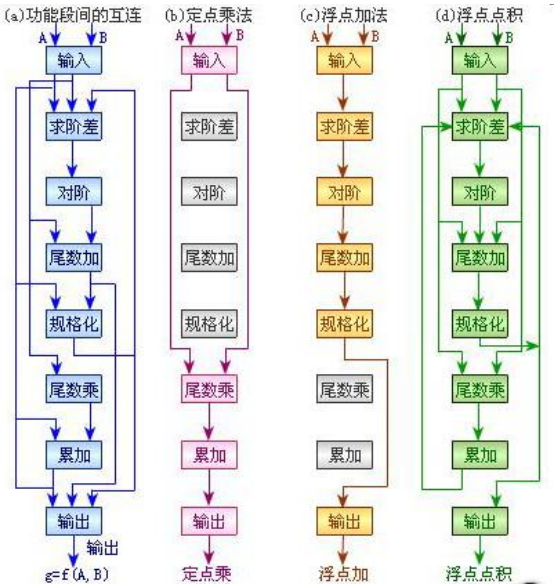
1、(复习题综合第 1 题)计算机运行以下指令：

线性多功能静态流水线，输入任务是不连续的情况，计算流水线的吞吐率、加速比和效率。用 TI－ASC 计算机的多功能静态流水线计算两个向量的点积：

$$Z=AB+CD+EF+GH$$



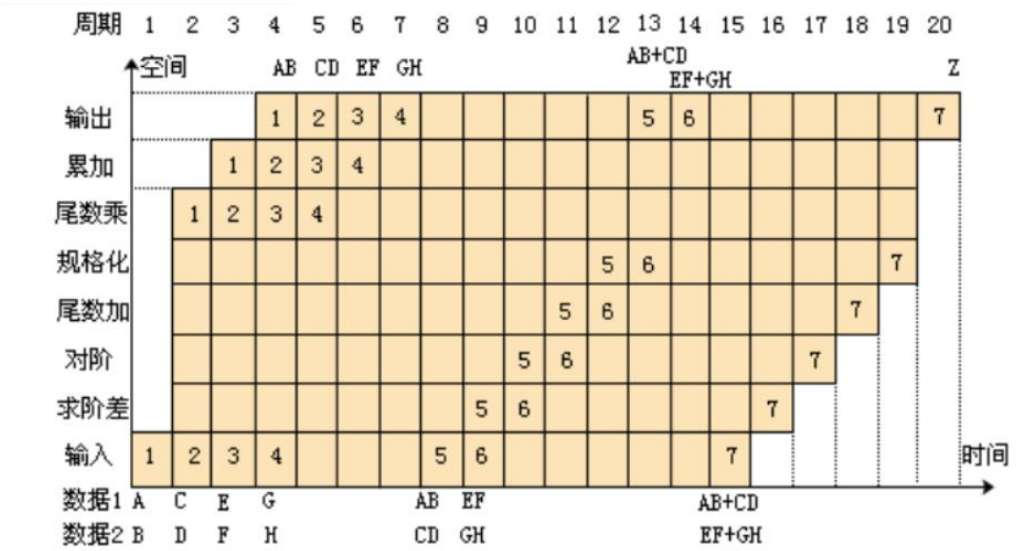
PS：题干里面的四个图用作参考，不一定全部用得到，需要哪个用哪个

解答：

先对整体的操作进行分模块，设计流水线如何进行，谨记一个原则：尽量减少数据相关性。

1. 先算四个点乘：AB；CD；EF；GH
2. 再算加法：AB + CD；EF + GH；(AB + CD) + (EF + GH)

