计算机体系结构

周学海
xhzhou@ustc.edu.cn
0551-63492149
中国科学技术大学

Review

- **硬件方法挖掘ILP**
  - 编译阶段无法确定的相关性，在程序执行时，用硬件方法判定
  - 可以使得程序代码在其他机器上有效地执行
- **记分牌的主要思想：允许stall后的指令继续**
  - 乱序执行(out-of-order execution) => 乱序完成(out-of-order completion)
  - 发射前检测结构相关和WAW相关
  - 读操作数前检测RAW相关
  - 写结果前处理WAR相关

CDC 6600 Scoreboard

CDC 6600 scoreboard的主要缺陷:

- **功能部件数较少，指令窗口较小**
- **没有定向数据通路**
- **仅局限于基本块内的动态指令流调度**
  - Branch类指令执行完成后，才能Issue下一条指令
- **结构冲突时不能发射**
- **WAR相关是通过等待解决的**
- **WAW相关时，不会进入Issue阶段**

第5章 指令级并行

5.1 指令级并行的基本概念及静态指令流调度

ILP及挑战性问题

软件方法挖掘指令集并行

基本块内的指令集并行

5.2硬件方法挖掘指令级并行

5.2-1 指令流动态调度方法之一：Scoreboard

5.2-2 指令流动态调度方法之二：Tomasulo

（教材3.4，3.5）

5.3 分支预测方法

5.4 基于硬件的推测执行

5.5 存储器访问冲突消解及多发射技术

5.6 多线程技术

5.2-2 指令流动态调度方法：Tomasulo

Tomasulo技术要点 → 算法运行示例 → Tomasulo循环展开示例

1、硬件结构
2、主要数据结构
3、流水线控制过程

动态调度方案之二：Tomasulo Algorithm

- **该算法首次在 IBM 360/91（1968/01）上使用（CDC6600,1964, Scoreboard）**
- **目标: 在没有专用编译器的情况下，提高系统性能**
- **IBM 360 & CDC 6600 ISA的差别**
  - IBM360只有 2位寄存器描述符 vs. CDC 6600寄存器描述符3位
  - IBM360　4个FP 寄存器 vs. CDC 6600 8个
  - IBM 360 有memory-register 操作
- **Alpha 21264, HP 8000, MIPS 10000, Pentium II, PowerPC 604，…**

Tomasulo Algorithm vs. Scoreboard

- **控制和缓存：分布在各部件中 vs. 集中在记分牌**
  - FU 缓存称“Reservation Stations”; 保存待用操作数
- **寄存器重命名：Tomasulo 有 vs. Scoreboard无**
  - 指令中的寄存器在RS中用寄存器值或指向RS的指针代替（称为register renaming）
  - 避免 WAR, WAW hazards
- **定向路径：Tomasulo 有 vs. Scoreboard无**
  - 传给FU的结果从RS来而不是从寄存器来
  - FU的计算结果通过Common Data Bus 以广播方式发向所有功能部件
- **控制相关处理：Tomasulo分支可跨越 vs. Scoreboard不可跨越**
  - 可以跨越分支，允许FP操作队列中FP操作不仅仅局限于基本块
- **Load和Store部件也看作带有RS的功能部件**

Tomasulo Organization

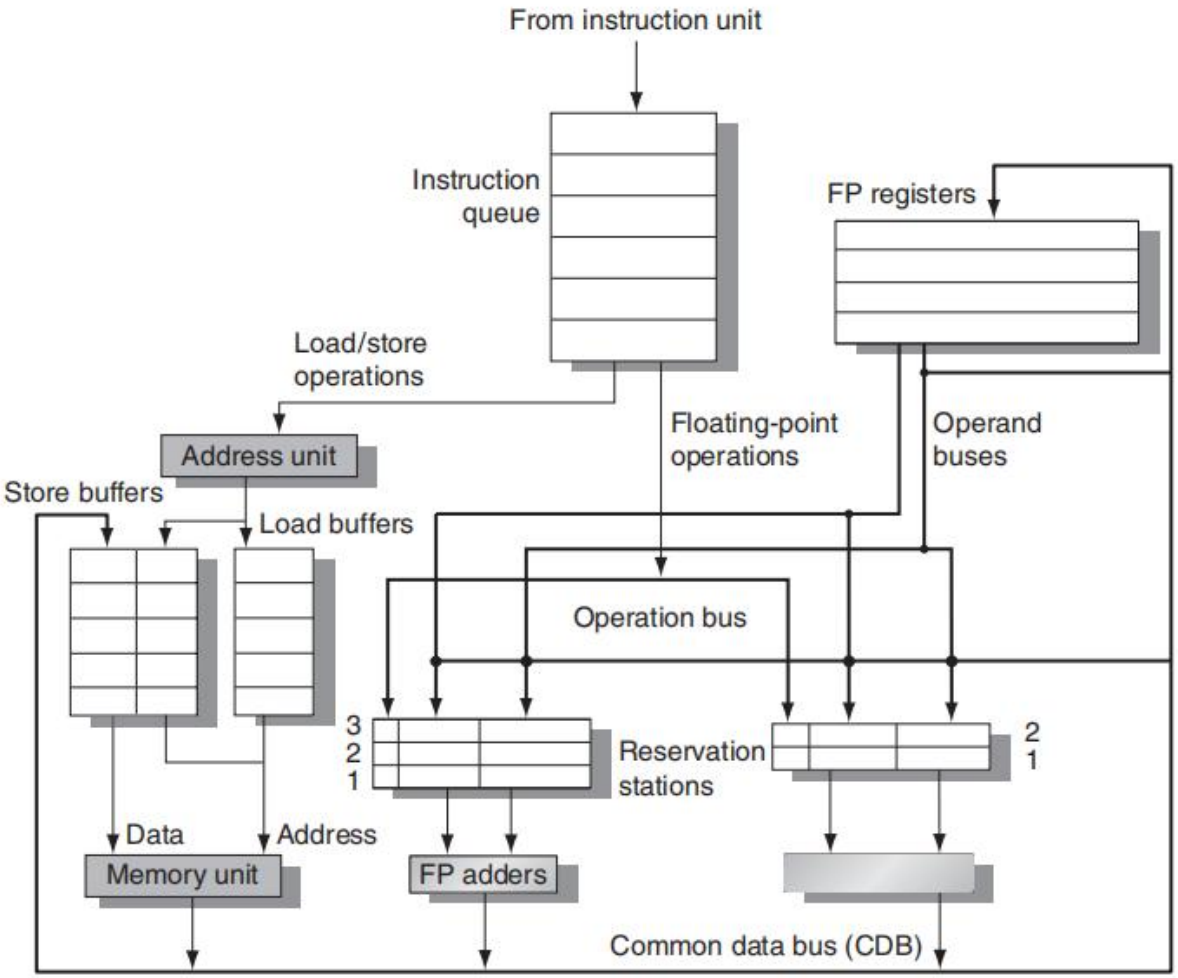


Figure 3.6 The basic structure of a MIPS floating-point unit using Tomasulo's algorithm.

