# 第7章 指令流水线

**7.2 选择题**

（1）C（2）A （3）D **（4）B** （5）C （6）A （7）D （8）A （9）C （10）B

**7.8**

**解：**

| CLKs | 取指 IF | 译码 ID | 执行 EX | 访存 MEM | 写回 WB |
|---|---|---|---|---|---|
| 1 | addi $s0, $s0,4 | | | | |
| 2 | lw $s1, ($s0) | addi $s0, $s0,4 | | | |
| 3 | add $s2, $s2, $s1 | lw $s1, ($s0) | addi $s0, $s0,4 | | |
| 4 | add $s2, $s2, $s1 | lw $s1, ($s0) | **Bubble** | addi $s0, $s0,4 | |
| 5 | add $s2, $s2, $s1 | lw $s1, ($s0) | **Bubble** | **Bubble** | addi $s0, $s0,4 |
| 6 | and $s3, $s1, $s2 | add $s2, $s2, $s1 | lw $s1, ($s0) | **Bubble** | **Bubble** |
| 7 | and $s3, $s1, $s2 | add $s2, $s2, $s1 | **Bubble** | lw $s1, ($s0) | **Bubble** |
| 8 | sub $s4, $s2, $s2 | add $s2, $s2, $s1 | **Bubble** | **Bubble** | lw $s1, ($s0) |
| 9 | Next Instr | sub $s4, $s2, $s2 | add $s2, $s2, $s1 | **Bubble** | **Bubble** |

**7.9**

**解：**

| CLKs | 取指 IF | 译码 ID | 执行 EX | 访存 MEM | 写回 WB |
|---|---|---|---|---|---|
| 1 | addi $s0, $s0,4 | | | | |
| 2 | lw $s1, ($s0) | addi $s0, $s0,4 | | | |
| 3 | add $s2, $s2, $s1 | lw $s1, ($s0) | addi $s0, $s0,4 | | |
| 4 | and $s3, $s1, $s2 | add $s2, $s2, $s1 | lw $s1, ($s0) | addi $s0, $s0,4 | |
| 5 | and $s3, $s1, $s2 | add $s2, $s2, $s1 | **Bubble** | lw $s1, ($s0) | addi $s0, $s0,4 |
| 6 | sub $s4, $s2, $s2 | and $s3, $s1, $s2 | add $s2, $s2, $s1 | **Bubble** | lw $s1, ($s0) |
| 7 | Next Instr | sub $s4, $s2, $s2 | and $s3, $s1, $s2 | add $s2, $s2, $s1 | **Bubble** |

**7.10**

**解：**程序执行周期数=508。

**7.11**

**解：**程序执行周期数=310。

**7.12**

**解：**程序执行周期数=80。

**7.13**

**解：**程序执行周期数=62。

**7.14**

**解：** $T_{min\_clk} = \max(T_{if\_max}, T_{id\_max}, T_{ex\_max}, T_{ex\_max}, T_{mem\_max}, T_{wb\_max})$，所以应该优化最慢的功能段，这里 ID 段最慢，应该优化其中的寄存器堆读延迟 $T_{RF\_read}$=150ps，当这个时间延迟优化到 100 时，ID 段时延和 IF、EX、MEM 相同，再进一步优化没有意义，只会增加成本。

**7.15**

**解：**（1）$[x]_{补}$=1111 1101 1111 1111B，即指令执行前(R1)=FDFFH，右移 1 位后为 1111 1110 1111 1111B，即指令执行后(R1)=FEFFH。

（2）至少需要5+(n-1)=5+(4-1)=8个时钟周期

（3）$I_3$的ID段被阻塞的原因：因为$I_3$与$I_1$和$I_2$都存在数据相关，需等到$I_1$和$I_2$将结果写回寄存器后，$I_3$才能读寄存器内容，所以$I_3$的ID段被阻塞，流水线中插入了3个气泡。$I_4$的IF段被阻塞的原因：因为$I_4$的前一条指令$I_3$在ID段被阻塞，所以$I_4$的IF段被阻塞。

需要注意的是此题中寄存器读写不能在一个时钟周期内完成，所以ID段和WB段的数据相关也必须插入气泡，相比教材上的方式多插入一个气泡。

（4）因 2*x 操作有左移和加法两种实现方法，故 x=x*2+a 对应的指令序列为

```
I1    LOAD     R1，[x]
I2    LOAD     R2，[a]
I3    SHL      R1        //或者ADD     R1，R1
I4    ADD      R1，R2
I5    STORE    R2，[x]
```