流水线时空图：



PS：这是一个静态流水线，需要等待所有的点乘完成后，再进行加法。

从流水线时空图中看到，用 20 个时钟周期完成了 7 个运算。当每一个流水段的延迟时间都为△t 时：

流水线的吞吐率 TP 为：7/20t

如果采用顺序执行方式，完成一次乘法要用 4 个 t，完成一次加法要用 6 个 t，则完成全部运算要用：4*4t＋3*6t=34t

则流水线的加速比 S 为：34t/20t=1.70

整个流水线共有 8 段：

流水线效率 E 为:34t/8*20t=0.21（时空图的面积比）

2、(复习题综合第 3 题)一条有 4 个流水段的非线性流水线，每一段的延迟时间相等，预约表如下：

时间 / 流水段	1	2	3	4	5	6	7
S_1	×						×
S_2		×				×	
S_3			×		×		
S_4				×			

（1）写出禁止向量和冲突向量

（2）画出调度状态图

（3）求出最大吞吐量

（4）按最优调度连续输入 8 个任务，实际吞吐量，加速比和效率各为多少

解答：

（1）

求禁止启动距离：

对于 S1：7 - 1 = 6

对于 S2：6 - 2 = 4

对于 S3：5 - 3 = 2

对于 S4：没有

禁止向量 F：是一个流水线中所有禁止启动距离的集合，本题中 F 即为上面所求的禁止启动距离的集合：

禁止向量 F=（6，4，2）

冲突向量 C：由禁止向量得到初始冲突向量，禁止向量中最大的禁止启动时间作为初始冲突向量的位数。如果启动距离为禁止距离，则该位为 1 否则为 0。

本题中最大禁止时间是 6，所以是 6 位二进制，且从右起第 2，4，6 位为 1。

初始冲突向量 C=（101010）

（2）首先根据初始冲突向量 C 推算出全部冲突向量，根据 F=（6，4，2）可知，分别可以间隔 1，3，5 拍进入指令，

1. 间隔 1 拍输入第二条指令:初始冲突向量 C 向右移动 1 位,左边用 0 补齐,得到 010101,与初始冲突向量 C，进行一个并，得到最终冲突向量：C1=111111，

由于 C1=111111 所以第三条指令只能间隔 7 拍后进入，C1 向右挪 7 位，左边用 0 补齐，得到 000000，与 C 进行一个并，得到最终冲突向量：C=101010，回到 C 形成闭环。

2. 间隔 3 拍输入第二条指令：初始冲突向量 C 向右移动 3 位，左边用 0 补齐，得到 000101，与初始冲突向量 C，进行一个并，得到最终冲突向量：C2=101111，

第三条指令间隔 5 拍后进入，C2 向右挪 5 位，左边用 0 补齐，得到 000001，进行一个并，得到最终冲突向量：C3=101011，或者第三条指令间隔 7 拍后进入，C2 向右挪 7 位，左边用 0 补齐，得到 000000，进行一个并，得到最终冲突向量：C=101010，回到 C 形成闭环。

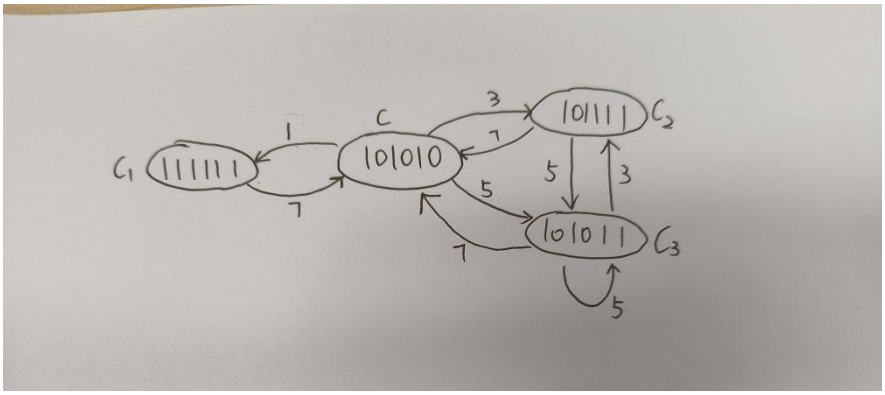
第四条指令间隔 3 拍后进入，C3 向右挪 3 位，左边用 0 补齐，得到 000101，与 C 进行一个并，得到最终冲突向量：C2=101111，形成闭环。

