计算机系统结构

练习题解析

2017/05/27

1.7 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	指令执行数量/条	平均时钟周期数
整数	45 000	1
数据传送	75 000	2
浮点	8 000	4
分支	1 500	2

求该计算机的有效CPI、MIPS和程序执行时间。

解：

每条指令的平均时钟周期数CPI

= 执行程序所需的时钟周期数 / 所执行的指令条数

= (45000×1＋75000×2＋8000×4＋1500×2) / 129500

= 1.776

MIPS速率 = f / CPI = 400 / 1.776 = 225.225 MIPS

程序执行时间

= (45000×1＋75000×2＋8000×4＋1500×2) / 400

=575s

1.9 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40%，则采用此改进方法后，能使整个系统的性能提高多少？

解：

由题可知：

可改进比例$F_e = 40\% = 0.4$　　部件加速比$S_e = 20$

根据Amdahl定律可知：

$$S_n = \frac{1}{(1-F_e)+\frac{F_e}{S_e}} = \frac{1}{(1-0.4)+\frac{0.4}{20}} = 1.613$$

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，R2，R3　；R0←(R2)+(R3)

画出指令执行过程的流水线时空图，并计算完成这三条指令共需要多少个时钟周期？

解：

由题意程序指令执行过程的流水线时空图如下图所示。

指令＼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

共插入了3个时钟周期的停顿，共需要9个时钟周期。

3.7 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 Δt，问：

（1）当在流水线的输入端连续地每 Δt 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 $2\Delta t$ 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？

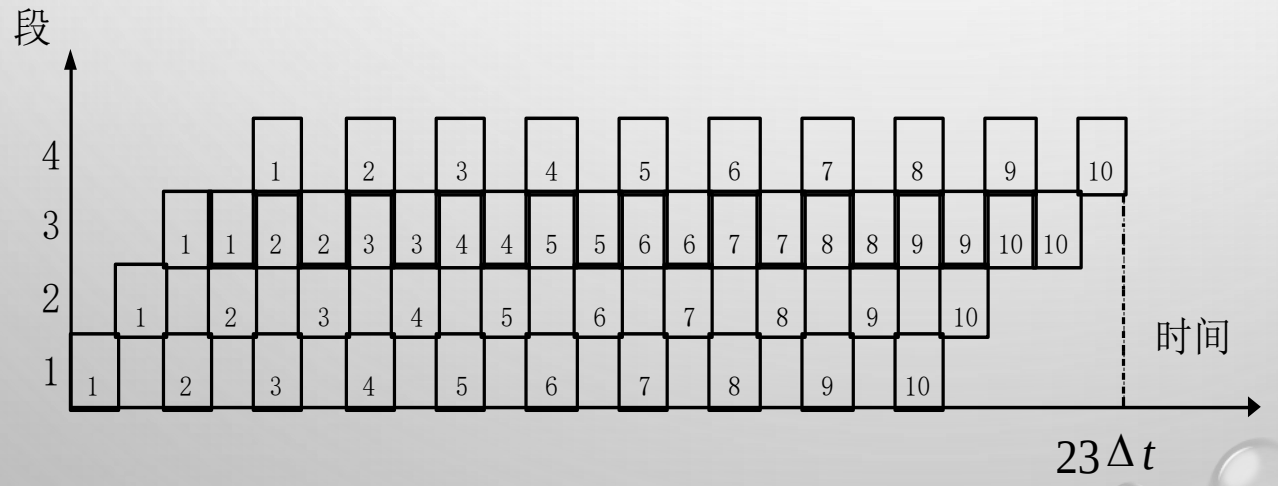
（3）当每段时间不变时，如何提高该流水线的吞吐率？仍连续处理10个任务时，其吞吐率提高多少？

