Lecture 4: Tomasulo Algorithm and Dynamic Branch Prediction

Professor David A. Patterson
Computer Science 252
Spring 1998

Review: Summary

- Instruction Level Parallelism (ILP) in SW or HW
- Loop level parallelism is easiest to see
- SW parallelism dependencies defined for program, hazards if HW cannot resolve
- SW dependencies/compiler sophistication determine if compiler can unroll loops
  - Memory dependencies hardest to determine
- HW exploiting ILP
  - Works when can't know dependence at run time
  - Code for one machine runs well on another
- Key idea of Scoreboard: Allow instructions behind stall to proceed (Decode => Issue instr & read operands)
  - Enables out-of-order execution => out-of-order completion
  - ID stage checked both for structural & data dependencies

Review: Three Parts of the Scoreboard

1. Instruction status—which of 4 steps the instruction is in
2. Functional unit status—Indicates the state of the functional unit (FU). 9 fields for each functional unit
   - Busy—Indicates whether the unit is busy or not
   - Op—Operation to perform in the unit (e.g., + or –)
   - Fi—Destination register
   - Fj, Fk—Source-register numbers
   - Qj, Qk—Functional units producing source registers Fj, Fk
   - Rj, Rk—Flags indicating when Fj, Fk are ready
3. Register result status—Indicates which functional unit will write each register, if one exists. Blank when no pending instructions will write that register

Review: Scoreboard Example Cycle 3

Instruction status

		j	k	Read operands	Execution complete	Write Result
LD	F6	34+	R2	1	2	3
LD	F2	45+	R3			
	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F6		R2				Yes
Mult1	No								
Mult2	No								
Add	No								
Divide	No								

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
3	FU				Integer				

- Issue MULT? No, stall on structural hazard

Review: Scoreboard Example Cycle 9

Instruction status				Read operands	Execution complete	Write Result
		j	k			
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9		
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	Qj	Qk	FU for k Rj	Fj? Rk	Fk?
	Integer	No									
10	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes	
	Mult2	No									
2	Add	Yes	Sub	F8	F6	F2			Yes	Yes	
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
9	FU	Mult1				Add	Divide			

- **Read operands for MULT & SUBD? Issue ADDD?**

Review: Scoreboard Example Cycle 17

Instruction status				Read operands	Execution complete	Write Result	
		j	k				
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2	13	14	16	

	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	Qj	Qk	FU for k Rj	Fj? Rk	Fk?
	Integer	No									
2	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes	
	Mult2	No									
	Add	Yes	Add	F6	F8	F2			Yes	Yes	
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes	

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
17	FU	Mult1				Add	Divide			

- **Write result of ADDD? No, WAR hazard**

Review: Scoreboard Example Cycle 62

Instruction status				Read operands	Execution complete	Write Result
		j	k			
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
	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

	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	Qj	Qk	FU for k Rj	Fj? Rk	Fk?
	Integer	No									
	Mult1	No									
	Mult2	No									
	Add	No									
0	Divide	No									

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
62	FU								

- **In-order issue; out-of-order execute & commit**

Review: Scoreboard Summary

- **Speedup 1.7 from compiler; 2.5 by hand BUT slow memory (no cache)**
- **Limitations of 6600 scoreboard**
  - **No forwarding (First write regsiter then read it)**
  - **Limited to instructions in basic block (small *window*)**
  - **Number of functional units(structural hazards)**
  - **Wait for WAR hazards**
  - **Prevent WAW hazards**

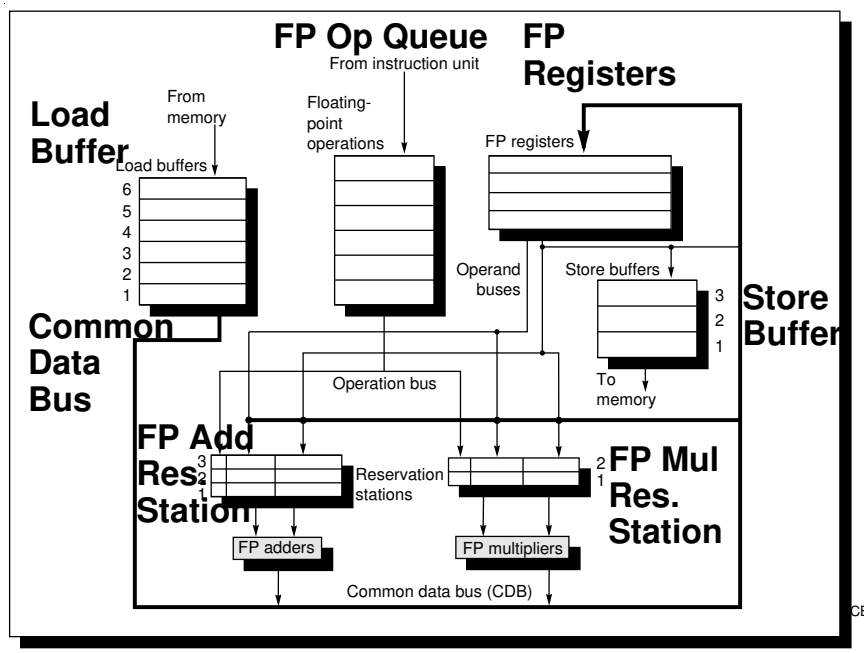
Another Dynamic Algorithm: Tomasulo Algorithm