第四条指令间隔 5 拍后进入，C3 向右挪 5 位，左边用 0 补齐，得到 000001，与 C 进行一个并，得到最终冲突向量：C3=101011，形成闭环。

3. 间隔 5 拍输入第二条指令：初始冲突向量 C 向右移动 5 位，左边用 0 补齐，得到 000001，与初始冲突向量 C，进行一个并，得到最终冲突向量：C3=101011，后续操作与 2.一致

至此，所有的都形成闭环（必须全部都要收束回 C 初始冲突向量）

可以画出调度状态图：



（3）最优调度是（1，7），所以最小平均延迟时间是(1+7)/2=4，最大吞吐量为最小平均延迟时间的倒数，即 1/4

（4）8 个任务总耗时：7+1+7+1+7+1+7+1=32t

实际吞吐量：8/32t

加速比：(8*7)/32

效率：(8*7)/(32*4)

3、(复习题综合第 6 题)有一个 Cache 存储器，主存有 8 块(0-7)，Cache 有 4 块(0-3)，采用组相联映像，组内块数为 2 块。采用 LRU（近期最久未使用）替换算法。(1)指出主存各块与 Cache 各块之间的映像关系。(2)某程序运行过程中，访存的主存块地址流为：

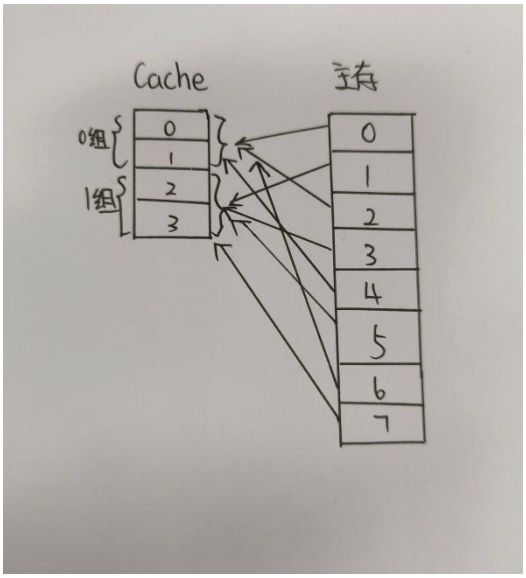
2， 3， 4， 1， 0， 7， 5， 3， 6， 1， 5， 2， 3， 7， 1

说明该程序访存对 Cache 的块位置的使用情况，指出发生块失效且块争用的时刻，计算 Cache 命中率。

解答：

（1） 由于是组相联映像，可以得到：主存的 0, 2, 4, 6 块对应于 Cache 中的 0, 1 块主存的 1, 3, 5, 7 块对应于 Cache 中的 2, 3 块。

按块分交叉映射



Cache 组相联相关知识群里电子版教材图 5.7

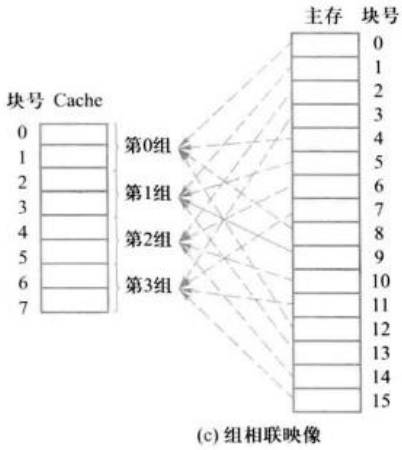
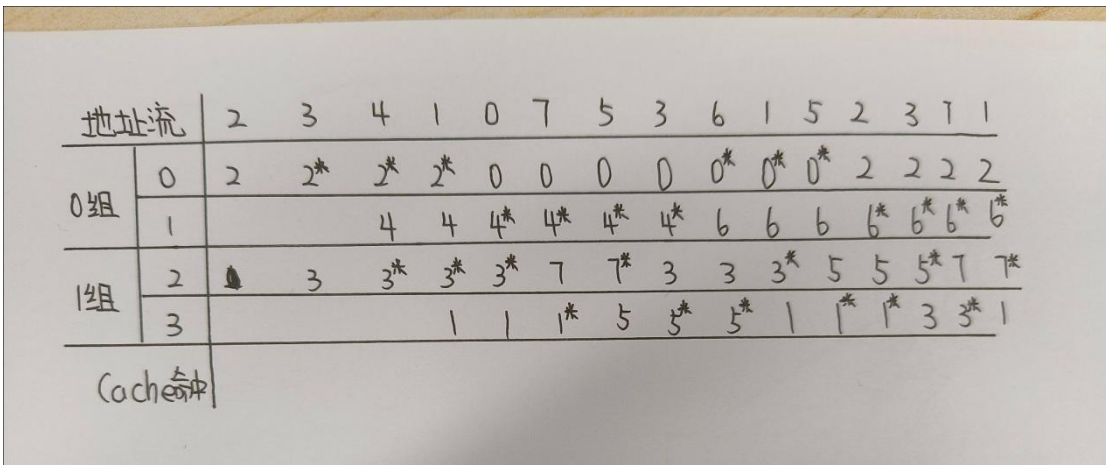


