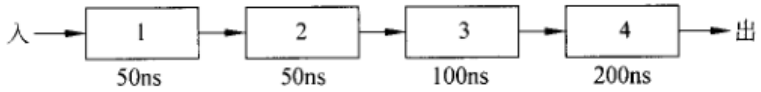
计算机系统结构作业四

3.6 有一条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是多少？

解答：

(1) 执行 10 条指令所需总时间 $T_k = 50+50+100+200\times10 = 2200$ ns

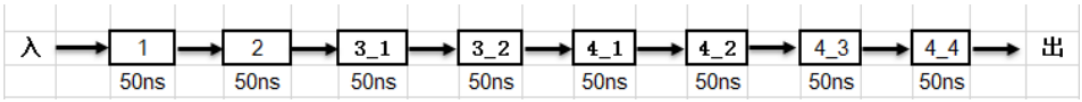
吞吐率 $TP = n / T_k = 10 / 2200\text{ns} = 4.55\times10^6\ s^{-1}$

实际效率 $E = T_0 / (k\times T_k) = 10\times(50+50+100+200) / (4\times2200) = 45.45\%$。

(2) 该流水线“瓶颈”是第 3 段和第 4 段，分别用时 100ns、200ns。

方法 1：细分瓶颈段

把第 3 段分成 2 个 50ms 的流水段，第 4 段分成 4 个 50ms 的流水段。



执行 10 条指令所需总时间 $T_k = 50\times8+(10-1)\times50 = 850$ns

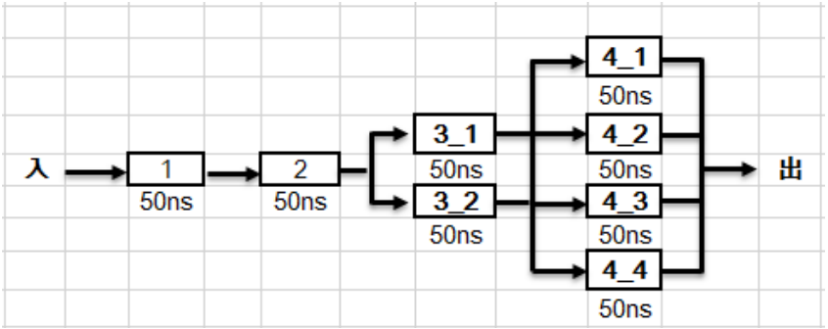
吞吐率 $TP = n / T_k = 10 / 850\text{ns} = 1.18\times10^7\ s^{-1}$

实际效率 $E = T_0 / (k\times T_k) = 10\times(50+50+100+200) / (8\times850)$

$= 58.82\%$。

方法 2：重复设置流水线

把第 3 段设置成 2 个并行的 100ms 的流水段，第 4 段设置成 4 个并行的 200ms 的流水段。



执行 10 条指令所需总时间 $T_k = 50\times8+(10-1)\times50 = 850$ns

吞吐率 $TP = n / T_k = 10 / 850\text{ns} = 1.18\times10^7\ s^{-1}$

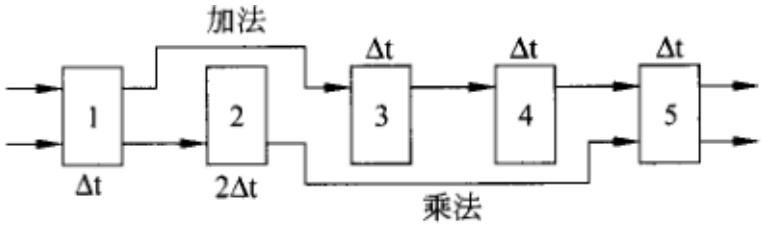
实际效率 $E = T_0 / (k\times T_k) = 10\times(50+50+100+200) / (8\times850)$

$= 58.82\%$。

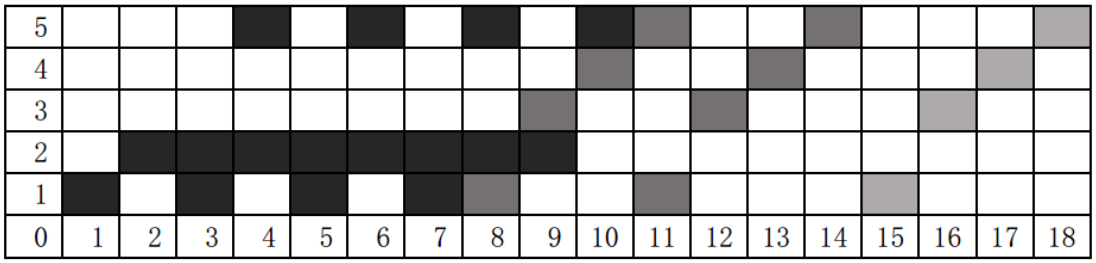
它的实际吞吐率和效率与第一种改进方法相同。

PS：其实改进方案有很多种，答案并不唯一。比如只改进第 4 段，将其细分为两段 100ns，或是重复设置；再或者有的同学将第 1、2、3 段进行合并。大家可以看到不同的方法会导致吞吐率和效率的不同变化，这两种衡量指标很可能一个高一个低，大家可以思考一下为什么~~~

