第一次作业

1、 某台主频为 400MHz 的计算机执行标准测试程序，程序中指令类型、执行数量和平均时钟周期数如下：

| 指令类型 | 指令数 | 时钟周期数 |
| --- | --- | --- |
| 整数运算 | 45000 | 1 |
| 数据传送 | 35000 | 2 |
| 浮点 | 15000 | 2 |
| 控制传送 | 5000 | 4 |

求该计算机的有效 CPI、MIPS 和程序执行时间。

解题思路：CPI是指令的平均时钟周期数，套用公式： $CPI = \sum_{i=1}^{n}(CPI_i \times IC_i / IC)$

MIPS是指每秒（百万条）指令数，套用公式：MIPS=f/(CPI*1000000)

1、 某台主频为 400MHz 的计算机执行标准测试程序，程序中指令类型、执行数量和平均时钟周期数如下：

| 指令类型 | 指令数 | 时钟周期数 |
| --- | --- | --- |
| 整数运算 | 45000 | 1 |
| 数据传送 | 35000 | 2 |
| 浮点 | 15000 | 2 |
| 控制传送 | 5000 | 4 |

求该计算机的有效 CPI、MIPS 和程序执行时间。

答案：
CPI=（45000*1+35000*2+15000*2+5000*4）/（45000+35000+15000+5000）=1.65（周期/指令）
MIPS=400*10^6/（1.65*10^6）=242.42（百万条指令/秒）
程序执行时间 Tcpu=（45000*1+35000*2+15000*2+5000*4）/（400*10^6）=0.0004125（s）

2、将计算机系统中某一功能的处理速度加快 20 倍，但该功能的处理时间仅为整个系统运行时间的 50%，则采用此增强功能方法后，能使整个系统的性能提高多少？

解题思路：

Amdahl定律:加快某部件执行速度所获得的系统性能(加速比)，与该部件在系统中总执行时间的比例有关。

系统加速比定义

$$加速比=\frac{系统性能_{改进后}}{系统性能_{改进前}}=\frac{总执行时间_{改进前}}{总执行时间_{改进后}}$$

系统性能加速比与该部件在系统中的总执行时间有关。

单部件加速主要依赖于两个因素：

部件加速比（$S_e$）：部件改进前的执行时间与改进后执行时间的比值 。 $S_e$ 通常大于1。

可改进比例（Fe）：可改部件的执行时间在总的执行时间中的比例总是小于等于1。。

2、将计算机系统中某一功能的处理速度加快 20 倍，但该功能的处理时间仅为整个系统运行时间的 50%，则采用此增强功能方法后，能使整个系统的性能提高多少？

解题思路：

（改进后）改进部分的时间缩短为原来的1/ Se，不可改进的部分（1- Fe ）执行时间没有改变，故改进后程序的总执行时间Tn

$$T_n = T_0\left(1 - Fe + \frac{Fe}{Se}\right)$$

即

$$S_n = \frac{T_0}{T_n} = \frac{1}{(1 - Fe) + \frac{Fe}{Se}}$$

答案：由题可知： 可改进比例 = 50% = 0.5 部件加速比 = 20

根据Amdahl定律可知：

Sn=1/[（1-0.5）+（0.5/20）]=1.905

3、计算机系统中有三个部件可以改进，这三个部件的部件加速比为： 部件1加速比 $S1=30$；部件2加速比 $S2=20$；部件3加速比 $S3=10$

（1）如果三个部件的可改进比例分别为 30%、30%和 20%，三个部件同时改进，那么系统中不可加速部分的执行时间在总执行时间中占的比例是多少？

（2）如果部件 1 和部件 2 的可改进比例均为 30%，那么当部件 3 的可改进比 例为多少时，系统加速比才可以达到 10？

解题思路：

与上题类似，

$$\text{加速比}=\frac{\text{系统性能}_{\text{改进后}}}{\text{系统性能}_{\text{改进前}}}=\frac{\text{总执行时间}_{\text{改进前}}}{\text{总执行时间}_{\text{改进后}}}$$

对于多部件的Amdahl定理，不可加速部分的执行时间为 $(1-\sum_i Fi)$

改进后程序的总执行时间Tn等于 $(1-\sum_i Fi)+\sum_i \frac{Fi}{Si}$

3、计算机系统中有三个部件可以改进，这三个部件的部件加速比为： 部件1加速比 S1＝30；部件2加速比 S2＝20；部件3加速比 S3＝10

（1）如果三个部件的可改进比例分别为 30%、30%和 20%，三个部件同时改进，那么系统中不可加速部分的执行时间在总执行时间中占的比例是多少？（2）如果部件 1 和部件 2 的可改进比例均为 30%，那么当部件 3 的可改进比 例为多少时，系统加速比才可以达到 10？

答案：（1）设系统改进前的执行时间为 T，不可改进部分的执行时$(1-Fe1-Fe2-Fe3)=0.2T$。

已知 3 个部件改进后的加速比分别为 S1＝30，S2＝20，S3＝10，因此 3 个部件 改进后的执行时间为：

改进后整个系统的执行时间为：$\mathbf{Tn}=(1-Fe1-Fe2-Fe3)+\dfrac{Fe1}{Se1}+\dfrac{Fe2}{Se2}+\dfrac{Fe3}{Se3}$

$$=0.045T+0.2T=0.245T$$

那么系统中不可改进部分的执行时间在总执行时间中占的比例是：0.2/0.245=81.6%

3、计算机系统中有三个部件可以改进，这三个部件的部件加速比为： 部件1加速比 S1＝30；部件2加速比 S2＝20；部件3加速比 S3＝10

（1）如果三个部件的可改进比例分别为 30%、30%和 20%，三个部件同时改进，那么系统中不可加速部分的执行时间在总执行时间中占的比例是多少？（2）如果部件 1 和部件 2 的可改进比例均为 30%，那么当部件 3 的可改进比 例为多少时，系统加速比才可以达到 10？

解题思路：由上问已知改进后程序的总执行时间Tn，可得 $S_n=\frac{T_0}{T_n}$

即 $$S=\frac{1}{(1-\sum_i Fi)+\sum_i \frac{Fi}{Si}}$$

答案：（2）在多个部件可改进情况下，Amdahl 定理的扩展

$$S_n=\frac{T_0}{T_n}=\frac{1}{(1-Fe1-Fe2-Fe3)+\frac{Fe1}{Se1}+\frac{Fe2}{Se2}+\frac{Fe3}{Se3}}$$

已知 S1＝30，S2＝20，S3＝10，Sn＝10，F1＝0.3，F2＝0.3，得： 得 F3＝0.36，即部件 3 的可改进比例为 36%。

4、假设浮点数指令FP指令的比例为 30%，其中浮点数平方根FPSQR占全部指令的比例为4%，FP操作的CPI为5，FPSQR操作的CPI为 20，其他指令的平均CPI为1.25。现有两种改进方案，第一种是把FPSQR操作的CPI减至3，第二种是把所有的FP操作的CPI减至3，比较两种方案对系统性能的提高程度

解题思路：

通过调整前后的CPI值进行比较，采取公式： $CPI = \sum_{i=1}^{n}(CPI_i \times IC_i / IC)$

针对那种操作进行的修改，对系统性能的提高可以通过观察对CPI的影响来进行判断

第一种方案是把FPSQR操作的CPI减至3，每次FPSQR操作的CPI减少17，而其中浮点数平方根FPSQR占全部指令的比例为4%。通过计算可以得出改进的程度

第二种方案是把所有的FP操作的CPI减至3，每次FP操作的CPI减少2，而其中浮点数指令FP指令的比例为 30%，通过计算可以得出改进的程度

4、假设浮点数指令FP指令的比例为 30%，其中浮点数平方根FPSQR占全部指令的比例为4%，FP操作的CPI为5，FPSQR操作的CPI为 20，其他指令的平均CPI为1.25。现有两种改进方案，第一种是把FPSQR操作的CPI减至3，第二种是把所有的FP操作的CPI减至3，比较两种方案对系统性能的提高程度

答案：改进前：CPI=5*30%+1.25*（1-30%）=2.375

方案 1：（20-3）*4%=0.68

方案 2：（5-3）*30%=0.6

0.68>0.6

所以方案一提升较大

5、考虑条件分支指令的两种不同设计方法：（1）CPU1：对比较指令设置条件码，然后测试条件码进行分支。（2）CPU2：在分支指令中包括比较过程。假设在两种CPU中，条件分支指令都占用2个时钟周期，所有其他指令占用1个时钟周期。对CPU1，执行的指令中分支指令占20%；每条分支指令之前都需要有比较指令，即比较指令也占20%。由于CPU1在分支时不需要比较，因此CPU2的时钟周期时间时CPU1的1.3倍。问：哪一个方案的CPU更快?

解题思路：

总CPU时间=CPI*IC*时钟周期

对于CPU1来说，执行的指令中分支指令占20%，且都占用2个时钟周期，剩余指令占80%，所有剩余指令占用1个时钟周期，可以通过这个计算CPI1。

对于CPU2来说，由于CPU2没有独立的比较指令，且CPU中的比较指令占20%，所以CPU2的指令数量为CPU1的80%，即 IC2=0.8IC1，分支指令占比则变为0.2/0.8=0.25，剩余指令占0.75，可以通过这个计算CPI2.

由题可知， CPU2的时钟周期时间时CPU1的1.3倍。即T2=1.3T1

通过对总CPU时间计算可以得出结论

5、考虑条件分支指令的两种不同设计方法：（1）CPU1：对比较指令设置条件码，然后测试条件码进行分支。（2）CPU2：在分支指令中包括比较过程。假设在两种CPU中，条件分支指令都占用2个时钟周期，所有其他指令占用1个时钟周期。对CPU1，执行的指令中分支指令占20%；每条分支指令之前都需要有比较指令，即比较指令也占20%。由于CPU1在分支时不需要比较，因此CPU2的时钟周期时间时CPU1的1.3倍。问：哪一个方案的CPU更快？

对于CPU1，其CPI为：

$$CPI_1 = 0.2 * 2 + 0.8 * 1 = 1.2$$

则CPU1的性能为：（设T1为CPU1时钟周期）

$$总CPU时间_1 = 1.2 * IC_1 * T_1$$

CPU2的分支指令比例和其CPI为：

$$20\%/80\% = 25\%$$

$$CPI_2 = 0.25 * 2 + 0.75 * 1 = 1.25$$

由于CPU2不执行比较，那么

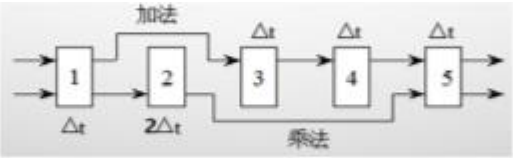
$$IC_2 = 0.8 * IC_1$$

那么CPU2的总性能为

$$总CPU时间_2 = IC_2 * 1.25 * T_2 = 0.8 * IC_1 * 1.25 * 1.3\, T_1 = 1.3\, IC_1 * T_1$$

