



華東理工大學
EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY

《 计算机体系结构 》

实验报告二

班 级： 计 203

学 号： 20002462

姓 名： 刘子言

指导教师： 梁建宁

信息科学与工程学院

2022 年 11 月

实验名称 Tomasulo 算法

实验地点 信息楼 418 实验日期 2022.11.23

一、实验目的

- 1、加深对指令级并行性及其开发的理解；
- 2、加深对 Tomasulo 算法的理解；
- 3、掌握 Tomasulo 算法在指令流出、执行、写回各阶段对浮点指令和 Load/ Store 指令进行的操作；
- 4、掌握采用了 Tomasulo 算法的浮点处理部件的结构；
- 5、掌握保留站的结构；
- 6、给定被执行代码片段，对于具体某个时钟周期，能够写出保留站、指令状态表以及浮点寄存器状态表内容的变化情况。

二、实验设备

个人 PC 机，实验平台采用 Tomasulo 算法模拟器。

三、实验原理

1、指令集并行性及其开发

- (1) 由于指令可以并行执行，所以指令之间可能实现的重叠执行称为指令级并行（ILP）。
- (2) 基础块是一段顺序执行的代码，除了入口外没有其他的转入分支，除了出口外没有其他的转出分支。一个基础块一般包含 3-6 条指令，基础块类的指令一般互相依赖，挖掘并行性很受限制，所以一般需要在不同的基础块之间寻找指令并行性。
- (3) 指令级并行大体有两种不同的开发方法：一是依靠硬件来帮助动态发现和开发并行，二是依靠软件技术在编译时静态发现并行。
- (4) 挖掘指令级并行性的目的：最大化 IPC，或者最小化 CPI。

2、Tomasulo 算法基本步骤

(1) 发射

从 FIFO 指令缓冲栈中获取下一条指令，如果保留栈有空闲，将指令发射到保留栈中；如果操作数尚未获得，暂时让指令停顿；重命名寄存器，能够消除 WAR、WRW 冒险。

(2) 执行

当操作数可用时，将其保存到正等待该数的保留栈中；当所有操作数都准备好了，执行指令（要避免 RAW 冒险）；为了防止存储器冒险，Load 和 Store 指令维持原程序中的顺序；为了维持异常行为跟原来一样，指令不允许开始执行，一直到前面所有的分支指令都完成。

(3) 写回

通过公共数据通道 CDB 将结果写到保留栈和寄存器堆；Store 指令保留在 Store 缓冲栈，需要一直等到存储值和存储地址可用为止，然后在有空闲存储器单元时，立即写入结果。

四、实验操作及运行结果

- 1、假设浮点功能部件的延迟时间为：加减法 2 个时钟周期，乘法 10 个时钟周期，除法 40

个时钟周期，Load 部件 2 个时钟周期。

① 对于下面的代码段，给出当指令 MUL.D 即将写回时，保留站、Load 缓冲器以及寄存器状态表中的内容。

```
L.D      F6, 24(R2)
L.D      F2, 12(R3)
MUL.D    F0, F2, F4
SUB.D    F8, F6, F2
DIV.D    F10, F0, F6
ADD.D    F6, F8, F2
```

在第 15 个时钟周期时，指令 MUL.D 即将写回（该指令在第 16 个时钟周期写结果），其保留站、Load 缓冲器以及寄存器状态表中的内容如下图所示：

指令	流出	执行	写结果
L.D F6, 24(R2)	1	2~3	4
L.D F2, 12(R3)	2	3~4	5
MUL.D F0, F2, F4	3	6~15	
SUB.D F8, F6, F2	4	6~7	8
DIV.D F10, F0, F6	5		
ADD.D F6, F8, F2	6	9~10	11

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	MULT.D	M2	R[F4]		
	Mult2	Yes	DIV.D		M1	Mult1	

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值		M2		M4	M3											

当前周期: 15

转移至

② 按步进方式执行上述代码，利用模拟器的“小三角按钮”的对比显示功能，观察每一个时钟周期前后各信息表中内容的变化情况。（针对步进过程中的一些比较重要的周期，在截图下方都附有文字解释）

第二步: 用右边的按钮, 控制指令的执行

指令	流出	执行	写结果
L.D F6, 24(R2)	1		
L.D F2, 12(R3)			
MUL.D F0, F2, F4			
SUB.D F8, F6, F2			
DIV.D F10, F0, F6			
ADD.D F6, F8, F2			

