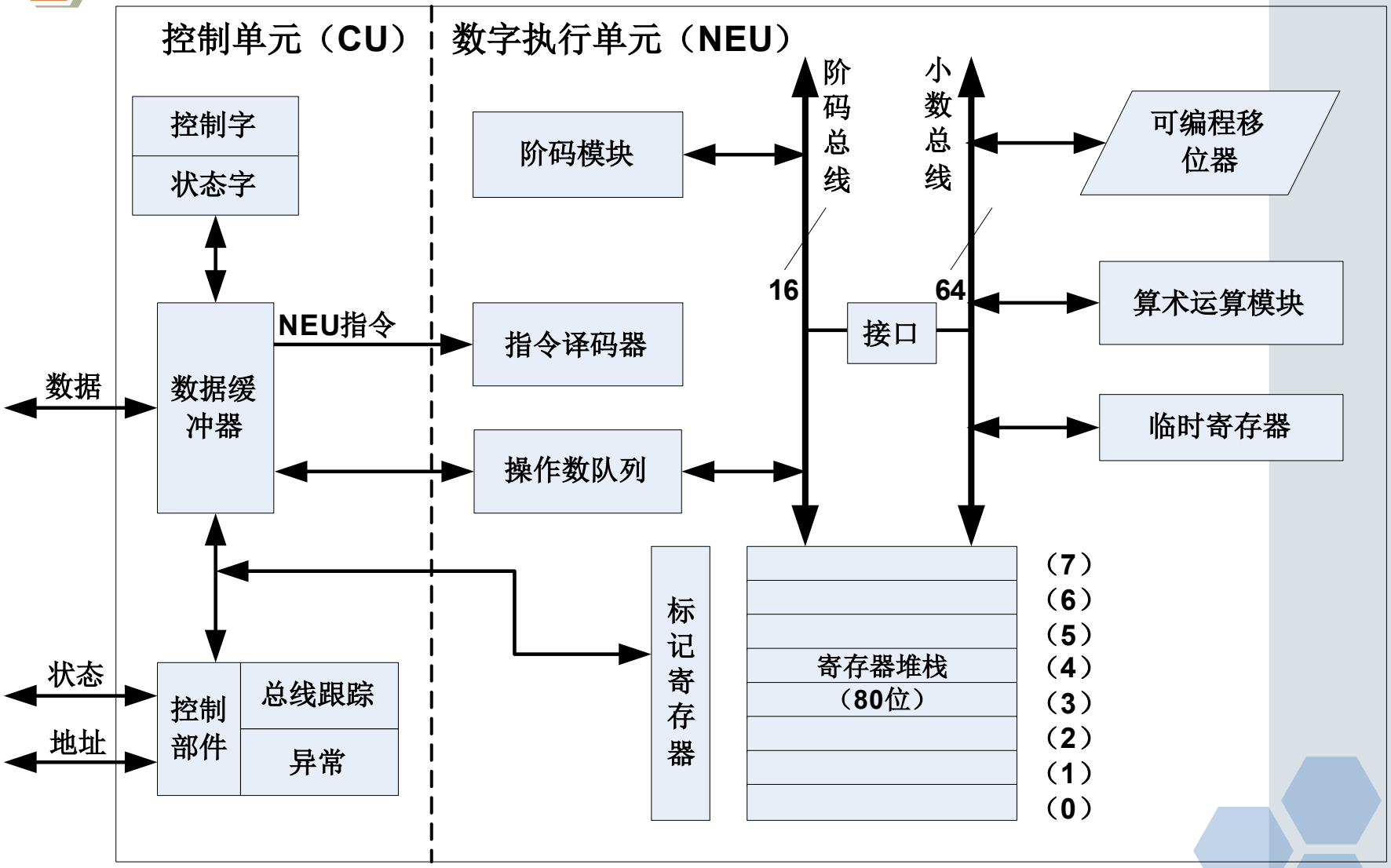
S— 符号位； 3 种浮点数均符合 IEEE754 标准，即阶码的底为 2，阶码值用移码表示，尾数用原码表示