解：

（1）会发生流水线阻塞的情况。

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）



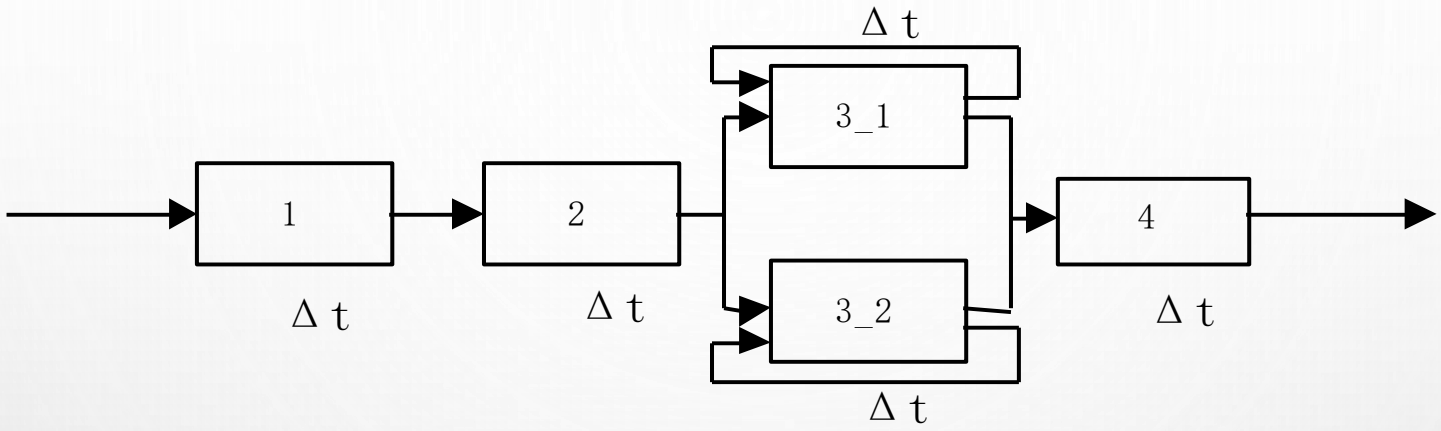
$$TP_{max} = \frac{1}{2\Delta t}$$

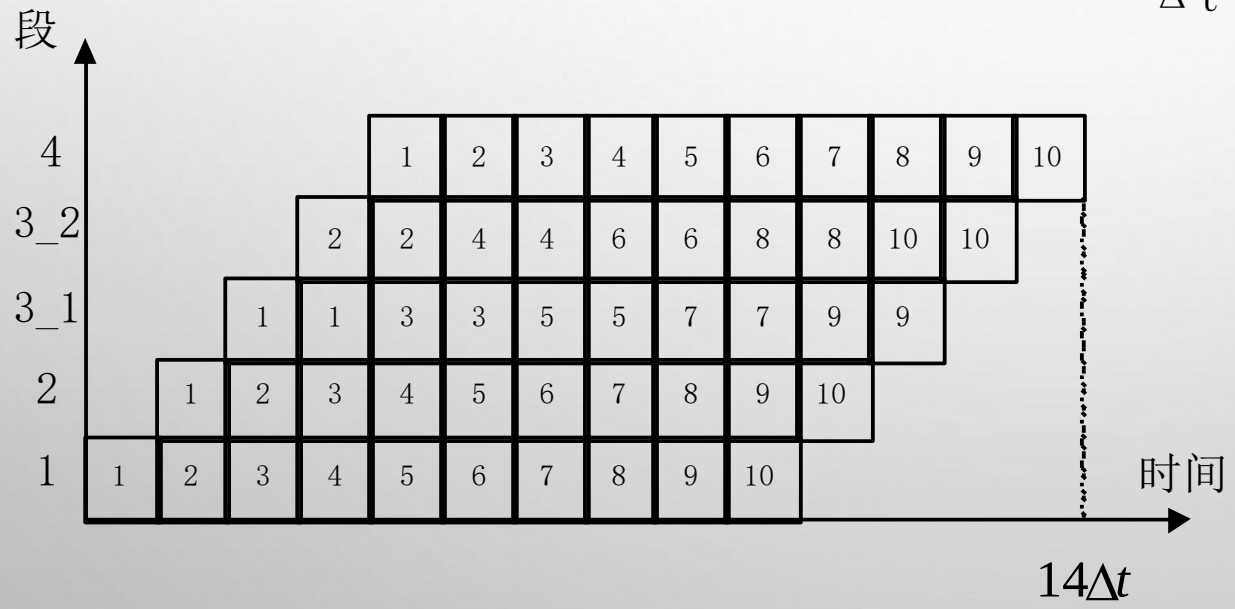
$$T_{pipeline} = 23\Delta t$$

$$TP = {}^{n}\!/\!_{T_{pipeline}} = {}^{10}\!/\!_{23\Delta t}$$

$$\Delta E = TP \cdot {}^{5\Delta t}\!/\!_{4} = {}^{50}\!/\!_{92} \approx 54.35\%$$