Reservation Station 结构

Op: 部件所进行的操作

Vj, Vk: **源操作数的值**。Store 缓冲区有Vk域，用于存放要写入存储器的值

A: 存放存储器地址。开始存立即数，计算出有效地址后，存放有效地址

Qj, Qk: 产生源操作数的RS

注：没有记分牌中的准备就绪标志， Qj, Qk=0 => ready

Store 缓存区中Qk表示产生结果的RS

Busy: 标识RS或FU是否空闲

Register result status—如果存在对寄存器的写操作，指示对该寄存器进行写操作的部件.

Qi: **保留站的编号**

Tomasulo 算法的三阶段

- **1. Issue—从FP操作队列中取指令**
  - 如果RS空闲(no structural hazard), 则控制发射指令和操作数(renames registers). **消除WAR，WAW相关**
- **2. Execution—operate on operands (EX)**
  - 当两操作数就绪后，就可以执行
    如果没有准备好，则监测Common Data Bus 以获取结果。通过推迟指令执行**避免RAW相关**
- **3. Write result—finish execution (WB)**
  - 将结果通过Common Data Bus传给所有等待该结果的部件;
    标识RS可用
- **数据通信：功能部件产生结果的传送**
  - 通常的数据总线: data + destination （“go to” bus)
  - Common data bus: data + source（“come from” bus)
    - 64 bits 数据线 + 4 bits 功能部件源地址（ FU source address)
    - 产生结果的部件如果与RS中等待的部件匹配，就接收数据
    - 广播方式传送

Tomasulo 算法流水线控制

rs, rt : 源寄存器名；rd:目的寄存器名
RS: 保留站数据结构；r:保留站编号
RegisterStat：寄存器结果状态表
Reg： 寄存器组

1. Issue

FP Operation:

Wait until： Station r empty

Action or bookkeeping:

1st 操作数

if(RegisterStat[rs].Qi≠0) {RS[r].Qj ← RegisterStat[rs].Qi}

2nd 操作数

else {RS[r].Vj ←Reg[rs]; RS[r].Qj ←0 }

if(RegisterStat[rt].Qi≠0) {RS[r].Qk ← RegisterStat[rt].Qi}

else {RS[r].Vk ←Reg[rt]; RS[r].Qk←0 }

RS[r].Busy ← yes; RegisterStat[rd].Qi = r;

Load or Store:

Wait until: Buffer r empty

Action or bookkeeping:

基址寄存器

if(RegisterStat[rs].Qi≠0)

{RS[r].Qj ← RegisterStat[rs].Qi}

else {RS[r].Vj ←Reg[rs]; RS[r].Qj ←0 }

RS[r].A ←imm; RS[r].Busy ← yes;

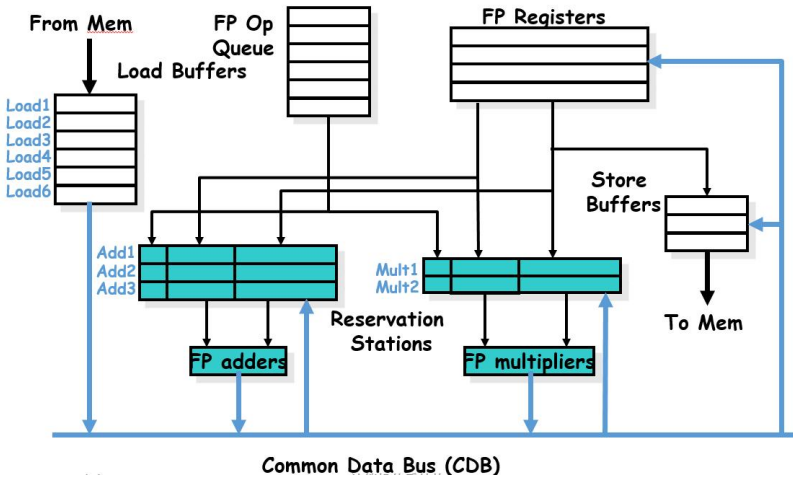
Load only: RegisterStat[rt].Qi = r;

Store only:

需写入的数据寄存器

if(RegisterStat[rt].Qi≠0) {RS[r].Qk ← RegisterStat[rt].Qi}

else {RS[r].Vk ←Reg[rt]; RS[r].Qk←0 }



注意：Load操作在EXE阶段分两步

2、Execute

FP Operation

wait until: (RS[r].Qj=0) and (RS[r].Qk=0)

Action or bookkeeping:

computer result: Operands are in Vj and Vk

Load-store step1

wait until: RS[r].Qj =0 & r is head of load-store queue

Action or bookkeeping:

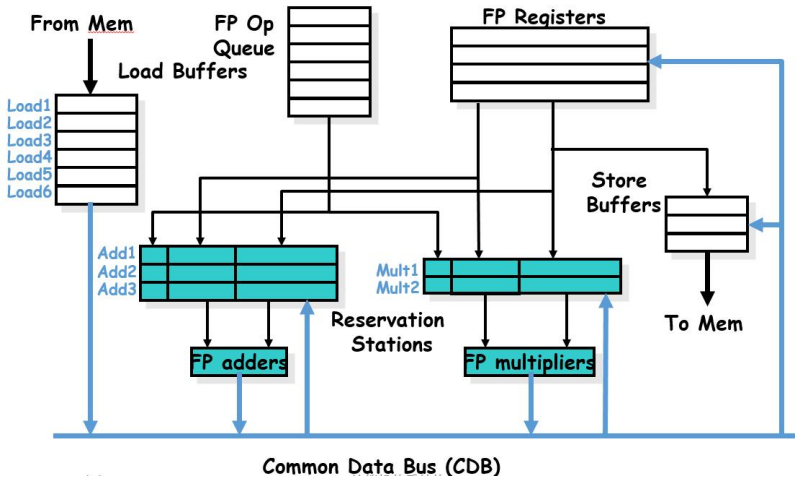
RS[r].A ← RS[r].Vj + RS[r].A;

Load step2

wait until: Load Step1 complete

Action or bookkeeping:

Read from Mem[RS[r].A]



3、Write result

FP Operation or Load

Wait until: Execution complete at r & CDB available

Action or bookkeeping

∀x (if (RegisterStat[x].Qi=r) {Regs[x] ← result; RegisterStat[x].Qi ← 0})

∀ x (if(RS[x].Qj =r) {RS[x].Vj ←result; RS[x].Qj ←0});

∀ x (if(RS[x].Qk =r) {RS[x].Vk ←result; RS[x].Qk ←0});

RS[r].Busy ← no;

