CPU 的流水线结构

龙文光
(内江师范学院 计算机系, 四川 内江 641112)

摘要: 根据流水线的基本原理, 阐述了 64 位 RISC CPU 的 5 级流水线结构和功能. 重点介绍了流水线的功能单元以及各单元的基本操作; 流水线暂停和异常的处理方法; 利用向前传送单元和数据回写单元提高流水线的效率.

关键词: 流水线; 异常; 暂停; 向前传送单元和数据回写单元
中图分类号: TP332　　文献标识码: A　　文章编号: 1001-8395(2003)03-0319-04

1　流水线概述

计算机执行程序是按顺序的方式进行的, 即程序中各条机器指令是按顺序串行执行的.

顺序执行的优点是控制简单, 但机器各部分利用率不高. 如果把两条指令或更多指令在时间上重叠起来进行, 将大幅提高程序的执行效率.

如果每个部件完成操作所需时间为 T, 那么每条指令的执行时间为 $4T$, 当每一条指令处理完后, 每隔 T 时间就能得到一条指令的处理结果, 平均数率提高了 4 倍. 如果把 5 条指令在时间上重叠起来进行, 就构成了 5 级流水线结构, 每级分两相位, 如图 1 所示[1].

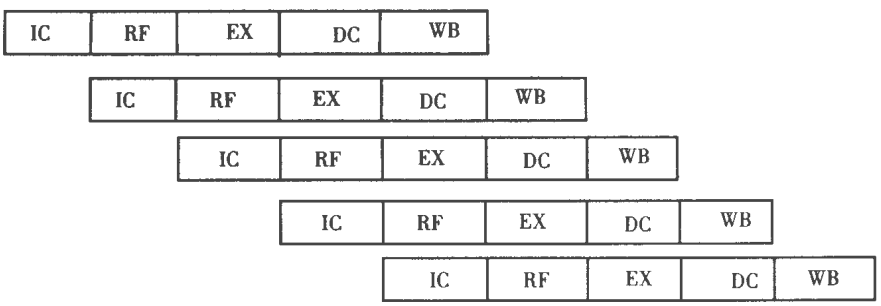

图 1　5 级流水线

对一条指令而言, CPU 从 IC 级开始操作, 从指令高速缓存取指令: RF 级从寄存器文件中取数据; EX 级执行程序; DC 级从数据高速缓访问数据; WB 级将执行结果写回到寄存器文件. 通过 5 级流水线操作, 能使 CPU 中同时有 5 条指令在同一时钟周期并行运行, 每一时钟周期执行 5 条指令, 成倍的提高 CPU 的速度.

2　流水线操作

CPU 指令分成 5 级来完成, 在流水线的每一级执行相应的操作. 在流水线的每一级, 都有相应的功能单元对指令或数据进行操作, 完成程序所要求的功能.

2.1　流水线功能单元　如图 2 所示.

(1) 下一条指令虚拟地址计算单元: 执行下一条指令虚拟地址计算.

(2) 程序计数寄存器, 锁存下一条指令的 PC 值, 保证正确的指令地址下被破坏.

(3) 指令存储器: 通过 PC 值指向相应的存储器地址, 取出指令.

(4) 指令寄存器: 保存从指令存储器中取出的指令.

(5) 寄存器文件: 有 32 个 64 位的通用寄存器. A 数据总线, B 数据总线表示 64 位的数据总线.

收稿日期: 2002-05-08
作者简介: 龙文光(1970-), 男, 讲师

(6) 将输出到ALU或数据存储器.

(7) ALU为算术逻辑运算单元.

(8) 数据存储器.

(9) 回写总线:将数据寄存器的数据或ALU的结果写回寄存器文件.

2.2 流水线操作 IC级操作:在相位1,生成取指令的虚拟地址,在时钟的下降缘将PC值锁入PC寄存器.在相位2,ICF—去指令高速缓存取指令;ITLB—同时指令地址超前转换缓冲器将虚拟地址转换为物理地址.

