

4.6 下面的一段 MIPS 汇编程序是计算高斯消去法中的关键一步,用于完成下面公式的计算:

$$Y = a \times X + Y$$

其浮点指令延迟如表 4.3 所示,整数指令均为 1 个时钟周期完成,浮点和整数部件均采用流水。整数操作之间以及与其他所有浮点操作之间的延迟为 0,转移指令的延迟为 0。X 中的最后一个元素存放在存储器中的地址为 DONE。

```
FOO:  L.D      F2,0(R1)
      MUL.D   F4,F2,F0
      L.D      F6,0(R2)
      ADD.D   F6,F4,F6
      S.D      F6,0[R2]
      DADDIU  R1,R1,#8
      DADDIU  R2,R2,#8
      DSUBIU  R3,R1,#DONE
      BNEZ    R3,FOO
```

- (1) 对于标准的 MIPS 单流水线,上述循环计算一个 Y 值需要多少时间? 其中有多少空转周期?
- (2) 对于标准的 MIPS 单流水线,将上述循环顺序展开 4 次,不进行任何指令调度,计算一个 Y 值平均需要多少时间? 加速比是多少? 其加速是如何获得的?
- (3) 对于标准的 MIPS 单流水线,将上述循环顺序展开 4 次,优化和调度指令,使循环处理时间达到最优,计算一个 Y 值平均需要多少时间? 加速比是多少?
- (4) 对于采用图 4.9 的前瞻执行机制的 MIPS 处理器(只有一个整数部件),当第二次循环执行到  
BNEZ R3,FOO  
时,写出前面所有指令的状态,包括指令使用的保留站、指令起始节拍、执行节拍和写结果节拍,并写出处理器当前的状态。
- (5) 对于两路超标量的 MIPS 流水线,设有两个指令流出部件,可以流出任意组合的指令,系统中的功能部件数量不受限制。将上述循环展开 4 次,优化和调度指令,使循环处理时间达到最优。计算一个 Y 值平均需要多少时间? 加速比是多少?
- (6) 对于如图 4.13 结构的超长指令字 MIPS 处理器,将上述循环展开 4 次,优化和调度指令,使循环处理时间达到最优。计算一个 Y 值平均需要多少时间? 加速比是多少?

表 4.3 本节使用的浮点流水线的延迟

产生结果的指令	使用结果的指令	延迟(时钟周期数)
浮点计算	另一个浮点计算	3
浮点计算	浮点 store(S.D)	2
浮点 load(L.D)	浮点计算	1
浮点 load(L.D)	浮点 store(S.D)	0

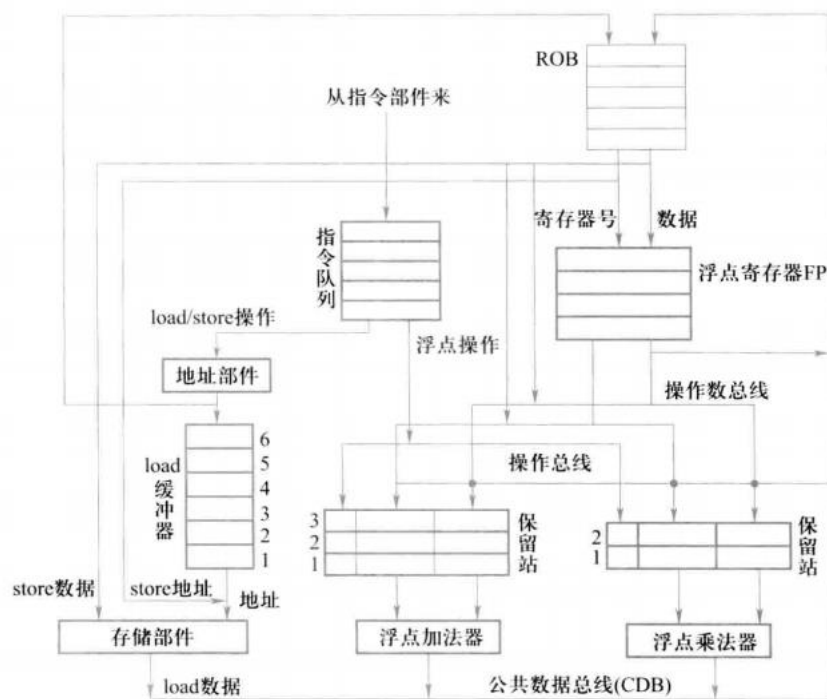
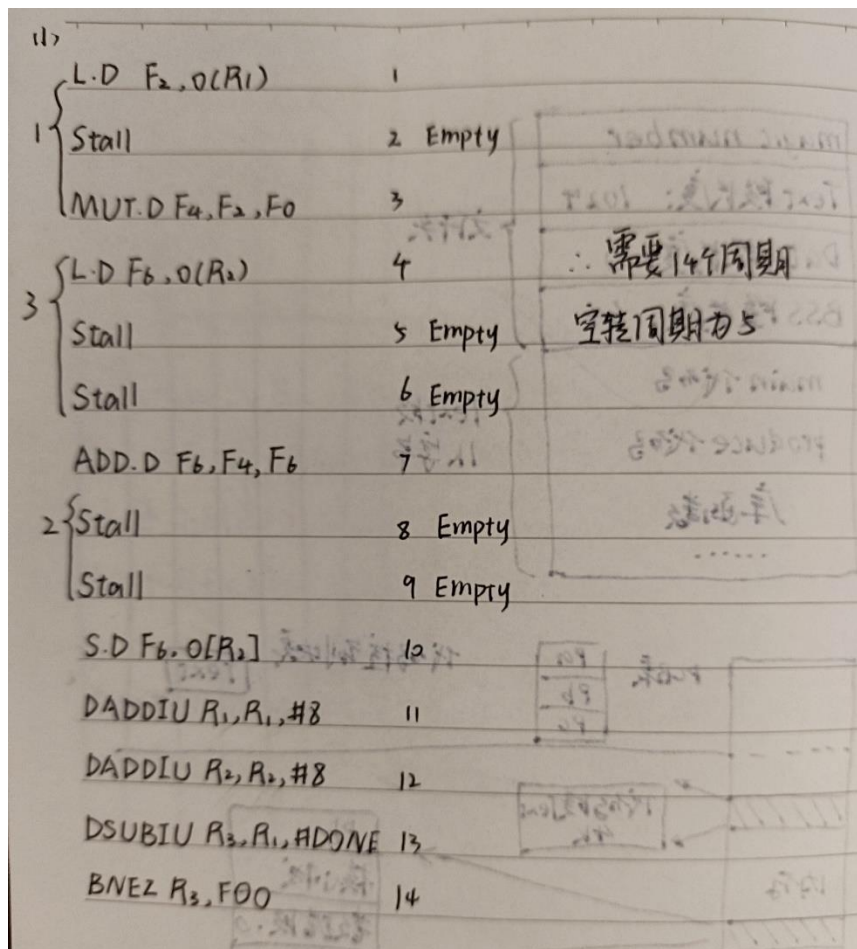