Store

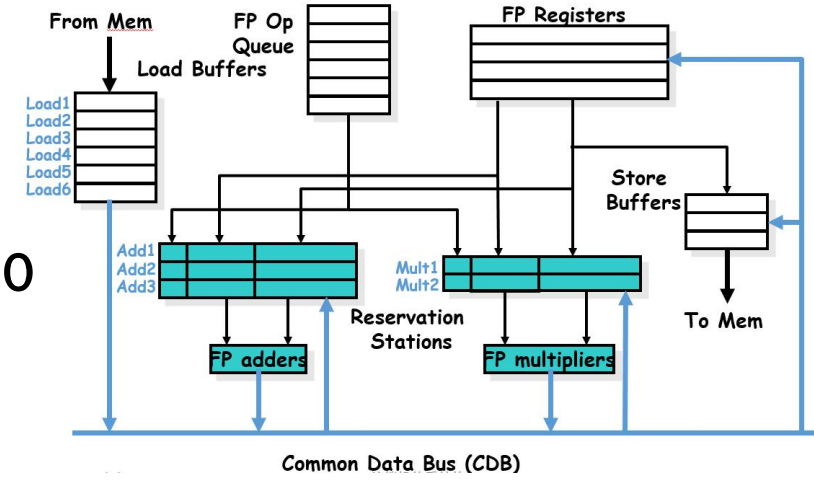
wait until:

Execution complete at r & RS[r].Qk = 0

Action or bookkeeping

Mem[RS[r].A] ← RS[r].Vk;

RS[r].Busy ← no;



Tomasulo 算法的特点

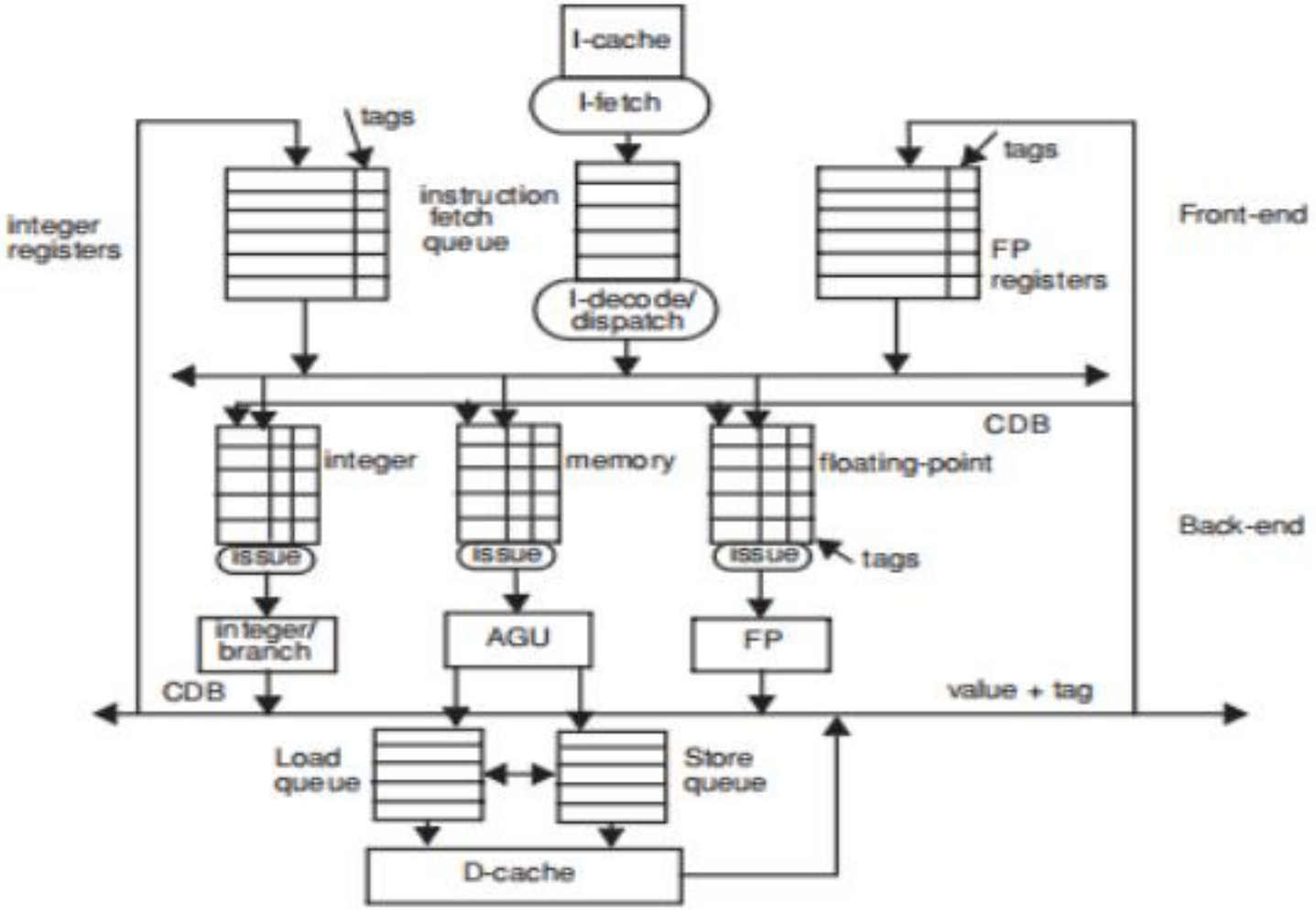
- **控制和缓存分布在各部件中**
  - FU 缓存称“reservation stations”; 保存待用操作数
- **指令中的寄存器在RS中用寄存器值或指向RS的指针代替（称为 register renaming）**
  - 避免 WAR, WAW hazards
- **传给FU的值从RS来而不是从寄存器来，FU的计算结果通过 Common Data Bus 以广播方式发向所有功能部件**
- **Load和Store部件也看作带有RS的功能部件**
- **可以跨越分支，允许FP操作队列中操作不仅仅局限于基本块**

5.2-2 指令流动态调度方法：Tomasulo

Tomasulo技术要点 → 算法运行示例 → Tomasulo循环展开示例

1、硬件结构
2、主要数据结构
3、流水线控制过程

Tomasulo Organization



说明：1、Tomasulo 算法中的ISSUE 对应图中的dispatch，该图中的issue指数据准备好了可送到执行部件执行。2、memory访问分为两部 （1）AGU 计算地址 （2）访存

Tomasulo Example

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2			
LD	F2	45+	R3			
MULTD	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

Register result status:

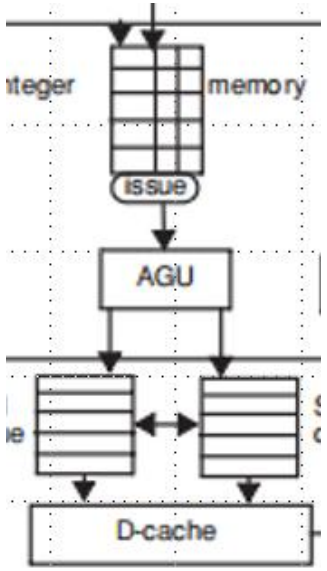
Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
0	*FU*									

Tomasulo Example Cycle 1

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result
LD	F6	34+	R2	1		
LD	F2	45+	R3			
MULTD	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	Yes	34+R2
Load2	No	
Load3	No	



Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
1	FU			Load1					

Tomasulo Example Cycle 2

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1		
LD	F2	45+	R3	2		
MULTD	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
2	*FU*		Load2		Load1					

Note: Unlike 6600, can have multiple loads outstanding

Tomasulo Example Cycle 3

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result
LD	F6	34+	R2	1	3	
LD	F2	45+	R3	2		
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	MULTD		R(F4)	Load2	
	Mult2	No					

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
3	FU	Mult1	Load2		Load1					

- Note: F4在保留站中被重命名；MULT issued vs. scoreboard
- Load1 准备写结果; 哪条指令正等待load1的结果?