（3）重复设置部件

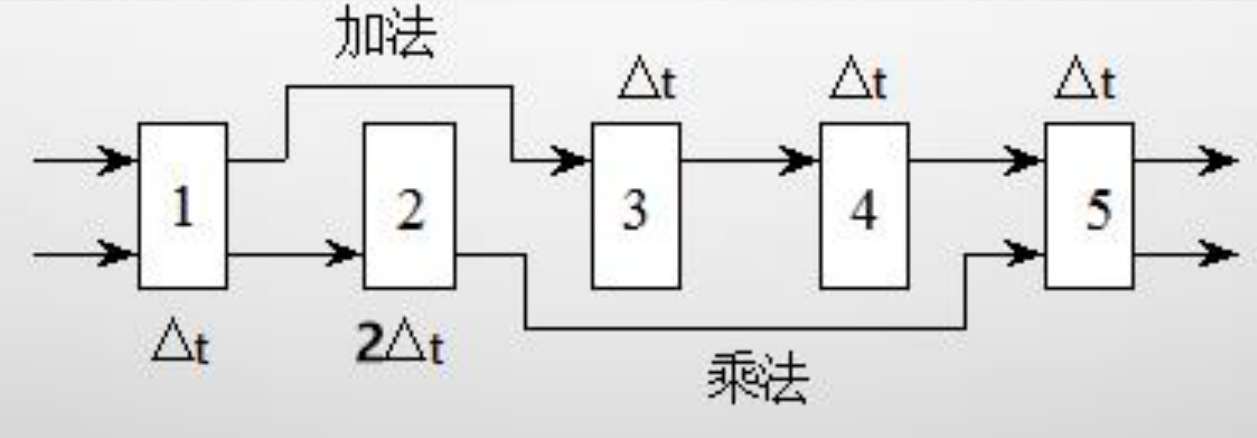




$$TP = \frac{n}{T_{pipeline}} = \frac{10}{14 \cdot \Delta t} = \frac{5}{7 \cdot \Delta t}$$