80387 的主要性能和结构

2. 80387 的结构：分为两个单元。

- **控制单元**（Control Unit，CU）将协处理器连接到微处理器系统数据总线上。微处理器和协处理器均监视指令流，如果为 ESC 指令，则由协处理器予以执行。
- **数字执行单元**（Numeric Execution unit，NEU）负责执行所有协处理器指令。
  - NEU 中有一个由 8 个 80 位寄存器构成的堆栈，用于存储算术指令的操作数和结果。
  - NEU 中还包含状态寄存器、控制寄存器、标记寄存器和异常指针寄存器。
  - FSTSWAX 指令是协处理器允许通过 AX 寄存器和微处理器直接通信的唯一指令。

80387 内部结构框图

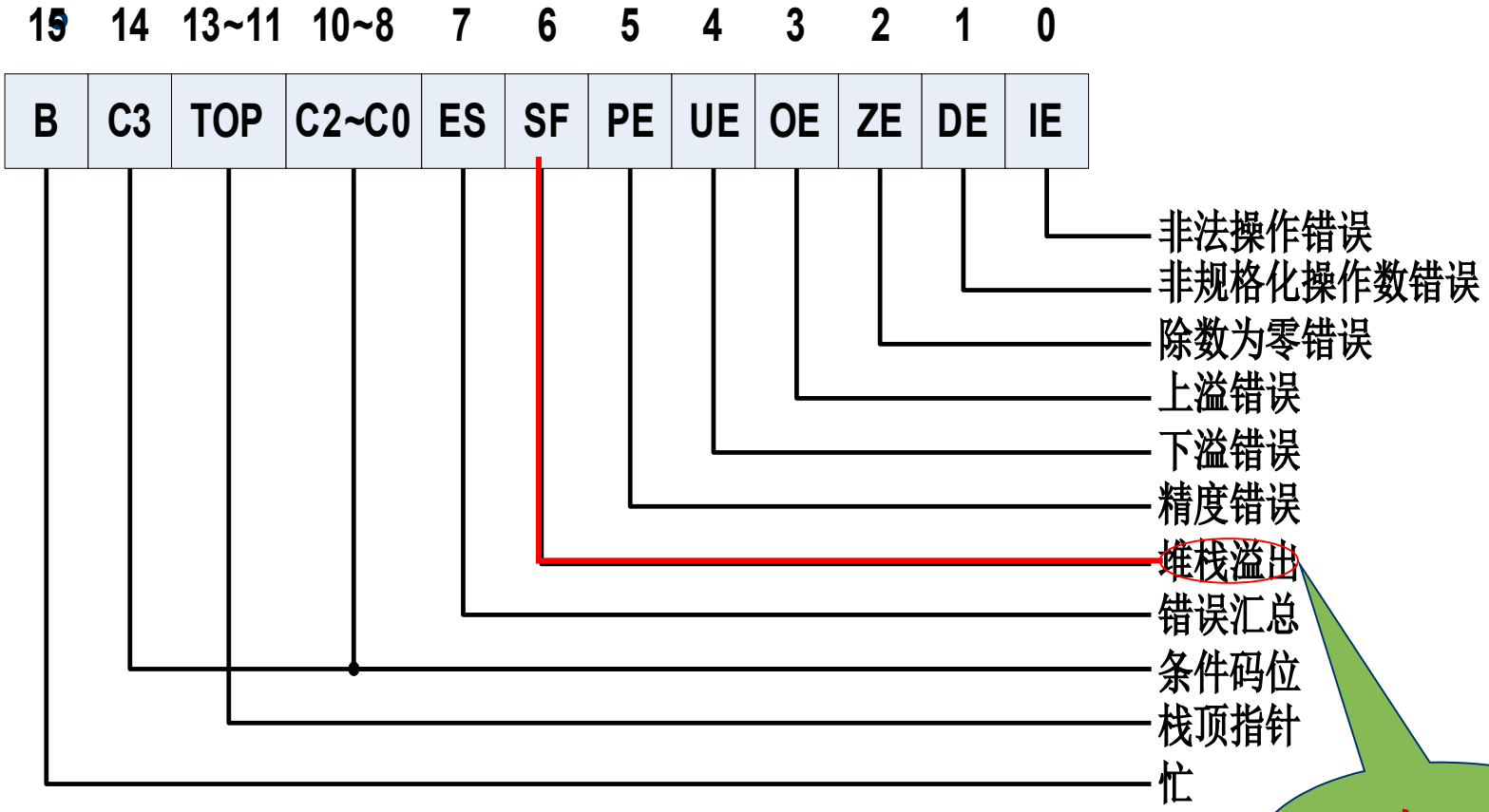


寄存器堆栈

- 80387 包含 8 个寄存器，每个为 80 位宽，它们首尾相接，组成一个“先进后出”的寄存器堆栈。这些堆栈寄存器中总是包含一个 80 位的扩展精度浮点数。数据只有驻留在内存时才可能是任何其他格式。当数据从内存中移到协处理器的寄存器堆栈中时，协处理器将这些带符号的整数、BCD 数、单精度或双精度数转换为扩展精度浮点数。
- 8 个寄存器编号为 0-7，处于栈顶的寄存器称为栈顶寄存器，它的编号由状态寄存器的 TOP 字段指出。在协处理器指令中，用 ST 表示栈顶寄存器，用 ST（i）（i=1-7）访问相对于栈顶寄存器偏移量为 i 的寄存器，即 i 是偏移量，而不是寄存器实际的编号。

状态寄存器

- 状态寄存器反映协处理器所有指令的运行情况



只在 80187 以上的更高型号中使用

状态寄存器