Tomasulo Example Cycle 4

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	No	
Load2	Yes	45+R3
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	Yes	SUBD	M(A1)			Load2
	Add2	No					
	Add3	No					
	Mult1	Yes	MULTD		R(F4)	Load2	
	Mult2	No					

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
4	FU	Mult1	Load2		M(A1)	Add1				

- Load2 准备写结果；哪条指令等待其结果？ **Answer：Add1 和 Mult1**

Tomasulo Example Cycle 5

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2			

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
2	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	No					
	Add3	No					
10	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
5	*FU*	Mult1	M(A2)		M(A1)	Add1	Mult2			

Tomasulo Example Cycle 6

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
1	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	Yes	ADDD		M(A2)	Add1	
	Add3	No					
9	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
6	*FU*	Mult1	M(A2)		Add2	Add1	Mult2			

- Issue ADDD here vs. scoreboard?

Tomasulo Example Cycle 7

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4	7	
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
0	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	Yes	ADDD		M(A2)	Add1	
	Add3	No					
8	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
7	FU	Mult1	M(A2)		Add2	Add1	Mult2			

- **Add1 准备写结果; 哪条指令正等待该结果?**

Answer: Add2

Tomasulo Example Cycle 8

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
2	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
7	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
8	*FU*	Mult1	M(A2)		Add2	(M-M)	Mult2			

Tomasulo Example Cycle 9

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
1	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
6	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
9	*FU*	Mult1	M(A2)		Add2	(M-M)	Mult2			

Tomasulo Example Cycle 10

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	10	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
0	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
5	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
10	*FU*	Mult1	M(A2)		Add2	(M-M)	Mult2			

- **Add2 准备写结果; 哪条指令正在等待该结果?**

Tomasulo Example Cycle 11

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
4	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
11	*FU*	Mult1	M(A2)		(M-M+M	(M-M)	Mult2			

- ADDD写结果，scoreboard此时ADDD能写结果吗？
- All quick instructions complete in this cycle!

Tomasulo Example Cycle 12

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
3	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
12	*FU*	Mult1	M(A2)		(M-M+M	(M-M)	Mult2			

Tomasulo Example Cycle 13

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
2	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
13	*FU*	Mult1	M(A2)		(M-M+M	(M-M)	Mult2			

Tomasulo Example Cycle 14

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
1	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
14	*FU*	Mult1	M(A2)		(M-M+M	(M-M)	Mult2			

Tomasulo Example Cycle 15

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3	15	
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
15	*FU*	Mult1	M(A2)		(M-M+M	(M-M)	Mult2			

Tomasulo Example Cycle 16

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3	15	16
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
40	Mult2	Yes	DIVD	M*F4	M(A1)		

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
16	*FU*	M*F4	M(A2)		(M-M+M	(M-M)	Mult2			

Faster than light computation
(skip a couple of cycles)

Tomasulo Example Cycle 55

Instruction status:

Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3	15	16
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
1	Mult2	Yes	DIVD	M*F4	M(A1)		

Register result status:

Clock		*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
55	*FU*	M*F4	M(A2)		(M-M+M	(M-M)	Mult2			

Tomasulo Example Cycle 56

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3	15	16
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5	56	
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

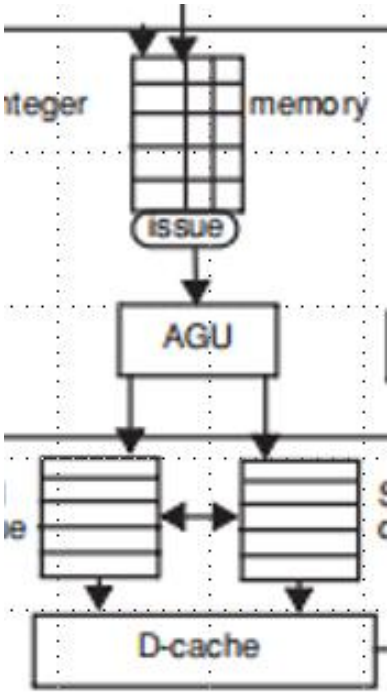
Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
0	Mult2	Yes	DIVD	M*F4	M(A1)		

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
56	FU	M*F4	M(A2)		(M-M+M	(M-M)	Mult2			

- Mult2 is completing; what is waiting for it?

No.	Inst.	i	j	k	Issue	Exec-start	Exec-End	Cache	WR(CDB)
1	LD	F6	34+	R2	1	2		3	4
2	LD	F2	45+	R3	2	3		4	5
3	MULTD	F0	F2	F4	3	6	15		16
4	SUBD	F8	F6	F2	4	6	7		8
5	DIVD	F10	F0	F6	5	17	56		57
6	ADDD	F6	F8	F12	6	9	10		11



- **TIPS：访存顺序约定不同，结果会不同**
- **本例中的访存约定：**
  - 所有访存指令1个计算地址队列（Memory队列），实际访存操作时分为load队列和store队列顺序计算访存地址
  - 顺序Load访存，顺序Store访存
  - Load访存可以跨越Store访存先行（Load的地址与Store地址不冲突时）
- **其他约定时，会怎样?**
  - 分别有LoadBuffer和StoreBuffer，但计算地址和实际访存buffer合并；顺序Load访存、顺序Store访存，Load访存可以跨越Store访存先行

Tomasulo Example Cycle 57

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3	15	16
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5	56	57
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
0	Mult2	Yes	DIVD	M*F4	M(A1)		

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
56	FU	M*F4	M(A2)		(M-M+M	(M-M)	Mult2			

- Once again: In-order issue, out-of-order execution and completion.

