# 计算机系统结构作业五

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 9Δt 才能完成一个任务，其预约表如下所示。

| 功能段 \ 时间 | 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。

解答：

(1) 由预约表得出禁止表：

$$F = \{8,4,3,1\}$$

初始冲突向量：

$$C_0 = (10001101)$$

$C_0$的后继状态有：

$C_1 = SHR^{(2)}(C_0) \vee C_0 = (00100011) \vee (10001101) = (10101111),$

$C_2 = SHR^{(5)}(C_0) \vee C_0 = (00000100) \vee (10001101) = (10001101) = C_0,$

$C_3 = SHR^{(6)}(C_0) \vee C_0 = (00000010) \vee (10001101) = (10001111),$

$C_4 = SHR^{(7)}(C_0) \vee C_0 = (00000001) \vee (10001101) = (10001101) = C_0,$

$C_1$的后继状态有：

$C_5 = SHR^{(5)}(C_1) \vee C_0 = (10001101) = C_0,$

$C_{61} = SHR^{(7)}(C_1) \vee C_0 = (10001101) = C_0,$

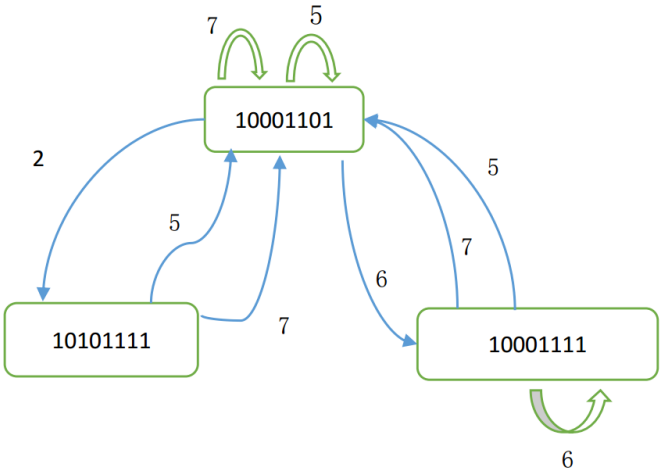
$C_3$的后继状态有：

$C_7 = SHR^{(5)}(C_3) \vee C_0 = (10001101) = C_0,$

$C_8 = SHR^{(6)}(C_3) \vee C_0 = (10001111) = C_3,$

$C_9 = SHR^{(7)}(C_3) \vee C_0 = (10001101) = C_0,$

因此，状态转移图：




（2）

由状态转移图可以看出最优调度策略为（2,5）；最大吞吐率为

$$\frac{2}{7\Delta t}$$

（3）

6 个任务实际吞吐率为

$$TP = \frac{6}{(2+5+2+5+2+9)\Delta t} = \frac{6}{25\Delta t}$$

3.11　在 MIPS 流水线上运行如下代码序列：

```
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。

解答：

(1) 时空图如下：

| 指令 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LW | IF | ID | EX | M | WB | | | | | | | | | | | | | | | | | |
| DADDIU | | IF | S | S | ID | EX | M | WB | | | | | | | | | | | | | | |
| SW | | | | | IF | S | S | ID | EX | M | WB | | | | | | | | | | | |
| DADDIU | | | | | | | | IF | ID | EX | M | WB | | | | | | | | | | |
| DSUB | | | | | | | | | IF | S | S | ID | EX | M | WB | | | | | | | |
| BNEZ | | | | | | | | | | | | IF | S | S | ID | EX | M | WB | | | | |
| LW | | | | | | | | | | | | | | | IF | S | S | IF | ID | EX | M | WB |

总的时钟周期数：(98×17) + 18 = 1684

(2) 时空图如下：

| 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 |

总的时钟周期数：(98×10) + 11 = 991

PS：关于“预测错误”的理解不同，这里我们按照只要遇到分支预测都不转移来处理，所以这题按(98×8) + 11 = 795 来算的也没有判错。如果考试出了，请按照参考答案的来

(3) 修改指令为：

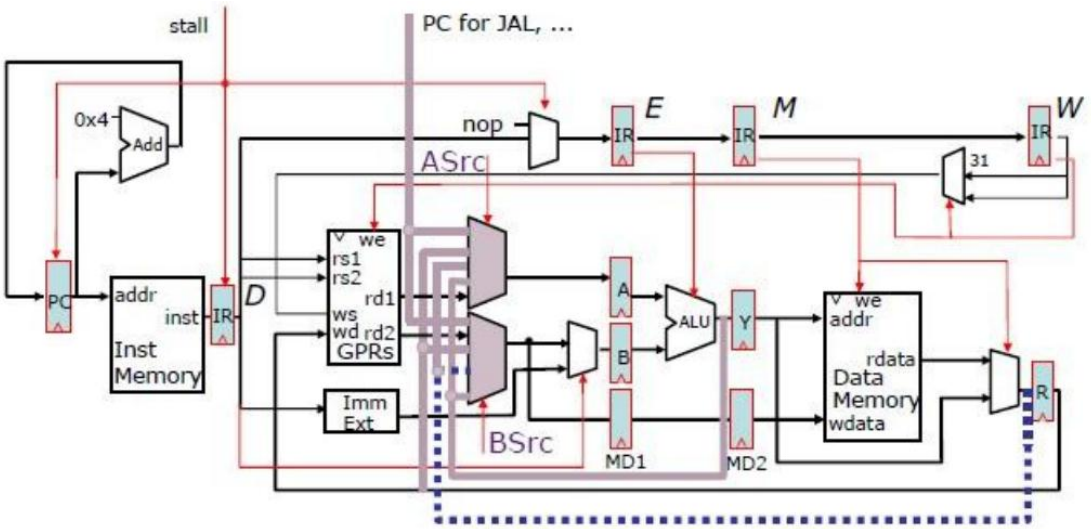
```
LOOP:
LW R1，  0(R2)
