

# 第一题

---

1) 流水线处理器与非流水线处理器的时钟周期分别是什么？

(a) 流水线时钟周期:  $500\text{ ps}$  , 非流水线时钟周期:  $300 + 400 + 350 + 500 + 100 = 1650\text{ ps}$

(b) 流水线时钟周期:  $200\text{ ps}$  , 非流水线时钟周期:  $200 + 150 + 120 + 190 + 140 = 800\text{ ps}$

2) **1w** 指令在流水线处理器和非流水线处理器中的总延迟分别是多少？

(a) 流水线处理器:  $500 \times 5 = 2500\text{ ps}$  , 非流水线处理器:  $1650\text{ ps}$

(b) 流水线处理器:  $200 \times 5 = 1000\text{ ps}$  , 非流水线处理器:  $800\text{ ps}$

3) 如果可以将原流水线数据通路的一级划分为两级, 每级的延迟是原级的一半, 会选择哪一级进行划分? 划分后处理器的时钟周期

(a) 选择 *MEM* 级, 划分后的延迟为  $250\text{ ps}$  , 处理器的时钟周期为  $400\text{ ps}$

(b) 选择 *IF* 级, 划分后的延迟为  $100\text{ ps}$  , 处理器的时钟周期为  $190\text{ ps}$

4) 假设没有阻塞和冒险, 数据存储器的利用率是多少 (占总周期数的百分比) ?

(a)  $15\% + 10\% = 25\%$

(b)  $30\% + 15\% = 45\%$

5) 假设没有阻塞和冒险, 寄存器堆的写寄存器端口的利用率是多少?

(a)  $50\% + 15\% = 65\%$

(b)  $30\% + 30\% = 60\%$

## 第二题

1) 指出指令序列中存在的相关及其类型。

(a) *RAW*:  $I_1$  和  $I_3$  的 \$1,  $I_2$  和  $I_3$  的 \$6

*WAR*:  $I_1$  和  $I_2$  的 \$6

(b) *RAW*:  $I_1$  和  $I_2, I_3$  的 \$5

*WAR*:  $I_2$  和  $I_3$  的 \$5

*WAW*:  $I_1$  和  $I_3$  的 \$5

2) 假设该流水线处理器没有转发, 指出指令序列中存在的冒 `nop` 指令以消除冒险。

只有 *RAW* 会引发数据冒险, 加入 `nop` 的指令序列如下

(a)

```
lw $1, 40($6)
add $6, $2, $2
nop
nop
sw $6, 50($1)
```

(b)

```
lw $5, -16($5)
nop
nop
sw $5, -16($5)
add $5, $5, $5
```

3) 假设该流水线处理器中有充分的转发。指出指令序列中存在的冒险并加入 `nop` 指令以消除冒险。

在有充分转发机制的流水线处理器, 需要注意 `lw` 的下一个指令后是否有 *RAW* 的情况, 若有就在那加 `nop`, 加入 `nop` 的指令序列如下

(a)

```
lw $1, 40($6)
add $6, $2, $2
sw $6, 50($1)
```

(b)

```
lw $5, -16($5)
nop
sw $5, -16($5)
add $5, $5, $5
```

4) 该指令序列在无转发和充分的转发时总执行时间分别是多少？后者相对于前者的加速比是多少。

(a) 无转发:  $9 \times 300 = 2700 \text{ ps}$  , 充分的转发:  $7 \times 400 = 2800 \text{ ps}$  , 加速比:  $\frac{2700}{2800} = 0.96$

(b) 无转发:  $9 \times 200 = 1800 \text{ ps}$  , 充分的转发:  $8 \times 250 = 2000 \text{ ps}$  , 加速比:  $\frac{1800}{2000} = 0.9$

5) 如果仅有 *ALU* 至 *ALU* 的转发 (没有从 *MEM* 到 *EX* 的转发) , 如何加入 **nop** 指令以消除可能的冒险?

只能在 *ALU* 计算后的结果转发到 *E* 级

(a)

```
lw $1, 40($6)
add $6, $2, $2
nop
sw $6, 50($1)
```

(b)

```
lw $5, -16($5)
nop
nop
sw $5, -16($5)
add $5, $5, $5
```

6) 该指令序列在仅有 *ALU* 至 *ALU* 的转发时总执行时间分别是多少？与无转发的情况相比，加速比是多少？

(a) 无转发:  $2700 \text{ ps}$  , *ALU* 至 *ALU* 的转发:  $8 \times 360 = 2880 \text{ ps}$  , 加速比:  $\frac{2700}{2880} = 0.94$

(b) 无转发:  $1800 \text{ ps}$  , *ALU* 至 *ALU* 的转发:  $9 \times 220 = 1980 \text{ ps}$  , 加速比:  $\frac{1800}{1980} = 0.91$

## 第三题

1) 为了将这条新指令增加到 MIPS 指令集，必须对流水线数据通路做什么改动？

(a) 将 Branch 改为二位，01 表示 beq 有效信号，10 表示 bezi 有效信号，若 Branch = 2'b10 且 Mem[RS] = 0 则执行

$$PC = PC + 4 + Offs$$

(b) 在 GRF 增加一个输出端口 RD3 输出 Rd 的数据，并且在 DM 前加一个 MUX 并控制 Rd 和 Rt 的数据写入 DM

2) 需要在第 1) 问的数据通路上增加哪些控制信号？

(a) 需将 Branch 增加一位

(b) 需增加 writeRd 信号在 DM 前的 MUX

3) 对新指令的支持是否会引入新的冒险？已有冒险导致的阻塞是否会更加严重？

(a) 会，bezi 指令会引来一个新的控制冒险，他的 PC 会到 W 级才会知道需不需要修改，所以会引起更严重的阻塞

(b) 不会，因为 swi 不会对寄存器进行任何修改

## 第四题

1) 如果没有转发或冒险检测电路，请插入 `nop` 指令以保证正确执行。

(a)

```
lw $1, 40($6)
nop
nop
add $2, $3, $1
add $1, $6, $4
nop
sw $2, 20($4)
and $1, $1, $4
```

(b)

```
add $1, $5, $3
nop
nop
sw $1, 0($2)
lw $1, 4($2)
nop
nop
add $5, $5, $1
sw $1, 0($2)
```

2) 重做第 1) 问，这次仅当通过改变或重排序指令都不能避免冒险时才插入 `nop` 指令。假设可以使用寄存器 `R7` 作为临时寄存器。

(a)

```
lw $7, 40($6)
add $1, $6, $4
nop
add $2, $3, $7
and $1, $1, $4
nop
sw $2, 20($4)
```

(b)

```
add $7, $5, $3
lw $1, 4($2)
nop
sw $7, 0($2)
add $5, $5, $1
sw $1, 0($2)
```

**3)** 如果处理器中存在转发，但忘了实现冒险检测单元（以为实现了），代码执行时会发生什么情况？

(a) I2 的 `add $2,$3,$1` 会读入 I1 的 `$1` 值，因为 I1 的 `lw` 还来不及在 I2 执行时写入 `$1`

(b) I2 的 `sw $1,0($2)` 会读入 I1 的 `$1` 值，因为 I1 的 `add` 还来不及在 I2 执行时写入 `$1`

**4)** 如果没有转发，对图中的冒险检测单元来说还需要哪些新的输入输出信号？以该指令序列为例，说明为什么需要这些信号。

输入信号：冒险检测单元需要在 EX 中检查 R 型指令和 lw 指令的 Rd 寄存器，在 MEM 中检查目标寄存器号，因此需要添加 ID/EX 流水线寄存器的 Rd 和 EX/MEM 的输出寄存器作为输入信号。