这 5 条指令在流水线中执行过程如下图所示。

| | 时间单元 | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 指令 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| I1 | IF | ID | EX | M | WB | | | | | | | | | | | | |
| I2 | | IF | ID | EX | M | WB | | | | | | | | | | | |
| I3 | | | IF | | | ID | EX | M | WB | | | | | | | | |
| I4 | | | | | | IF | | | | ID | EX | M | WB | | | | |
| I5 | | | | | | | | | | IF | | | | ID | EX | M | WB |

故执行 x=x*2+a 语句最少需要 17 个时钟周期。

**7.16**

**解：**

(1)已知计算机 M 采用 32 位定长指令字，一条指令占 4B，观察表中各指令的地址均

为 32 位，且相邻指令的地址差为 4 个地址单位，正好对应指令长度 4B，所以该计算机是按字节编址的。

(2)左移 2 位相当于以乘 4，根据汇编代码可知数组间的数据间隔为 4 个地址单位，而计算机按字节编址，所以数组 A 中每个元素占 4B。

(3)bne 指令的机器代码为 1446FFFAH，根据题目给出的指令格式，后 2B 的内容为 OFFSET 字段，所以该指令的 OFFSET 字段为 FFFAH，用补码表示，值为-6。

当系统执行到 bne 指令时，PC 自动加 4，PC 的内容就为 08048118H，而跳转的目标是 08048100H，两者相差了 18H，即-24 个字节距离，-24/-6=4。可知 bne 指令的转移目标地址计算公式为(PC)+4+OFFSET*4。

(4)由于数据相关而发生阻塞的指令为第 2、3、4、6 条，第 6 条指令会发生控制冒险。当前循环的第 5 条指令与下次循环的第 1 条指令虽然有数据相关，但由于第 6 条指令后有 3 个时钟周期的阻塞，因而消除了该数据相关。

## 7.2 选择题

1. C

线吞吐率：

最大吞吐率：对于 m 段的指令流水线而言，若各段的时间均为Δt，则最大吞吐率为

$T_{pmax} = \frac{1}{\Delta t}$

实际吞吐率：指流水线完成 n 条指令的实际吞吐率：

$$T_p = \frac{1}{\Delta t[1 + (m-1)/n])}$$

$\Delta t = 1/1.03\text{GHz}$ m=4, n=100

代入公式后即为 $1.0 \times 10^9$条指令/秒

2. A CPU 时钟周期由各段中时间最长的决定

3. D 最慢的功能段+流水线寄存器的延迟 80ps+20ps=100ps

4. B I2 写 R5，I3 读 R5 发生了 RAW 写后读冲突 插入 2 个气泡

I1: ADD R1, R2, R3 ； r2+r3->r1

I2: ADD R5, R2, R4 ； r2+r4->r5

I3: ADD R4, R5, R3 ； r5+r3->r4

I4: ADD R5, R2, R6 ； r2+r6->r5

| CLKs | IF | ID | EX | MEM | WB |
|---|---|---|---|---|---|
| 1 | I1 | | | | |
| 2 | I2 | I1 | | | |
| 3 | I3 | I2 | I1 | | |
| 4 | | I3 (访问 R5) | I2 | I1 | |
| 5 | | | | I2 | I1 |
| 6 | I4 | I3 | | | I2 |
| 7 | | I4 | I3 | | |
| 8 | | | I4 | I3 | |
| 9 | | | | I4 | I3 |
| 10 | | | | | I4 |

5. C 无冒险的是 I2 和 I4

I1：add s2,s1,s0

I2: load s3,0(t2)

I3: add s2, s2,s3

I4: store s2,0(t2)

| CLKs | IF | ID | EX | MEM | WB |
|---|---|---|---|---|---|
| 1 | I1 | | | | |
| 2 | I2 | I1 | | | |
| 3 | I3 | I2 | I1 | | |
| 4 | | I3 (因访问 R3 产生冲突而阻塞 | I2 | I1 | |
| 5 | | | | I2 | I1 |
| 6 | I4 | I3 访问 R3 | | | I2 写回 s3 |
| 7 | | I4 | I3 | | |
| 8 | | | I4 | I3 | |
| 9 | | | | I4 访存阻塞 | I3 写回 S2 |
| 10 | | | | I4 | |
| 11 | | | | | I4 |

8. A 错误的是：包含生成控制信号的控制部件

9. C 超标量流水技术， 采用了多发技术，CPI<1 ,但是流水线功能段的处理时间不变

7.8 正常完成需要 5+4=9T，实际用了 17T， 插入气泡=17-9=8

Addi $s0,$s0,4

Lw $s1,($s0)