Compare to Scoreboard Cycle 62

Instruction status:

Instruction		j	k	Issue	Read Oper	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	62
ADDD	F6	F8	F2	13	14	16	22

Issue	Exec Comp	Write Result
1	3	4
2	4	5
3	15	16
4	7	8
5	56	57
6	10	11

- **为什么scoreboard/6600所需时间较长?**
  - 结构冲突
  - WAR,WAW冲突
  - 没有定向技术

Tomasulo v. Scoreboard (IBM 360/91 v. CDC 6600)

流水化的功能部件	多个功能部件
(6 load, 3 store, 3 +, 2 x/÷)	(1 load/store, 1 + , 2 x, 1 ÷,...)
指令窗口大小: 较大	较小
有结构冲突时不发射	有结构冲突时不发射
WAR: 用寄存器重命名避免	stall 来避免
WAW:用寄存器重命名避免	停止发射
从FU广播结果写寄存器方式	
Control: RS集中式scoreboard	

Tomasulo 算法的特点

- **控制和缓存分布在各部件中**
  - FU 缓存称“reservation stations”; 保存待用操作数
- **指令中的寄存器在RS中用寄存器值或指向RS的指针代替（称为 register renaming）**
  - 避免 WAR, WAW hazards
- **传给FU的结果从RS来而不是从寄存器来，FU的计算结果通过Common Data Bus 以广播方式发向所有功能部件**
- **Load和Store部件也看作带有RS的功能部件**
- **可以跨越分支，允许FP操作队列中FP操作不仅仅局限于基本块**

Tomasulo 缺陷

- **复杂**
  - delays of 360/91, MIPS 10000, IBM 620?
- **要求高速CDB**
  - 性能受限于Common Data Bus

教材：Ch. 3.4-3.5

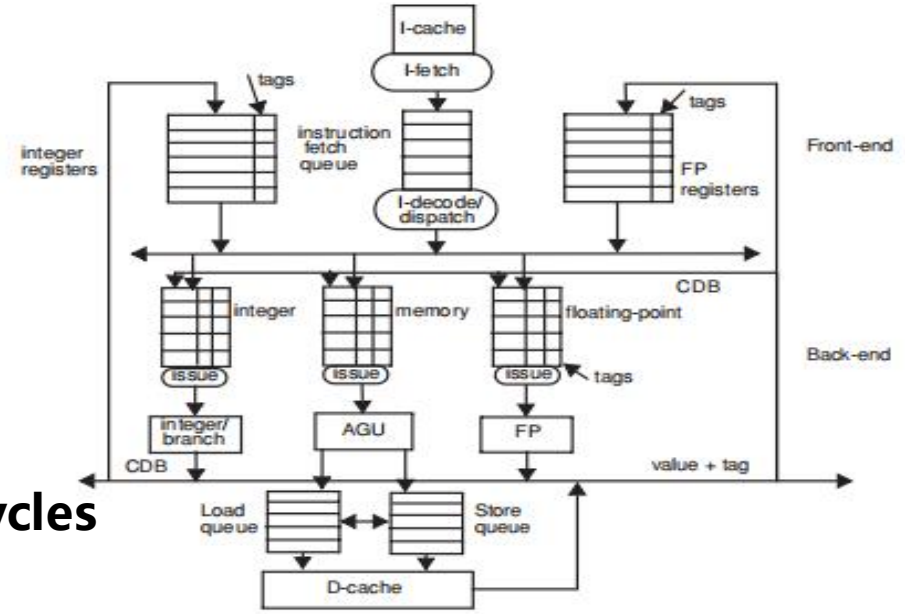
5.2-2 指令流动态调度方法：Tomasulo

Tomasulo技术要点 → 算法运行示例 → Tomasulo循环展开示例

1、硬件结构
2、主要数据结构
3、流水线控制过程

Tomasulo Loop Example

Loop:	LD	F0, 0(R1)
	MULTD	F4, F0, F2
	SD	F4, 0(R1)
	SUBI	R1,R1,#8
	BNEZ	R1 Loop



- **假设循环3次，设Multiply执行阶段4 cycles**
- **访存操作分为计算地址和访存两阶段**
  - **计算地址需1个cycle**
  - **第1次load时Cache未命中， 访存需7个cycles (cache miss), 第2次以后均命中，访存操作需1cycles**
- **访存顺序的约定：**
  - **所有访存指令1个计算地址队列，实际访存操作时分为load队列和store队列**
  - 计算访存地址按序，Load操作之间按序，Store操作按序
  - Load访存操作如果与store访存操作没有冲突，可以先行
- **为清楚起见，下面我们也列出SUBI, BNEZ的时钟周期**

Loop Example

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1				Load1	No		
	1	MULTD	F4	F0	F2				Load2	No		
	1	SD	F4	0	R1				Load3	No		
	2	LD	F0	0	R1				Store1	No		
	2	MULTD	F4	F0	F2				Store2	No		
	2	SD	F4	0	R1				Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	No						SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	0	80	FU									

Loop Example Cycle 1

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1			Load1	Yes	80	
	1	MULTD	F4	F0	F2				Load2	No		
	1	SD	F4	0	R1				Load3	No		
	2	LD	F0	0	R1				Store1	No		
	2	MULTD	F4	F0	F2				Store2	No		
	2	SD	F4	0	R1				Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	No						SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	1	80	FU	Load1								

Loop Example Cycle 2

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~		Load1	Yes	80+0	
	1	MULTD	F4	F0	F2	2			Load2	No		
	1	SD	F4	0	R1				Load3	No		
	2	LD	F0	0	R1				Store1	No		
	2	MULTD	F4	F0	F2				Store2	No		
	2	SD	F4	0	R1				Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	YES	Multd		R(F2)	Load1		SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	2	80	FU	Load1		Mult1						

Loop Example Cycle 3

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~		Load1	Yes	80	
	1	MULTD	F4	F0	F2	2			Load2	No		
	1	SD	F4	0	R1	3			Load3	No		
	2	LD	F0	0	R1				Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2				Store2	No		
	2	SD	F4	0	R1				Store3	No		