# 第二次作业

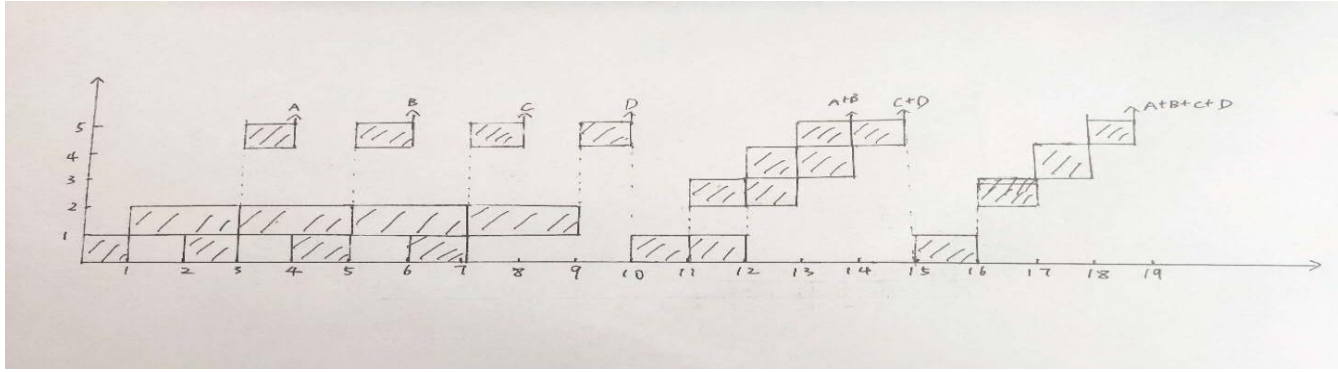
1、有一条静态多功能流水线由 5 段组成，加法用 1、3、4、5 段，乘法用 1、2、5 段，第 2 段的时间为 2△t，其余各段的时间均为△t，而且流水线的输出可以直接返回输入端或暂存于相应的流水寄存器中。现要在该流水线上计算 $\sum_{i=1}^{4}(A_i \times B_i)$ ，画出其时空图，并计算其吞吐率、加速比和效率



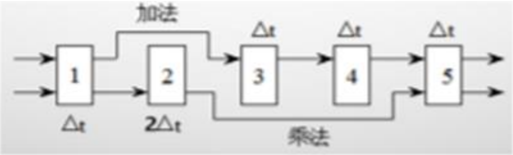

解题思路：

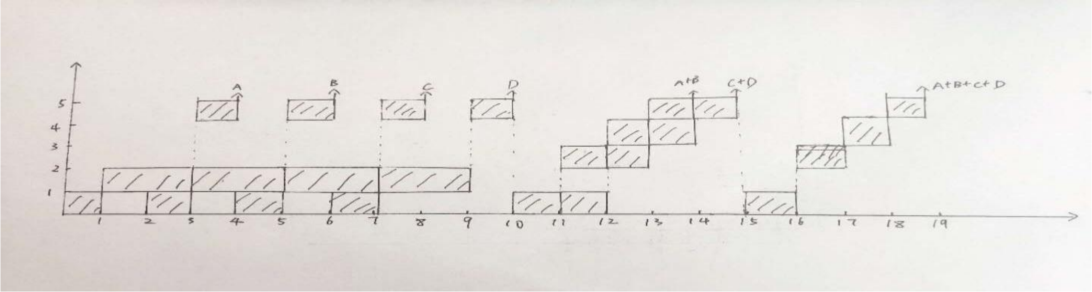
首先，应选择适合于流水线工作的算法。对于本题，应先计算 A1 × B1、 A2 × B2、A3 × B3 和 A4 × B4；再计算(A1 × B1) ＋(A2 × B2)和(A3 × B3) ＋ (A4 × B4)； 然后求总的结果。

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。



1、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 2 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 $\sum_{i=1}^{4}(A_i \times B_i)$ ，画出其时空图，并计算其吞吐率、加速比和效率




由图可见，它在 19 个△t 时间中，给出了 7 个结果。所以吞吐率为：

$$Tp=7/19\ \triangle t$$

如果不用流水线，由于一次求积需 4△t，一次求和需 4△t，则产生上述 7 个 结果共需（4×4+3×4）△t =28△t。所以加速比为：

$$S=28\ \triangle t\ /19\ \triangle t\ =1.47$$

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：

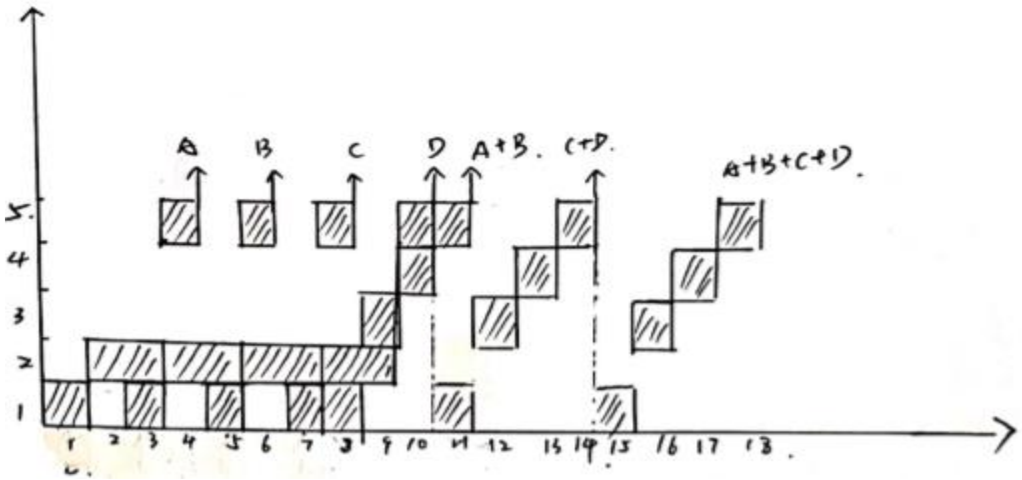
$$E=28\ \triangle t\ /\ (5\times 19\ \triangle t\ )\ =0.29$$

1、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 2 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 $\sum_{i=1}^{4}(A_i \times B_i)$ ，画出其时空图，并计算其吞吐率、加速比和效率

常见错误：对静态流水线的定义不清晰（ppt第三章31页有类似例题）

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。



2、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△t，问：
（1）当在流水线的输入端连续地每 △t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 2△t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？

解题思路：对流水线进行画图分析

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | S2 | stall | stall | S3 | S3 | S4 | | |
| 第 4 个任务 | | | | S1 | S2 | stall | stall | stall | S3 | S3 | S4 |

答案：（1）会发生流水线阻塞情况。（如上图）

（2）计算最大吞吐率：由上图可知，除了第一个任务意外，每多一个任务，就多两个△t

所以，完成n个任务所需时间为（5+2n-2）△t=（2n+3）△t

所以吞吐率 Tp=n/[（2n+3）△t]

由此可知，当n趋近于无穷时，吞吐率达到最大

Tpmax=lim n/[（2n+3）△t]=1/(2△t)

2、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△t，问：

（1）当在流水线的输入端连续地每 △t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 2△t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？

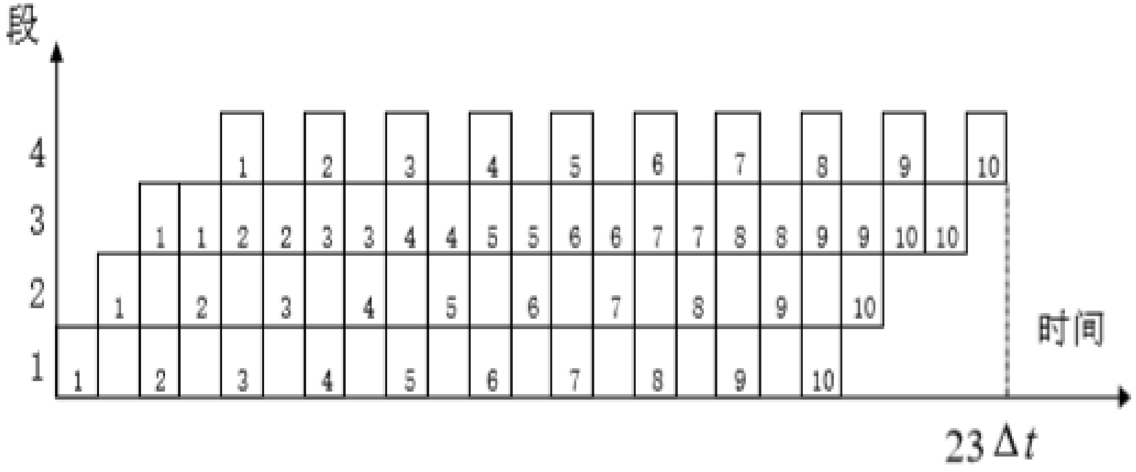
（3）当每段时间不变时，如何提高该流水线的吞吐率？仍连续处理10 个任务时，其吞吐率提高多少？

答案：

（2）当n=10时，由下图可知，总共时间为23 △t，阴影部分面积为50

吞吐率 Tp=10/[（2*10+3）△t]=10/（23 △t）

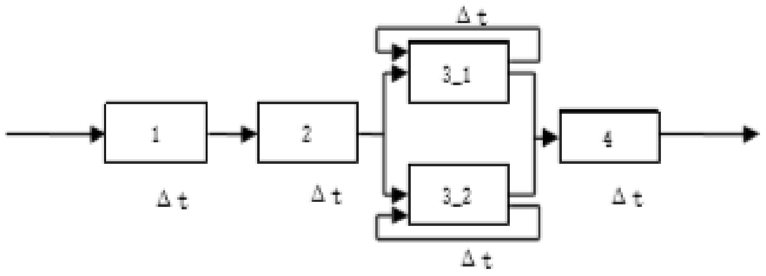
效率 E=50 △t/23*4 △t=0.54

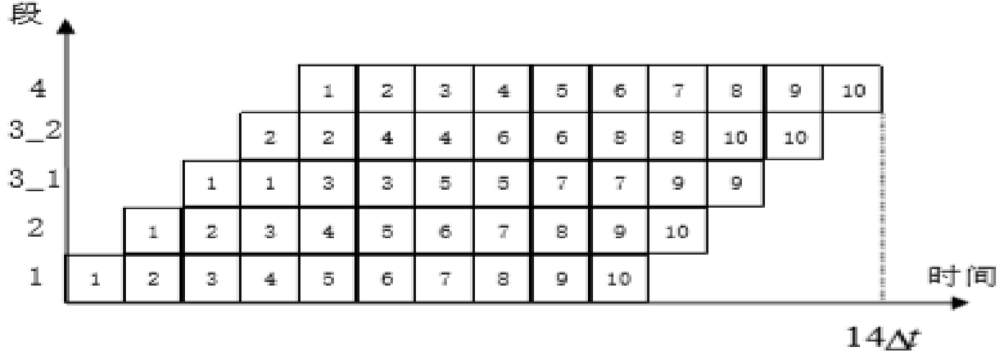


有一个流水线由4段组成，其中每当流经第3段时，总要在该段循环一次，然后才能流到第4段。如果每段经过一次所需要的时间都是△t，问：
（1）当在流水线的输入端连续地每 △t时间输入任务时，该流水线会发生 什么情况？
（2）此流水线的最大吞吐率为多少？如果每 2△t输入一个任务，连续处理10个任务时的实际吞吐率和效率是多少？
（3）当每段时间不变时，如何提高该流水线的吞吐率？仍连续处理10 个任务时，其吞吐率提高多少？

解题思路：本题的瓶颈在三四段，即由于第三段需要循环一次，第四段总需要等待，但是如果通过重复设置部件的方式，可以使任务错开，如下图，在第一个任务使用3-1部件时，第二个任务使用3-2部件，等第一个任务使用完后，第三个任务正好可以使用3-1部件。

答案：（3）重复设置部件




吞吐率 Tp=10/（14 △t ）

提升倍数= 10/（14 △t ）/10/（23 △t ）=1.64

# 第三次作业

3-1 针对所给出的代码段，给出对该代码段进行静态指令调度之后的时空图（MIPS五段流水线模型，支持“定向传送”）。静态指令调度后的加速比是多少?

```
LOAD    R1,M(A)
LOAD    R2,M(B)
ADD     R3,R1,R2
STORE   R3,M(C)
LOAD    R4,M(X)
LOAD    R5,M(Y)
ADD     R6,R4,R5
STORE   R6,M(Z)
```