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 2Δt，其余各段时间均为 Δt，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。若在该流水线上计算$\sum_{i=1}^{4}(A_i \times B_i)$，试计算其吞吐率、加速比和效率。



解答：选择适合流水线的算法，先计算$A_1 \times B_1$、$A_2 \times B_2$、$A_3 \times B_3$、$A_4 \times B_4$；再计算$A_1 \times B_1 + A_2 \times B_2$，$A_3 \times B_3 + A_4 \times B_4$；最后算总的累加结果。时空图如下：



从时空图可得：
T_k = 18Δt，　n = 7　那么吞吐率 TP = 7/(18Δt)；
又一次加法和一次乘法都是用时 4Δt，因此不用流水线时总时间为 28Δt。
加速比 S = 28 / 18 = 14/9 = 1.556
实际效率 E = 28 / (5 ×18) = 31.11%

PS：在计算加法的时候要等所需要的两个输入数计算完才可以计算哦~~另外不需要等待$A_4 \times B_4$计算完再计算$A_1 \times B_1 + A_2 \times B_2$，只要在$A_2 \times B_2$计算完成之后就可以计算~

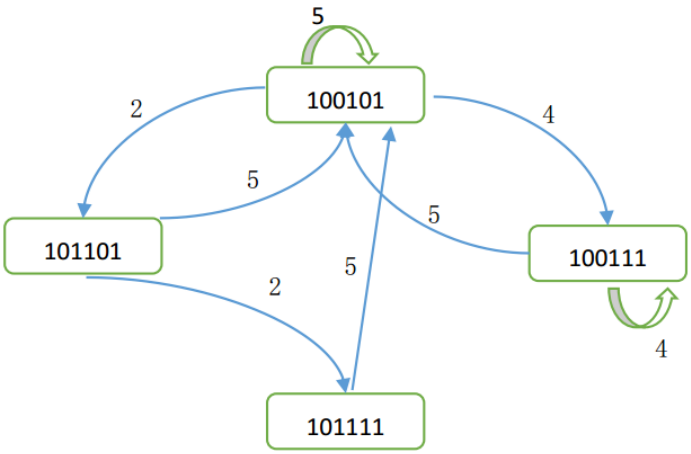
3.10：有一个 5 段流水线，各段执行时间均为 Δt，其预约表如下所示：

功能段 \ 时间	1	2	3	4	5	6	7
S1	√						√
S2		√			√		
S3			√	√			
S4				√			√
S5					√	√	

（1）画出流水线任务调度的状态转移图。
（2）分别求出允许不等时间间隔调度和等时间间隔调度的两种最优调度策略，计算着两种调度策略的流水线最大吞吐率。
（3）若连续输入 10 个任务，分别求采用这两种调度策略的流水线的实际吞吐率和加速比。

解答：

(1) 根据预约表得到禁止启动集合 F = {1,3,6}；
初始冲突向量 C0 = (100101)；
初始冲突向量逻辑右移 1、3、6 位时，不作处理；
逻辑右移 2、4、5 位要进行处理；
初始冲突向量 C0 逻辑右移 2 位之后：001001V100101=101101=C1；
初始冲突向量 C0 逻辑右移 4 位之后：000010V100101=100111=C2；
初始冲突向量 C0 逻辑右移 5 位之后：000001V100101=100101=C0；
对于中间冲突向量 C1：
逻辑右移 2 位之后：001011V100101=101111=C3；
逻辑右移 5 位之后：000001V100101=100101=C0；
对于中间冲突向量 C2：
逻辑右移 4 位之后：000010V100101=100111=C2；
逻辑右移 5 位之后：000001V100101=100101=C0；
对于中间冲突向量 C3：
逻辑右移 5 位之后：000001V100101=100101=C0；
则流水线任务调度的状态转移图为：



(2) 采用允许不等时间间隔调度的最优调度策略为（2,2,5），此时
平均延迟时间 T = (2+2+5)×Δt/3 = 3Δt
最大吞吐率 TP_{max} = 1/(3Δt)
采用等时时间间隔调度的最优调度策略为（4），此时
平均延迟时间 T = 4Δt
最大吞吐率 TP_{max} = 1/(4Δt)

(3) 采用允许不等时间间隔调度的最优调度策略为（2,2,5），此时
总执行时间 T_k = (7+9×3)×Δt = 34Δt

实际吞吐率 $TP = 10/(34\Delta t) = 5/(17\Delta t)$
加速比 $S = 10 \times 7\Delta t/T_k = 2.059$

采用等时时间间隔调度的最优调度策略为（4），此时
总执行时间 $T_k = (7+9\times4)\times\Delta t = 43\Delta t$
实际吞吐率 $TP = 10/(43\Delta t)$
加速比 $S = 10 \times 7\Delta t/T_k = 1.628$