图 4.9 基于 Tomasulo 算法的支持前瞻执行的浮点部件的结构

遍数	指 令	流出	执行	访存	写 CDB	说明
1	L. D      F0,0(R1)	1	2	3	4	流出第一条指令
1	ADD. D    F4,F0,F2	1	5		8	等待 L. D 的结果
1	S. D      F4,0(R1)	2	3	9		等待 ADD. D 的结果
1	DADDIU   R1,R1,#-8	2	4		5	等待 ALU(计算指令 S. D 的有效地址也是用该 ALU)
1	BNE      R1,R2,Loop	3	6			等待 DADDIU 的结果
2	L. D      F0,0(R1)	4	7	8	9	等待 BNE 完成
2	ADD. D    F4,F0,F2	4	10		13	等待 L. D 的结果
2	S. D      F4,0(R1)	5	8	14		等待 ADD. D 的结果
2	DADDIU   R1,R1,#-8	5	9		10	等待 ALU
2	BNE      R1,R2,Loop	6	11			等待 DADDIU 的结果
3	L. D      F0,0(R1)	7	12	13	14	等待 BNE 完成
3	ADD. D    F4,F0,F2	7	15		18	等待 L. D 的结果
3	S. D      F4,0(R1)	8	13	19		等待 ADD. D 的结果
3	DADDIU   R1,R1,#-8	8	14		15	等待 ALU
3	BNE      R1,R2,Loop	9	16			等待 DADDIU 的结果

图 4.13 基于 Tomasulo 算法的双流出超标量流水线中指令的流出、执行和写 CDB 时钟分析



(2)

12) 前 10 个周期执行 4 次, 后 4 个周期执行 1 次

$$4 \times 10 + 4 = 44 \Rightarrow \text{平均 } \frac{44}{4} = 11 \text{ 个时钟周期}$$

加速比:  $\frac{14}{11} = 1.27$

加速来源: 少执行了 3 次后 4 个周期

(3)

未经优化的:

```
L.D F2, 0(R1)
Stall
MUT.D F4, F2, F0
L.D F6, 0(R2)
Stall
Stall
ADD.D F6, F4, F6
Stall
Stall
S.D F6, 0[R2]
```

```
L.D F2, 8(R1)
Stall
MUT.D F4, F2, F0
L.D F6, 8(R2)
Stall
Stall
ADD.D F6, F4, F6
Stall
Stall
S.D F6, 8[R2]
```

```
L.D F2, 16(R1)
Stall
MUT.D F4, F2, F0
L.D F6, 16(R2)
Stall
Stall
ADD.D F6, F4, F6
Stall
Stall
S.D F6, 16[R2]
```

```
L.D F2, 24(R1)
Stall
MUT.D F4, F2, F0
L.D F6, 24(R2)
Stall
Stall
ADD.D F6, F4, F6
Stall
Stall
S.D F6, 24[R2]
```

```
DADDIU R1, R1, #32
DADDIU R2, R2, #32
DSUBIU R3, R1, #DONE
BNEZ R3, F00
```