按照原先的指令顺序:

出现多个冲突，多次停顿

$$R3 = R2 + R1$$

$$R6 = R4 + R5$$

# 方式一：

先全部load，再add，再store

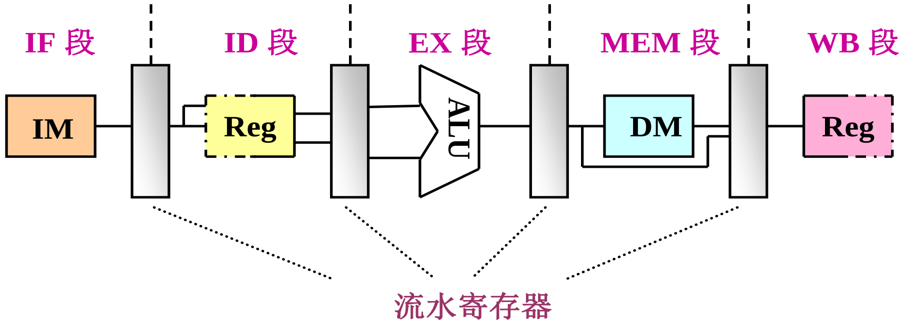
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| load R1 | IF | ID | EX | MEM | WB | | | | | | | |
| load R4 | | IF | ID | EX | MEM | WB | | | | | | |
| load R2 | | | IF | ID | EX | MEM | WB | | | | | |
| Load R5 | | | | IF | ID | EX | MEM | WB | | | | |
| add R3 | | | | | IF | ID | EX | MEM | WB | | | |
| add R6 | | | | | | IF | ID | EX | MEM | WB | | |
| storeR3 | | | | | | | IF | ID | EX | MEM | WB | |
| storeR6 | | | | | | | | IF | ID | EX | MEM | WB |

加速比

$S = (8\times5)/12 = 3.33$

# 方式二：

## 交换 load 和 add




| 指令 | 指令周期 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LOAD $R_1$, M(A) | IF | ID | EX | ME | RB | | | | | | | | |
| LOAD $R_2$, M(B) | | IF | ID | EX | ME | RB | | | | | | | |
| LOAD $R_4$, M(X) | | | IF | ID | EX | ME | RB | | | | | | |
| ADD $R_3$, $R_1$, $R_2$ | | | | IF | ID | EX | ME | RB | | | | | |
| LOAD $R_5$, M(Y) | | | | | IF | ID | EX | ME | RB | | | | |
| STORE $R_3$, M(C) | | | | | | IF | ID | EX | ME | RB | | | |
| ADD $R_6$, $R_4$, $R_5$ | | | | | | | IF | ID | EX | ME | RB | | |
| STORE $R_6$, M(Z) | | | | | | | | | IF | ID | EX | ME | RB |

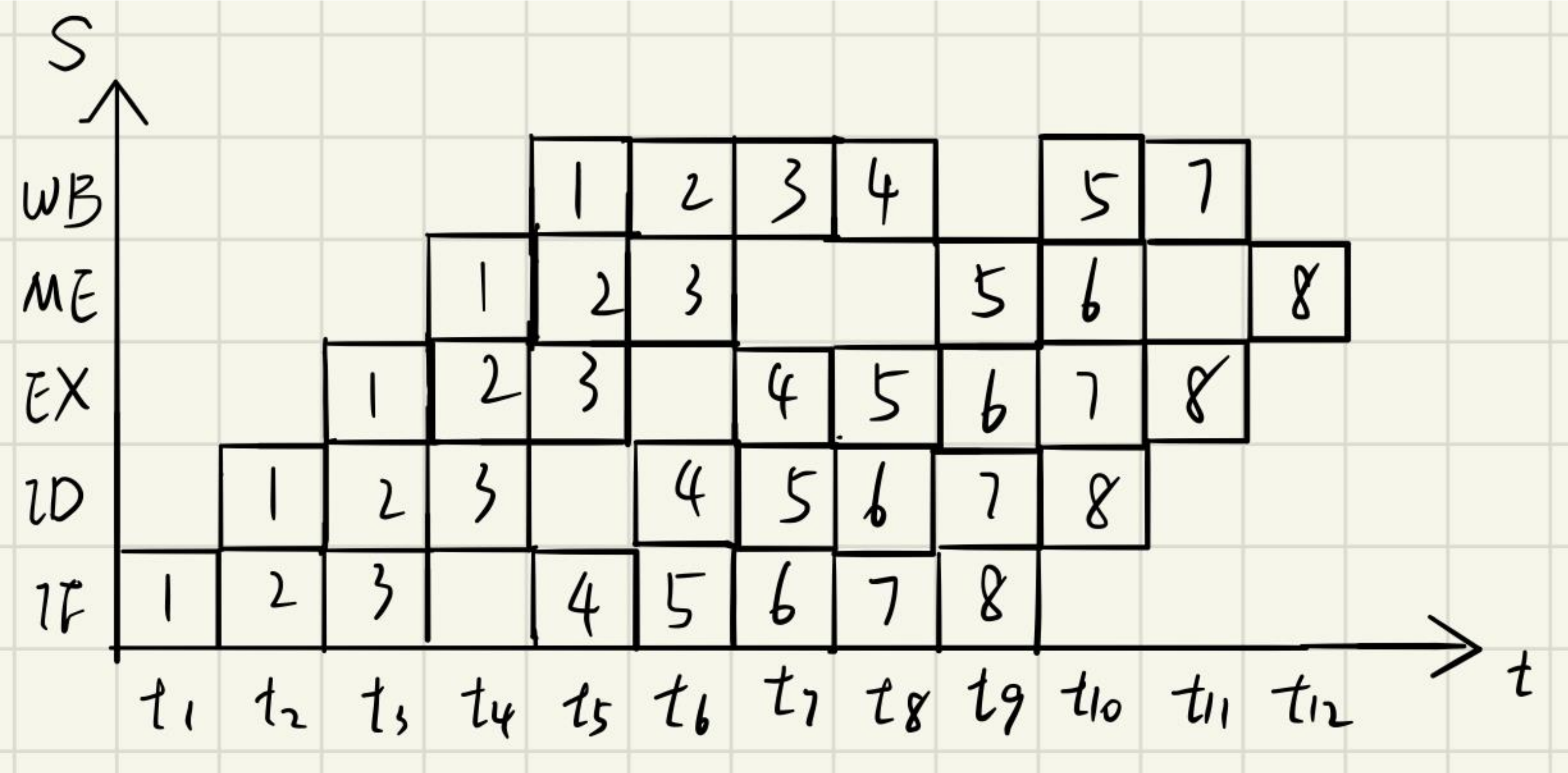
add 在 EX 段计算出 R6

store 在 ID 段读 R6

所以两者不能在同一时间周期内

加速比 $S = \frac{8 \times 5}{13} = \frac{40}{13}$

# 常见错误情况：




错误原因

虽然是空操作，但还是需要有这个周期的存在。

**3-2** 在一条单流水线处理机上执行下面程序。每条指令都要经过“取指”，“译码”，“执行”和“写结果”4个流水段。每个流水段的延迟时间都是5ns。在“执行”流水段，LS部件完成LOAD或STORE操作,其它操作在ALU部件完成，两个操作部件输出端有直接数据通路与任一操作部件的输入端相连，ALU部件产生的条件码能直接送入控制器，若采用静态分支预测技术，每次预测转移成功。画出指令流水线的时空图，计算流水线的吞吐率、加速比和效率。

```
1:          SUB   R0, R0       ;R0←0
2:          LOAD  R1,#8        ;向量长度 8
3:  LOOP:   LOAD  R2,A         ;R2 ← A 向量的一个元素
4:          MUL   R2,R1        ;R2 ←(R2)*(R1)
5:          ADD   R0,R2        ;R0 ←(R0)+(R2)
6:          DJNE  R1,LOOP      ;R1 ←(R1)-1 若(R1)≠0 则转
7:          STORE R0,S         ;保存结果
```

```
1:              SUB   R0, R0
2:              LOAD R1,#8
--------------------------------
3:      LOOP:   LOAD R2,A
4:              MUL R2,R1
5:              ADD R0,R2
6:              DJNE  R1,LOOP
--------------------------------
7:              STORE R0,S
```

循环8次

任务数： 35

总完成时间： 40×5ns

不采用流水线的总时间： 35×4×5ns

设备的实际使用时间： 35×4×5ns

吞吐率：

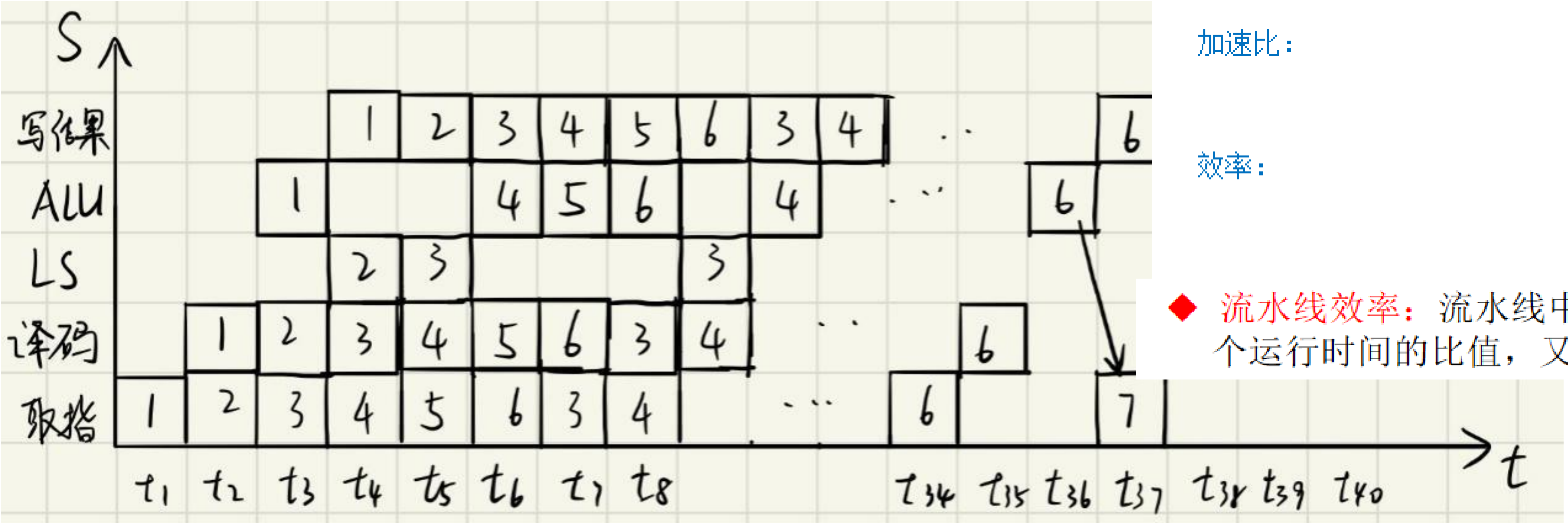
$$TP = \frac{35}{40 \times 5ns} = 175\ (MIPS)$$

加速比：

$$S = \frac{35 \times 4 \times 5ns}{40 \times 5ns} = 3.5$$

效率：

$$E = \frac{35 \times 4}{40 \times 5} = 0.7$$




◆ 流水线效率：流水线中的设备实际使用时间与整个运行时间的比值，又称流水线设备利用率。

3.12 假设各种分支指令数占所有指令数的百分比如下。

| | |
|---|---|
| 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?

