

3. 假定某计算机中有一条转移指令，采用相对寻址方式，共占两个字节，第一字节是操作码，第二字节是相对位移量（用补码表示），CPU每次从内存只能取一个字节。假设执行到某转移指令时PC的内容为200，执行该转移指令后要求转移到100开始的一段程序执行，则该转移指令第二字节的内容应该是多少？
3. 假设地址为1200H的内存单元中的内容为12FCH，地址为12FCH的内存单元的内容为38B8H，而38B8H单元的内容为88F9H。说明以下各情况下操作数的有效地址和操作数各是多少？
- (1) 操作数采用变址寻址，变址寄存器的内容为12，指令中给出的形式地址为1200H。
 - (2) 操作数采用一次间接寻址，指令中给出的地址码为1200H。
 - (3) 操作数采用寄存器间接寻址，指令中给出的寄存器编号为8，8号寄存器的内容为1200H。
5. 通过查资料了解Intel 80x86微处理器和MIPS处理器中各自提供了哪些加法指令，说明每条加法指令的汇编形式、指令格式和功能，并比较加、减运算指令在这两种指令系统中不同的设计方式，包括不同的溢出处理方式。
6. 某计算机指令系统采用定长指令字格式，指令字长16位，每个操作数的地址码长6位。指令分二地址、单地址和零地址三类。若二地址指令有 k_2 条，无地址指令有 k_0 条，则单地址指令最多有多少条？
- 7.3. 有些计算机提供了专门的指令，能从32位寄存器中抽取其中任意一个位串置于一个寄存器的低位有效位上，并高位补0，如下图所示。MIPS指令系统中没有这样的指令，请写出最短的一个MIPS指令序列来实现这个功能，要求 $i=5, j=22$ ，操作前后的寄存器分别为 $\$s_0$ 和 $\$s_2$ 。
- 8.3. 以下程序段是某个过程对应的指令序列。入口参数int a和int b分别置于 $\$a_0$ 和 $\$a_1$ 中，返回参数是该过程的结果，置于 $\$v_0$ 中。要求为以下MIPS指令序列加注释，并简单说明该过程的功能。

```

                                add    $t0, $zero, $zero
loop:                          beq     $a1, $zero, finish
                                add     $t0, $t0, $a0
                                sub     $a1, $a1, 1
                                j        loop
finish:                        addi    $t0, $t0, 100
                                add     $v0, $t0, $zero

```

- 9.3. 下列指令序列用来对两个数组进行处理，并产生结果存放在 $\$v_0$ 中。假定每个数组有2500 个字，其数组下标为0到2499。两个数组的基地址分别存放在 $\$a_0$ 和 $\$a_1$ 中，数组长度分别存放在 $\$a_2$ 和 $\$a_3$ 中。要求为以下MIPS指令序列加注释，并简单说明该过程的功能。假定该指令序列运行在一个时钟频率为2GHz的处理器上，add、addi和sll指令的CPI为1；lw和bne指令的CPI为2，则最坏情况下运行所需时间是多少秒？

```

                                sll     $a2, $a2, 2
                                sll     $a3, $a3, 2
                                add     $v0, $zero, $zero
                                add     $t0, $zero, $zero
outer:                          add     $t4, $a0, $t0
                                lw      $t4, 0($t4)

```

```

            add    $t1, $zero, $zero
inner:      add    $t3, $a1, $t1
            lw     $t3, 0($t3)
            bne    $t3, $t4, skip
            addi   $v0, $v0, 1
            skip:  addi $t1, $t1, 4
            bne    $t1, $a3, inner
            addi   $t0, $t0, 4
            bne    $t0, $a2, outer

```

3. 假定在一个五级流水线（如图7.5所示）处理器中，各主要功能单元的操作时间为：存储单元：200ps；ALU和加法器：150ps；寄存器堆读口或写口：50ps。请问：

（1）若执行阶段EX所用的ALU操作时间缩短20%，则能否加快流水线执行速度？如果能的话，能加快多少？如果不能的话，为什么？

（2）若ALU操作时间增加20%，对流水线的性能有何影响？

（3）若ALU操作时间增加40%，对流水线的性能有何影响？

4. 假定某计算机工程师想设计一个新CPU，一个典型程序的核心模块有一百万条指令，每条指令执行时间为100ps。请问：

（1）在非流水线处理器上执行该程序需要花多长时间？

（2）若新CPU是一个20级流水线处理器，执行上述同样的程序，理想情况下，它比非流水线处理器快多少？

（3）实际流水线并不是理想的，流水段间数据传送会有额外开销。这些开销是否会影响指令执行时间（Instruction latency）和指令吞吐率（Instruction throughput）？

