

Tomasulo 算法模拟器设计与分析实验

计 63

孙亚男

2016011285

一、实验要求

1.1 基本要求

1.1.1 能够正确接受任意 NEL 汇编语言编写的指令序列作为输入（1000 行以内，保证符合附录 中定义的文法）

1.1.2 能够正确输出每一条指令发射的时间周期，执行完成的时间周期，写回结果的时间周期。

1.1.3 能够正确输出各时间周期的寄存器数值。

1.1.4 能够正确输出各时间周期保留站状态、LoadBuffer 状态和寄存器结果状态。

1.2 扩展要求

1.2.1 设计美观的交互界面。

1.2.2 实现高效的模拟算法，能够支持更大规模的 NEL 指令序列。

1.2.3 调研不同的分支预测技术，实现或改进部分技术，设计自己的分支预测方案等。

1.2.4 丰富 NEL 语言，为它添加更多的指令支持，并能够模拟这些指令的执行。

二、设计思路

2.1 Tomasulo 算法模拟器以时钟周期作为循环依据，实现了所有的基础功能。

2.2 具体执行过程

在每个周期内先根据上一周期的各个结构的状态进行写回操作检查，若存在某个保留站或 LoadBuffer 执行完成，则检查该类型中是否有指令在保留站内等待功能部件空闲；若存在，则使得该指令开始执行。即若当前周期有指令完成，则立即使得下一条就绪指令开始执行（对应重要说明中第二点：在例子中，第 4 周期，第一条 LD 指令执行完后，立刻有一个 LoadUnit 变得空闲，第 5 周期应当直接使第三条 LD 进入此单元开始执行，而是不应当像之前那样额外再等待一个周期，也就是说在第 5 周期，LD,F1,0x2 写回的时候，LD,F3,0xFFFFFFFF 就应当可以开始执行了）。

对于需要写回的指令，写入指令的写回周期，更新所有等待该指令结果的保留站和寄存器的状态，并清空该保留站内的内容。由于写回周期得到的空闲保留站需要等到下一周期才可以给下一条发射的指令使用，故不更新该类型保留站数量，等到该周期最后再进行更新。

进行指令发射操作，在不考虑分支预测的情况下，若当前已发射一条 JUMP 指令，则不会再发射指令。若没有已发射的指令，则根据保留站的空闲状态决定是否发射下一条指令。

进行指令执行操作，检查所有保留站，若正在运行状态，则更新运行时间，若时间为 0，则写入指令的执行完成周期。根据功能部件的空闲状况和保留站的等待状况判断是否有就绪指令，若存在，则依据先进先出原则选择序号最小的指令开始执行。

最后更新保留站空闲数量，时钟周期加一，进入下一周期继续执行上述过程。

2.3 附加功能实现

2.3.1 交互界面

将要求中的各项数据依次显示在界面上，实现按周期显示寄存器、保留站和 LoadBuffer 状态，按指令序号查找其发射、执行和写回周期。此外，还对一些非法情况如对应序号指令不存在，时钟周期为负等，进行了检测和提示。若查询的时钟周期大于实际运行周期，则返回最后一个周期的结果。

为了界面的美观，我设计了三款皮肤可供切换，具体交互界面如下：



`Inst := OPREX','REGISTER','REGISTER ','' INTEGER`

`Inst := "NOP"`

`OPREX := "ADDI" | "SUBI" | "SHL" | "SHR" | "SAL" | "SAR"`

具体实现如下：

ADDI 和 SUBI：增加对这两条指令的识别，在发射指令时，直接将立即数的加数作为就绪数据准备，其他操作同 ADD 系列指令相同。

SHL、SHR、SAL 和 SAR：增加对这四条指令的识别，在发射指令时，直接将表示移位数的立即数作为就绪数据准备，在计算时使用 java 自带>>、>>>和<<操作符进行移位计算。