$$\text{吞吐率提高倍数} = \frac{5/7\ \Delta t}{10/23\ \Delta t} = 1.64$$

3.8 有一条动态多功能流水线由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：

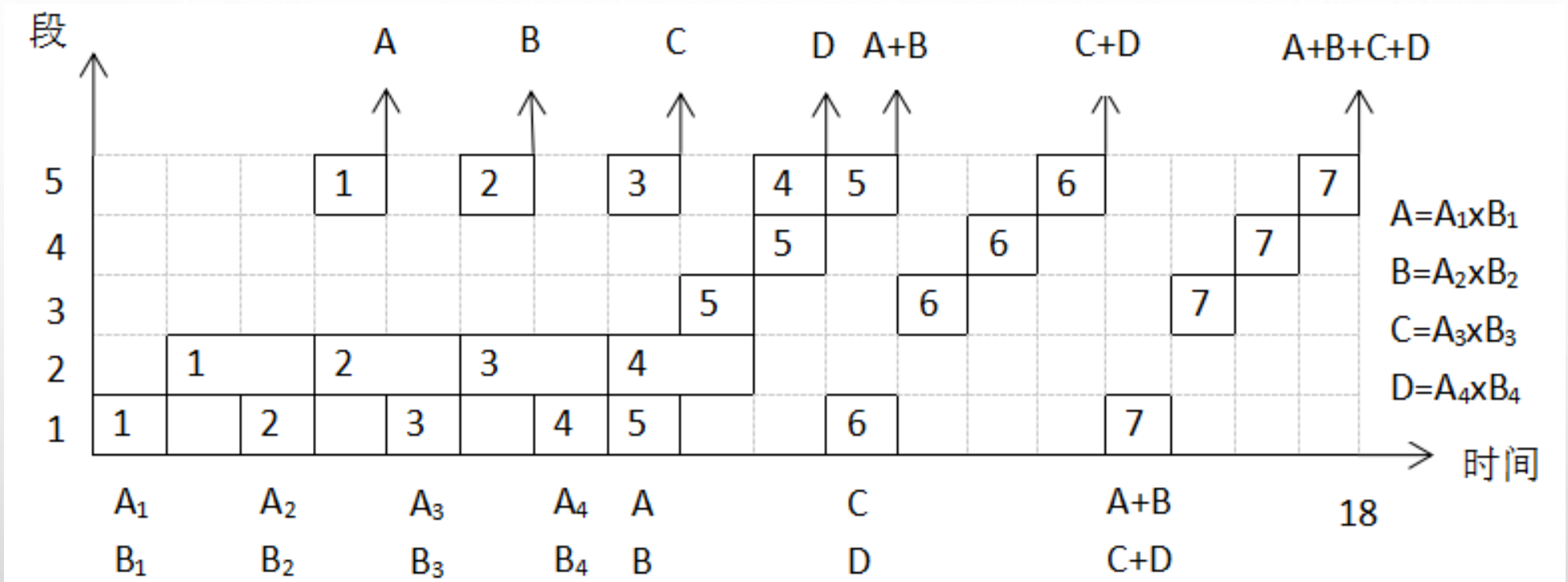
（1）选择适合于流水线工作的算法。

- 应先计算 $A_1 \times B_1$、$A_2 \times B_2$、$A_3 \times B_3$和$A_4 \times B_4$；
- 再计算 $(A_1 \times B_1) + (A_2 \times B_2)$
  $(A_3 \times B_3) + (A_4 \times B_4)$；
- 然后求总的累加结果。

（共计4次乘法，3次加法）

（2）画出时空图

（3）计算性能



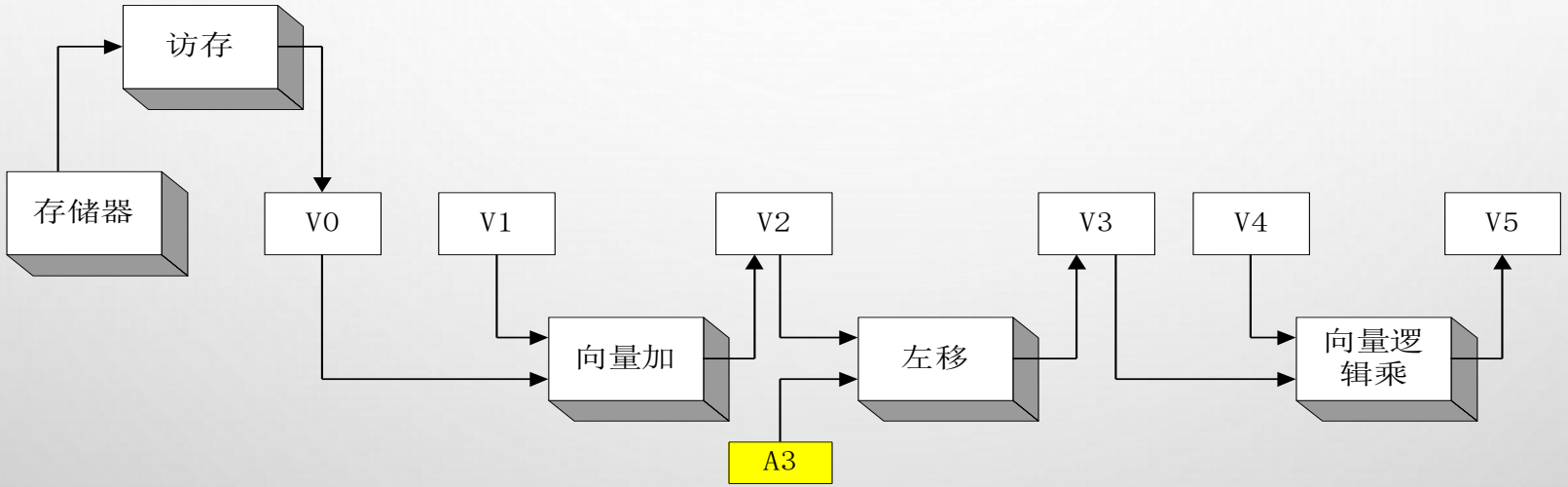
$$TP = \frac{7}{18\Delta t} \qquad S = \frac{28\Delta t}{18\Delta t} \approx 1.56 \qquad E = \frac{4\times 4 + 4\times 3}{5\times 18} \approx 0.311$$