优化后的:

//以下相互关联的指令之间至少间隔了4条指令, 不发生冲突

```
L.D F2, 0(R1)
L.D F8, 8(R1)
L.D F14, 16(R1)
L.D F20, 24(R1)//没有延迟
```

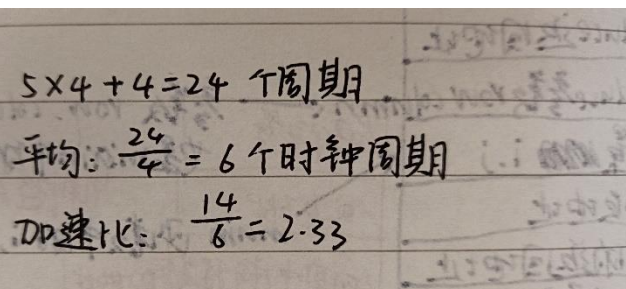
```
MUT.D F4, F8, F0
MUT.D F10, F10, F0
MUT.D F16, F12, F0
MUT.D F22, F14, F0//没有延迟, 因为使用的是来自不同寄存器的源操作数
```

```
L.D F6, 0[R2]
L.D F12, 8[R2]
L.D F18, 16[R2]
L.D F24, 24[R2]//没有延迟
```

```
ADD.D F6, F4, F6
ADD.D F12, F10, F12
ADD.D F18, F16, F18
ADD.D F24, F22, F24//没有延迟
```

```
S.D F6, 0[R2]
S.D F12, 8[R2]
S.D F18, 16[R2]
S.D F24, 24[R2]//没有延迟
```

```
DADDIU R1, R1, #32
DADDIU R2, R2, #32
DSUBIU R3, R1, #DONE
BNEZ R3, F00
```



(4)

指令状态:

指令	执行时钟		
	发射	执行	写回结果
L.D F2, 0(R1)	1	2	3
MUL.D F4, F2, F0	2	4	5
L.D F6, 0(R2)	3	4	6
ADD.D F6, F4, F6	4	8	9
S.D F6, 0[R2]	5	11	12
DADDIU R1, R1, #8	6	7	8
DADDIU R2, R2, #8	7	8	9
DSUBIU R3, R1, #DONE	8	9	10
BNEZ R3, FOO	9	10	
L.D F2, 0(R1)	10	11	12
MUL.D F4, F2, F0	11	13	14
L.D F6, 0(R2)	12	13	15
ADD.D F6, F4, F6	13	17	18
S.D F6, 0[R2]	14	20	21
DADDIU R1, R1, #8	15	16	17
DADDIU R2, R2, #8	16	17	18
DSUBIU R3, R1, #DONE	17	18	19
BNEZ R3, FOO	18		

名称	保留站						
	Busy	Op	Vj	Vk	Qj	Qk	A
Load1							
Load2							
Store1							
Store2							
Add1	Y	ADD	Reg[F4]	Reg[F6]			
Add2							
Add3							
Mult1							
Mult2							

字 段	浮点寄存器状态							
	F0	F2	F4	F6	F8	F10	.....	F30
Qi				Add1				

(5)

访存/整数指令	浮点指令	
L.D F2, 0(R1)		1
L.D F8, 8(R1)		2
L.D F14, 16(R1)	MUT.D F4, F2, F0	3
L.D F20, 24(R1)	MUT.D F10, F8, F2	4
L.D F6, 0(R2)	MUT.D F14, F12, F0	5
L.D F12, 8(R2)	MUT.D F22, F20, F0	6
L.D F18, 16(R2)	ADD.D F6, F4, F0	7
L.D F24, 24(R2)	ADD.D F12, F10, F2	8
DADDIU R1, R1, #32	ADD.D F18, F16, F18	9
S.D F6, 0(R2)	ADD.D F24, F22, F2	10
S.D F12, 8(R2)		11
S.D F18, 16(R2)		12
S.D F24, 24(R2)		13
DADDIU R2, R2, #32		14
DSUBIU R3, R1, #DONE		15
BNEZ R3, F00		16

平均指令周期为:  $\frac{16}{4} = 4$

加速比:  $\frac{14}{4} = 3.5$

(6)

16)

访存1	访存2	浮点1	浮点2	整数
L.D F2, 0(R1)	L.D F8, 8(R1)			
L.D F14, 16(R1)	L.D F20, 24(R1)			
L.D F6, 0(R2)	L.D F12, 8(R2)	MUT.D F4, F2, F0	MUT.D F10, F8, F0	
L.D F18, 16(R2)	L.D F24, 24(R2)	MUT.D F14, F12, F0	MUT.D F22, F20, F0	
		ADD.D F6, F4, F0	ADD.D F12, F10, F2	
		ADD.D F18, F16, F18	ADD.D F24, F22, F2	DADDIU R1, R1, #32
				DADDIU R2, R2, #32
				DSUBIU R3, R1, #DONE
				BNEZ R3, F00

平均指令周期:  $\frac{11}{4} = 2.75$

加速比:  $\frac{14}{2.75} = 5.09$