RF级操作:在相位1,物理地址的高位与指令cache的TAG标记比较,相同就将指令送到指令寄存器,否则产生ITLB异常.在相位2,从寄存器文件读取数据,用于EX级执行;交且同时指令译码器输出相应的控制信号.

EX级操作:在相位1,Branch指令判断,同时进行Branch指令地址计算;执行ALU,计算数据虚拟地址.

DC级操作:在相位1,读取数据高速缓存,同时在DTLB进行虚拟地址到物理地址的转换.

调整数据:物理地址的高位与数据cache的TAG标记比较,相同就将从数据高速缓存读取数据或存储数据到数据高速缓存,否则产生DTLB异常.

WB级操作:在相位1,回写数据到寄存器文件;存储数据到数据高速缓存.其流水线操作如图3所示.

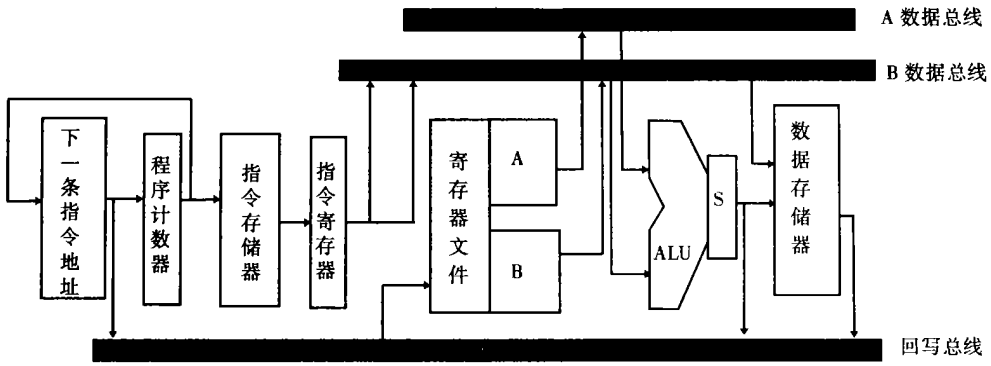

图2 流水线功能单元

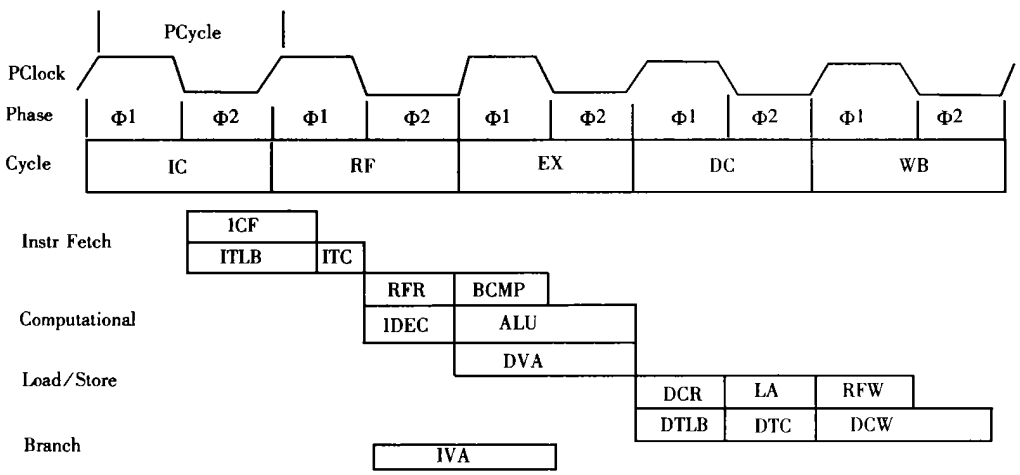

图3 流水线操作

3 流水线暂停和异常处理

当没在Cache找到所需指令或数据时,称为Cache没命中(miss);或者因为数据的相关性发生,则正常的CPU指令流水线将被中断.如果中断需用硬件来处理,则称为互锁(interlock),如果中断需用软件处理,则称为异常(exception).在每一时钟周期,所有在流水线的指令的异常和互锁的产生条件被检测,当条件满足,就在流水线的相应级产生异常和互锁.如果产生互锁,则暂停(stall)流水线;如果产生异常,则流水线将去掉(flush或abort)异常原因指令和该指令之后的所有指令[2].

