第3章 流水线技术

3.1解释下列术语

**流水线：**将一个重复的时序过程，分解成为若干个子过程，而每一个子过程都可有效地在其专用功能段上与其它子过程同时执行。

单功能流水线：指流水线的各段之间的连接固定不变、只能完成一种固定功能的流水线。

多功能流水线：指各段可以进行不同的连接，以实现不同的功能的流水线。

静态流水线：指在同一时间内，多功能流水线中的各段只能按同一种功能的连接方式工作的流水线。当流水线要切换到另一种功能时，必须等前面的任务都流出流水线之后，才能改变连接。

动态流水线：指在同一时间内，多功能流水线中的各段可以按照不同的方式连接，同时执行多种功能的流水线。它允许在某些段正在实现某种运算时，另一些段却在实现另一种运算。

部件级流水线：把处理机中的部件进行分段，再把这些部件分段相互连接而成。它使得运算操作能够按流水方式进行。这种流水线也称为运算操作流水线。

处理机级流水线：又称指令流水线。它是把指令的执行过程按照流水方式进行处理，即把一条指令的执行过程分解为若干个子过程，每个子过程在独立的功能部件中执行。

处理机间流水线：又称为宏流水线。它是把多个处理机串行连接起来，对同一数据流进行处理，每个处理机完成整个任务中的一部分。前一个处理机的输出结果存入存储器中，作为后一个处理机的输入。

线性流水线：指各段串行连接、没有反馈回路的流水线。数据通过流水线中的各段时，每一个段最多只流过一次。

非线性流水线：指各段除了有串行的连接外，还有反馈回路的流水线。

顺序流水线：流水线输出端任务流出的顺序与输入端任务流入的顺序完全相同。

乱序流水线：流水线输出端任务流出的顺序与输入端任务流入的顺序可以不同，允许后进入流水线的任务先完成。这种流水线又称为无序流水线、错序流水线、异步流水线。

**吞吐率：**在单位时间内流水线所完成的任务数量或输出结果的数量。

**流水线的加速比：**使用顺序处理方式处理一批任务所用的时间与按流水处理方式处理同一批任务所用的时间之比。

**流水线的效率：**即流水线设备的利用率，它是指流水线中的设备实际使用时间与整个运行时间的比值。

**数据相关：**考虑两条指令*i*和*j*，*i*在*j*的前面，如果下述条件之一成立，则称指令*j*与指令*i*数据相关：

（1）指令*j*使用指令*i*产生的结果；

（2）指令*j*与指令*k*数据相关，而指令*k*又与指令*i*数据相关。

**名相关：**如果两条指令使用了相同的名，但是它们之间并没有数据流动，则称这两条指令存在名相关。

**控制相关：**是指由分支指令引起的相关。它需要根据分支指令的执行结果来确定后面该执行哪个分支上的指令。

反相关：考虑两条指令*i*和*j*，*i*在*j*的前面，如果指令*j*所写的名与指令*i*所读的名相同，则称指令*i*和*j*发生了反相关。

输出相关：考虑两条指令*i*和*j*，*i*在*j*的前面，如果指令*j*和指令*i*所写的名相同，则称指令*i*和*j*发生了输出相关。

换名技术：名相关的两条指令之间并没有数据的传送，只是使用了相同的名。可以把其中一条指令所使用的名换成别的，以此来消除名相关。

**结构冲突：**因硬件资源满足不了指令重叠执行的要求而发生的冲突。

**数据冲突：**当指令在流水线中重叠执行时，因需要用到前面指令的执行结果而发生的冲突。

**控制冲突：**流水线遇到分支指令或其它会改变PC值的指令所引起的冲突。

定向：用来解决写后读冲突的。在发生写后读相关的情况下，在计算结果尚未出来之前，后面等待使用该结果的指令并不见得是马上就要用该结果。如果能够将该计算结果从其产生的地方直接送到其它指令需要它的地方，那么就可以避免停顿。

写后读冲突：考虑两条指令i和j，且i在j之前进入流水线，指令j用到指令i的计算结果，而且在i将结果写入寄存器之前就去读该寄存器，因而得到的是旧值。

读后写冲突：考虑两条指令i和j，且i在j之前进入流水线，指令j的目的寄存器和指令i的源操作数寄存器相同，而且j在i读取该寄存器之前就先对它进行了写操作，导致i读到的值是错误的。

写后写冲突：考虑两条指令i和j，且i在j之前进入流水线，，指令j和指令i的结果单元（寄存器或存储器单元）相同，而且j在i写入之前就先对该单元进行了写入操作，从而导致写入顺序错误。这时在结果单元中留下的是i写入的值，而不是j写入的。