- For IBM 360/91 about 3 years after CDC 6600 (1966)
- Goal: High Performance without special compilers
- Differences between IBM 360 & CDC 6600 ISA
  - IBM has only 2 register specifiers/instr vs. 3 in CDC 6600
  - IBM has 4 FP registers vs. 8 in CDC 6600
- Why Study? lead to Alpha 21264, HP 8000, MIPS 10000, Pentium II, PowerPC 604, …

Tomasulo Algorithm vs. Scoreboard

- Control & buffers distributed with Function Units (FU) vs. centralized in scoreboard;
  - FU buffers called "reservation stations"; have pending operands
- Registers in instructions replaced by values or pointers to reservation stations(RS); called register renaming ;
  - avoids WAR, WAW hazards
  - More reservation stations than registers, so can do optimizations compilers can't
- Results to FU from RS, not through registers, over Common Data Bus that broadcasts results to all FUs
- Load and Stores treated as FUs with RSs as well
- Integer instructions can go past branches, allowing FP ops beyond basic block in FP queue

Tomasulo Organization



Reservation Station Components

Op—Operation to perform in the unit (e.g., + or –)

Vj, Vk—Value of Source operands

- Store buffers has V field, result to be stored

Qj, Qk—Reservation stations producing source registers (value to be written)

- Note: No ready flags as in Scoreboard; Qj,Qk=0 => ready
- Store buffers only have Qi for RS producing result

Busy—Indicates reservation station or FU is busy

Register result status—Indicates which functional unit will write each register, if one exists. Blank when no pending instructions that will write that register.

Three Stages of Tomasulo Algorithm

1. **Issue—get instruction from FP Op Queue**

   If reservation station free (no structural hazard), control issues instr & sends operands (renames registers).

2. **Execution—operate on operands (EX)**

   When both operands ready then execute; if not ready, watch Common Data Bus for result

3. **Write result—finish execution (WB)**

   Write on Common Data Bus to all awaiting units; mark reservation station available

- **Normal data bus: data + destination ("go to" bus)**
- **Common data bus: data + source ("come from" bus)**
  - **64 bits of data + 4 bits of Functional Unit source address**
  - **Write if matches expected Functional Unit (produces result)**
  - **Does the broadcast**

Tomasulo Example Cycle 0

Instruction status									
Instruction		j	k	Issue	Execution complete	Write Result		Busy	Address
LD	F6	34+	R2				Load1	No	
LD	F2	45+	R3				Load2	No	
	F0	F2	F4				Load3	No	
SUBD	F8	F6	F2						
DIVD	F10	F0	F6						
ADDD	F6	F8	F2						

Reservation Stations						
	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
0	Mult1	No				
0	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	Execution complete	Write Result		Busy	Address
LD	F6	34+	R2	1			Load1	Yes	34+R2
LD	F2	45+	R3				Load2	No	
	F0	F2	F4				Load3	No	
SUBD	F8	F6	F2						
DIVD	F10	F0	F6						
ADDD	F6	F8	F2						

Reservation Stations						
	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
	Add3	No				
0	Mult1	No				
0	Mult2	No				

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

CS 252 Administrivia

- **Get your photo taken by Joe Gebis! (or give URL)**
- **Class videos review next door (201 McLaughlin)**
- **Reading Assignments for Lectures 3 to 7**
  - **Computer Architecture: A Quantitative Approach, Chapter 4, Appendix B**
- **Exercises for Lectures 3 to 7**
  - **Due Thursday Febuary 12 at 5PM homework box in 283 Soda (building is locked at 6:45 PM)**
  - **4.2, 4.10, 4.19**
  - **4.14 parts c) and d) only**
  - **B.2**
  - **Done in pairs, but both need to understand whole assignment**
  - **Study groups encouraged, but pairs do own work**