图 5.7 3 种映像规则

（2） 命中率 0，如图所示



4、(复习题综合第 8 题)tomasulo 算法的第 3 个时钟周期的指令状态，保留站状态，和寄存器结果状态如下图所示；

（其中 Op 表示现在保留站中正在工作的指令,Vj，Vk 表示已经准备好的操作数，Qj,Qk 表示已发射但未准备好的操作数）。已知 load 执行延时 2 个 cycles，add（sub）执行延时 2 个 cycles，mul 执行延时 10 个 cycles，div 执行延时 40 个 cycles。（1）写出 tomasulo 算法的核心思想。（2）写出第 5 个时钟周期的指令运行状态，保留站状态，和寄存器结果状态，并说明原因。

解答：

（1）第四章 ppt26 页

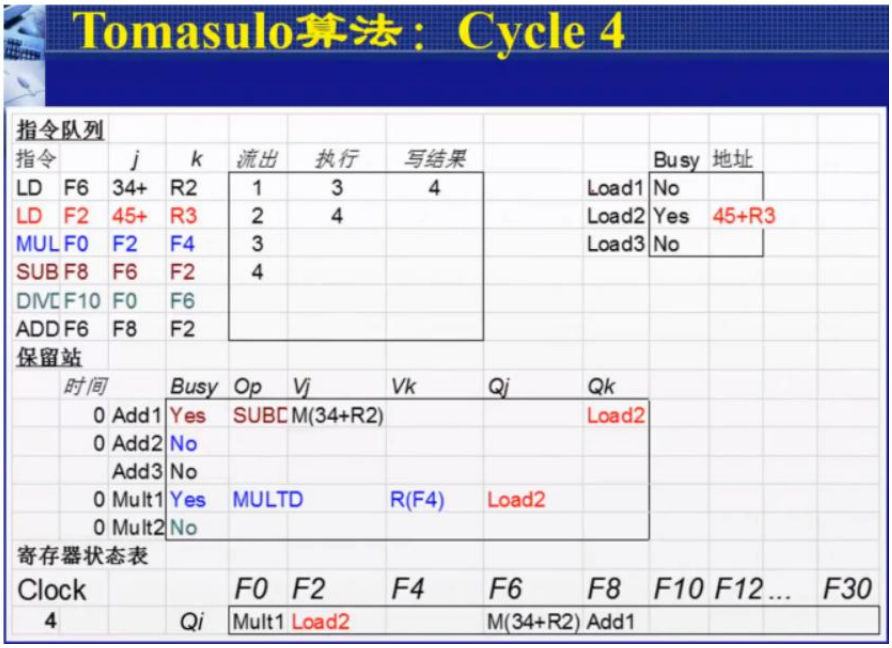
（2）参考 ppt 例 4.1 和例 4.2，或者参考链接里面 3.1https://zhuanlan.zhihu.com/p/499978902