**链接技术：**具有先写后读相关的两条指令，在不出现功能部件冲突和Vi冲突的情况下，可以把功能部件链接起来进行流水处理，以达到加快执行的目的。

**分段开采：**当向量的长度大于向量寄存器的长度时，必须把长向量分成长度固定的段，然后循环分段处理，每一次循环只处理一个向量段。

半性能向量长度：向量处理机的性能为其最大性能的一半时所需的向量长度。

向量长度临界值：向量流水方式的处理速度优于标量串行方式的处理速度时所需的向量长度的最小值。

**3.2 简述流水线技术的特点。**

答：流水技术有以下特点：

（1） 流水线把一个处理过程分解为若干个子过程，每个子过程由一个专门的功能部件来实现。因此，流水线实际上是把一个大的处理功能部件分解为多个独立的功能部件，并依靠它们的并行工作来提高吞吐率。

（2） 流水线中各段的时间应尽可能相等，否则将引起流水线堵塞和断流。

（3） 流水线每一个功能部件的前面都要有一个缓冲寄存器，称为流水寄存器。

（4） 流水技术适合于大量重复的时序过程，只有在输入端不断地提供任务，才能充分发挥流水线的效率。

（5） 流水线需要有通过时间和排空时间。在这两个时间段中，流水线都不是满负荷工作。

**3.3 解决流水线瓶颈问题有哪两种常用方法？**

答：细分瓶颈段与重复设置瓶颈段

**3.4 减少流水线分支延迟的静态方法有哪些？**

答：（1）预测分支失败：沿失败的分支继续处理指令，就好象什么都没发生似的。当确定分支是失败时，说明预测正确，流水线正常流动；当确定分支是成功时，流水线就把在分支指令之后取出的指令转化为空操作，并按分支目标地址重新取指令执行。

（2）预测分支成功**：**当流水线ID段检测到分支指令后，一旦计算出了分支目标地址，就开始从该目标地址取指令执行。

（3）延迟分支：主要思想是从逻辑上“延长”分支指令的执行时间。把延迟分支看成是由原来的分支指令和若干个延迟槽构成。不管分支是否成功，都要按顺序执行延迟槽中的指令。

3种方法的共同特点：它们对分支的处理方法在程序的执行过程中始终是不变的。它们要么总是预测分支成功，要么总是预测分支失败。

**3.5 简述延迟分支方法中的三种调度策略的优缺点。**

|  |  |  |
| --- | --- | --- |
| 调度策略 | 对调度的要求 | 对流水线性能改善的影响 |
| 从前调度 | 分支必须不依赖于被调度的指令 | 总是可以有效提高流水线性能 |
| 从目标处调度 | 如果分支转移失败，必须保证被调度的指令对程序的执行没有影响，可能需要复制被调度指令 | 分支转移成功时，可以提高流水线性能。但由于复制指令，可能加大程序空间 |
| 从失败处调度 | 如果分支转移成功，必须保证被调度的指令对程序的执行没有影响 | 分支转移失败时，可以提高流水线性能 |

**3.6列举出下面循环中的所有相关，包括输出相关、反相关、真相关。**

for (i=2; i<100; i=i+1)

a[i]=b[i]+a[i] ;/\* s1 \*/

c[i+1]=a[i]+d[i] ; /\* s2 \*/

a[i-1]=2\*b[i] ; /\* s3 \*/

b[i+1]=2\*b[i] ;/\* s4 \*/

解：展开循环两次：

a[i] = b[i] + a[i] ; /\* s1 \*/

c[i+1] = a[i] + d[i] ; /\* s2 \*/

a[i-1] = 2 \* b[i] ; /\* s3 \*/

b[i+1] = 2 \* b[i] ; /\* s4 \*/

a[i+1] = b[i+1] + a[i+1] ; /\* s1’ \*/

c[i+2] = a[i+1] + d[i+1] ; /\* s2 ‘\*/

a[i] = 2 \* b[i+1] ; /\* s3 ‘\*/

b[i+2] = 2 \* b[i+1] ; /\* s4 ‘\*/

输出相关：无

反相关：无

真相关：S1&S2

由于循环引入的相关：S4&S4’（真相关）、S1’&S4（真相关）、S3’&S4（真相关）、S1&S3’（输出相关、反相关）、S2&S3’（反相关）。

3.7 简述三种向量处理方式，它们对向量处理机的结构要求有何不同？

答 (1)横向处理方式：若向量长度为N，则水平处理方式相当于执行N次循环。若使用流水线，在每次循环中可能出现数据相关和功能转换，不适合对向量进行流水处理。 (2)纵向处理方式：将整个向量按相同的运算处理完毕之后，再去执行其他运算。适合对向量进行流水处理，向量运算指令的源/目向量都放在存储器内，使得流水线运算部件的输入、输出端直接与存储器相联，构成M-M型的运算流水线。 (3)纵横处理方式：把长度为N的向量分为若干组，每组长度为n，组内按纵向方式处理，依次处理各组，组数为「N/n」，适合流水处理。可设长度为n的向量寄存器，使每组向量运算的源/目向量都在向量寄存器中，流水线的运算部件输入、输出端与向量寄存器相联，构成R-R型运算流水线。