3.1 流水线互锁 RF级互锁:ITM—指令TLB miss.ITLB是指指令地址超前转换器(instruction translation look-aside buffer)在ITLB没有找到与虚拟指令对应的物理地址对应的物理地址,流水线控制器产

生 stall 信号暂停流水线操作，ITLB 再从联合 TLB（joint TLB）查找. 这将产生 3 个时钟的暂停. 如果在 JTLB 没查找到，则产生 ITLB 异常. ICB—指令 cache 忙. 指令 cache miss，没能在 cache 中找到所需指令. 流水线控制器产生 stall 信号暂停流水线操作，直到整条 cache 槽被写进指令.

EX 级互锁：MCI—多周期指令互锁. 如果一条指令执行的时间超过一个时钟周期，如乘法指令或除法指令，则流水线控制器产生 stall 信号暂停流水线向前执行，直到乘法或除法完成. LDI-load 指令互锁. load 指令装载的数据被一下条指令所使用，则产生一个时钟周期的 stall.

DC 级互锁：DCM—数据 cache miss. 去数据 cache 取数据，发生数据 cache miss. 流水线控制器产生 stall 信号暂停流水线操作，直到整条 cache 槽被写进数据. DCM—数据 cache 忙. 如果 store 指令的下一条指令需要用数据 cahce，则产生 stall 信号. 如果去数据 cache 取数据，发生数据 cache miss，则产生 stall 信号. COP—cahce 指令操作产生的 stall 信号.

3.2　流水线异常　当流水线发生异常时，流水线将暂停 2 个时钟周期. 而使产生流水线异常的指令和该指令之后的指令将被去除掉. 在去除掉流水线指令后，流水线异常处理单元将按照预定义从一指定的地址取指令，进行异常处理[3].

流水线异常分为两大类：一类是与指令不相关的异常（复位，NMI 和中断）；一类是由流水线执行特定的指令引起的异常.

例如，CPU 流水线在执行存储指令：

```
lw   $2, 20( $5)   //load data,   load-exception happens
sw a3, 0x(a0)      //store data to data-cache
addiu a1, a1, 4    //modify load-address
addiu a0, a0, 4    //modify store-address
bnea1, a2, copyloop1//jump target address
```

lw 指令以寄存器 $5 的值加偏移量 20 为地址，将相应地址的存储器的值装载到寄存器 $2 中，其中%5 加 20 生成的地址是虚拟地址，必须通过 TLB 转换为物理地址. 如果在 TLB 中没有找到对应的物理地址，则产生 load TLB 异常.

当发生异常时，CPU 将进行异常处理，而 CPO 中的寄存器将保存现场数据. 其中 EPC 或 ErrorEPC 寄存器将保留返回的 PC 值，Cause 寄存器将保存异常原因，Status 寄存器将保存异常发生时的流水线状态，如图 4 所示.