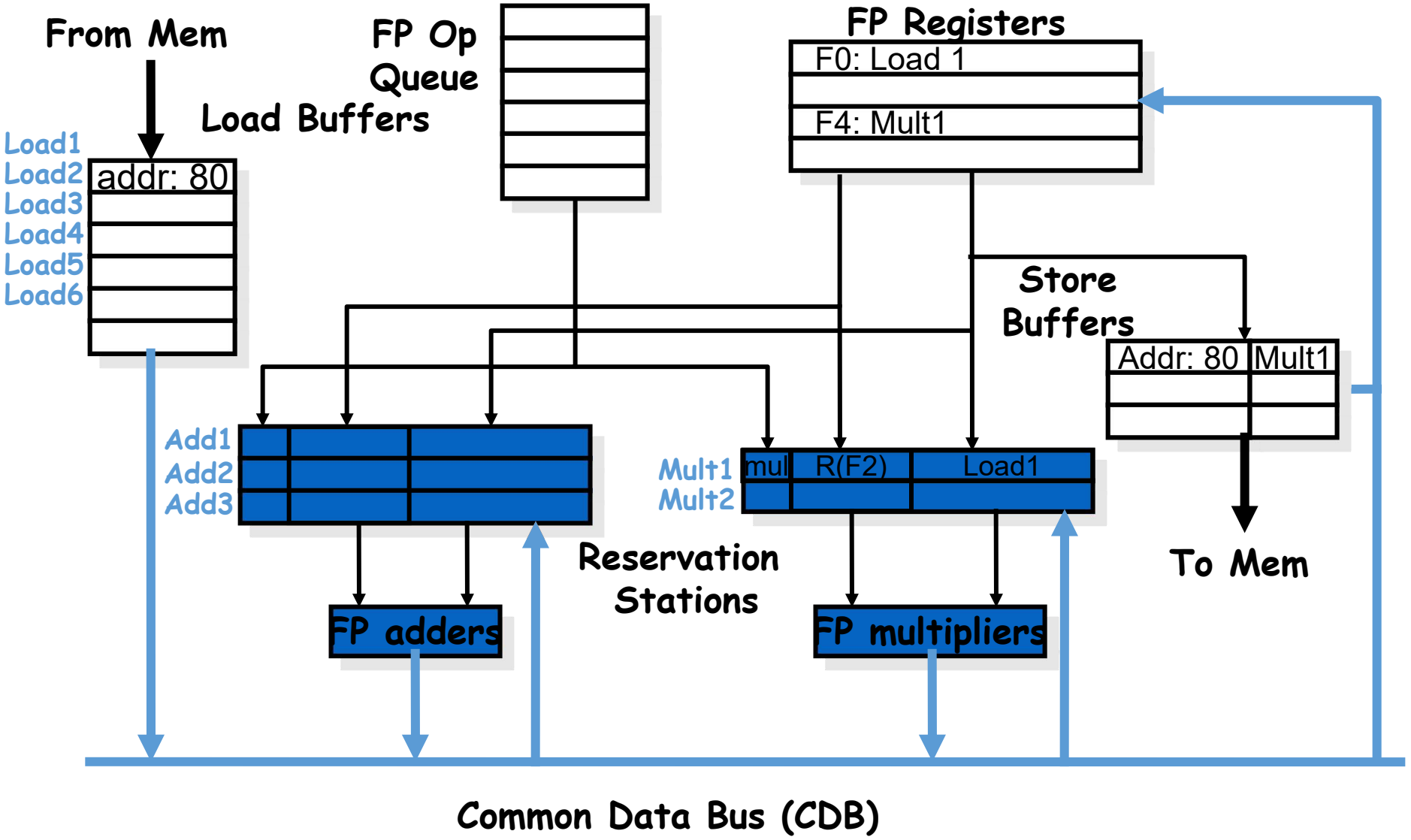
Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	YES	Multd		R(F2)	Load1		SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	3	80	FU	Load1		Mult1						

What does this mean physically?



Loop Example Cycle 4

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~		Load1	Yes	80	
	1	MULTD	F4	F0	F2	2			Load2	No		
	1	SD	F4	0	R1	3	4		Load3	No		
	2	LD	F0	0	R1				Store1	YES	80+0	Mult1
	2	MULTD	F4	F0	F2				Store2	No		
	2	SD	F4	0	R1				Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	YES	Multd		R(F2)	Load1		SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	4	80	FU	Load1		Mult1						

- Dispatching SUBI Instruction

Loop Example Cycle 5

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~		Load1	Yes	80	
	1	MULTD	F4	F0	F2	2			Load2	No		
	1	SD	F4	0	R1	3	4		Load3	No		
	2	LD	F0	0	R1				Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2				Store2	No		
	2	SD	F4	0	R1				Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	YES	Multd		R(F2)	Load1		SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	5	80	FU	Load1		Mult1						

- And, BNEZ instruction

Loop Example Cycle 6

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~		Load1	Yes	80	
	1	MULTD	F4	F0	F2	2			Load2	Yes	72	
	1	SD	F4	0	R1	3	4		Load3	No		
	2	LD	F0	0	R1	6			Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2				Store2	No		
	2	SD	F4	0	R1				Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	YES	Multd		R(F2)	Load1		SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	6	72	FU	Load2		Mult1						

- 注意：F0 不是从80地址处装载的值 R1=72

Loop Example Cycle 7

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~		Load1	Yes	80	
	1	MULTD	F4	F0	F2	2			Load2	Yes	72+0	
	1	SD	F4	0	R1	3	4		Load3	No		
	2	LD	F0	0	R1	6	7,		Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2	7			Store2	No		
	2	SD	F4	0	R1				Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	Yes	Multd		R(F2)	Load1		SUBI	R1	R1	#8
		Mult2	Yes	Multd		R(F2)	Load2		BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	7	72	FU	Load2		Mult2						

- 对寄存器文件的操作都是第2次循环的指令

Loop Example Cycle 8

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~		Load1	Yes	80	
	1	MULTD	F4	F0	F2	2			Load2	Yes	72	
	1	SD	F4	0	R1	3	4		Load3	No		
	2	LD	F0	0	R1	6	7,		Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2	7			Store2	Yes	72	Mult2
	2	SD	F4	0	R1	8			Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	Yes	Multd		R(F2)	Load1		SUBI	R1	R1	#8
		Mult2	Yes	Multd		R(F2)	Load2		BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	8	72	FU	Load2		Mult2						

- 第1次循环与第2次循环重叠执行

Loop Example Cycle 9

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9		Load1	Yes	80	
	1	MULTD	F4	F0	F2	2			Load2	Yes	72	
	1	SD	F4	0	R1	3	4		Load3	No		
	2	LD	F0	0	R1	6	7,		Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2	7			Store2	Yes	72+0	Mult2
	2	SD	F4	0	R1	8	9		Store3	No		

Reservation Station:

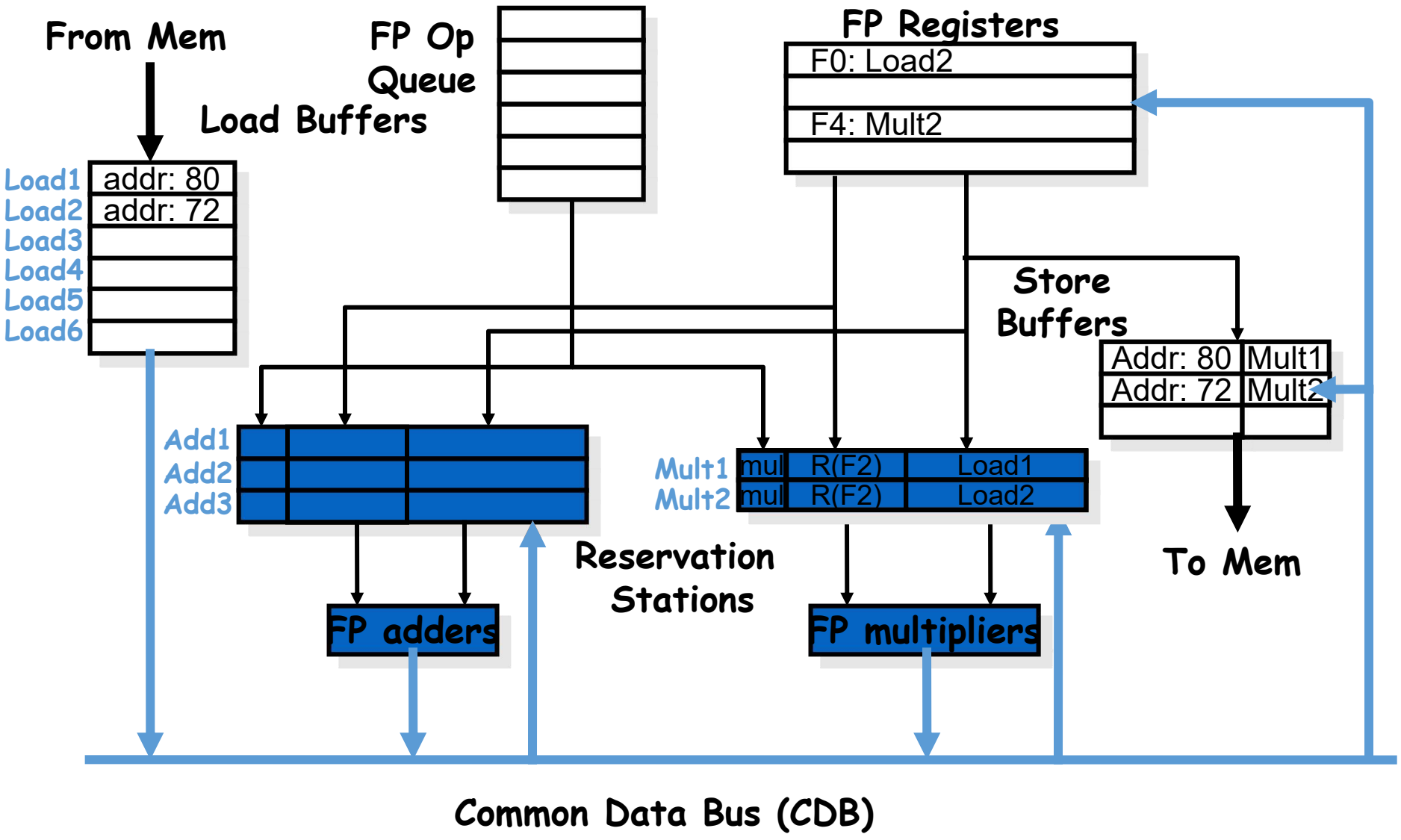
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	Yes	Multd		R(F2)	Load1		SUBI	R1	R1	#8
		Mult2	Yes	Multd		R(F2)	Load2		BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	9	72	FU	Load2		Mult2						

- Dispatching SUBI，Load1执行完毕

What does this mean physically?



Loop Example Cycle 10

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2			Load2	Yes	72	
	1	SD	F4	0	R1	3	4		Load3	No		
	2	LD	F0	0	R1	6	7,		Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2	7			Store2	Yes	72	Mult2
	2	SD	F4	0	R1	8	9		Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	4	Mult1	Yes	Multd	M[80]	R(F2)			SUBI	R1	R1	#8
		Mult2	Yes	Multd		R(F2)	Load2		BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	10	64	FU	Load2		Mult2						

- Dispatching BNEZ，Load1写结果

Loop Example Cycle 11

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~		Load2	Yes	72	
	1	SD	F4	0	R1	3	4		Load3	Yes	64	
	2	LD	F0	0	R1	6	7, 11		Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2	7			Store2	Yes	72	Mult2
	2	SD	F4	0	R1	8	9		Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	3	Mult1	Yes	Multd	M[80]	R(F2)			SUBI	R1	R1	#8
		Mult2	Yes	Multd		R(F2)	Load2		BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	11	64	FU	Load3		Mult2						

- Load3发射，F0 由第3次循环的Load装载地址为64单元的内容

Loop Example Cycle 12

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~		Load2	No		
	1	SD	F4	0	R1	3	4		Load3	Yes	64+0	
	2	LD	F0	0	R1	6	7, 11	12	Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2	7			Store2	Yes	72	Mult2
	2	SD	F4	0	R1	8	9		Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	2	Mult1	Yes	Multd	M[80]	R(F2)			SUBI	R1	R1	#8
	4	Mult2	Yes	Multd	M[72]	R(F2)			BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	12	64	FU	Load3		Mult2						

- Load2写结果(CDB)；Mult2就绪；Load3计算地址（AGU）

为什么不能发射Mult3?

Loop Example Cycle 13

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~		Load2	No		
	1	SD	F4	0	R1	3	4		Load3	Yes	64	
	2	LD	F0	0	R1	6	7, 11	12	Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2	7	13~		Store2	Yes	72	Mult2
	2	SD	F4	0	R1	8	9		Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	1	Mult1	Yes	Multd	M[80]	R(F2)			SUBI	R1	R1	#8
	3	Mult2	Yes	Multd	M[72]	R(F2)			BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	13	64	FU	Load3		Mult2						

Load3访存

Loop Example Cycle 14

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~14		Load2	No		
	1	SD	F4	0	R1	3	4		Load3	No		
	2	LD	F0	0	R1	6	7, 11	12	Store1	YES	80	Mult1
	2	MULTD	F4	F0	F2	7	13~		Store2	Yes	72	Mult2
	2	SD	F4	0	R1	8	9		Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	0	Mult1	Yes	Multd	M[80]	R(F2)			SUBI	R1	R1	#8
	2	Mult2	Yes	Multd	M[72]	R(F2)			BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	14	64	FU			Mult2						

Mult1执行完毕；Load3写结果；哪条指令等待其结果； Answer： Store1

Loop Example Cycle 15

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~14	15	Load2	No		
	1	SD	F4	0	R1	3	4, 15		Load3	No		
	2	LD	F0	0	R1	6	7, 11	12	Store1	Yes	80	[80]*F2
	2	MULTD	F4	F0	F2	7	13~		Store2	Yes	72	Mult2
	2	SD	F4	0	R1	8	9		Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	No						SUBI	R1	R1	#8
	1	Mult2	Yes	Multd	M[72]	R(F2)			BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	15	64	FU			Mult2						

- Mult1写结果

Loop Example Cycle 16

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~14	15	Load2	No		
	1	SD	F4	0	R1	3	4, 15	16	Load3	No		
	2	LD	F0	0	R1	6	7, 11	12	Store1	No		
	2	MULTD	F4	F0	F2	7	13~16		Store2	Yes	72	Mult2
	2	SD	F4	0	R1	8	9		Store3	No		

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	4	Mult1	Yes	Multd		R(F2)	R(F0)		SUBI	R1	R1	#8
	0	Mult2	Yes	Multd	M[72]	R(F2)			BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	16	64	FU			Mult3						

- Mult2执行完毕，SD1写结果，发射Mult3

Loop Example Cycle 17

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~14	15	Load2	No		
	1	SD	F4	0	R1	3	4, 15	16	Load3	No		
	2	LD	F0	0	R1	6	7, 11	12	Store1	No		
	2	MULTD	F4	F0	F2	7	13~16	17	Store2	Yes	72	[72]*R2
	2	SD	F4	0	R1	8	9		Store3	Yes	64	Mult3