**解题思路:**

$$\frac{\text{加速比（有控制相关）}}{\text{加速比（无控制相关）}} = \frac{\text{CPI（有控制相关）}}{\text{CPI（无控制相关）}}$$

没有控制相关的情况: CPI=1

# 有控制相关的情况

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

## 1. 排空流水线

由上图可知，对于条件分支，有2个时钟周期的延迟，对于无条件分支，有1个个时钟周期的延迟。

所以：

$$CPI=1+20\%\times 2+5\%\times 1=1.45$$

$$S=CPI/1=1.45$$

# 有控制相关的情况

## 2. 预测分支失败

对于失败的条件分支没有延迟；
对于成功的条件分支，有2个时钟周期的延迟；
对于无条件分支，有1个时钟周期的延迟。

$$CPI=1+20\%\times60\%\times2+5\%\times1=1.29$$

$$S=CPI/1=1.29$$

## 3. 预测分支成功

对于成功分支需要等待到第二个时钟周期得到跳转地址，所以有1个时钟周期的延迟；对于失败的条件分支，有2个时钟周期的延迟；对于无条件分支，有1个时钟周期的延迟。

$$CPI=1+(20\%\times60\%+5\%)\times1+20\%\times40\%\times2=1.33$$

$$S=CPI/1=1.33$$

第四次作业：

3.5 在一条单流水线多操作部件的处理机上执行下面的程序，取指令、指令译码各需一个时钟周期，MOVE、ADD 和 MUL 操作各需要 2 个、3 个和 4 个时钟周期。每个操作都在第一个时钟周期从通用寄存器中读操作数，在最后一个时钟周期把运算结果写到通用寄存器中。

```
k:     MOVE   R1,R0       ;R1←(R0)
k+1:   MUL    R0,R2,R1    ;R0←(R2)*(R1)
k+2:   ADD    R0,R3,R2    ;R0←(R2)+(R1)
```

画出指令执行的流水线时空图，并计算执行完三条指令共使用了多少个时钟周期。

该题的流水线可总结为：

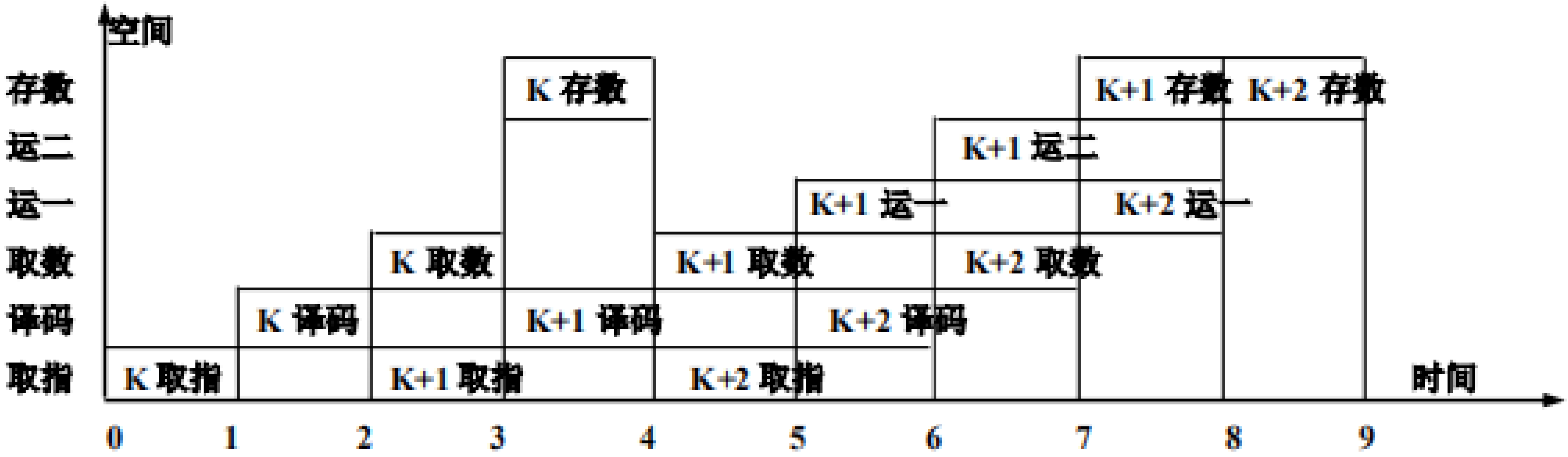
取指令+指令译码+操作（取数+运算+存寄存器）

3.5 在一条单流水线多操作部件的处理机上执行下面的程序，取指令、指令译码各需一个时钟周期，MOVE、ADD 和 MUL 操作各需要 2 个、3 个和 4 个时钟周期。每个操作都

得到时空图如下：

注意：一是“先写后读”，k 和 k+1

一是“先写后写”，k+1 和 k+2




共需9个时钟周期

3.5 在一条单流水线多操作部件的处理机上执行下面的程序，取指令、指令译码各需一个时钟周期，MOVE、ADD 和 MUL 操作各需要 2 个、3 个和 4 个时钟周期。每个操作都

注意： 一是“先写后读”， k 和 k+1

一是“先写后写”， k+1 和 k+2

得到时空图如下：

流水线时空图

| 指令 \ 时钟周期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| K | IF | ID | EX | EX | | | | | |
| K+1 | | IF | ID | stall | EX | EX | EX | EX | |
| K+2 | | | IF | stall | ID | EX | EX | stall | EX |

3.6 有一指令流水线如图 3.34 所示。

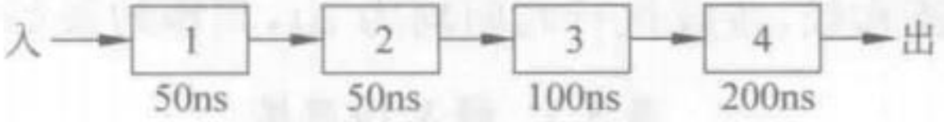



图 3.34 题 3.6 图

(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？

（1）

一般地，各段时间不等的流水线的实际吞吐率为

$$\mathrm{TP}=\frac{n}{\sum_{i=1}^{k}\Delta t_i+(n-1)\max(\Delta t_1,\Delta t_2,\cdots,\Delta t_k)} \tag{3.6}$$

书61，62页

在各段时间不等的情况下，$k$ 段流水线连续处理 $n$ 个任务的流水线效率为

$$E=\frac{n\cdot\sum_{i=1}^{k}\Delta t_i}{k\left[\sum_{i=1}^{k}\Delta t_i+(n-1)\cdot\max(\Delta t_1,\Delta t_2,\cdots,\Delta t_k)\right]} \tag{3.21}$$

$$\begin{aligned}T_k&=(50+50+100+200)+9\times200\\&=2200(\mathrm{ns})\end{aligned}$$

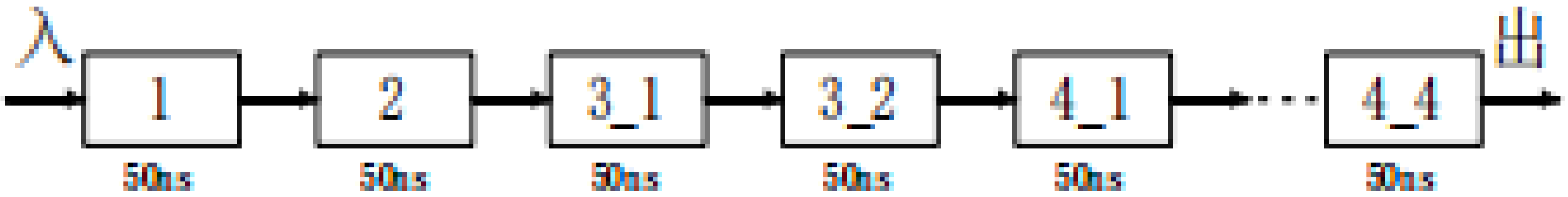
$$\mathrm{TP}=n/T_k=1/220(\mathrm{ns}^{-1})$$

$$E=\mathrm{TP}\cdot\frac{\sum_{i=1}^{m}\Delta t_i}{k}=\mathrm{TP}\cdot\frac{400}{4}=\frac{5}{11}\approx45.45\%$$

（2） 因为前2段时间一样，第3段是其2倍，第4段是其4倍，所以瓶颈在3,4段

书59,60页

**方法1： 细分流水段（变成8段时间相等的）**

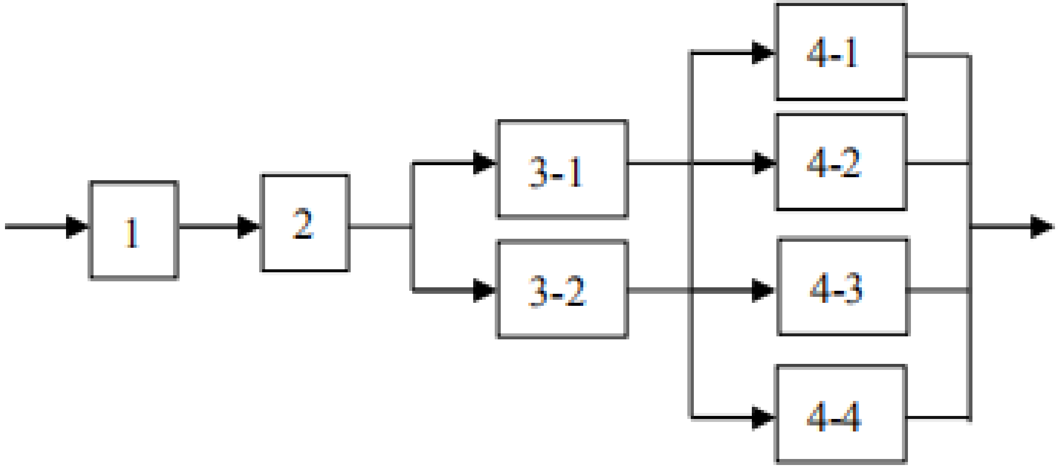



$$T_k = \sum_{i=1}^{k} \Delta t_i + (n-1)\Delta t_{max}$$
$$= 50 \times 8 + 9 \times 50$$
$$= 850(ns)$$

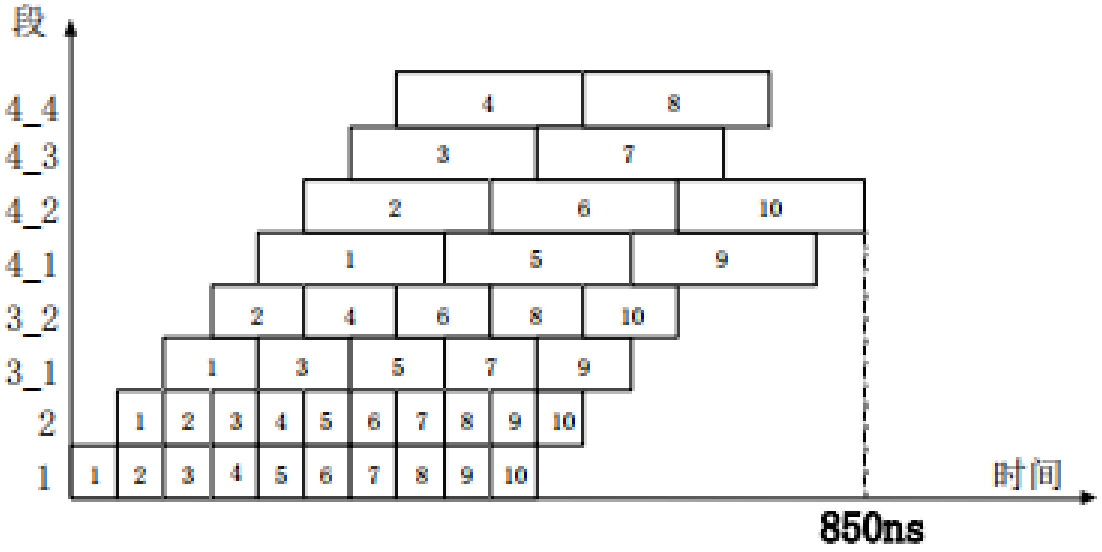
$$TP = n/T_k = 1/85\,(ns^{-1})$$

$$E = TP \cdot \frac{\sum_{i=1}^{m} \Delta t_i}{m} = TP \cdot \frac{400}{8} = \frac{10}{17} \approx 58.82\%$$