- B　忙位（busy bit）表明协处理器正忙于执行一项任务，通过检测状态寄存器或者使用FWAIT指令均可测试忙位。由于较新的协处理器自动与微处理器同步，所以在执行其他协处理器任务之前不必测试忙标志。
- C3～C0　条件码位（condition code bit），表明了协处理器的条件。
- TOP栈顶（top-of-stack， ST）位表示当前寻址为栈顶的寄存器，通常是寄存器ST（0）。
- ES 错误汇总（error summary）位，当任何一个非屏蔽的错误位（PE、UE、OE、ZE、DE或IE）被置位时，则ES被置位。在8087协处理器中该位也可引起协处理器中断。但从80187开始，不再有协处理器中断。

状态寄存器

- **SF 堆栈标志**（stack flag）位用于区分堆栈上溢和下溢的非法操作。当该位被置 1 时，再根据标志位 B9 （C1）来区分上溢和下溢两种情况。
- **PE 精度错误**（precision error）表明结果或操作数超过了设定的精度范围。
- **UE 下溢错误**（underflow error）表明一个非 0 的结果太小，以致于不能用由控制字选择的当前精度来表示。
- **OE 上溢错误**（overflow error）表明结果太大而不能被表示出来，如果此错误被屏蔽，则协处理器对上溢错误就会产生一个无穷大。

状态寄存器

- ZE 被零除错误（zero error）表明当被除数是非无穷大和非零时，除数是零。
- DE 非规格化操作数错误（denormalized error）表明至少有一个操作数是非规格化的。
- IE 非法操作错误（Invalid error）表明堆栈有上溢或下溢错误，是不确定的形式（0÷0、+∞和－∞等），或者使用了NAN作为操作数。此标志表明诸如对负数开平方等类似的错误。