4.5 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。



$T_{通过}=（1+7+1）+（1+3+1）+（1+4+1）+（1+2）=23(拍)$

$T_{总共}=T_{通过}+64=23+64=87(拍)$

4.8　在一台向量处理机上实现A＝B×s操作，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。计算A的总执行时间。

解：

因为向量长度超过了向量寄存器的长度，所以要采取分段开采方法。

每次循环主要由下面三条向量指令组成：

LV　　V_1，R_b　　；取向量B

MULTVS　V_2，V_1，F_s；向量和标量相乘

SV　　R_a，V_2　　；存向量

假设A和B的分别放在R_a和R_b之中，s在F_s中。

三条指令之间存在有写读数据相关，因此必须把它们分成3个编队，T_{chime}= 3。

$$T_{200}=4\times(15+T_{start})+200\times3$$
$$=60+(4\times T_{start})+600$$
$$=660+(4\times T_{start})$$

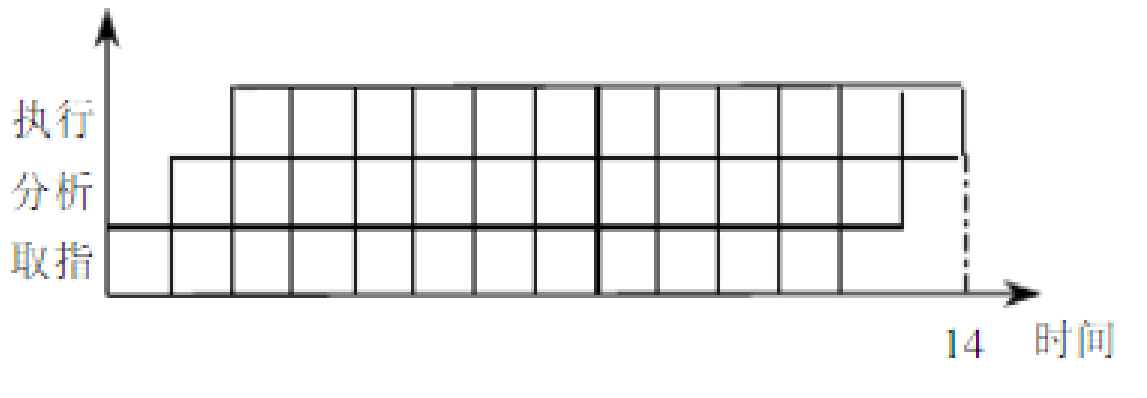
其中：$T_{start}=12+7+12=31$，

因此，$T_{200}=660+4\times31=784$

5.11　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。

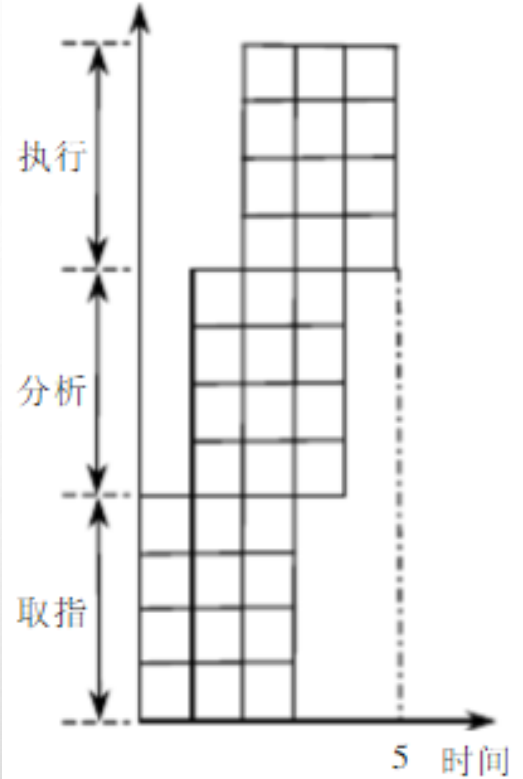
解：

标量流水处理机的时空图：



执行完12条指令需T1＝14△t。

超标量流水处理机的时空图：

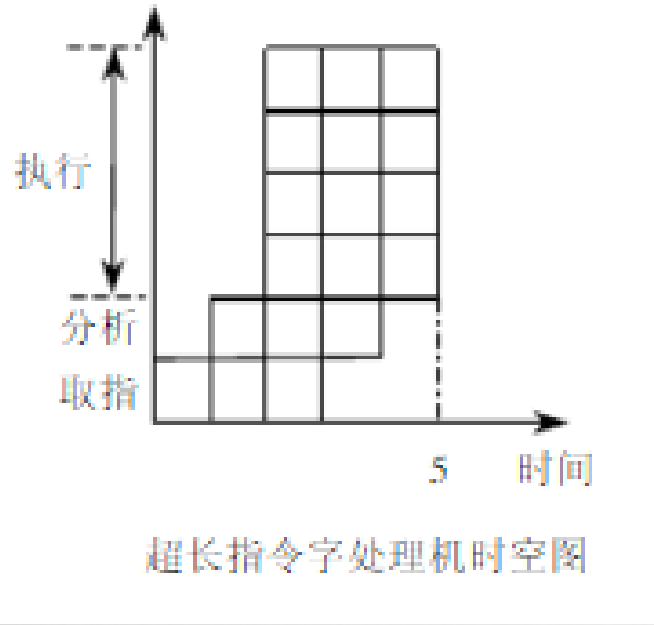