Add $s2,$s2,$s1

And $s3,$s1,$s2

Sub $s4,$s2,$s2

| CLKs | IF | ID | EX | MEM | WB |
|---|---|---|---|---|---|
| 1 | addi | | | | |
| 2 | lw | addi | | | |
| 3 | add | Lw 访问$s0<br>出现数据冒险 | addi | | |
| 4 | add | lw | nop | addi | |
| 5 | add | lw | nop | nop | Addi 写回 s0 |
| 6 | | add 冒险 | lw | nop | nop |

| | | | | | |
|---|---|---|---|---|---|
| 7 | | add | nop | lw | nop |
| 8 | and | Add 访问$s1 | nop | nop | Lw 写回$s1 |
| 9 | | and | add | nop | nop |
| 10 | | and | nop | add | nop |
| 11 | Sub | And 访问$s2 | nop | nop | Add 写回$s2 |
| 12 | | Sub(访问$s2) | and | nop | nop |
| 13 | | sub | nop | and | nop |
| 14 | | Sub(访问$s2) | nop | nop | And 写回$s2 |
| 15 | | | Sub | nop | nop |
| 16 | | | | sub | nop |
| 17 | | | | | sub |

7.10 五级流水，分支跳转指令在 EX 段执行， 无分支预测，无分支延迟槽技术，计算程序的执行周期

```
Add $S0, $0,100   ；i=100
While_loop:
   Beq $s0, $0, done
   Addi $s0, $s0, -1
   J while_loop
done:
```

解答：

(1). i from 100 to 1 循环体执行 100 次，包括 beq, addi, J 指令，这三条指令，对于 Beq 而言，均不会跳转，因而不会引发 bubble，但是 J 指令会带入 2 个 bubble(因为在 EX 段才能知道跳转的结果)，相当于两条 Nop 空操作指令，所以，循环体一次相当于 3+2=5 条指令； 5*100=500 条，流水线内 CPI=1，故相当于 500cc

(2). 循环体外还有一条指令 ；1cc

(3). I=0 时，beq 执行一次，退出循环，执行 done 开始的分支地址指令，但是此时代入 2 个 bubble，故执行完此 3 条指令需要 m+(n-1)=5+2=7cc

所以，程序执行周期=1+500+7=508

I=2 时，1+5*2+5+（3-1）=18

| CLKs | IF | ID | EX | MEM | WB |
|---|---|---|---|---|---|
| 1 | Add | | | | |
| 2 | beq | add | | | |
| 3 | addi | beq | add | | |
| 4 | J | addi | beq | add | |
| 5 | A | j | addi | beq | add |
| 6 | B | A | j | addi | beq |
| 7 | beq | bubble | bubble | j | addi |
| 8 | addi | beq | bubble | bubble | J |
| 9 | J | addi | beq | bubble | bubble |
| 10 | A | J | addi | beq | bubble |
| 11 | B | A | J | Addi | beq |
| 12 | Beq | bubble | bubble | J | addi |
| 13 | Addi | beq | bubble | bubble | J |
| 14 | J | addi | Beq | bubble | bubble |
| 15 | Done | | | beq | bubble |
| 16 | | Done | | | beq |
| 17 | | | Done | | |
| 18 | | | | Done | |
| 19 | | | | | done |

| CLKs | 取值 IF | 译码 ID | 执行 EX | 访存 MEM | 写回 WB | 说明 |
|---|---|---|---|---|---|---|
| 1 | add | | | | | |
| 2 | beq | add | | | | |
| 3 | addi | beq | add | | | |
| 4 | j | addi | beq | add | | |
| 5 | A | j | addi | beq | add | |
| 6 | B | A | j | addi | Beq | j 跳转，同步清空前两部流水线 |
| 7 | beq | / | / | j | Addi | |
| 8 | addi | beq | / | / | j | |
| 9 | j | addi | beq | / | / | 仅在第 100 次时 beq 选择跳转，同步清空前两部流水线，退出循环 |
| 重复第 5~9 步，经历 100 次 addi 指令后 i=0，循环也进行了 100 次 | | | | | | |
| 4+1+5*100=505 | A | / | / | beq | / | |
| 506 | B | A | / | / | beq | |
| 507 | | B | A | / | / | |
| 508 | | | B | A | / | |