(2)　　方法2：　重复设置瓶颈段




画出时空图为：




$$TP = n/Tk = 1/85\,(ns^{-1})$$

$$E = \frac{400 \times 10}{850 \times 8} = \frac{10}{17} \approx 58.82\%$$

3.11 在 MIPS 流水线上运行以下代码序列。

```
LOOP: LW      R1,0(R2)
      DADDIU  R1,R1,#1
      SW      R1,0(R2)
      DADDIU  R2,R2,#4
      DSUB    R4,R3,R2
      BNEZ    R4,LOOP
```

R1 ← 取数
R1 = R1 + 1
R1 → 存数
R2 = R2 + 4
R4 = R3 - R2
当R4不为0时，发生转移

其中,R3 的初值是 R2+396。假设：在整个代码序列的执行过程中,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写操作和读操作可以通过分别把它们安排在前半个时钟周期和后半个时钟周期来实现。请问：

(1) 在没有任何其他定向(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。

（1）在没有任何其他定向（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？

1、“先读后存”， lw在MEM段存数，add在ID段读

2、排空流水线，等到MEM段分支判断完，才可进行下一指令

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LOOP: LW R1 0(R2) | IF | ID | EX | M | WB | | | | | | | | | | | | | | |
| ADDI R1 R1 #1 | | IF | S | S | ID | EX | M | WB | | | | | | | | | | | |
| SW 0(R2) R1 | | | | | IF | S | S | ID | EX | M | WB | | | | | | | | |
| ADDI R2 R2 #4 | | | | | | | | IF | ID | EX | M | WB | | | | | | | |
| SUB R4 R3 R2 | | | | | | | | | IF | S | S | ID | EX | M | WB | | | | |
| BNZ R4 LOOP | | | | | | | | | | | | IF | S | S | ID | EX | M | WB | |
| LW R1 0(R2) | | | | | | | | | | | | | | | IF | S | S | IF | |

总循环数：396÷4 = 99，　每轮需要17个周期（最后一轮18个周期）

总时钟周期 = 99×17 + 1 = 98×17 + 18

（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？

1、“先读后存”， lw在MEM段存数，add在ID段读，对同一寄存器的读和写可在同一周期进行

2、预测失败时，需等到MEM段

| 指令 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **LW** | IF | ID | EX | M | WB | | | | | | | | | | |
| **DADDIU** | | IF | ID | S | EX | M | WB | | | | | | | | |
| **SW** | | | IF | S | ID | EX | M | WB | | | | | | | |
| **DADDIU** | | | | | IF | ID | EX | M | WB | | | | | | |
| **DSUB** | | | | | | IF | ID | EX | M | WB | | | | | |
| **BNEZ** | | | | | | | IF | ID | EX | M | WB | | | | |
| **LW** | | | | | | | | IF | miss | miss | IF | ID | EX | M | WB |

总循环数：396÷4 = 99， 每轮需要10个周期； 最后一轮需要11个周期

总时钟周期 = 99×10 + 1 = 98×10 + 11

（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。

可重新组织指令为：

```
LOOP:   LW      R1, 0(R2)
        ADDI    R1, R1, #1
        SW      0(R2), R1
        ADDI    R2, R2, #4
        SUB     R4, R3, R2
        BNZ     R4, LOOP
```

```
LOOP:   LW      R1, 0(R2)
        ADDI    R2, R2, #4
        ADDI    R1, R1, #1
        SW      0(R2), R1
        SUB     R4, R3, R2
        BNZ     R4, LOOP
```

```
LOOP:   LW      R1, 0(R2)
        ADDI    R2, R2, #4
        ADDI    R1, R1, #1
        SW      -4(R2), R1
        SUB     R4, R3, R2
        BNZ     R4, LOOP
```

```
LOOP:   LW      R1, 0(R2)
        ADDI    R2, R2, #4
        ADDI    R1, R1, #1
        SUB     R4, R3, R2
        BNZ     R4, LOOP
        SW      -4(R2), R1
```

```
lw      r1,0(r2)        ;加法寄存器R1←取数(R2)
addi    r2,r2,#4        ;指针R2←指针R2+4
addi    r1,r1,#1        ;R1←R1+1
Sub     r4,r3,r2        ;R4←R3-R2
bnez    r4,Loop         ;若R4≠0, 循环
sw      -4(r2),r1       ;分支延迟槽， 存数(R2-4)←R1
```

(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。

时空图为：

| Instruction | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| lw r1,0(r2) | IF | ID | EX | M | WB | | | | | | |
| addi r2,r2,#4 | | IF | ID | EX | M | WB | | | | | |
| addi r1,r1,#1 | | | IF | ID | EX | M | WB | | | | |
| sub r4,r3,r2 | | | | IF | ID | EX | M | WB | | | |
| bnz r4,loop | | | | | IF | ID | EX | M | WB | | |
| sw r1,-4(r2) | | | | | | IF | ID | EX | M | WB | |
| lw r1,0(r2) | | | | | | | IF | ID | EX | M | WB |

总循环数：396÷4 = 99，　每轮需要6个周期，最后一轮需要10个周期

总时钟周期 = 98×6 + 10 = 99×6 + 4

# 习题讲解

——第4、5、9章

4-1 在CRAY-1机器上执行下述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，如果向量长度为64，问最快需要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：4拍）

V3←V2<A3　　（按（A3）左移：4拍）

V5←V3∧V2　　（向量逻辑乘：2拍 ）

注意点1:　寄存器冲突

前三条指令链接执行，
最后一条指令串行

**注意点2：**　如果题目中有关于数据传送的信息，那么就完全按照题目要求。

否则可按照Cray-1的默认要求：

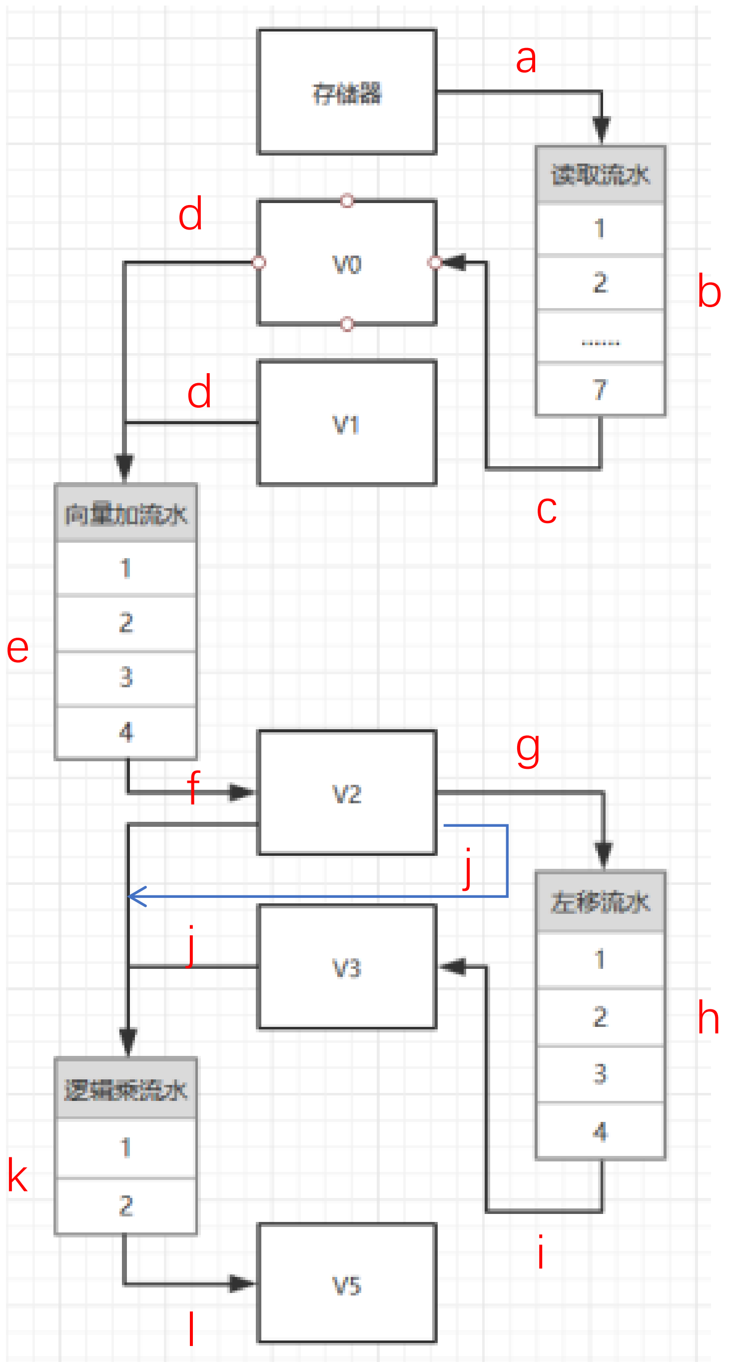
由于同步的要求，链接时，Cray-1 中把向量数据元素送往向量功能部件以及把结果存入向量寄存器都需要一拍时间，从存储器中把数据送入访存功能部件也需要一拍时间。

根据题意，可画出链接示意图：

[（1+7+1）+（1+4+1）+（1+4+1）+63 ]

+ [ (1+2+1)+63 ]

= 151 或 150



4.6　在 Cray-1 上，按照链接方式执行下述 5 条向量指令(括号中给出了相应功能部件的时间)，如果向量寄存器和功能部件之间数据传输需要一拍，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，则需要多少拍才能得到全部结果。

| | |
|---|---|
| $V_0 \leftarrow$ 存储器 | //从存储器中取数：7 拍 |
| $V_2 \leftarrow V_0 + V_1$ | //向量加：3 拍 |
| $V_3 \leftarrow V_2 < A_3$ | //按 (A3) 左移：5 拍 |
| $V_5 \leftarrow V_3 \wedge V_4$ | //向量逻辑乘：2 拍 |
| 存储器 $\leftarrow V_5$ | //向存储器中存数：7 拍 |