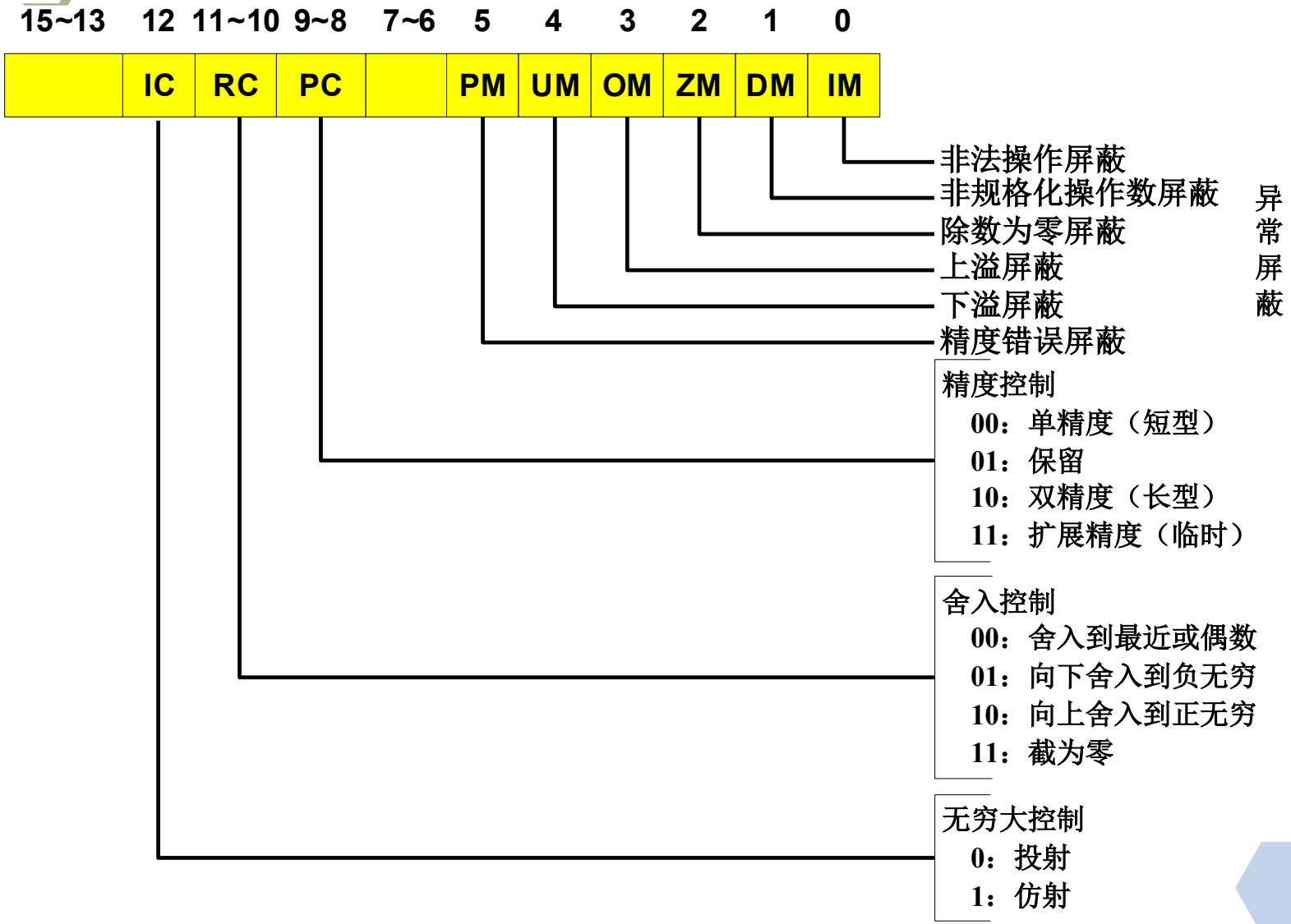
状态寄存器

- 执行 FSTSW 指令就可以访问状态寄存器，此指令将状态寄存器中的内容存人内存的一个字单元中。在 80187 或 80187 以上的协处理器中， FSTSW AX 指令可将状态寄存器中的内容直接复制到微处理器的 AX 寄存器中。一旦状态寄存器的状态被存储到内存或 AX 寄存器中，则可以使用常规软件检测状态寄存器中的各位，譬如使用以下两种方法测试状态寄存器的各位。一种方法是使用 TEST 指令来测试状态寄存器的各位，另一种方法是使用 SAHF 指令将状态寄存器中最左边的 8 位传送到微处理器的标志寄存器中。
- 协处理器和微处理器之间的通信在 80187 和 80287 中是通过 I / O 端口 00FAH ～ 00FFH 实现的，而在 80387 ～ Pentium4 中是通过 I / O 端口 800000FAH ～ 800000FFH 实现的。

控制寄存器

- 控制寄存器包括精度控制、舍入控制和无穷大控制，它也可以屏蔽或者不屏蔽与状态寄存器最右边 6 位对应的异常位。FLDCW 指令用于给控制寄存器赋值。
- 控制寄存器中各位及各个组合位的功能：
  1. IC　　无穷大控制（infinity control）
  2. RC　　舍入控制（rounding control）
  3. PC　　精度控制（precision control）