超标量处理机时空图

执行完12条指令需T2＝5△t，相对于标量流水处理机的加速比为：

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

超长指令字处理机的时空图：

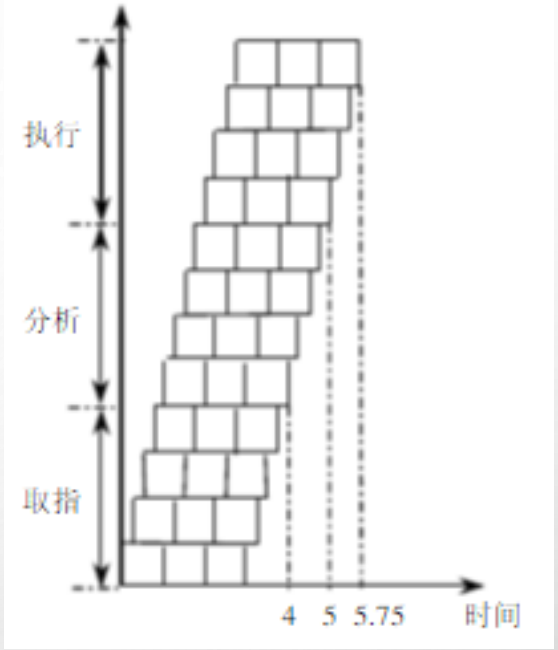


超长指令字处理机时空图

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T3＝5△t，相对于标量流水处理机的加速比为：

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

超流水处理机的时空图：



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：

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

补充一：下述指令，第一条指令已经执行并写入了结果，第二条指令已完成正等待写结果，给出记分牌（或Tomasulo)法所用的指令、功能部件（或保留站）和结果寄存器状态表。

```
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
```

解：记分牌算法：

(1) 指令状态表

指令	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				

由题意可知，第一条指令MULT.D指令已经完全执行完，其结果已写入F0中；第二条指令L.D指令也已经执行完，但是结果还没有写入目的寄存器F6中；第二条指令L.D与SUB.D和DIV.D指令之间存在关于寄存器F6的先写后读(RAW)相关,因此SUB.D和DIV.D在流出段等待，不能进入流水线的读操作数阶段。ADD.D指令与SUB.D之间存在关于加法器的结构相关。