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	3	Mult1	Yes	Multd		R(F2)	R(F0)		SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	17	64	FU			Mult3						

- Mult2写结果。可以发射SD3吗？

可以：SD3访问Memory队列，Mult2访问store buffer队列

Loop Example Cycle 18

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~14	15	Load2	No		
	1	SD	F4	0	R1	3	4, 15	16	Load3	Yes	64	
	2	LD	F0	0	R1	6	7, 11	12	Store1	No		
	2	MULTD	F4	F0	F2	7	13~16	17	Store2	Yes	72	[72]*F2
	2	SD	F4	0	R1	8	9	18	Store3	Yes	64+0	Mult3

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	2	Mult1	Yes	Multd		R(F2)	R(F0)		SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	18	64	FU			Mult2						

- SD2访存（store buffer），SD3计算有效地址（AGU,Memory队列）

Loop Example Cycle 19

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~14	15	Load2	No		
	1	SD	F4	0	R1	3	4, 15	16	Load3	No		
	2	LD	F0	0	R1	6	7, 11	12	Store1	No		
	2	MULTD	F4	F0	F2	7	13~16	17	Store2	No		
	2	SD	F4	0	R1	8	9	18	Store3	Yes	64	Mult3

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	1	Mult1	Yes	Multd		R(F2)	R(F0)		SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	19	64	FU			Mult2						

Loop Example Cycle 20

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~14	15	Load2	No		
	1	SD	F4	0	R1	3	4, 15	16	Load3	No		
	2	LD	F0	0	R1	6	11	12	Store1	No		
	2	MULTD	F4	F0	F2	7	13~16	17	Store2	No		
	2	SD	F4	0	R1	8	9	18	Store3	Yes	64	Mult3

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
	0	Mult1	Yes	Multd	M[64]	R(F2)			SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	20	64	FU	Load3		Mult2						

Mult3准备写结果；

Loop Example Cycle 21

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~14	15	Load2	No		
	1	SD	F4	0	R1	3	4, 15	16	Load3	No		
	2	LD	F0	0	R1	6	11	12	Store1	No		
	2	MULTD	F4	F0	F2	7	13~16	17	Store2	No		
	2	SD	F4	0	R1	8	9	18	Store3	Yes	64	[64]*F2

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	No						SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	21	64	FU									

- Mult3写结果，Mult3 17~20

Loop Example Cycle 22

Instruction Status

	ITER	Inst.	i	j	k	Issue	Exec	WR		Busy	Addr	Fu
	1	LD	F0	0	R1	1	2~9	10	Load1	No		
	1	MULTD	F4	F0	F2	2	11~14	15	Load2	No		
	1	SD	F4	0	R1	3	4, 15	16	Load3	No		
	2	LD	F0	0	R1	6	11	12	Store1	No		
	2	MULTD	F4	F0	F2	7	13~16	17	Store2	No		
	2	SD	F4	0	R1	8	9	18	Store3	Yes	64	[64]*F2

Reservation Station:

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	Code			
		Add1	No						LD	F0	0	R1
		Add2	No						MULTD	F4	F0	F2
		Add3	No						SD	F4	0	R1
		Mult1	No						SUBI	R1	R1	#8
		Mult2	No						BNEZ	R1	Loop	

Register Result Status

	Clock	R1		F0	F2	F4	F6	F8	F10	F12	.....	F30
	22	64	FU									

- SD3 访存；

Summary-Loop Example

ITER	Inst.	i	j	k	Issue	Exec-start	Exec-End	Cache	WR(CDB)
1	LD	F0	0	R1	1	2		3~9	10
1	MULTD	F4	F0	F2	2	11	14		15
1	SD	F4	0	R1	3	4	-	16	
1	SUB				4				
1	BNEZ				5				
2	LD	F0	0	R1	6	7		11	12
2	MULTD	F4	F0	F2	7	13	16		17
2	SD	F4	0	R1	8	9		18	
2	SUB				9				
2	BNEZ				10				
3	LD	F0	0	R1	11	12		13	14
3	MULTD	F4	F0	F2	16	17	20		21
3	SD	F4	0	R1	17	18		22	
3	....								

本例访存约定：

- 顺序计算访存地址
- 顺序Load访存
- 顺序Store访存
- Load访存可以跨越Store访存先行（Load的地址与Store地址不冲突时）

- **TIPS： 不同的存储器访问序的约定会产生不同结果。**
- **其他约定?**
  - 例如：分别有LoadBuffer和StoreBuffer，但计算地址和实际访存buffer合并；顺序Load访存、顺序Store访存，Load访存可以跨越Store访存先行
  - 例如：简单约定所有访存指令按序执行（计算地址队列和实际访存buffer合并，并且顺序访存）；

Summary

- **Tomasulo Algorithm 三阶段**
- **1. Issue—从FP操作队列中取指令**
  - 如果RS空闲(no structural hazard), 则控制发射指令和操作数 (renames registers).
- **2. Execution—operate on operands (EX)**
  - 当两操作数就绪后，就可以执行
    如果没有准备好，则监测Common Data Bus 以获取结果
- **3. Write result—finish execution (WB)**
  - 将结果通过Common Data Bus传给所有等待该结果的部件;
    表示RS可用
- **基本数据结构**
- **1. Instruction Status**
- **2. Reservation Station**
- **3. Register Result Status**
- **注意:**
  - CDB冲突、Loadbuffer和Storebuffer操作冲突

Summary

- **Reservations stations: 寄存器重命名，缓冲源操作数**
  - 避免寄存器成为瓶颈
  - 避免了Scoreboard中无法解决的 WAR, WAW hazards
  - 允许硬件做循环展开
- **不限于基本块(分支指令后的指令可以继续发射)**
- **贡献**
  - Dynamic scheduling
  - Register renaming
  - Load/store disambiguation
- **360/91 后 Pentium II; PowerPC 604; MIPS R10000; HP-PA 8000; Alpha 21264使用这种技术**

Summary：Tomasulo算法实现循环重叠执行?

- **寄存器重命名技术**
  - 不同的循环使用不同的物理寄存器 (dynamic loop unrolling).
  - 将代码中的静态寄存器名修改为动态寄存器指针“pointers”
  - 有效地增加了寄存器文件的大小
- **关键: 分支指令后的指令可以继续发射，以便能发射多个循环中的操作**

Acknowledgements

- **These slides contain material developed and copyright by:**
  - John Kubiatowicz (UCB)
  - Krste Asanovic (UCB)
  - John Hennessy (Standford)and David Patterson (UCB)
  - Chenxi Zhang (Tongji)
  - Muhamed Mudawar (KFUPM)
- **UCB material derived from course CS152、CS252、CS61C**
- **KFUPM material derived from course COE501、COE502**