名称	Busy	地址	值
Load1	Yes	24	
Load2	No		
Load3	No		

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi				Load1												
值																

当前周期: 1

转移至

第二步：用右边的按钮，控制指令的执行

步进 退1步 前进5个周期 后退5个周期 执行到底 退出

指令状态

指令	流出	执行	写结果
L.D F6, 24(R2)	1	2	
L.D F2, 12(R3)	2		
MULT.D F0, F2, F4			
SUB.D F8, F6, F2			
DIV.D F10, F0, F6			
ADD.D F6, F8, F2			

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi		Load2		Load1												
值																

Load部件

名称	Busy	地址	值
Load1	Yes	R[R2]+24	
Load2	Yes	12	
Load3	No		

当前周期： 2

转移至 go

第二步：用右边的按钮，控制指令的执行

步进 退1步 前进5个周期 后退5个周期 执行到底 退出

指令状态

指令	流出	执行	写结果
L.D F6, 24(R2)	1	2~3	
L.D F2, 12(R3)	2	3	
MULT.D F0, F2, F4	3		
SUB.D F8, F6, F2			
DIV.D F10, F0, F6			
ADD.D F6, F8, F2			

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	MULT.D		R[F4]	Load2	
	Mult2	No					

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi		Mult1	Load2		Load1											
值																

Load部件

名称	Busy	地址	值
Load1	Yes	R[R2]+24	M[R[R2]+24]
Load2	Yes	R[R3]+12	
Load3	No		

当前周期： 3

转移至 go

第二步：用右边的按钮，控制指令的执行

步进 退1步 前进5个周期 后退5个周期 执行到底 退出

指令状态

指令	流出	执行	写结果
L.D F6, 24(R2)	1	2~3	4
L.D F2, 12(R3)	2	3~4	
MULT.D F0, F2, F4	3		
SUB.D F8, F6, F2	4		
DIV.D F10, F0, F6			
ADD.D F6, F8, F2			

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	Yes	SUB.D	M1			Load2
	Add2	No					
	Add3	No					
	Mult1	Yes	MULT.D		R[F4]	Load2	
	Mult2	No					

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi		Mult1	Load2		Load1	Add1										
值					M1											

Load部件

名称	Busy	地址	值
Load1	No		
Load2	Yes	R[R3]+12	M[R[R3]+12]
Load3	No		

当前周期： 4

转移至 go

前4个周期中，第1条load指令完成了流出-执行-写结果，第2条load指令完成了流出-执行过程，尚未写回。寄存器F6、F2、F0、F8重命名为Load1、Load2、Mult1、Add1。寄存器F6的值更新，存进寄存器状态表中（M1），并在保留站中暂存MULT和SUB的相关操作数的信息。由于第2条Load指令与MULT指令和SUB指令之间存在数据相关（寄存器F2），所以第4周期开始时MULT指令还没有进入执行阶段。

SUB 指令与 ADD 指令之间存在数据相关（寄存器 F8），所以第 7 周期开始时 ADD 指令还没有进入执行阶段。

第二步：用右边的按钮，控制指令的执行

步进 退1步 前进5个周期 后退5个周期 执行到底 退出

指令状态

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT.D F0, F2, F4	3	6~	
SUB.D F8, F6, F2	4	6~7	8
DIV.D F10, F0, F6	5		
ADD.D F6, F8, F2	6		

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	Yes	ADD.D	M3	M2		
	Add3	No					
7	Mult1	Yes	MULT.D	M2	R[F4]		
	Mult2	Yes	DIV.D		M1	Mult1	

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值		M2		M1	M3											

当前周期： 8

转移至 go

到第 8 个周期，SUB 指令完成写结果。寄存器 F8 的值更新，存进寄存器状态表中 (M3)；ADD 指令可以正常往下执行，并将对应操作数的值 M3 存进保留站中。

第二步：用右边的按钮，控制指令的执行

步进 退1步 前进5个周期 后退5个周期 执行到底 退出

指令状态

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT.D F0, F2, F4	3	6~	
SUB.D F8, F6, F2	4	6~7	8
DIV.D F10, F0, F6	5		
ADD.D F6, F8, F2	6	9~	

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
1	Add2	Yes	ADD.D	M3	M2		
	Add3	No					
6	Mult1	Yes	MULT.D	M2	R[F4]		
	Mult2	Yes	DIV.D		M1	Mult1	

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值		M2		M1	M3											

当前周期： 9

转移至 go

第二步：用右边的按钮，控制指令的执行

步进 退1步 前进5个周期 后退5个周期 执行到底 退出

指令状态

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT.D F0, F2, F4	3	6~	
SUB.D F8, F6, F2	4	6~7	8
DIV.D F10, F0, F6	5		
ADD.D F6, F8, F2	6	9~10	