3.8 可采用哪些方法来提高向量处理机的性能？

答：可采用多种方法：

1. 设置多个功能部件，使它们并行工作；
2. 采用链接技术，加快一串向量指令的执行；
3. 采用循环开采技术，加快循环的处理；
4. 采用多处理机系统，进一步提高性能。

**3.9 有一指令流水线如下所示**



1. 求连续输入10条指令，该流水线的实际吞吐率和效率；
2. 该流水线的“瓶颈”在哪一段？请采取两种不同的措施消除此“瓶颈”。对于你所给出的两种新的流水线，连续输入10条指令时，其实际吞吐率和效率各是多少？

解：（1）









=10\*（50+50+100+200）/4\*（50+50+100+200+9\*200）

（2）瓶颈在3、4段。

* 变成八级流水线（细分）









* 重复设置部件

1

2

3-1

3-2

4-1

4-2

4-3

4-4







**3.10有一个**流水线由4段组成，其中每当流经第3段时，总要在该段循环一次，然后才能流到第4段。如果每段经过一次所需要的时间都是，问：

1. 当在流水线的输入端连续地每时间输入任务时，该流水线会发生什么情况？
2. 此流水线的最大吞吐率为多少？如果每输入一个任务，连续处理10个任务时的实际吞吐率和效率是多少？
3. 当每段时间不变时，如何提高该流水线的吞吐率？仍连续处理10个任务时，其吞吐率提高多少？

解：（1）会发生流水线阻塞情况。

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 第1个任务 | S1 | S2 | S3 | S3 | S4 |  |  |  |  |  |  |
| 第2个任务 |  | S1 | S2 | stall | S3 | S3 | S4 |  |  |  |  |
| 第3个任务 |  |  | S1 | stall | S2 | stall | S3 | S3 | S4 |  |  |
| 第4个任务 |  |  |  |  | S1 | stall | S2 | stall | S3 | S3 | S4 |

（2）





（3）重复设置部件





吞吐率提高倍数＝＝1.64

**3.11 有一条静态多**功能流水线由5段组成，加法用1、3、4、5段，乘法用1、2、5段，第3段的时间为2△t，其余各段的时间均为△t，而且流水线的输出可以直接返回输入端或



暂存于相应的流水寄存器中。现要在该流水线上计算 ，画出其时空图，并计算其吞吐率、加速比和效率。



解：首先，应选择适合于流水线工作的算法。对于本题，应先计算A1＋B1、A2＋B2、A3＋B3和A4＋B4；再计算(A1＋B1) ×(A2＋B2)和(A3＋B3) ×(A4＋B4)；然后求总的结果。

其次，画出完成该计算的时空图，如图所示，图中阴影部分表示该段在工作。



由图可见，它在18个△*t*时间中，给出了7个结果。所以吞吐率为：

**

如果不用流水线，由于一次求积需3△*t*，一次求和需5△*t*，则产生上述7个结果共需（4×5+3×3）△*t* =29△*t*。所以加速比为：



该流水线的效率可由阴影区的面积和5个段总时空区的面积的比值求得：



**3.12 动态多功能流水线**由6个功能段组成，如下图：



其中，S1、S4、S5、S6组成乘法流水线，S1、S2、S3、S6组成加法流水线，各个功能段时间均为50ns，假设该流水线的输出结果可以直接返回输入端，而且设置有足够的缓冲寄存器，若以最快的方式用该流水计算：

1. 画出时空图；
2. 计算实际的吞吐率、加速比和效率。

解：机器一共要做10次乘法，4次加法。



**3.13 在MIPS流水线上运行如下代码序列：**

LOOP： LW R1，0（R2）

DADDIU R1，R1，#1

SW R1， 0（R2）

DADDIU R2，R2，#4

DSUB R4，R3，R2

BNEZ R4，LOOP

其中：R3的初值是R2+396。假设：在整个代码序列的运行过程中，所有的存储器访问都是命中的，并且在一个时钟周期中对同一个寄存器的读操作和写操作可以通过寄存器文件“定向”。问：

1. 在没有任何其它定向（或旁路）硬件的支持下，请画出该指令序列执行的流水线时空图。假设采用排空流水线的策略处理分支指令，且所有的存储器访问都命中Cache，那么执行上述循环需要多少个时钟周期？
2. 假设该流水线有正常的定向路径，请画出该指令序列执行的流水线时空图。假设采用预测分支失败的策略处理分支指令，且所有的存储器访问都命中Cache，那么执行上述循环需要多少个时钟周期？
3. 假设该流水线有正常的定向路径和一个单周期延迟分支，请对该循环中的指令进行调度，你可以重新组织指令的顺序，也可以修改指令的操作数，但是注意不能增加指令的条数。请画出该指令序列执行的流水线时空图，并计算执行上述循环所需要的时钟周期数。