5. 假定最复杂的一条指令所用的组合逻辑分成6块，依次为A~F，其延迟分别为80ps、30ps、60ps、50ps、70ps、10ps。在这些组合逻辑块之间插入必要的流水段寄存器就可实现相应的指令流水线，寄存器延迟为20ps。理想情况下，以下各种方式所得到的时钟周期、指令吞吐率和指令执行时间各是多少？应该在哪里插入流水线寄存器？

（1）插入一个流水段寄存器，得到一个两级流水线

（2）插入两个流水段寄存器，得到一个三级流水线

（3）插入三个流水段寄存器，得到一个四级流水线

（4）吞吐量最大的流水线

6. 以下指令序列中，哪些指令对发生数据相关？假定采用“取指、译码/取数、执行、访存、写回”五段流水线方式，那么不用“转发”技术的话，需要在发生数据相关的指令前加入几条nop指令才能使这段程序避免数据冒险？如果采用“转发”是否可以完全解决数据冒险？不行的话，需要在发生数据相关的指令前加入几条nop指令才能使这段程序不发生数据冒险？

```

add    $s3, $s1, $s0
        sub    $t2, $s0, $s3
        lw     $t1, 0($t2)
        add    $t1, $t1, $t2

```

7. 假定以下MIPS指令序列在图7.18所示的流水线数据通路中执行：

```

addu    $s3, $s1, $s0
subu    $t2, $s0, $s3
lw      $t1, 0($t2)
add     $t3, $t1, $t2
add     $t1, $s4, $s5

```

请问：（1）上述指令序列中，哪些指令的哪个寄存器需要转发，转发到何处？

（2）上述指令序列中，是否存在load-use数据冒险？

（3）第5周期结束时，各指令执行状态是什么？哪些寄存器的数据正被读出？哪些寄存器将被写入？

8. 假定有一个程序的指令序列为“lw, add, lw, add, ...”。add指令仅依赖它前面的lw指令，而lw指令也仅依赖它前面的add指令，寄存器写口和寄存器读口分别在一个时钟周期的前、后半周期内独立工作。请问：（1）在带转发的五段流水线中执行该程序，其CPI为多少？

（2）在不带转发的五段流水线中执行该程序，其CPI为多少？

10. 在一个采用“取指、译码/取数、执行、访存、写回”的五段流水线中，若检测结果是否为“零”的操作在执行阶段进行，则分支延迟损失时间片（即分支延迟槽）为多少？以下一段MIPS指令序列中，在考虑数据转发的情况下，哪些指令执行时会发生流水线阻塞？各需要阻塞几个时钟周期？

```

1 loop:  add $t1, $s3, $s3
          2          add $t1, $t1, $t1
          3          add $t1, $t1, $s6
          4          lw   $t0, 0($t1)
          5          bne $t0, $s5, exit
          6          add $s3, $s3, $s4
          7          j     loop
          8          exit:

```

11. 假设数据通路中各主要功能单元的操作时间为：存储单元：200ps；ALU和加法器：100ps；寄存器堆读口或写口：50ps。程序中指令的组成比例为：取数25%、存数10%、ALU52%、分支11%、跳转2%。假设时钟周期取存储器存取时间的一半，MUX、控制单元、PC、扩展器和传输线路等的延迟都忽略不计，则下面的实现方式中，哪个更快？快多少？

- (1) 单周期方式：每条指令在一个固定长度的时钟周期内完成；
- (2) 多周期方式：每类指令时钟数：取数-7，存数-6，ALU-5，分支-4，跳转-4；
- (3) 流水线方式：取指1、取指2、取数/译码、执行、存取1、存取2、写回7段流水线；没有结构冒险；数据冒险采用“转发”技术处理；load指令与后续各指令之间存在依赖关系的概率分别1/2、1/4、1/8、...；分支延迟损失时间片为2，预测准确率为75%；不考虑异常、中断和访问失效引起的流水线冒险。

12. 假设有一段程序的核心模块中有五条分支指令，该模块将会被执行成千上万次，在其中一次执行过程中，五条分支指令的实际执行情况如下（T：Taken；N：not Taken）。

分支指令1 (B1)：T-T-T。

分支指令2 (B2)：N-N-N-N。

分支指令3 (B3)：T-N-T-N-T-N。

分支指令4 (B4)：T-T-T-N-T。

分支指令5 (B5)：T-T-N-T-T-N-T。

假定各个分支指令在每次模块执行过程中实际执行情况都一样，并且动态预测时，每个分支指令都有各自的预测表项，每次执行时的初始预测位都相同。请给出以下几种预测方案的预测准确率。

- (1) 静态预测，总是预测转移（Taken）。
- (2) 静态预测，总是预测不转移（not Taken）。
- (3) 一位动态预测，初始预测转移（Taken）。
- (4) 二位动态预测，初始预测弱转移（Taken）。