从首条指令 add 进入 ID，到所有指令和插入的气泡都走到 WB（A,B 本就是后加的辅助指令，

7.11 采用动态分支预测技术 BTB 表

Beq 和 J 指令首次第一轮，BTB 表为空，所以，结果等同于未做预测

故指令数：i=100， 1+5（循环体内）=6cc

I from 99 to 1, 查 BTB 表，利用了动态分支预测技术，消除气泡

故指令数：3（循环体内）*99=297

I=0: beq 跳转，预测错误，引发 2 个 bubble，

5+（3-1）=7cc

所以，6+297+7=310cc

| CLKs | 取值 IF | 译 码 ID | 执行 EX | 访存 MEM | 写回 WB | 说明 |
|---|---|---|---|---|---|---|
| 1 | add | | | | | |
| 2 | beq | add | | | | 首次读入 beq，查表无果 |
| 3 | addi | beq | add | | | |
| 4 | j | addi | beq | add | | 首次读入 j，查表无果；beq 指令入表，预测不分枝 |
| 5 | A | j | addi | beq | add | 首次 addi,执行后 i=99 |
| 6 | B | A | j | addi | Beq | 首次执行 j，同步清空前两部流水线，j 指令入表跳转地址为 beq 地址 |
| 7 | beq | / | / | j | Addi | 再次读入 beq，表中查得跳转目标为 addi,预测跳转 |
| 8 | addi | beq | / | / | j | |
| 9 | j | addi | beq | / | / | 再次执行 beq，判断知预测正确，不浪费周期；<br>再次读入 j，查表，直接跳转，不浪费周期 |
| 10 | beq | j | addi | beq | / | |
| 11 | addi | beq | j | addi | beq | |
| 重复第 9~11 步，经历 99 次 addi 指令后 i=0，循环也进行了 99 次 | | | | | | |
| 8+1+3*99=306 | j | addi | beq | j | addi | 执行 beq，发现预测错误，同步清空前两部流水线 |
| 307 | A | / | / | beq | j | |
| 308 | B | A | / | / | beq | |
| 309 | | B | A | / | / | |
| 310 | | | B | A | / | |





7.12

```
Addi $s0, $0,0
Addi $s1, $0,0
For_loop:
   Slti $t1, $s0,10
   Beq$t1, $0, done
   Addi $s1, $s1, $s0
   Addi $s0, $0,1
 J for_loop
done:
```

解答类似 7.10

2+（5+2）*10+1+ 5+(3-1) =80cc

7.13

第一轮，i=0， 相当于无预测， 故指令数：2+5+2（循环体内，2 为 J 产生的 2 个气泡）

I from 1 to 9 ， 利用 BTB 表进行预测，指令数 9*5（循环体内，无气泡）

I=10 时， beq 跳出，预测错误，代入 2 个 bubble，故 5+（3-1）=7cc

此外，beq 之前还有一条指令，所以，共计 2+5+2+9*5+7+1=62cc

7.16 按序发射 按序完成

正常无阻塞如下表

| CLKs | IF | ID | EX | MEM | WB |
|---|---|---|---|---|---|
| 1 | Sll | | | | |
| 2 | add | Sll | | | |
| 3 | load | Add 读 R4 | sll | | |
| 4 | add | Load 读 R4 | add | sll | |
| 5 | add | add | load | add | Sll 写回 R4 |
| 6 | bne | Add | add | load | Add 写回 R4 |
| 7 | | bne | add | add | Load 写回 R5 |
| 8 | | | bne | add | Add 写回 R1 |
| 9 | | | | bne | Add R2 自增 1 |
| 10 | | | | | Bne 分支转移 |

出现冒险

分支指令 bne 的执行会引起 3cc 的阻塞，也就是其下一次迭代的前三条指令要

等到 3cc 以后才能进入流水线，而此时，add R2，R2，1 已经执行完毕，故不会发生冲突

| CLKs | IF | ID | EX | MEM | WB |
|---|---|---|---|---|---|
| 1 | Sll | | | | |
| 2 | add | Sll | | | |
| 3 | load | | sll | | |
| 4 | add | | | sll | |
| 5 | add | Add 读 R4 | | | Sll 写回 R4 |
| 6 | bne | load | add | | |
| 7 | Loop: sll | | | add | |
| 8 | | Load 读 R4 | | | Add 写回 R4 |
| 9 | | Add 读 R5 | load | | |
| 10 | | | | load | |
| 11 | | Add 读 R5 | | | Load 写回 R5 |
| 12 | | add | add | | |
| 13 | | bne | add | add | |
| 14 | | | | add | Add 写回 R1 |
| 15 | | | | | Add 写回 R2 |
| 16 | | Bne 读 R2 | | | |
| 17 | | | bne | | |
| 18 | | | | bne | |
| 19 | | | | | bne |
| 20 | Sll 进入 pipeline | | | | |