解：

(2) 功能部件状态表

部件名称	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

由于乘法操作已经执行结束并写入结果中，所以乘法部件（Mult）的Busy字段为no，表示处于非忙状态；整数部件（Integer）的Busy字段为yes，表示部件正在忙，Op字段表示其正在执行L.D指令，加法部件（Add）的Busy字段为yes，其正在被SUB.D指令占用， F6寄存器还在被L.D指令占用中，所以Rj的状态为no，同理Divide部件。

解：

(3) 结果寄存器状态表

	F0	F2	F4	F6	F8	F10
部件名称				Integer	Add	Divide

结果寄存器状态表记录了当前机器状态下将把结果写入该寄存器的功能部件的名称。当前写F6的为Integer部件，写F8的为Add部件，写F10的为Divide部件。

解：

Tomasulo算法基本思想：

（1）记录与检测指令间的相关，操作数一旦就绪就立即执行，把发生RAW冲突的可能性降到最小。

（2）通过寄存器换名来消除WAR和WAW冲突。

当指令流出时，如果操作数还没有计算出来，则将该指令中相应的寄存器号换名为将产生这个操作数的保留站的标识。

指令流出到保留站后，其操作数寄存器号或者换成了数据本身（如果该数据已经就绪），或者换成了保留站的标识，不再与寄存器有关系。

解： Tomasulo算法：

(1) 指令执行状态

指令	流出	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			

由题意可知，第一条指令MULT.D指令已经完全执行完，其结果已写入CBD中；第二条指令L.D指令也已经完成有效地址计算，正在等待储存器的响应；ADD.D指令与SUB.D之间存在关于加法器的结构相关。

解：

(2) 保留站

部件名称	Busy	Op	Vj	Vk	Qj	Qk	A
Load	Yes	L.D					34+Reg[R2]
Mult	No						
Add	Yes	SUB.D		Reg[F2]	Load		
Divide	Yes	DIV.D	Reg[F0]			Load	

由于乘法操作已经执行结束并写入结果中，所以乘法部件（Mult）的Busy字段为no，表示处于非忙状态；Load的Busy字段为yes，表示部件正在忙，Op字段表示其正在执行L.D指令，完成有效地址计算，写入A字段中，加法部件（Add）的Busy字段为yes，其正在被SUB.D指令占用，第一操作数的F6还没有就绪，需要进行寄存器换名，将产生该操作数的保留站Load放入当前保留站的Qj中，第二操作数以及就绪，提取寄存器F2中操作数到当前保留站的Vk中，置Qk为0，同理Divide。

解：

(3) 结果寄存器状态表

	F0	F2	F4	F6	F8	F10
部件名称				Load	Add	Divide

结果寄存器状态表记录了当前机器状态下将把结果写入该寄存器的功能部件的名称。当前写F6的为Load部件，写F8的为Add部件，写F10的为Divide部件。

补充二：在含有一个PE的SISD机和一个含有 16个PE且连接成的超立方体结构的阵列处理机上计算下列求内积表达式 $S=\sum_{i=0}^{31}(A_i \times B_i)$

假定每个PE完成一次加法需要2Δt，完成一次乘法需要4Δt。在相邻PE之间传送一个数据需要1Δt。操作数A_i和B_i最初始存放在PE_i mod 16中，其中i=0，1…，31。取指令、取操作数、译码和写结果等时间均忽略不计。问：

(1)在SISD机上计算S时间？

(2)在超立方体结构上计算S的时间是多少？

解：

（1）在SISD机上计算s需要串行计算32次乘法和31次加法。

共需要时间：

$$32 \times 4 + 31 \times 2 = 190\Delta t$$

（2）共有16个PE，每个PE各自有A_i、B_i对两组。

每个PE内两次乘法和一次加法用时：

$$2 \times 4\Delta t + 1 \times 2\Delta t = 10\Delta t$$

每次得到立方体传输PE并相加用时：

$$\Delta t + 2\Delta t = 3\Delta t$$

立方体降到0维共需log16=4次，用时：$4 \times 3\Delta t = 12\Delta t$

总时间：

$$10\Delta t + 12\Delta t = 22\Delta t$$