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	Yes	ADD.D	M3	M2		
	Add3	No					
5	Mult1	Yes	MULT.D	M2	R[F4]		
	Mult2	Yes	DIV.D		M1	Mult1	

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值		M2		M1	M3											

当前周期： 10

转移至 go

第二步：用右边的按钮，
控制指令的执行

步进

退1步

前进5个周期

后退5个周期

执行到底

退出

指令状态

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT D F0, F2, F4	3	6~	
SUB D F8, F6, F2	4	6~7	8
DIV D F10, F0, F6	5		
ADD D F6, F8, F2	6	9~10	11

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

当前周期： 14

转移至

go

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
1	Mult1	Yes	MULT D	M2	R[F4]		
	Mult2	Yes	DIV D		M1	Mult1	

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值		M2		M4	M3											

第二步：用右边的按钮，
控制指令的执行

步进

退1步

前进5个周期

后退5个周期

执行到底

退出

指令状态

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT D F0, F2, F4	3	6~15	
SUB D F8, F6, F2	4	6~7	8
DIV D F10, F0, F6	5		
ADD D F6, F8, F2	6	9~10	11

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

当前周期： 15

转移至

go

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	MULT D	M2	R[F4]		
	Mult2	Yes	DIV D		M1	Mult1	

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值		M2		M4	M3											

第二步：用右边的按钮，
控制指令的执行

步进

退1步

前进5个周期

后退5个周期

执行到底

退出

指令状态

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT D F0, F2, F4	3	6~15	16
SUB D F8, F6, F2	4	6~7	8
DIV D F10, F0, F6	5		
ADD D F6, F8, F2	6	9~10	11

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

当前周期： 16

转移至

go

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	Yes	DIV D	M5	M1		

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值	M5	M2		M4	M3											

从第 6-15 个周期，MULT 指令始终处于执行阶段（乘法需要 10 个周期延迟），到第 16 个周期，MULT 指令完成写结果。保留站中清除 MULT 指令相关记录，Mult1 的状态 Busy 还原为 No。寄存器 F0 的值更新，存进寄存器状态表中（M5）；DIV 指令可以正常开始执行，并将对应操作数的值 M5 存进保留站中。

第二步：用右边的按钮，控制指令的执行

步进 退1步 前进5个周期 后退5个周期 执行到底 退出

指令状态

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT D F0, F2, F4	3	6~15	16
SUB D F8, F6, F2	4	6~7	8
DIV D F10, F0, F6	5	17	
ADD D F6, F8, F2	6	9~10	11

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
39	Mult2	Yes	DIV D	M5	M1		

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值	M5	M2		M4	M3											

当前周期： 17

转移至 go

此处省略从第 18-55 个周期的截图，此阶段（第 17-56 周期）为 DIV 指令的执行阶段（除法器需要延迟 40 个周期），其他指令、保留站、寄存器状态表、Load 部件表等没有变化。

第二步：用右边的按钮，控制指令的执行

步进 退1步 前进5个周期 后退5个周期 执行到底 退出

指令状态

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT D F0, F2, F4	3	6~15	16
SUB D F8, F6, F2	4	6~7	8
DIV D F10, F0, F6	5	17~56	
ADD D F6, F8, F2	6	9~10	11

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	Yes	DIV D	M5	M1		

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值	M5	M2		M4	M3											

当前周期： 56

转移至 go

第二步：用右边的按钮，控制指令的执行

步进 退1步 前进5个周期 后退5个周期 执行到底 退出

指令状态

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT D F0, F2, F4	3	6~15	16
SUB D F8, F6, F2	4	6~7	8
DIV D F10, F0, F6	5	17~56	57
ADD D F6, F8, F2	6	9~10	11

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值	M5	M2		M4	M3	M6										

当前周期： 57

转移至 go

到第 57 个周期，DIV 指令完成写结果。保留站中清除 DIV 指令相关记录，Mult2 的状态 Busy 还原为 No。寄存器 F10 的值更新，存进寄存器状态表中（M6）。至此，该代码段的所有指令全部执行完毕。

2、对于与上面相同的延迟时间和代码段:

① 给出在第 3 个时钟周期时保留站的内容。

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	MULT.D		R[F4]	Load2	
	Mult2	No					