和上题一样的思路，只是这道题的 5 条指令不存在部件和寄存器冲突，且指令间都存在先写后读相关。可采用链接执行。

一次的通过时间：　(1+7+1) + (1+3+1) + (1+5+1) + (1+2+1) + (1+7+1) = 34 或 33

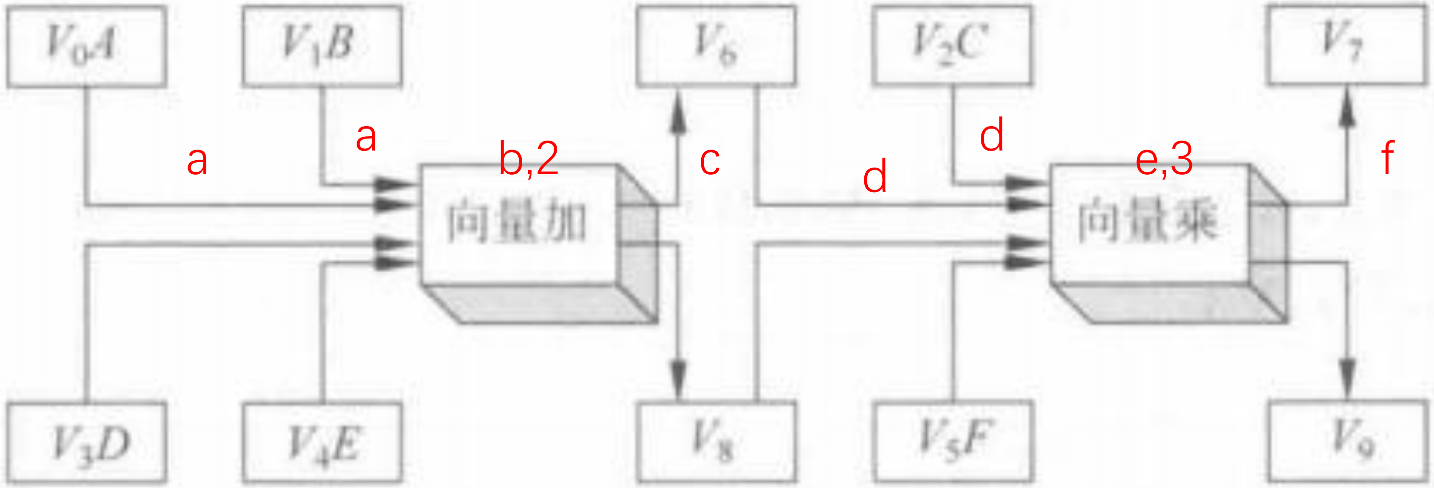
得到全部结果的时间：34/33 + 63 = 97/ 96

4.7　某向量处理机有 16 个向量寄存器，其中 $V_0 \sim V_5$ 中分别放有向量 $\boldsymbol{A}$、$\boldsymbol{B}$、$\boldsymbol{C}$、$\boldsymbol{D}$、$\boldsymbol{E}$、$\boldsymbol{F}$，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 Cray-1 的链接技术，先计算 $(\boldsymbol{A}+\boldsymbol{B})\times\boldsymbol{C}$，在流水线不停流的情况下，接着计算 $(\boldsymbol{D}+\boldsymbol{E})\times\boldsymbol{F}$。

(1) 求此链接流水线的通过时间(设寄存器入、出各需一拍)。

根据题意，可将其描述为：

$$V_6 \leftarrow A+B$$
$$V_7 \leftarrow V_6 \times C$$
$$V_8 \leftarrow D+E$$
$$V_9 \leftarrow V_8 \times F$$

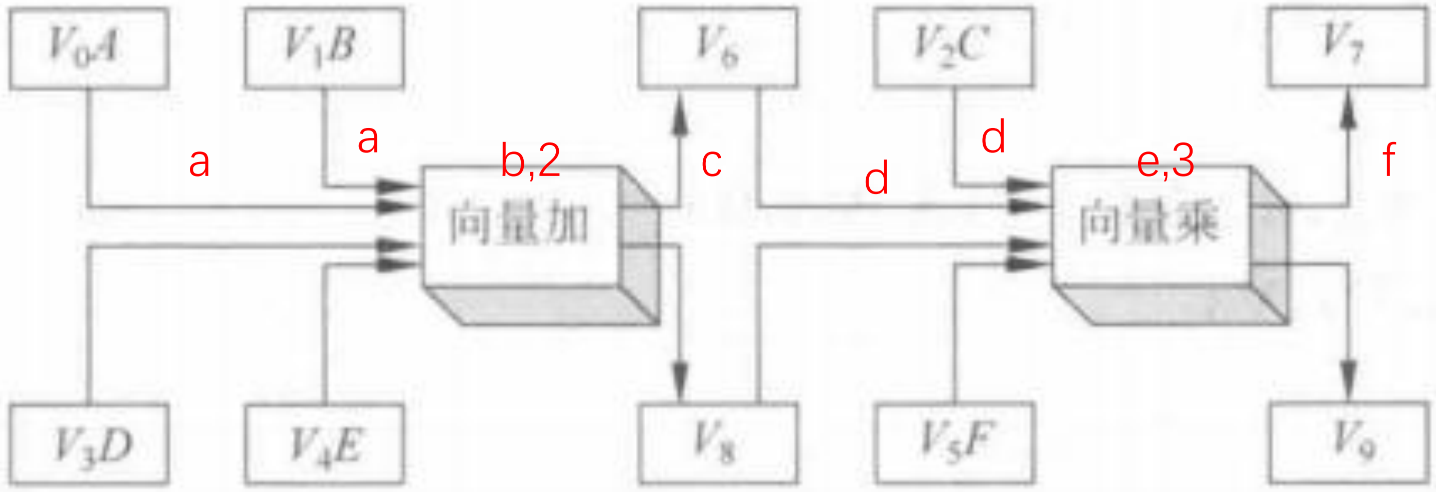



通过时间即经历一次加和乘的时间：　(a+b+c+d+e+f) = 1+2+1+1+3+1 = 9

（2）假如每拍时间为 50ns，完成这些计算并把结果存入相应的寄存器，此处理部件的实际吞吐率为多少 MFLOPS？

另一种想法：（A+B）×C 与（D+E）×F 只是一个执行**（X+Y）×Z** 的前后两部分

得到全部结果所需时间：9+（8+8-1）=24 拍 = 1200 ns




MFLOPS：每秒百万次浮点运算数

吞吐率：$$\frac{4\times 8}{1200ns}=\frac{32}{1200\times 10^{-9}}\times\frac{1}{10^{6}}\approx 26.67 \text{ MFLOPS}$$

4.8 在一台向量处理机上实现 $A=B\times S$ 计算，其中 $A$ 和 $B$ 是长度为 $N=200$ 的向量，$S$ 是一个标量。向量寄存器长度 $MVL=64$，各功能部件的启动时间为：取数和存数部件为 12 个时钟周期，乘法部件为 7 个时钟周期，执行标量代码的开销 $T_{loop}=15$ 个时钟周期，对一个向量元素执行一次操作的时间 $T_g=$ 一个时钟周期。求计算 $A$ 的总执行时间。

根据题意，得向量指令如下：（假设向量A和B存放在寄存器Ra和Rb中，标量S存放在R0中）

| | | | | |
|---|---|---|---|---|
| (1) | LV | V1, Rb | // 取向量B | (12) |
| (2) | MULTSV | V2, R0, V1 | // B ×S | (7) |
| (3) | SV | V2, Ra | // 结果存到向量A中 | (12) |

**$T_{start}$ = 12+7+12 = 31**

**$T_{loop}$ = 15**

**m = 3**

**n = 64**

分析：

(1) 指令相关 --> 分成3个编队 --> $T_{all} = T_{start} + T_{loop} + m\times n$

(2) 向量长度大于寄存器长度 --> 分段开采 --> 200 / 64 = 3×64 + 8，分4段

$$T_{总} = T_{step} \times 3 + T_{last}$$

前3段循环：$T_{step} = [\,T_{start} + T_{loop} + 3\times 64\,] \times 3 = 714$

最后一段：$T_{last} = T_{start} + T_{loop} + 3\times 8 = 70$

总时间：$T = T_{step} + T_{last} = 784$

$$T = T_{start} + T_{loop} + m\times n$$

$T_{start}$ **= 12+7+12 = 31**

$T_{loop}$ **= 15**

**m = 3**

**n = 64**

$$\begin{aligned}
T_{all} &= T_{step} \times p + T_{last} \\
&= (T_{start} + T_{loop} + m \times \mathrm{MVL}) \times p + (T_{start} + T_{loop} + m \times q) \\
&= (p+1) \times (T_{start} + T_{loop}) + m(\mathrm{MVL} \times p + q) \\
&= \left\lceil \frac{n}{\mathrm{MVL}} \right\rceil \times (T_{start} + T_{loop}) + mn \qquad (4.7)
\end{aligned}$$

$= 4 \times (31 + 15) + 3 \times 200 \times 1 = 784$

4.9 向量处理机 Cray Y-MP/8 的机器周期时间为 6ns，一个周期可以完成一次加和一次乘法运算。另外，8 台处理机在最好的情况下可以同时运算而互不干扰。计算 Cray Y-MP/8 的峰值性能。

$R_\infty$ 表示当向量长度为无穷大时，向量处理机的最大性能，也称为峰值性能，单位是 MFLOPS。

$$R_\infty = \lim_{n \to \infty} \frac{\text{向量指令序列中浮点运算次数} \times \text{时钟频率}}{\text{向量指令序列执行所需的时钟周期数}} \tag{4.8}$$

分析可知:

峰值性能 = 一个周期时间内浮点运算次数

$$R_\infty = \frac{(1+1)\times 8}{6\times 10^{-9}} \times \frac{1}{10^6} \approx 2667 \text{ MFLOPS}$$

MFLOPS：每秒百万次浮点运算数

4.10 $A$、$B$ 两个向量存放于存储器中，其向量长度为 64。设流水加法器有 4 级，流水线时钟周期为 10ns，读出 $A$、$B$ 向量第一对元素到流水线始端所需的时钟周期数为 2，求执行向量加法指令 ADDV 所需的时间。

考察向量指令的处理时间的原始定义

在向量处理机上，执行一条向量长度为 n 的向量指令所需的时间为：

$$T = [s + e + (n - 1)] \times T_c$$

书103页

s：向量部件流水线的建立所需时钟周期 2

e：向量流水线通过时间，即第一对向量元素通过流水线产生第一个结果所需的时钟周期 4

n：向量长度 64

Tc：流水线一个时钟周期的时间 10ns

$$\mathbf{T = (2 + 4 + 63) \times 10\ ns = 690ns}$$

# 第七八九次作业

# 第七次作业

下述指令，第一条指令已经执行并完成，第二条指令已完成正等待写结果，试给出记分牌法所用的指令状态表、功能部件状态表和结果寄存器状态表。假定：浮点流水线延迟如下：取数1个时钟周期，加法2个时钟周期，乘法10个时钟周期，除法40个时钟周期（功能部件状态表中所列的各部件均为1个）。

```
MULT.D    F0,  F2,  F4
L.D       F6,  34(R2)
SUB.D     F8,  F6,  F2
DIV.D     F10, F0,  F6
ADD.D     F6,  F8,  F2
```

下述指令，第一条指令已经执行并完成，第二条指令已完成正等待写结果，试给出记分牌法所用的指令状态表、功能部件状态表和结果寄存器状态表。假定：浮点流水线延迟如下：取数1个时钟周期，加法2个时钟周期，乘法10个时钟周期，除法40个时钟周期（功能部件状态表中所列的各部件均为1个）。

- 解题思路：
- 记分牌原理：把5段流水线中ID段分解成流出和读操作数两个段，执行段采用多功能部件，允许多条指令在执行段中并行操作

尽早执行没有结构冲和数据冲突的指令。

实施判断是否有WR、RW、WW相关存在。

建立互锁机制，阻止数据冲突发生。

通过一个称为记分牌的硬件实现对指令的动态调度。

下述指令，第一条指令已经执行并完成，第二条指令已完成正等待写结果，试给出记分牌法所用的指令状态表、功能部件状态表和结果寄存器状态表。假定：浮点流水线延迟如下：取数1个时钟周期，加法2个时钟周期，乘法10个时钟周期，除法40个时钟周期（功能部件状态表中所列的各部件均为1个）。

- 解题思路：

```
MULT.D    F0,  F2,  F4
L.D       F6,  34(R2)
SUB.D     F8,  F6,  F2
DIV.D     F10, F0,  F6
ADD.D     F6,  F8,  F2
```