Computers in the News

- The first Alpha 21264 chips are sampling now and will enter volume production in the spring of 1998
  - 15.2 million transistors
  - 64 KB on-chip data and instruction caches
  - superscalar: ≤ 4 instructions per clock cycle to be issued to 4 integer execution units and 2 floating point units
  - Out-of-order instruction execution
  - Improved branch prediction through "intuitive execution"
- Performance will begin at an estimated 40 SPECint95 and 60 SPECfp95 and will reach more than 100 SPECint95 and 150 SPECfp95, and operate at more than 1000 MHz by the year 2000.
  - FYI: Intel Pentium II 333 MHz Pentium II (1998) 13 SPECint95, 9 SPECfp95

Tomasulo Example Cycle 2

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1		
LD	F2	45+	R3	2		
	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

	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
	Add3	No				
0	Mult1	No				
0	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	Execution complete	Write Result
LD	F6	34+	R2	1	3	
LD	F2	45+	R3	2		
	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

	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
	Add3	No				
0	Mult1	Yes	MULTD	R(F4)	Load2	
0	Mult2	No				

Register result status

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

- Note: registers names are removed ("renamed") in Reservation Stations; MULT issued vs. scoreboard
- Load1 completing; what is waiting for Load1?

Tomasulo Example Cycle 4

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	
	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

	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	Yes	SUBD	M(34+R2)		Load2
0	Add2	No				
	Add3	No				
0	Mult1	Yes	MULTD	R(F4)	Load2	
0	Mult2	No				

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
4	FU	Mult1	Load2		M(34+R2)	Add1			

- Load2 completing; what is waiting for it?

Tomasulo Example Cycle 5

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4						
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2							

Reservation Stations	Name		Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
2	Add1	Yes	SUBD	M(34+R2)	M(45+R3)		
0	Add2	No					
	Add3	No					
10	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status									
Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
5	FU	Mult1	M(45+R3)		M(34+R2)	Add1	Mult2		

Tomasulo Example Cycle 6

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4						
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6						

Reservation Stations	Name		Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
1	Add1	Yes	SUBD	M(34+R2)	M(45+R3)		
0	Add2	Yes	ADDD		M(45+R3)	Add1	
	Add3	No					
9	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status									
Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
6	FU	Mult1	M(45+R3)		Add2	Add1	Mult2		

- **Issue ADDD here vs. scoreboard?**

Tomasulo Example Cycle 7

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4	7					
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6						

Reservation Stations	Name		Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	Yes	SUBD	M(34+R2)	M(45+R3)		
0	Add2	Yes	ADDD		M(45+R3)	Add1	
	Add3	No					
8	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status									
Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
7	FU	Mult1	M(45+R3)		Add2	Add1	Mult2		

- **Add1 completing; what is waiting for it?**

Tomasulo Example Cycle 8

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6						

Reservation Stations	Time	Name		Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
	0	Add1	No					
	2	Add2	Yes	ADDD	M()-M()	M(45+R3)		
	0	Add3	No					
	7	Mult1	Yes		M(45+R3)	R(F4)		
	0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status									
Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
8	FU	Mult1	M(45+R3)		Add2	M()-M()	Mult2		

Tomasulo Example Cycle 9

Instruction status					Execution	Write				
Instruction		j	k	Issue	complete	Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6						

Reservation Stations			S1	S2	RS for j	RS for k
Time	Name	Op	Vj	Vk	Qj	Qk
0	Add1	No				
1	Add2	Yes	ADDD	M()–M()	M(45+R3)	
0	Add3	No				
6	Mult1	Yes	MULTD	M(45+R3)	R(F4)	
0	Mult2	Yes	DIVD		M(34+R2)	Mult1

Register result status									
Clock		F0	F2	F4	F6	F8	F10	F12	... F30
9	FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2		

Tomasulo Example Cycle 10

Instruction status					Execution	Write				
Instruction		j	k	Issue	complete	Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	10					

Reservation Stations			S1	S2	RS for j	RS for k
Time	Name	Op	Vj	Vk	Qj	Qk
0	Add1	No				
0	Add2	Yes	ADDD	M()–M()	M(45+R3)	
0	Add3	No				
5	Mult1	Yes	MULTD	M(45+R3)	R(F4)	
0	Mult2	Yes	DIVD		M(34+R2)	Mult1