解：

寄存器读写可以定向，无其他旁路硬件支持。排空流水线。



第i次迭代（i＝0..98）开始周期：1＋（i×17）

总的时钟周期数：1+（98×17）＋18-1（因与上次循环重叠1个周期，以下同）＝1684

有正常定向路径，预测分支失败。



第i次迭代（i＝0..98）开始周期：1＋（i×10）

总的时钟周期数：1+（98×10）＋11-1＝991

有正常定向路径。单周期延迟分支。

LOOP: LW R1，0(R2)

DADDIU R2，R2，#4

DADDIU R1，R1，#1

DSUB R4，R3，R2

BNEZ R4，LOOP

SW R1，-4(R2)

第i次迭代（i ＝0..98）开始周期：1＋（i ×6 ）

总的时钟周期数：1+（98×6）＋10-1＝598



**3.14 假设各种分支指令数占所有指令数的百分比如下：（书3.14）**

|  |  |
| --- | --- |
| 条件分支 | 20%（其中的60%是分支成功的） |
| 跳转和调用 | 5% |

现有一条段数为4的流水线，无条件分支在第二个时钟周期结束时就被解析出来，而条件分支要到第三个时钟周期结束时才能够被解析出来。第一个流水段是完全独立于指令类型的，即所有类型的指令都必须经过第一个流水段的处理。请问在没有任何控制相关的情况下，该流水线相对于存在上述控制相关情况下的加速比是多少？

解：没有控制相关时流水线的平均CPI＝1

存在控制相关时：由于无条件分支在第二个时钟周期结束时就被解析出来，而条件分支

要到第3个时钟周期结束时才能被解析出来。所以：

（1）若使用排空流水线的策略，则对于条件分支，有两个额外的stall，对无条件分支，有一个额外的stall：

CPI = 1+20%\*2+5%\*1 = 1.45

加速比S=CPI/1 = 1.45

（2） 若使用预测分支成功策略，则对于不成功的条件分支，有两个额外的stall，对无条件分支和成功的条件分支，有一个额外的stall 1：

CPI = 1+20%\*(60%\*1+40%\*2) +5%\*1 = 1.33

加速比S=CPI/1 = 1.33

（3）若使用预测分支失败策略，则对于成功的条件分支，有两个额外的stall；对无条件分支，有一个额外的stall；对不成功的条件分支，其目标地址已经由PC 值给出，不必等待，所以无延迟：

CPI = 1+20%\*(60%\*2 + 40%\*0) +5%\*1 = 1.29

加速比S=CPI/1 = 1.29

**3.15 在CRAY-1机器上**，按照链接方式执行下述4条向量指令（括号中给出了相应功能部件的执行时间），如果向量寄存器和功能部件之间的数据传送需要1拍，试求此链接流水线的通过时间是多少拍？如果向量长度为64，则需多少拍才能得到全部结果？

V0←存储器 （从存储器中取数：7拍）

V2←V0+V1 （向量加：3拍）

V3←V2<A3 （按（A3）左移：4拍）

V5←V3∧V4 （向量逻辑乘：2拍）

解：通过时间就是每条向量指令的第一个操作数执行完毕需要的时间，也就是各功能流水线由空到满的时间，具体过程如下图所示。要得到全部结果，在流水线充满之后，向量中后继操作数继续以流水方式执行，直到整组向量执行完毕。





**3.16 某向量处理机**有16个向量寄存器，其中V0~V5中分别放有向量A、B、C、D、E、F，向量长度均为8，向量各元素均为浮点数；处理部件采用两条单功能流水线，加法功能部件时间为2拍，乘法功能部件时间为3拍。采用类似于CARY-1的链接技术，先计算（A+B）\*C，在流水线不停流的情况下，接着计算（D+E）\*F。

1. 求此链接流水线的通过时间？（设寄存器入、出各需1拍）
2. 假如每拍时间为50ns，完成这些计算并把结果存进相应寄存器，此处理部件的实际吞吐率为多少MFLOPS？

解：（1）我们在这里假设A＋B的中间结果放在V6中，（A＋B）×C地最后结果放在V7中，D＋E地中间结果放在V8中，（D＋E）×F的最后结果放在V9中。具体实现参考下图：



通过时间应该为前者（（A＋B）×C）通过的时间：

T通过= (1+2+1)+(1+3+1) =9（拍）

（2）在做完（A＋B）×C之后，作（C＋D）×E就不需要通过时间了。

V6←A＋B

V7←V6×C

V8←D＋E

V9←V8×F