- L.D F6,34(R2)和 SUB.D  F8, F6, F2、DIV.D  F10, F0, F6 之间存在 RAW 冲突，所以在流出段等待，SUB.D  F8, F6, F2 与 ADD.D  F6, F8, F2 之间存在结构相关，流出段也不能做。

下述指令，第一条指令已经执行并完成，第二条指令已完成正等待写结果，试给出记分牌法所用的指令状态表、功能部件状态表和结果寄存器状态表。假定：浮点流水线延迟如下：取数1个时钟周期，加法2个时钟周期，乘法10个时钟周期，除法40个时钟周期（功能部件状态表中所列的各部件均为1个）。

- 解答：
- 指令的执行状态表：记录（已取指）指令的执行状态

| 指令 | 指令状态表 | | | |
|---|---|---|---|---|
| | 流出 | 读操作数 | 执行 | 写结果 |
| MULT.D F0，F2，F4 | √ | √ | √ | √ |
| L.D F6，34(R2) | √ | √ | √ | |
| SUB.D F8，F6，F2 | √ | | | |
| DIV.D F10，F0，F6 | √ | | | |
| ADD.D F6，F8，F2 | | | | |

| 指令 | 指令状态表 | | | |
| --- | --- | --- | --- | --- |
| | 流出 | 读操作数 | 执行 | 写结果 |
| MULT.D F0, F2, F4 | √ | √ | √ | √ |
| L.D F6, 34(R2) | √ | √ | √ | |
| SUB.D F8, F6, F2 | √ | | | |
| DIV.D F10, F0, F6 | √ | | | |
| ADD.D F6, F8, F2 | | | | |

| 部件名称 | 功能部件状态表 | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | Busy | Op | Fi | Fj | Fk | Qj | Qk | Rj | Rk |
| Integer | yes | L.D | F6 | R2 | | | | no | |
| Mult | no | | | | | | | | |
| Add | yes | SUB.D | F8 | F6 | F2 | Integer | | no | yes |
| Divide | yes | DIV.D | F10 | F0 | F6 | | Integer | yes | no |

功能部件状态表：记录各功能部件的状态。

Busy：忙标志，功能部件是否忙 -Op：正在或将要执行的操作

Fi：目的寄存器（编号） Fj，Fk：源寄存器（编号）

Qj，Qk：向源寄存器Fj、Fk写数据的功能部件

Rj，Rk：标志位，“yes”表示Fj，Fk的操作数可用--就绪且未被取走。否则为“no”。

下述指令，第一条指令已经执行并完成，第二条指令已完成正等待写结果，试给出记分牌法所用的指令状态表、功能部件状态表和结果寄存器状态表。假定：浮点流水线延迟如下：取数1个时钟周期，加法2个时钟周期，乘法10个时钟周期，除法40个时钟周期（功能部件状态表中所列的各部件均为1个）

- 解答：

结果（目标）寄存器状态表：指出哪个功能部件将把结果写入该目标寄存器。

| | 结果寄存器状态表 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | F0 | F2 | F4 | F6 | F8 | F10 | … | F30 |
| 部件名称 | | | | Integer | Add | Divide | | |

第八次作业

1、下表为某处理机在给定时刻的指令流状态。给出该时刻在Tomasulo保留站和寄存器状态表的对应内容。

| 指令 | 指令状态 | | |
|---|---|---|---|
| | 流出 | 执行 | 写结果 |
| SUB.D F6,F4,F8 | √ | √ | √ |
| L.D F8,32(R3) | √ | √ | |
| ADD.D F0,F8,F4 | √ | | |
| MUL.D F2,F0,F6 | √ | | |

1、下表为某处理机在给定时刻的指令流状态。给出该时刻在Tomasulo保留站和寄存器状态表的对应内容。

- 解题思路：
- 基本思想：
- 指令执行的流水线需3段：
- 流出：从指令队列头部取一条指令，准备送到该保留站。
- 执行:通过保留站执行指令规定的操作。
- 写结果：结果放到CDB，等待寄存器或保留站获取。

1、下表为某处理机在给定时刻的指令流状态。给出该时刻在Tomasulo保留站和寄存器状态表的对应内容。

- 指令的流出：从指令队列的头部取一条指令

  如果指令操作的保留站有空闲，就把该指令送到保留站r 。

  如果其操作数已经在寄存器中，将操作数送入保留站r

  如果操作数没有就绪，把将产生操作数的保留站标识送入保留站。

  一旦被记录的保留站完成计算，直接把数据送给C保留站r。

  (通过寄存器换名和对操作数进行缓冲，消除WAR冲突)

  如果没有空闲的保留站，指令就不能流出。

  通过对目标寄存器的顺序预约，消除WAW冲突。

| 指令 | 指令状态 | | |
|---|---|---|---|
| | 流出 | 执行 | 写结果 |
| SUB.D F6,F4,F8 | √ | √ | √ |
| L.D F8,32(R3) | √ | √ | |
| ADD.D F0,F8,F4 | √ | | |
| MUL.D F2,F0,F6 | √ | | |

| 名称 | 保留站 | | | | | | |
|---|---|---|---|---|---|---|---|
| | Busy | Op | Vj | Vk | Qj | Qk | A |
| Load1 | yes | LD | | | | | 32+Regs[R3] |
| Load2 | No | | | | | | |
| Add1 | No | | | | | | |
| Add2 | yes | ADD | | Regs[F4] | Load1 | | |
| Add3 | No | | | | | | |
| Mult1 | yes | MUL | | Regs[F6] | Add2 | | |
| Mult2 | no | | | | | | |

- 第一条指令已完成并写入
- yes表示本保留站或缓冲单元忙
- Op 是本对源操作数进行的操作
- Vj，Vk：源操作数的值（或换名），对load，Vk字段用于保存偏移量。
- Qj，Qk：将产生源操作数的保留站号 （如Add1）
- A：仅load和store缓冲器有该字段。开始是存放指令中的立即数字段，地址计算后存放有效地址

1、下表为某处理机在给定时刻的指令流状态。给出该时刻在Tomasulo保留站和寄存器状态表的对应内容。

- 解题思路：
- 寄存器状态表：每个寄存器在该表中有对应的一项，存放将把结果写入该寄存器的保留站的站号。

| | 寄存器状态表 | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | F0 | F2 | F4 | F6 | F8 | F10 | … | F30 |
| Qi | Add2 | Mult1 | | | Load1 | | | |

2、假设有一条长流水线，仅仅对条件转移指令使用分支目标缓冲。假设分支预测错误的开销为 4 个时钟周期，缓冲不命中的开销为 3 个时钟周期。假设：命中率为 90%，预测精度为 90%，分支频率为 15%，没有分支的基本 CPI 为 1。

（1） 求程序执行的 CPI。

（2） 相对于采用固定的 2 个时钟周期延迟的分支处理，哪种方法程序执行速度更快？

- 解题思路：
- （1）程序执行的 CPI = 没有分支的基本 CPI（1） + 分支带来的额外开销
- 分支带来的额外开销是指在分支指令中， 缓冲命中但预测错误带来的开销与缓冲没有命 中带来的开销之和。
- 分支带来的额外开销 = 15% * (90% 命中× 10%预测错误× 4 + 10% 没命中× 3)= 0.099
- 所以，程序执行的 CPI ＝ 1 ＋ 0.099 = 1.099

2、假设有一条长流水线，仅仅对条件转移指令使用分支目标缓冲。假设分支预测错误的开销为 4 个时钟周期，缓冲不命中的开销为 3 个时钟周期。假设：命中率为 90%，预测精度为 90%，分支频率为 15%，没有分支的基本 CPI 为 1。
（1） 求程序执行的 CPI。
（2） 相对于采用固定的 2 个时钟周期延迟的分支处理，哪种方法程序执行速度更快?

- 集体思路:

（2）采用固定的 2 个时钟周期延迟的分支处理 CPI = 1 + 15% ×2 = 1.3

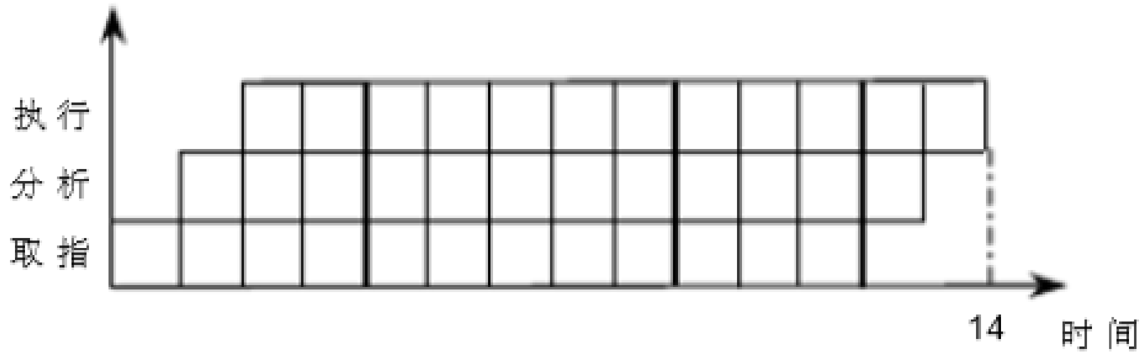
- 由（1)(2)可知分支目标缓冲方法执行速度快。

3、假设分支目标缓冲的命中率为 90%，程序中无条件转移指令的比例为 5%，没有无条件转移指令的程序 CPI 值为 1。假设分支目标缓冲中包含分支目标指令， 允许无条件转移指令进入分支目标缓冲，则程序的 CPI 值为多少？假设原来的CPI=1.1

- 集体思路：
- 设每条无条件转移指令的延迟为 x，则有：
- $1 + 5\% \times x = 1.1$
- $x = 2$
- 当分支目标缓冲命中时，无条件转移指令的延迟为 0。
- 所以 程序的 CPI $= 1 + 2 \times 5\% \times (1 - 90\%) = 1.01$

3、设指令流水线由取指令、分析指令和执行指令 3 个部件构成，每个部件经过的时间 为△ t，连续流入 12 条指令。分别画出标量流水处理机以及 ILP 均为 4 的超标量处理机、超长指令字处理机、超流水处理机的时空图，并分别计算它们相对于标量流水处理机的加速比。

- 解题思路:
- 标量流水处理机的时空图:

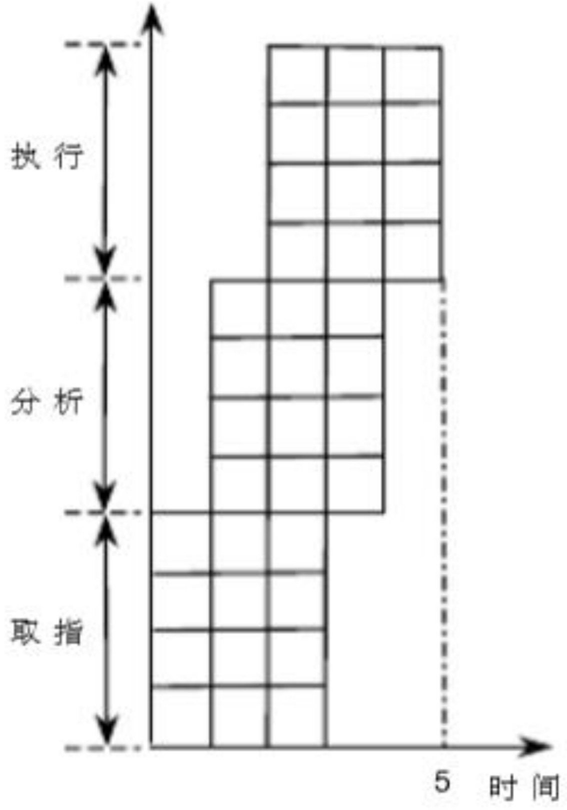



- 执行完 12 条指令需 T1＝14△t。

3、设指令流水线由取指令、分析指令和执行指令 3 个部件构成，每个部件经过的时间 为△ t，连续流入 12 条指令。分别画出标量流水处理机以及 ILP 均为 4 的超标量处理机、超长指令字处理机、超流水处理机的时空图，并分别计算它们相对于标量流水处理机的加速比。