Cycle 4：

对于 LD1，本周期结果写回，LD1 也就从 Load 缓冲器中释放，同时寄存器状态表中 F6 的值被改写为一个立即数，后续就与其无关了。

对于 LD2，本周期执行完毕，MUL 因操作数还没有就绪无法执行，不开启倒计时。

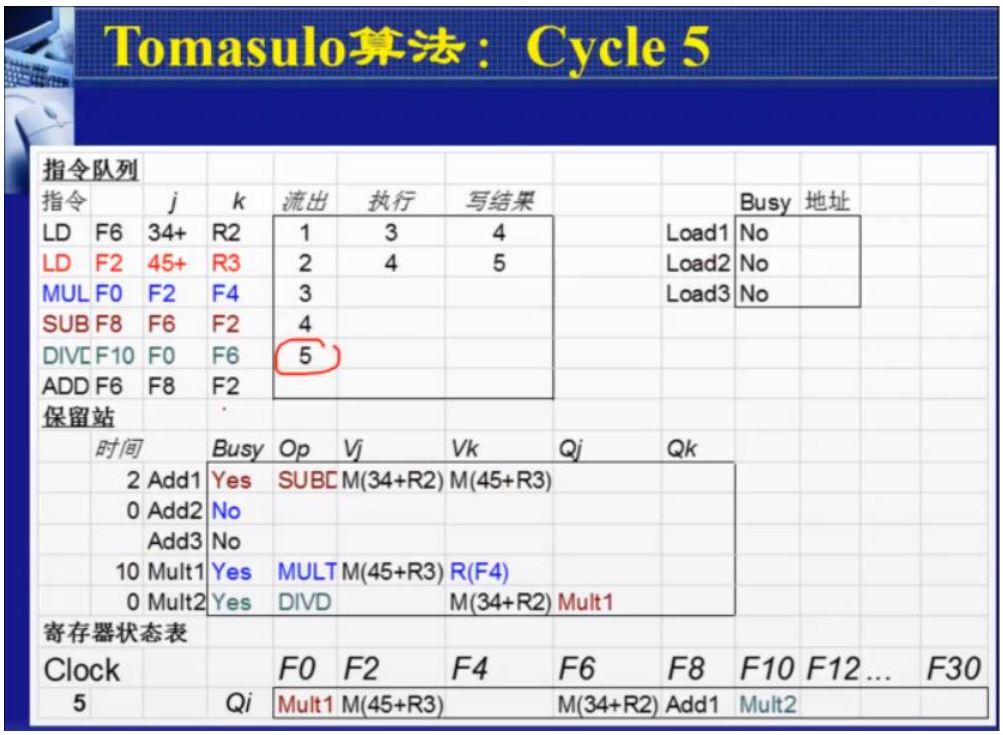
第四条指令 SUB 流入，先在指令队列中标识流出，然后在保留站中装入该指令，方法和之前是一样的，V_j 中可以装入立即数（因为已经操作完毕了），Q_k 中装入 LD2,同时在 F8 中装入 Add1。



Cycle 5：

对于 LD2，本周期写回结果完毕，从 Load 寄存器中释放，同时 F2 的值被改写,此时在保留站中的两条指令的操作数得到满足因此对于 Add1 而言，Q_k 清空，立即数装入 V_k 此刻这两条指令的操作数都已经就绪了，因此开始执行。

DIVD 正常流入，查询寄存器状态，可以发现 F0 仍被占用，F6 正常，因此 j,k 保留和上面相同的操作方式，同时在 F10 中留下 Mult2。



5、(复习题综合第 10 题)考虑考虑某两级 cache，第一级为 L1，第二级为 L2，两级 cache 的全局不命中率分别是 5%和 2%，假设 L2 的命中时间是 5 个时钟周期，L2 的不命中开销是 100 时钟周期，L1 的命中时间是 1 个时钟周期，平均每条指令访存 1.4 次，不考虑写操作的影响。求：