Register result status									
Clock		F0	F2	F4	F6	F8	F10	F12	... F30
10	FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2		

- **Add2 completing; what is waiting for it?**

Tomasulo Example Cycle 11

Instruction status					Execution	Write				
Instruction		j	k	Issue	complete	Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	10	11				

Reservation Stations			S1	S2	RS for j	RS for k
Time	Name	Op	Vj	Vk	Qj	Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
4	Mult1	Yes		M(45+R3)	R(F4)	
0	Mult2	Yes	DIVD		M(34+R2)	Mult1

Register result status									
Clock		F0	F2	F4	F6	F8	F10	F12	... F30
11	FU	Mult1	M(45+R3)		(M-M)+M()	M()–M()	Mult2		

- **Write result of ADDD here vs. scoreboard?**

Tomasulo Example Cycle 12

Instruction status					Execution	Write				
Instruction		j	k	Issue	complete	Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4	6	7				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	10	11				

Reservation Stations			S1	S2	RS for j	RS for k
Time	Name	Op	Vj	Vk	Qj	Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
3	Mult1	Yes	MULTD	M(45+R3)	R(F4)	
0	Mult2	Yes	DIVD		M(34+R2)	Mult1

Register result status									
Clock		F0	F2	F4	F6	F8	F10	F12	... F30
12	FU	Mult1	M(45+R3)		(M-M)+M()	M()–M()	Mult2		

- **Note: all quick instructions complete already**

Tomasulo Example Cycle 13

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	10	11				

Reservation Stations	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
	Add3	No				
2	Mult1	Yes	MULTD M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD	M(34+R2)	Mult1	

Register result status									
Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
13	FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2		

Tomasulo Example Cycle 14

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3				Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	10	11				

Reservation Stations	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
1	Mult1	Yes	MULTD M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD	M(34+R2)	Mult1	

Register result status									
Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
14	FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2		

Tomasulo Example Cycle 15

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3	15			Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	10	11				

Reservation Stations	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
	Add3	No				
0	Mult1	Yes	MULTD M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD	M(34+R2)	Mult1	

Register result status									
Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
15	FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2		

- **Mult1 completing; what is waiting for it?**

Tomasulo Example Cycle 16

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3	15	16		Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	10	11				

Reservation Stations	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
	Add3	No				
0	Mult1	No				
40	Mult2	Yes	DIVD	M*F4	M(34+R2)	

Register result status									
Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
16	FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2		

- **Note: Just waiting for divide**

Tomasulo Example Cycle 55

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3	15	16		Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	10	11				

Reservation Stations	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
	Add3	No				
0	Mult1	No				
1	Mult2	Yes	DIVD	M*F4	M(34+R2)	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
55	FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 56

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3	15	16		Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5	56					
ADDD	F6	F8	F2	6	10	11				

Reservation Stations	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
	Add3	No				
0	Mult1	No				
0	Mult2	Yes	DIVD	M*F4	M(34+R2)	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
56	FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

- **Mult 2 completing; what is waiting for it?**

Tomasulo Example Cycle 57

Instruction status				Issue	Execution complete	Write Result			Busy	Address
LD	F6	34+	R2	1	3	4		Load1	No	
LD	F2	45+	R3	2	4	5		Load2	No	
	F0	F2	F4	3	15	16		Load3	No	
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5	56	57				
ADDD	F6	F8	F2	6	10	11				

Reservation Stations	Name	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No				
0	Add2	No				
	Add3	No				
0	Mult1	No				
0	Mult2	No				

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
57	FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	M*F4/M			

- **Again, in-oder issue, out-of-order execution, completion**

Compare to Scoreboard Cycle 62

Instruction status				j	k	Read operanc	Execution comple	Write Result
LD	F6	34+	R2			1	2	3	4
LD	F2	45+	R3			5	6	7	8
	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

	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	Qj	FU for k Qk	Fj? Rj	Fk? Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
0	Divide	No								

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
62	FU									

- **Why takes longer on Scoreboard/6600?**

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

Pipelined Functional Units	Multiple Functional Units
(6 load, 3 store, 3 +, 2 x/÷)	(1 load/store, 1 + , 2 x, 1 ÷)
window size: ≤ 14 instructions	≤ 5 instructions
No issue on structural hazard	same
WAR: renaming avoids	stall completion
WAW: renaming avoids	stall completion
Broadcast results from FU	Write/read registers
Control: reservation stations	central scoreboard