  4. Exception Masks 异常屏蔽字段

80387 控制寄存器


15~13
12
11~10
9~8
7~6
5
4
3
2
1
0
IC
RC
PC
PM
UM
OM
ZM
DM
IM
非法操作屏蔽
非规格化操作数屏蔽
除数为零屏蔽
上溢屏蔽
下溢屏蔽
精度错误屏蔽
异常屏蔽
精度控制
00：单精度（短型）
01：保留
10：双精度（长型）
11：扩展精度（临时）
舍入控制
00：舍入到最近或偶数
01：向下舍入到负无穷
10：向上舍入到正无穷
11：截为零
无穷大控制
0：投射
1：仿射

标记寄存器

- 标记寄存器（tag register）表明协处理器堆栈中每个寄存器内容的状态特征，又叫特征寄存器。标记寄存器用每 2 位表示寄存器堆栈中 1 个寄存器的状态，即 TAG（i）表示堆栈寄存器 ST（i）的状态。TAG（i）特征值为 00~11 四种组合时分别表明相应的寄存器有正确数据、数据为 0、数据非法、无数据 4 种情况。
- 通过程序查看标记寄存器的唯一方法是使用 FSTENV、FSAVE 或 FRSTOR 指令来存储协处理器操作环境。其中每条指令均可将标记寄存器与其他协处理器数据一起存储。

标记寄存器

15~14	13~12	11~10	9~8	7~6	5~4	3~2	1~0
TAG(7)	TAG(6)	TAG(5)	TAG(4)	TAG(3)	TAG(2)	TAG(1)	TAG(0)