NOP：增加对该指令的识别，由于该指令是空指令，故设计该指令的发射、执行和写回周期均为发射时的时钟周期。

2.3.3 高效的模拟算法

为了优化模拟算法，提高计算效率，我建立了 HashSet 的数据结构，对于每一个保留站记录该保留站的所有被使用位置，这样当该保留站对应的指令进行写回操作时可以直接遍历集合中的所有元素并修改对应的保留站，而无需遍历整个保留站。

三、运行结果

3.1 test0.nel

该测例为助教给出详细输出的测例，依次输出每个周期的结果并与预期输出

对比，结果符合要求。下面给出其中几个周期的输出结果：

current instruction: 6

open run style

CLOCK: 6 INSTRU: 1

	Issue	Exec	Comp	Write	Result
LD, F2, 0x1	2	5	6		

F0	F1	F2	F3	F4	F5	F6	F7
Ars1	0x1	0x1	LB3	Mrs1			

F8	F9	F10	F11	F12	F13	F14	F15

F16	F17	F18	F19	F20	F21	F22	F23

F24	F25	F26	F27	F28	F29	F30	F31

isBusy	isRun	Time	Op	Vj	Vk	Qj	Qk	addr
add1	✓	✓	3	SUB	0x2	0x1		x
add2	✓	x	1	JUMP			Ars1	x
add3								x
add4								x
add5								x
add6								x
mul1	✓	x	4	DIV		LB3	Ars1	x
mul2								x
mul3								x
load1				x	x	x	x	x
load2				x	x	x	x	x
load3	✓	✓	1	x	x	x	x	0xffffffff

current instruction: 3

open run style

CLOCK: 13 INSTRU: 1

	Issue	Exec	Comp	Write	Result
LD, F2, 0x1	2	5	6		

F0	F1	F2	F3	F4	F5	F6	F7
0x1	0x1	0x1	0xffffffff	Mrs1			

F8	F9	F10	F11	F12	F13	F14	F15

F16	F17	F18	F19	F20	F21	F22	F23

F24	F25	F26	F27	F28	F29	F30	F31

isBusy	isRun	Time	Op	Vj	Vk	Qj	Qk	addr
add1	✓	✓	0	JUMP	0xffffffff			x
add2								x
add3								x
add4								x
add5								x
add6								x
mul1	✓	✓	1	DIV	0xffffffff	0x1		x
mul2								x
mul3								x
load1				x	x	x	x	x
load2				x	x	x	x	x
load3				x	x	x	x	x

current instruction: 6

open run style

CLOCK: 18 INSTRU: 1

	Issue	Exec	Comp	Write	Result
LD, F2, 0x1	2	5	6		

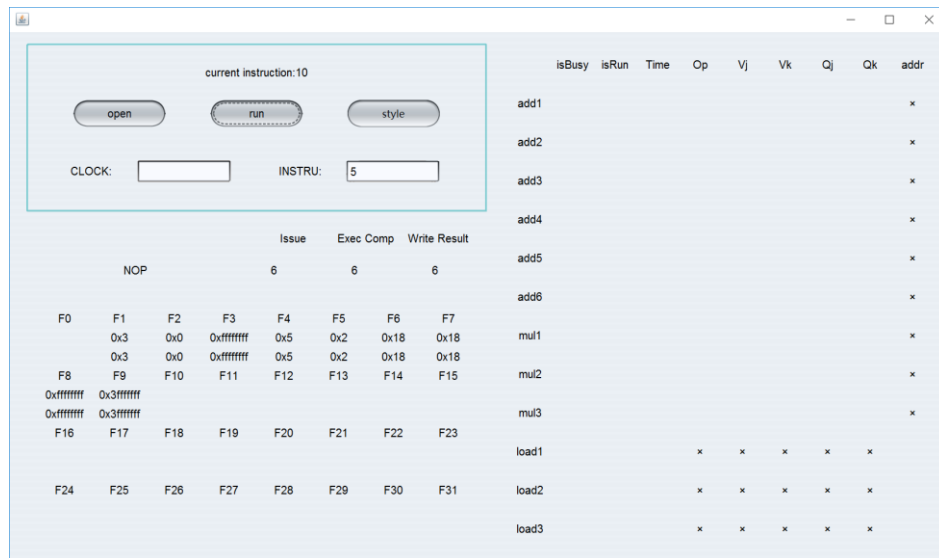
F0	F1	F2	F3	F4	F5	F6	F7
0x0	0x1	0x1	0xffffffff	Mrs1			

