

1. 目标

- 1.1 熟悉指令的动态调度方法；
- 1.2 熟悉 Tomasulo 算法；熟悉基于 Tomasulo 算法的前瞻执行算法。

2. 任务

- 2.1 分别使用 Tomasulo 算法、基于 Tomasulo 算法的前瞻执行算法模拟执行下述指令。请写出

#	code	
12	LD F6, 34(R2)	
2	LD F2, 45(R3)	
3	MUL.D F0, F2, F4	
4	SUB.D F8, F6, F2	
5	DIV.D F10, F0, F6	
6	ADD.D F6, F8, F2	
7	SUB.D F10, F6, F2	
8	ST F10, 24(R2)	

- 1) MUL.D 指令将要写结果时；2) DIV.D 指令将要写结果时的系统状态，包括指令状态表、功能部件状态表、以及结果寄存器状态表。

其中部件延迟与数量如下表

#	部件	延迟	数量
1	Load	2	3
2	add	2	3
3	Multi/Divide	10/40	3
4	Store	2	3
5	Branch	2	1

- 2.2 编写 Tomasulo 算法的演示程序，包括如下功能：

代码中包含分支指令，再执行分支指令时，假定使用流水线暂停技术：分支指令没有执行完毕，不允许下一条指令流出。

可以参考 github 上的代码 <https://github.com/dyfios/Tomasulos-Dynamic-Scheduling-Implementation/tree/master/src>，需要增加或者完善如下功能（编程语言不限）：

- 1) 读取以文件形式提供的指令序列，例如 program.txt，内容如下：

Loop:

LD F6, 34(R2)

LD F2, 45(R3)

```

SD      F2, 3, R3
LD      F25, 3(R0)
MULD    F0, F2, F4
SUBD    F8, F0, F6
DIVD    F10,F0, F6
ADDD    F6, F8, F2
BNE     Loop, F6, F10

```

- 2) 读取以文件形式提供的部件配置表，例如 parts.txt，内容如下：

```

#part  delay  number
Laod   2      3
Add    2      3
Multi  10,40  3
Store  2      3

```

- 3) 设置初始存储器的值

可以批量设置存储器的值，例如都设置为 10.

- 4) 设置寄存器的值

可以批量设置或单个设置寄存器的值，包括浮点寄存器和整数寄存器。

- 5) 界面上显示

指令队列、指令状态、保留站、寄存器状态

指令队列：已执行完毕的指令不显示。

Inst	Dest	J	K
ADDD	F6	F8	F2
BNE	-8	F6	F10

指令状态表：显示所有已流出的指令状态。

Inst	Dest	J	K	Issued	Exec'd	Wri...
LD	F6	34+	R2	1	4	7
LD	F2	45+	R3	2	5	-1
SD	F2	3	R3	3	6	-1
LD	F25	3+	0	4	7	-1
MULD	F0	F2	F4	5	-1	-1
SUBD	F8	F0	F6	6	-1	-1
DIVD	F10	F0	F6	7	-1	-1

保留站：显示所有 Load,Store,乘法部件，加法部件的保留站信息。

乘法器	存数					取数				
	Name	B...	Addr	Q	V	Name	B...	Addr		
	St...	T	3		3	Load1	F	-1		
	St...	F	-1			Load2	T	45		
加法器	St...	F	-1			Load3	T	3		
	Name	B...	Op	Vj	Vk	Qj	Qk	A		
	Mult1	T	MULD		5	L2		-1		
	Mult2	T	DIVD		35	M1		-1		
	Name	B...	Op	Vj	Vk	Qj	Qk	A		
	Add1	T	SUBD		35	M1		-1		
	Add2	F	UNINIT					-1		
	Add3	F	UNINIT					-1		

寄存器状态:

浮点寄存器	FU	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
	Mult1	2	Load2	4	5	6	35	8	Add1	10	Mult2	12	13	14	
整数寄存器	R0	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

以上界面的显示只作为例子，实际的数值应该以配置的部件数量和延迟为准。

6) 界面操作功能

重置(reset):恢复到初始状态;

单步(step): 每按一下, 时钟进1, 每条指令做相应的动作;

自动(auto): 自动执行, 直到程序运行完毕, 或者按下暂停(pause)键或停止(stop)键;

暂停(pause): 在自动执行时, 可以暂停自动执行; 再按一次, 则继续自动执行。

停止(stop): 停止程序的执行。

3. 提交要求

此次上机, 分为两次提交:

3.1 第1次提交 2.1 的内容

提交截止日期为 2018 年 5 月 18 日前。通过 Elearning 提交。

3.2 第2次提交为 2.2 的内容

提交截止日期为 2018 年 6 月 3 日。提交要求: 包括源代码、以及相应的文档说明, 并能够运行。通过 Elearning 提交。