DADDIU R2，  R2，  #4
DADDIU R1，  R1，  #1
DSUB R4，  R3，  R2
BNEZ R4，  LOOP
SW R1，  -4(R2)
```

时空图：

| 指令 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| LW | IF | ID | EX | M | WB | | | | | | |
| DADDIU | | IF | ID | EX | M | WB | | | | | |
| DADDIU | | | IF | ID | EX | M | WB | | | | |
| DSUB | | | | IF | ID | EX | M | WB | | | |
| BNEZ | | | | | IF | ID | EX | M | WB | | |
| SW | | | | | | IF | ID | EX | M | WB | |
| LW | | | | | | | IF | ID | EX | M | WB |

总的时钟周期数：(98×6) + 10 = 598

Question ： 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 ← (rs) func (rt) | rs, rt | rd |
| ALUi | rt ← (rs) op imm | rs | rt |
| LW | rt ← M [(rs) + imm] | rs | rt |
| SW | M [(rs) + imm] ← (rt) | rs, rt | |
| BZ | *cond* (rs) | | |
| | *true:* PC ← (PC) + imm | rs | |
| | *false:* PC ← (PC) + 4 | rs | |
| J | PC ← (PC) + imm | | |
| JAL | r31 ← (PC), PC ← (PC) + imm | | 31 |
| JR | PC ← (rs) | rs | |
| JALR | r31 ← (PC), PC ← (rs) | rs | 31 |

ws = *Case* opcode
- ALU ⇒rd
- ALUi, LW ⇒rt
- JAL, JALR R31

we = Case opcode
- ALU, ALUi, LW ⇒(ws≠0)
- JAL, JALR ⇒on
- ... ⇒off

$we\text{-}bypass_E$ = Case $opcode_E$
- ALU, ALUi, ⇒(ws≠0)
- ... ⇒off

$we\text{-}stall_E$ = Case $opcode_E$
- LW ⇒(ws≠0)
- JAL, JALR ⇒on
- ... ⇒off

re1 = *Case* opcode
- ALU, ALUi, LW, SW, BZ ⇒on
- JR, JALR ⇒on
- J, JAL ⇒off

re2 = *Case* opcode
- ALU, SW ⇒on
- … ⇒off

Q1： Write down the bypass condition for the path between M (Memory) -> D (Decode) stages into register B. (The path is shown with a dotted line in the figure.)

**Answer：**
Bypass $_{MEM->ID(B)}$=
($opcode_M$==LW) and ($opcode_{ID}$==ALU or SW) and ($opcode_M$[rt]==$opcode_{ID}$[rt])

Q2： Write down the stall condition in which stalls are only caused by data hazards.

**Answer：**
Stall =
($opcode_E$==LW) and ($opcode_{ID}$==ALU or SW) and ($opcode_E$[rt]==$opcode_{ID}$[rt])

Q3： Please write down an instruction sequence (with fewer than 5 instructions) which activates the bypass logic in Question 1.

**Answer：（不唯一，仅做参考）**
LW R1 R2 0
LW R4 R5 0
ALU R1 R2 R3