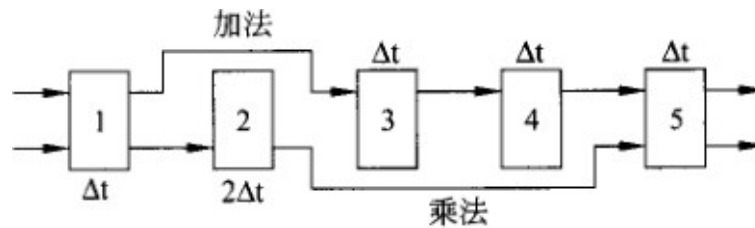


计算机系统结构作业六

1. (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 $2\Delta t$ ，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。



2. (3.9) 在一个 5 段流水线处理机上，各段执行时间均为 Δt ，需经 $9\Delta t$ 才能完成一个任务，其预约表如下所示。

时间 \ 功能段	1	2	3	4	5	6	7	8	9
S1	✓								✓
S2		✓	✓						
S3				✓			✓	✓	
S4				✓	✓				
S5						✓	✓		

- (1) 画出流水线任务调度的转移图。
- (2) 求流水线的最优调度策略和最大吞吐率。
- (3) 按最优调度策略输入 6 个任务，求流水线的实际吞吐率。

3. (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：

时间 \ 功能段	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。

4. (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
    
```

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。问：

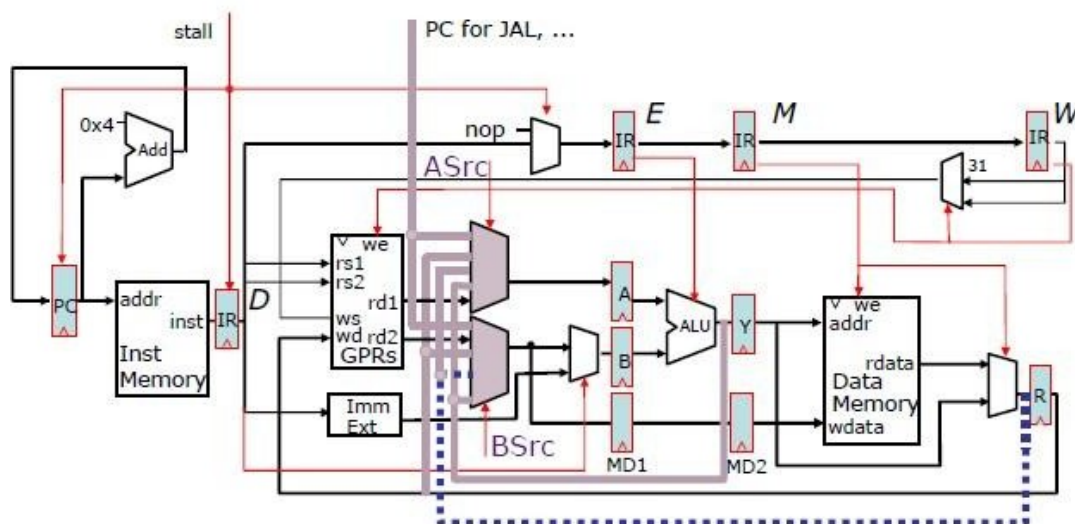
(1) 在没有任何其它定向（或旁路）硬件的支持下，请画出该指令序列执行的流水线时空图。假设采用排空流水线的策略处理分支指令，且所有的存储器访问都命中 Cache，那么执行上述循环需要多少个时钟周期？

(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。

5. 学习教材 3.5 节流水线的实现(page 83~90)，回答以下问题

A Fully-Bypassed Simple 5-stage Pipeline



R-type:

op	rs	rt	rd		func
----	----	----	----	--	------

I-type:

op	rs	rt	immediate16
----	----	----	-------------

J-type:

op	immediate26
----	-------------

		source(s)	destination
ALU	$rd \leftarrow (rs) \text{ func } (rt)$	rs, rt	rd
ALUi	$rt \leftarrow (rs) \text{ op } \text{imm}$	rs	rt
LW	$rt \leftarrow M[(rs) + \text{imm}]$	rs	rt
SW	$M[(rs) + \text{imm}] \leftarrow (rt)$	rs, rt	
BZ	$\text{cond}(rs)$		
	$\text{true: } PC \leftarrow (PC) + \text{imm}$	rs	
	$\text{false: } PC \leftarrow (PC) + 4$	rs	
J	$PC \leftarrow (PC) + \text{imm}$		
JAL	$r31 \leftarrow (PC), PC \leftarrow (PC) + \text{imm}$		31
JR	$PC \leftarrow (rs)$	rs	
JALR	$r31 \leftarrow (PC), PC \leftarrow (rs)$	rs	31

ws = Case opcode

ALU \Rightarrow rd
 ALUi, LW \Rightarrow rt
 JAL, JALR R31

re1 = Case opcode

ALU, ALUi,
 LW, SW, BZ
 JR, JALR \Rightarrow on
 J, JAL \Rightarrow off

we = Case opcode

ALU, ALUi, LW \Rightarrow (ws \neq 0)
 JAL, JALR \Rightarrow on
 ... \Rightarrow off

re2 = Case opcode

ALU, SW \Rightarrow on
 ... \Rightarrow off

we-bypass_E = Case opcode_E

ALU, ALUi, \Rightarrow (ws \neq 0)
 ... \Rightarrow off

we-stall_E = Case opcode_E

LW \Rightarrow (ws \neq 0)
 JAL, JALR \Rightarrow on
 ... \Rightarrow off

(1) Write down the bypass condition for the path between M (Memory) \rightarrow D(Decode) stages into register B. (The path is shown with a dotted line in the figure.)

Bypass MEM \rightarrow ID(B) =

(2) Write down the stall condition in which stalls are only caused by data hazards.

Stall =

(3) Please write down an instruction sequence (with fewer than 5 instructions) which activates the bypass logic in Question(1).