② 步进 5 个时钟周期, 给出这时保留站、Load 缓冲器以及寄存器状态表中的内容。

在①的基础上再步进 5 个周期，此时为第 8 个时钟周期，保留站、Load 缓冲器以及寄存器状态表中的内容如下图：

指令

指令	流出	执行	写结果
L D F6, 24(R2)	1	2~3	4
L D F2, 12(R3)	2	3~4	5
MULT D F0, F2, F4	3	6~	
SUB D F8, F6, F2	4	6~7	8
DIV D F10, F0, F6	5		
ADD D F6, F8, F2	6		

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	Yes	ADD D	M3	M2		
	Add3	No					
7	Mult1	Yes	MULT D	M2	R[F4]		
	Mult2	Yes	DIV D		M1	Mult1	

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值		M2		M1	M3											

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

当前周期: 8

转移至 go

③ 再步进 10 个时钟周期，给出这时保留站、Load 缓冲器以及寄存器状态表中的内容。

在②的基础上再步进 10 个周期，此时为第 18 个时钟周期，保留站、Load 缓冲器以及寄存器状态表中的内容如下图：

[illegible]

3、假设浮点功能部件的延迟为：加减法 3 个时钟周期，乘法 8 个时钟周期，除法 40 个时钟周期。自己编写一段程序（要在实验报告中给出），重复上述步骤 2 的工作。

(1) 自己编写的程序如下：

```
L.D      F6, 24(R2)
L.D      F2, 12(R3)
MUL.D    F0, F2, F4
ADD.D    F8, F2, F6
DIV.D    F10, F0, F6
SUB.D    F6, F8, F2
```

(2) 重复上述步骤 2 中的工作：

先在 Tomasulo 算法模拟器中重新输入指令，然后根据要求修改各个浮点功能部件的延迟时间，随后开始逐步执行程序。

① 给出在第 3 个时钟周期时保留站的内容。

保留站							
Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	MULT.D		R[F4]	Load2	
	Mult2	No					

② 步进 5 个时钟周期，给出这时保留站、Load 缓冲器以及寄存器状态表中的内容。

在①的基础上再步进 5 个周期，此时为第 8 个时钟周期，保留站、Load 缓冲器以及寄存器状态表中的内容如下图：

指令	流出	执行	写结果
L.D F6, 24(R2)	1	2~3	4
L.D F2, 12(R3)	2	3~4	5
MULT.D F0, F2, F4	3	6~	
ADD.D F8, F2, F6	4	6~8	
DIV.D F10, F0, F6	5		
SUB.D F6, F8, F2	6		

Load部件			
名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

当前周期： 8

转移至

保留站							
Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	Yes	ADD.D	M2	M1		
	Add2	Yes	SUB.D		M2	Add1	
	Add3	No					
5	Mult1	Yes	MULT.D	M2	R[F4]		
	Mult2	Yes	DIV.D		M1	Mult1	

寄存器																
字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值	M2			M1												

③ 再步进 10 个时钟周期，给出这时保留站、Load 缓冲器以及寄存器状态表中的内容。

在②的基础上再步进 10 个周期，此时为第 18 个时钟周期，保留站、Load 缓冲器以及寄存器状态表中的内容如下图：

指令	流出	执行	写结果
L.D F6, 24(R2)	1	2~3	4
L.D F2, 12(R3)	2	3~4	5
MULT.D F0, F2, F4	3	6~13	14
ADD.D F8, F2, F6	4	6~8	9
DIV.D F10, F0, F6	5	15~	
SUB.D F6, F8, F2	6	10~12	13

保留站

Time	名称	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
36	Mult2	Yes	DIV.D	M5	M1		

寄存器

字段	F0	F2	F4	F6	F8	F10	F12	F14	F16	F18	F20	F22	F24	F26	F28	F30
Qi	Mult1	Load2		Add2	Add1	Mult2										
值	M5	M2		M4	M3											

Load部件

名称	Busy	地址	值
Load1	No		
Load2	No		
Load3	No		

当前周期: 18

转移至 GO

五、实验中出现的问题和解决方法

- 实验中关于 Tomasulo 算法模拟器的使用

问题：该实验中运用 Tomasulo 算法模拟器，主要是为了更清楚地学习掌握 Tomasulo 算法的执行流程，体会该算法在动态调度中的优越性；而如果在输入代码段后直接采用“执行到底”按钮，则只能得到执行结果，无法观察到程序执行的细节部分，不利于我们对算法流程的掌握。

解决方案：先尽量多使用 Tomasulo 算法模拟器上的“步进”按钮，进行步进形式执行，仔细观察每一个时钟周期前后各信息表中的内容变化情况，了解 Tomasulo 算法中各程序执行的详细过程；有需要的情况下再灵活使用“退 1 步”“前进 5 个周期”“后退 5 个周期”等按钮，辅助观察程序执行情况。