（1）计算 L2 的局部不命中率

（2）计算 L1 的不命中开销是多少个时钟周期

（3）每次访存的平均访存时间是多少个时钟周期

（4）每次访存的平均停顿时间是多少个时钟周期

（5）每条指令的平均停顿时间是多少个时钟周期

解答：由于 L2 全局=L1 全局*L2 局部，所以 L2 局部不命中率=2%/5%=40%

L1 的不命中开销=5+40%*100=45 个时钟周期

平均访存时间= L1 命中+L1 不命中*(L2 命中+L2 局部不命中*L2 不命中开销)

= 1+5%*(5+40%*100)= 3.25 个时钟周期

由于平均每条指令访存 1.4 次,且每次访存的平均停顿时间为 3.25-1.0=2.25 个时钟周期

所以每条指令的平均停顿时间=2.25x1.4=3.16 个时钟周期

6、关于结构冲突添加气泡，电子版教材图 3.19

当功能部件不是完全流水或者资源份数不够时，往往容易发生冲突。例如，有些流水线处理机只有一个存储器，将数据和指令放在一起。在这种情况下，如果在某个时钟周期，既要完成某条指令的访存操作，又要完成其后某条指令的“取指令”，那么就会发生访存冲突(结构冲突)。如图 3.19 中的“M”所示。为了消除这个冲突，可以在前一条指令访问存储器时将流水线停顿一个时钟周期，推迟后面取指令的操作，如图3.20所示。该停顿周期往往被称为“流水线气泡”，简称“气泡”。

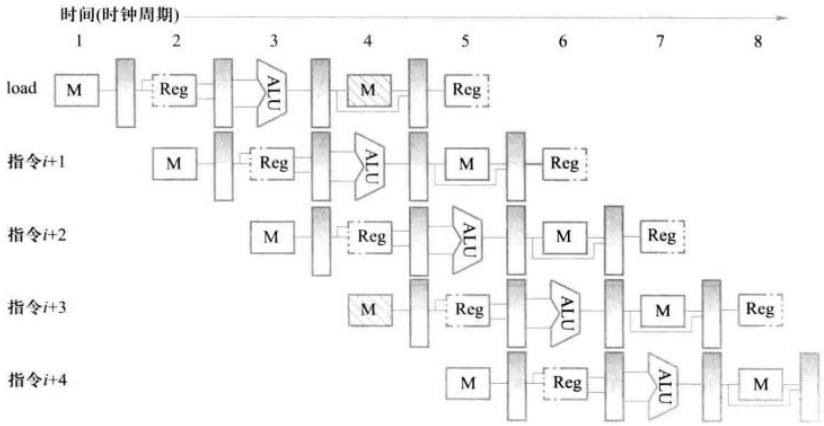


图 3.19　由于访问同一个存储器而引起的结构冲突

问：插入气泡后可能仍然存在冲突？

答：会的，在单一存储器的系统中，流水线暂停可以解决部分冲突问题，但如果暂停不足，后续周期可能仍会发生冲突。进一步暂停流水线操作或采用独立的存储器。

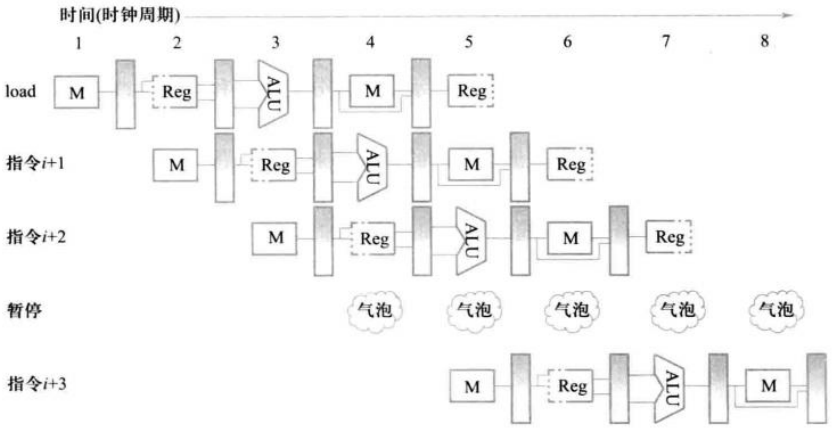


图 3.20　为消除结构冲突而插入的流水线气泡