TAG值：

00：合法

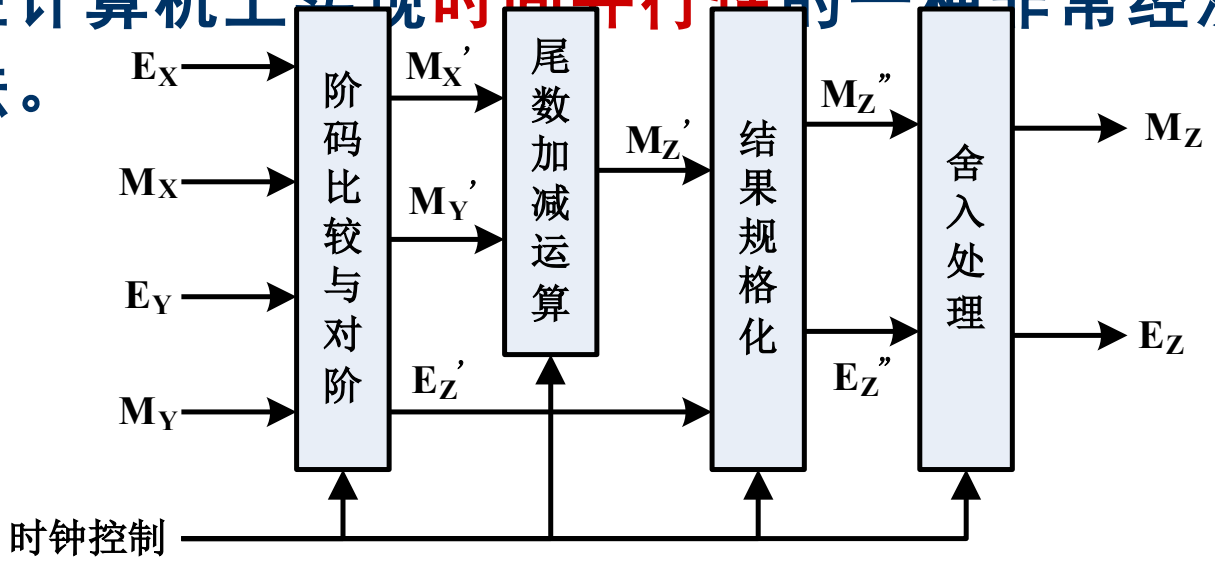
01：零

10：非法或无穷大

11：空

二、浮点运算流水线

- **流水线原理：把输入的任务分割为一系列的子任务，使各子任务能在流水线的各个阶段并发地执行。将任务连续不断地输入流水线，从而实现了子任务的并行。**
- **流水处理大幅度地改善了计算机的系统性能，是在计算机上实现时间并行性的一种非常经济的方法。**



二、浮点运算流水线

❖Pentium 浮点流水线

① 由预取 PF 、首次译码 D1 、二次译码 D2 、取操作数 EX 、首次执行 X1 、二次执行 X2 、浮点寄存器写入 WF 和出错报告 ER 共 8 个操作步骤组成。

② 由浮点接口、寄存器组及控制部件 FIRC 、浮点指数功能部件 FEXP 、浮点乘法部件 FMUL 、浮点加法部件 FADD 、浮点除法部件 FDIV 以及浮点舍入处理部件 PFRND 共 7 个部件组成。

二、浮点运算流水线

❖Pentium 浮点流水线

③ 浮点的取数、加减法、乘法和比较等“基本”操作，采用了新的算法并用硬件来实现，其执行速度是 80486 的 10 倍多，允许单周期通过，即能以每个时钟执行一条浮点指令的速度来执行。

④ 配置了直接支持 3 倍精度浮点计算的部件，极大地简化了微体系结构，并明显地改进了浮点部件的性能。

二、浮点运算流水线

❖Pentium 浮点流水线

⑤ 基于 NetBurst 微架构的 Pentium4 实现了被称作为流 SIMD 扩展 2 （ SSE2 ）的 144 条新 SIMD 指令，这些新指令支持 128 位的 SIMD 整数操作和 128 位 SIMD 双精度浮点操作。

⑥Core 构架拥有 2 个浮点执行单元同时处理向量和标量的浮点运算，其中一个浮点单元负责加减等简单的处理，而另一个浮点单元则负责乘除等运算。 Core 构架对浮点性能的改进效果是显而易见的。

本章小结

- 定点机器数的加减法运算：通过补码来实现
  - 补码的加减运算规则使得计算机中的减法转化为加法来运算，方便了硬件设计。
- 定点机器数的乘法运算
  - 乘法运算：原码一位乘法
  - 乘法器件可以采用基于串行乘法算法的乘法器，也可以采用高速的阵列乘法器。
- 定点机器数的除法运算
  - 除法运算：原码恢复余数除法、原码加减交替法
  - 除法的硬件实现中，陈列除法器大大地提高除法运算的速度。
- 浮点数的运算也均由定点数的运算复合而成。浮点运算器由阶码运算部件和尾数运算部件两部分构成。
- 本章重点为定点数和浮点数的运算方法。

作业

- P156：1，2，8（1），16

The End !