- 超标量：在每个时钟周期流出的指令条数不固定,但有上限，依代码的具体情况而定。设这个上限为n，就称该处理机为n-流出(发射)。
- 解答:
- 超标量流水处理机中，每一个时钟周期同时启动 4 条指令。执行完 12 条指令需 T 2＝5 △t，相对于标量流水处理机的加速比为:

$$S_2 = \frac{T_1}{T_2} = \frac{14\Delta t}{5\Delta t} = 2.8$$




超标量处理机时空图

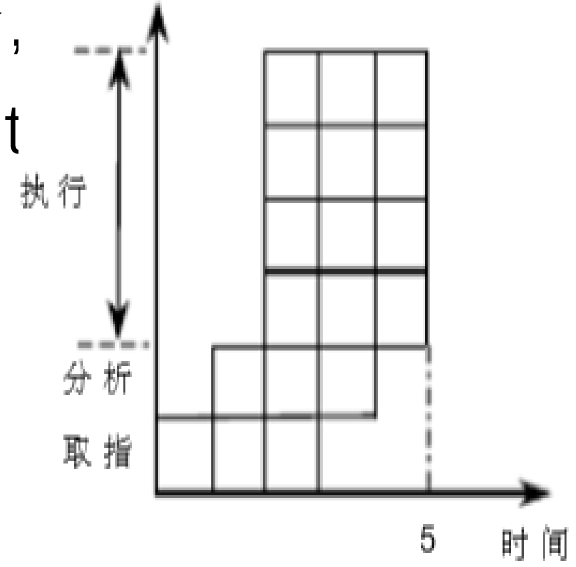
3、设指令流水线由取指令、分析指令和执行指令 3 个部件构成，每个部件经过的时间 为△ t，连续流入 12 条指令。分别画出标量流水处理机以及 ILP 均为 4 的超标量处理机、超长指令字处理机、超流水处理机的时空图，并分别计算它们相对于标量流水处理机的加速比。

超长指令技术：把能并行执行的多条指令组装成一条很长的指令

解答：

超长指令字处理机中，每 4 条指令组成一条长指令，共形成 3 条长指令。执行完 12 条 指令需 T 3 = 5△t，相对于标量流水处理机的加速比为：

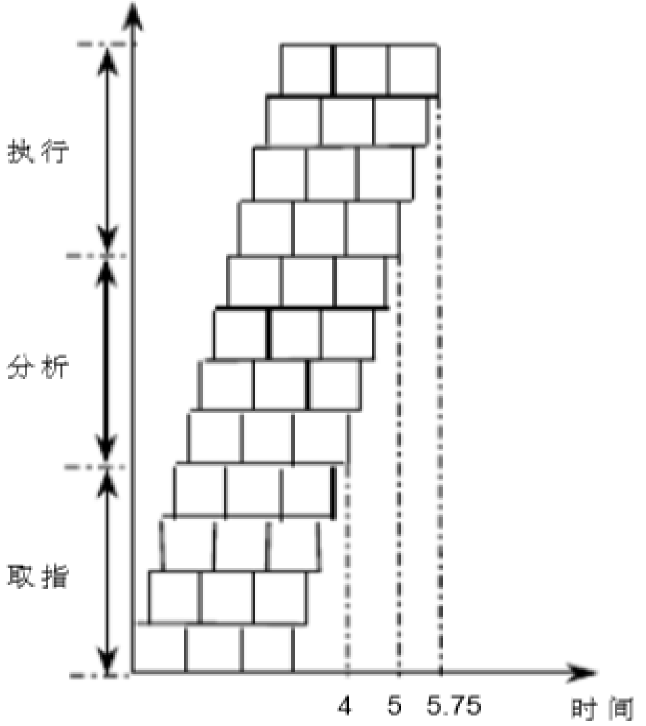
$$S_3 = \frac{T_1}{T_3} = \frac{14\Delta t}{5\Delta t} = 2.8$$



超长指令字处理机时空图

3、设指令流水线由取指令、分析指令和执行指令 3 个部件构成，每个部件经过的时间 为△t，连续流入 12 条指令。分别画出标量流水处理机以及 ILP 均为 4 的超标量处理机、超长指令字处理机、超流水处理机的时空图，并分别计算它们相对于标量流水处理机的加速比。

- 超流水处理机：将每个流水段进一步细分，这样在一个时钟周期内能够分时流出多条指令。这种处理机称为超流水线处理机。实际上该超流水线计算机的流水线周期为1/n个时钟周期。
- 解答：
- 超流水处理机中，每 1/4 个时钟周期启动一条指令。执行完 12 条指令需 T4＝5.75△t， 相对于标量流水处理机的加速比为：




$$S_4 = \frac{T_1}{T_4} = \frac{14\Delta t}{5.75\Delta t} = 2.435$$

第九次作业

1、在有16个处理机的混洗交换网络中，若要使第0号处理器与第15号处理器相连，需要经过多少次混洗和交换?

- 解题思路:
- 交换函数：实现二进制地址编码中第k位互反的输入端与输出端之间的连接。

$$E\left(x_{n-1}x_{n-2}\cdots x_{k+1}x_k x_{k-1}\cdots x_1 x_0\right)=x_{n-1}x_{n-2}\cdots x_{k+1}\bar{x}_k x_{k-1}\cdots x_1 x_0$$

- 混洗函数：把输入端的二进制编号循环左移一位

$$\sigma\left(x_{n-1}x_{n-2}\cdots x_1 x_0\right)=x_{n-2}x_{n-3}\cdots x_1 x_0 x_{n-1}$$

- 解答:

$$E_0(0000)=0001, \sigma(0001)=0010, E_0(0010)=0011, \sigma(0011)=0110, E_0(0110)=0111,$$
$$\sigma(0111)=1110, E_0(1110)=1111$$

- 故需要3次混洗，4次交换。

9. 设函数的自变量是十进制数表示的处理机编号。现有32台处理机，其编号为$0, 1, 2, \cdots, 31$。

（1）分别计算下列互连函数。

$\mathrm{Cube}_2(12)$  $\sigma(8)$  $\beta(9)$  $\mathrm{PM2I}_{+3}(28)$  $\mathrm{Cube}_0(\sigma(4))$  $\sigma(\mathrm{Cube}_0(18))$

（2）用$\mathrm{Cube}_0$和$\sigma$构成的混洗交换网（每步只能使用$\mathrm{Cube}_0$和$\sigma$一次），网络直径是多少？从5号处理机发送数据到7号处理机，最短路径要经过几步？请列出经过的处理机编号。

（3）采用移数网络构成互连网，网络直径是多少？节点度是多少？与2号处理机距离最远的是几号处理机？

- 解答:
- （1）共有32台处理机，所以表示处理机号的二进制地址应为5位
- 交换函数：$\mathrm{Cube}_2(12) = \mathrm{Cube}_2(01100) = 01000 = 8$
- 混洗函数：$\sigma(8) = \sigma(01000) = 10000 = 16$
- 碟式函数：$\beta(9) = \beta(01001) = 11000 = 24$
- PM2I函数：$\mathrm{PM2I}_{+3}(28) = (28 + 2^3) \bmod 32 = 4$

$$\mathrm{Cube}_0(\sigma(4)) = \mathrm{Cube}_0(\sigma(00100)) = \mathrm{Cube}_0(01000) = 01001 = 9$$

$$\sigma(\mathrm{Cube}_0(18)) = \sigma(\mathrm{Cube}_0(10010)) = \sigma(10011) = 00111 = 7$$

9. 设函数的自变量是十进制数表示的处理机编号。现有32台处理机，其编号为$0, 1, 2, \cdots, 31$。

（1）分别计算下列互连函数。

$\mathrm{Cube}_2(12)$ $\sigma(8)$ $\beta(9)$ $\mathrm{PM2I}_{+3}(28)$ $\mathrm{Cube}_0(\sigma(4))$ $\sigma(\mathrm{Cube}_0(18))$

（2）用$\mathrm{Cube}_0$和$\sigma$构成的混洗交换网（每步只能使用$\mathrm{Cube}_0$和$\sigma$一次），网络直径是多少？从5号处理机发送数据到7号处理机，最短路径要经过几步？请列出经过的处理机编号。

（3）采用移数网络构成互连网，网络直径是多少？节点度是多少？与2号处理机距离最远的是几号处理机？

- 解答:
- （2）2^n个结点的均匀洗牌交换网的网络直径是2n-1，所以32个结点的均匀洗牌交换网的网络直接是9。
- 从5号处理机发送到7号，最短路径要经过6步:

$$00101 \rightarrow 00100 \rightarrow 01000 \rightarrow 01001 \rightarrow 10010 \rightarrow 10011 \rightarrow 00111$$

即$5 \rightarrow 4 \rightarrow 8 \rightarrow 9 \rightarrow 18 \rightarrow 19 \rightarrow 7$

9. 设函数的自变量是十进制数表示的处理机编号。现有32台处理机，其编号为$0, 1, 2, \cdots, 31$。

（1）分别计算下列互连函数。

$\mathrm{Cube}_2(12)$ $\sigma(8)$ $\beta(9)$ $\mathrm{PM2I}_{+3}(28)$ $\mathrm{Cube}_0(\sigma(4))$ $\sigma(\mathrm{Cube}_0(18))$

（2）用$\mathrm{Cube}_0$和$\sigma$构成的混洗交换网（每步只能使用$\mathrm{Cube}_0$和$\sigma$一次），网络直径是多少？从5号处理机发送数据到7号处理机，最短路径要经过几步？请列出经过的处理机编号。

（3）采用移数网络构成互连网，网络直径是多少？节点度是多少？与2号处理机距离最远的是几号处理机？

- 解答:
- （3）移数网络网络直径为$\lceil n/2 \rceil = \lceil 5/2 \rceil = 3$，节点度为$2n-1 = 2 \times 5 - 1 = 9$，

根据公式，与2号处理机直接相连的处理机有0，1，3，4，6，10，18，26，30。

以此类推，找寻到上述处理机直接相连的的处理机，这样距离为2，由于已知直径为3，所以最远的就是距离为3的处理机，这样的处理机包括13，15，21，23。

10. $N=16$的互连网络的输入端号和输出端号分别为$0\sim15$。若互连网络实现的互连可以用互连函数表示为$f(x_3,x_2,x_1,x_0)=x_0x_1x_2x_3$，那么，是否可以用循环表示法表示该互连网络实现的互连？如果可以，请写出其循环表示。

$$f(0)=f(0000)=0000=0$$

$$f(1)=f(0001)=1000=8, f(8)=f(1000)=0001=1$$

$$f(2)=f(0010)=0100=4, f(4)=f(0100)=0010=2$$

$$f(3)=f(0011)=1100=12, f(12)=f(1100)=0011=3$$

$$f(5)=f(0101)=1010=10, f(10)=f(1010)=0101=5$$

$$f(6)=f(0110)=0110=6$$

$$f(7)=f(0111)=1110=14, f(14)=f(1110)=0111=7$$

$$f(9)=f(1001)=1001=9$$

$$f(11)=f(1011)=1101=13, f(13)=f(1101)=1011=11$$

$$f(15)=f(1111)=1111=15$$

故循环表示为$(0)(1\ \ 8)(2\ \ 4)(3\ \ 12)(5\ \ 10)(6)(7\ \ 14)(9)(11\ \ 13)(15)$。