

一、概念题

解释下列名词：

1.不精确异常：当执行指令 i 导致发生异常时，处理机的现场（状态）与严格按程序顺序执行时指令 i 的现场不同。

2.保留站：在采用 Tomasulo 算法的 MIPS 处理器浮点部件中，在运算部件的入口设置的用来保存一条已经流出并等待到本功能部件执行的指令（相关信息）。

3.超标量：一种多指令流出技术。它在每个时钟周期流出的指令条数不固定，依代码的具体情况而定，但有上限。

4.CDB：公共数据总线。

二、填空题

2.1 开发指令级并行的方法主要有两类：基于硬件的动态开发方法以及基于软件的静态开发方法。

2.2 对于正确地执行程序来说，必须保持的最关键的两个属性是：数据流和异常行为。

2.3 动态分支预测技术包括：采用分支历史表、采用分支目标缓冲和基于硬件的前瞻执行。

2.4 寄存器换名可以消除 WAR 冲突和 WAW 冲突

2.5 要扩充 Tomasulo 算法支持前瞻执行，需将 Tomasulo 算法中的“写结果”段分为写结果和指令确认两个段。

2.6 前瞻执行允许指令乱序执行，但要求程序顺序确认。

三、问答题

3.1 超标量处理机与 VLIW 处理机相比有哪些优点？

（1）超标量结构对程序员是透明的，处理机能自己检测下一条指令能否流出，不需要由编译器或专门的变换程序对程序中的指令进行重新排列；

（2）即使是没有经过编译器针对超标量结构进行调度优化的代码或是旧的编译器生成的代码也可以运行，当然运行的效果不会很好。

3.2 简述前瞻执行（speculation）的基本思想：

对分支指令的结果进行猜测，并假设这个猜测总是对的，然后按这个猜测结果继续取、流出和执行后续的指令。只是执行指令的结果不是写回到寄存器或存储器，而是写入一个称为再定序缓冲器 ROB（ReOrder Buffer）中。等到相应的指令得到“确认”（commit）（即确实是应该执行的）之后，才将结果写入寄存器或存储器。

四、应用题

4.1 对于下述指令序列：（注：于书中题目略有改动，第四条指令 F2，F6 位置互换）

L.D F6, 34(R2)
L.D F2, 45(R3)
MUL.D F0, F2, F4
SUB.D F8, **F6, F2**
DIV.D F10, F0, F6
ADD.D F6, F8, F2

(1)给出当第一条指令完成并写入结果时，Tomasulo 算法所用的各种信息表中的内容。

(2)假设各种操作的延迟为：

Load 1 个时钟周期
加法 2 个时钟周期
乘法 10 个时钟周期
除法 40 个时钟周期

给出 MUL.D 指令准备写结果时各状态表的内容。

（1）下面给出了当采用 Tomasulo 算法时，在上述给定的时刻，保留站、load 缓冲器以及寄存器状态表中的内容。标志 Add1 表示是第一个加法功能部件，Mult1 表示是第一个乘法功能部件，其余以此类推。

指令	指令执行状态		
	流出	执行	写结果
L.D F6, 34(R2)	√	√	√
L.D F2, 45(R3)	√	√	
MUL.D F0, F2, F4	√		
SUB.D F8, F6, F2	√		
DIV.D F10, F0, F6	√		
ADD.D F6, F8, F2	√		

名称	保留站						
	Busy	Op	Vj	Vk	Qj	Qk	A
Load1	no						
Load2	yes	L.D					45+Regs[R3]
Add1	yes	SUB.D	Mem[34+Regs[R2]]			Load2	

Add2	yes	ADD.D			Add1	Load2	
Add3	no						
Mult1	yes	MUL.D		Regs[F4]	Load2		
Mult2	yes	DIV.D		Mem[34+Regs[R2]]	Mult1		

域	寄存器状态							
	F0	F2	F4	F6	F8	F10	...	F30
Qi	Mult1	Load2		Add2	Add1	Mult2		

(2) MUL.D 指令准备写入结果时各状态表的内容如下所示。

这里，由于 ADD.D 指令与 DIV.D 指令的 WAR 冲突已经消除，ADD.D 可以先于 DIV.D 完成并写入 F6，不会出现错误。

指令		指令执行状态		
		流出	执行	写结果
L.D	F6, 34(R2)	√	√	√
L.D	F2, 45(R3)	√	√	√
MUL.D	F0, F2, F4	√	√	
SUB.D	F8, F6 , F2	√	√	√
DIV.D	F10, F0, F6	√		
ADD.D	F6, F8, F2	√	√	√

名称	保留站						
	Busy	Op	Vj	Vk	Qj	Qk	A
Load1	no						
Load2	no						
Add1	no						
Add2	no						
Add3	no						
Mult1	yes	MUL.D	Mem[45+Regs[R3]]	Regs[F4]			
Mult2	yes	DIV.D		Mem[34+Regs[R2]]	Mult1		

域	寄存器状态							
	F0	F2	F4	F6	F8	F10	...	F30
Qi	Mult1					Mult2		