F8	F9	F10	F11	F12	F13	F14	F15

F16	F17	F18	F19	F20	F21	F22	F23

F24	F25	F26	F27	F28	F29	F30	F31

isBusy	isRun	Time	Op	Vj	Vk	Qj	Qk	addr
add1								x
add2	✓	✓	1	JUMP	0x0			x
add3								x
add4								x
add5								x
add6								x
mul1	✓	✓	1	DIV	0xffffffff	0x1		x
mul2								x
mul3								x
load1				x	x	x	x	x
load2				x	x	x	x	x
load3				x	x	x	x	x



F6 与 F7 为 0x3 逻辑左移和算术左移 0x3 位, 即 $3 \times 8 = 24 = 0x18$;F8 为 0xffffffff 算术右移 0x2, 为 0xffffffff ; F9 为 0xffffffff 逻辑右移 0x2, 为 0x3ffffff ; NOP 的发射、执行和写回周期均为 6。符合对于汇编指令结果的预期。

3.5 test4_1simple.nel test4_2huge.nel test4_3hugeNop.nel

这三组测例为自己设计的测例, 为测试大规模指令下模拟器是否可以正常运行, 其中 1 号测例为一段简单的汇编代码, 2 号测例为 1 号测例重复 90 次, 3 号测例为 1 号测例加入一定数量的 NOP 指令至保留站和 LoadBuffer 清空后的汇编代码重复 58 次。

预期 3 号测例的运行总周期应为 1 号测例的 58 倍, 2 号测例的平均每段运行周期应小幅度小于 3 号测例的平均每段运行周期。具体运行结果如下:

current instruction:11

open

run

style

CLOCK:

19

INSTRU:

								isBusy	isRun	Time	Op	Vj	Vk	Qj	Qk	addr	
NOP																	
F0	F1	F2	F3	F4	F5	F6	F7										
	0xb	0xffff3c6f	0xffffffffa	0x2098	0xc3b5												x
	0xb	0xffff3c6f	0xffffffffa	0x2098	0xc3b5												x
F8	F9	F10	F11	F12	F13	F14	F15										x
																	x
F16	F17	F18	F19	F20	F21	F22	F23										
		0x1								x	x	x	x	x			
		0x1															
F24	F25	F26	F27	F28	F29	F30	F31			x	x	x	x	x			
										x	x	x	x	x			

add1

add2

add3

add4

add5

add6

mul1

mul2

mul3

load1

load2

load3

current instruction:990

open

run

style

CLOCK:

1176

INSTRU:

								isBusy	isRun	Time	Op	Vj	Vk	Qj	Qk	addr	
NOP																	
F0	F1	F2	F3	F4	F5	F6	F7										
	0xfffffb2	0xffff3c6f	0xffffffffa	0x2098	0xc3b5												x
	0xfffffb2	0xffff3c6f	0xffffffffa	0x2098	0xc3b5												x
F8	F9	F10	F11	F12	F13	F14	F15										x
																	x
F16	F17	F18	F19	F20	F21	F22	F23										
		0x1								x	x	x	x	x			
		0x1															
F24	F25	F26	F27	F28	F29	F30	F31			x	x	x	x	x			
										x	x	x	x	x			

add1

add2

add3

add4

add5

add6

mul1

mul2

mul3

load1

load2

load3

current instruction:986

open

run

style

CLOCK:

1102

INSTRU:

								isBusy	isRun	Time	Op	Vj	Vk	Qj	Qk	addr	
NOP																	
F0	F1	F2	F3	F4	F5	F6	F7										
	0xb	0xffff3c6f	0xffffffffa	0x2098	0xc3b5												x
	0xb	0xffff3c6f	0xffffffffa	0x2098	0xc3b5												x
F8	F9	F10	F11	F12	F13	F14	F15										x
																	x
F16	F17	F18	F19	F20	F21	F22	F23										
		0x1								x	x	x	x	x			
		0x1															
F24	F25	F26	F27	F28	F29	F30	F31			x	x	x	x	x			
										x	x	x	x	x			

add1

add2

add3

add4

add5

add6

mul1

mul2

mul3

load1

load2

load3

可以看出：

3 号测例的总运行周期为 1102，为 1 号测例的 58 倍。

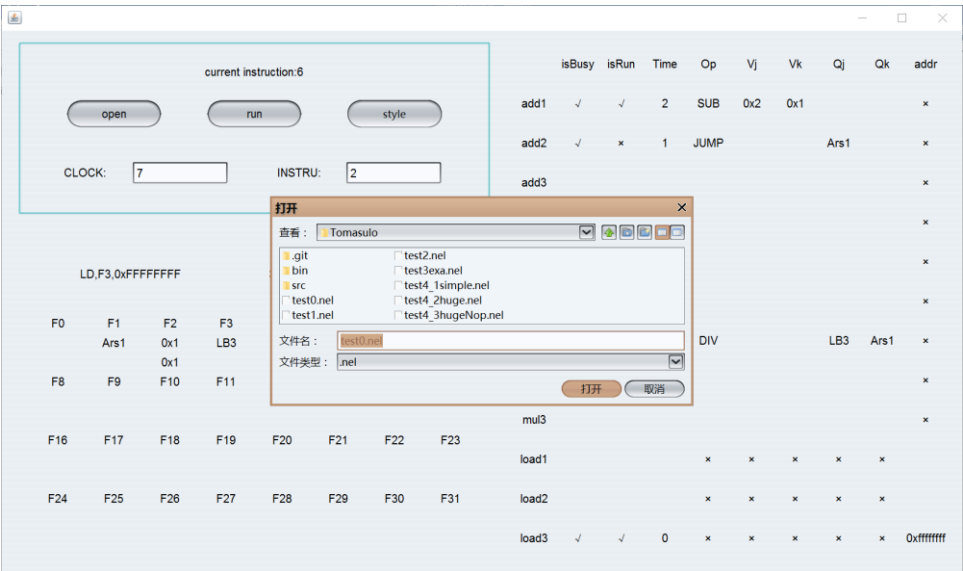
2 号测例的平均每段运行周期为 13.07，小于 3 号测例的平均每段运行周期 19。

四、代码运行方式

- 4.1 通过 eclipse 导入 java 项目；
- 4.2 将文件夹中的 substance.jar 皮肤美化包导入 java build path 中；
- 4.3 运行该项目。

五、界面使用方式

- 5.1 点击 open 按钮打开文件选择窗口， 并选择测例；



- 5.2 输入想要查看的周期，若没有输入，则默认为最后一个周期；

- 5.3 输入想要查看的指令，若没有输入，则不会显示指令相关信息；

5.4 点击 run 按钮开始运行，并查看结果。

六、实验总结

该实验旨在通过实际实现 Tomasolo 模拟器来深入理解 Tomasolo 算法的执行流程和原理，由于该算法细节较为复杂，实验要求开始时不够明确，导致我反复修改了五次代码，工作量有些大；但是完成实验的过程中，我对算法的理解更加深入，顺利完成实验后也非常有成就感。

第一年布置新的实验，过程确实十分艰苦，建议之后可以完善实验指导书中细节要求的部分，并优化测例。如：连续 Load 4 条指令，第四条指令应在第一条指令写回周期发射还是在写回周期的下一周期发射。

在实验完成的过程中遇到了一些具体实现细节的问题，在与助教沟通后得到解决，感谢助教提供的测例，也感谢老师的指导！