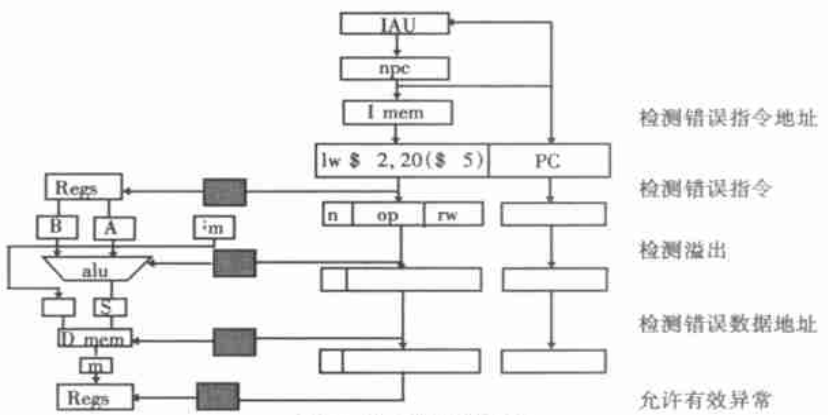

图 4　流水线异常框图

4　向前传送单元和数据回写单元

4.1　向前传送单元（forwarding unit）　有时，在流水线的 EX，DC 和 WB 级产生的数据或条件将用于下一条指令的 EX 级，通常执行单元是从寄存器文件取指令，这时，由于所需的数据没有准备就绪，流水线将产生数据危险（data hazard）. 如果使流水线处于等待状态，这就降低了流水线的运行效率. 因此，须通过向前传送单元直接将在流水线 EX 级，DC 级或 WB 级的数据或条件送达 EX 级，使 EX 级所需的操作数不再从寄存器文件读取，提高流水线的效率.

如图 5 所示，load 指令加载数据到 B 寄存器中，在正常操作时，load 指令需在 DC 级加载数据，在 WB 级写回到 B 寄存器中. 而现在在流水中，load 指令之后有一条 add A，B 指令在 EX 级需要用 load 指令的结果，因此产生数据危险，则需要使用向前传送单元将 load 指令的结果直接输入到 add A，B 指令的 EX 级.

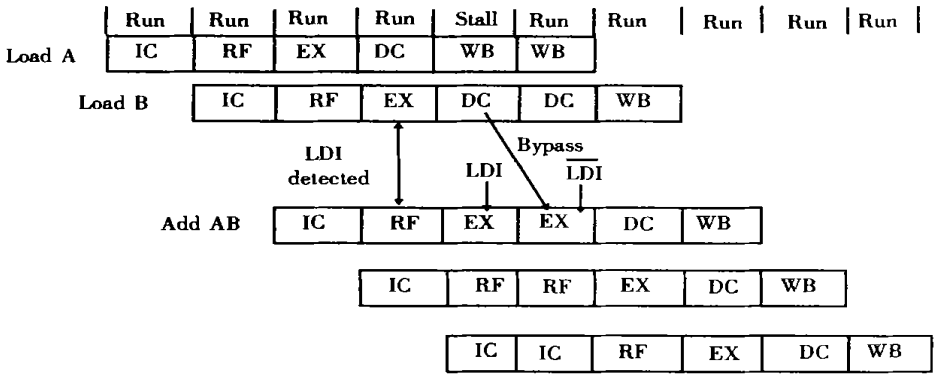

图 5 向前传送流程

4.2 数据回写单元(write-back unit) CPU有一片上回写缓冲器,作为临时的输出数据存储器.由于主存储器的速度比CPU和cache的速度慢,因此,在CPU向BIU输出数据或地址到外部存储器,则先将数据存放在回写缓冲器,至到有BIU申请时或BIU接收到命令信号,将数据通过BIU输出.它在一个时钟周期写入8个字节,共能存32个字节.有4个32位的物理地址,4位标记数据类型,4个8字节的存储区.

5 结语

流水线技术一开始就是在巨型机、大型机上应用,后来把这项技术运用于微处理器,这项技术的根本目的就是合理利用现有硬件,提高CPU处理能力.由于使用多种技术包括硬气件发展提高了微处理器的运算速度,同时也拓展了微型机的应用范围.随着硬气件的发展以后还会出现这种"自上向下"的发展,提高计算机的性能.

参考文献

[1] 李亚民. 计算机组成与系统结构[M]. 北京:清华大学出版社,2000.181~191.
[2] Stallings W. 计算机与结构,性能设计(第四版)[M]. 北京:清华大学出版社,1997.447~450.
[3] Mano M M. 计算机系统结构[M]. 北京:清华大学出版社,1998.302~310.

The Structure of Pipeline of CPU

LONG Weng-guang
(*Department of Computer Science, Neijiang Teacher's College, Neijiang* 641112, *Sichuan*)

Abstract: Based on the principle of pipeline, this paper analyzes the structure and function of five-deep pipeline of 64 bits and focuses on how to operate the functional unit, how to solve the stall and exception, and how to improve the efficiency with use of forwarding unit and write-back unit.

Key words: Pipeline; Exception; Stall; Forwarding unit write-back unit

(编辑 李德华)