Tomasulo Drawbacks

- Complexity
  - delays of 360/91, MIPS 10000, IBM 620?
- Many associative stores (CDB) at high speed
- Performance limited by Common Data Bus
  - Multiple CDBs => more FU logic for parallel assoc stores

Tomasulo Loop Example

```
Loop: LD     F0   0   R1
      MULTD  F4   F0  F2
      SD     F4   0   R1
      SUBI   R1   R1  #8
      BNEZ   R1   Loop
```

- Assume Multiply takes 4 clocks
- Assume first load takes 8 clocks (cache miss?), second load takes 4 clocks (hit)
- To be clear, will show clocks for SUBI, BNEZ
- Reality, integer instructions ahead

Loop Example Cycle 0

Instruction status

		j	k	iteration	Issue	Write Result
LD	F0	0	R1	1		
	F4	F0	F2	1		
SD	F4	0	R1	1		
LD	F0	0	R1	2		
	F4	F0	F2	2		
SD	F4	0	R1	2		

	Busy	Address	
Load1	No		
Load2	No		
Load3	No		Qi
Store1	No		
Store2	No		
Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	Qj	Qk	RS for k
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	No					
0	Mult2	No					

Code:

```
LD    F0   0  R1
      F4   F0 F2
SD    F4   0  R1
SUBI  R1   R1 #8
BNEZ  R1   Loop
```

Register result status

Clock	R1	F0	F2	F4	F6	F8	F12...
0	8 0	Qi					

Loop Example Cycle 1

Instruction status

		j	k	iteration	Issue	Write Result
LD	F0	0	R1	1	1	
	F4	F0	F2	1		
SD	F4	0	R1	1		
LD	F0	0	R1	2		
	F4	F0	F2	2		
SD	F4	0	R1	2		

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	Qj	Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
0	Mult1	No				
0	Mult2	No				

RS for k

Code:

```
LD   F0   0 R1
     F4   F0 F2
SD   F4   0 R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
1	80	Qi	Load1					

Loop Example Cycle 2

Instruction status

		j	k	iteration	Issue	Write Result
LD	F0	0	R1	1	1	
	F4	F0	F2	1	2	
SD	F4	0	R1	1		
LD	F0	0	R1	2		
	F4	F0	F2	2		
SD	F4	0	R1	2		

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	Qj	Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
0	Mult1	Yes	MULTD	R(F2)	Load1	
0	Mult2	No				

RS for k

Code:

```
LD   F0   0 R1
     F4   F0 F2
SD   F4   0 R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
2	80	Qi	Load1		Mult1			

Loop Example Cycle 3

Instruction status

		j	k	iteration	Issue	Write Result
LD	F0	0	R1	1	1	
	F4	F0	F2	1	2	
SD	F4	0	R1	1	3	
LD	F0	0	R1	2		
	F4	F0	F2	2		
SD	F4	0	R1	2		

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	Qj	Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
0	Mult1	Yes	MULTD	R(F2)	Load1	
0	Mult2	No				

RS for k

Code:

```
LD   F0   0 R1
     F4   F0 F2
SD   F4   0 R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
3	80	Qi	Load1		Mult1			

- **Note: MULT1 has no registers names in RS**

Loop Example Cycle 4

Instruction status

		j	k	iteration	Issue	Write Result
LD	F0	0	R1	1	1	
	F4	F0	F2	1	2	
SD	F4	0	R1	1	3	
LD	F0	0	R1	2		
	F4	F0	F2	2		
SD	F4	0	R1	2		

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	Qj	Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
0	Mult1	Yes	MULTD	R(F2)	Load1	
0	Mult2	No				

RS for k

Code:

```
LD   F0   0 R1
     F4   F0 F2
SD   F4   0 R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
4	72	Qi	Load1		Mult1			

Loop Example Cycle 5

Instruction status

Instruction		j	k	iteration	Issue	Write Result		Busy	Address	
LD	F0	0	R1	1	1		Load1	Yes	80	
	F4	F0	F2	1	2		Load2	No		
SD	F4	0	R1	1	3		Load3	No		Qi
LD	F0	0	R1	2			Store1	Yes	80	Mult1
	F4	F0	F2	2			Store2	No		
SD	F4	0	R1	2			Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	RS for k Qj	RS for k Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
0	Mult1	Yes	MULTD	R(F2)	Load1		SUBI	R1	R1 #8
0	Mult2	No					BNEZ	R1	Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
5	72	Qi	Load1		Mult1			

Loop Example Cycle 6

Instruction status

Instruction		j	k	iteration	Issue	Write Result		Busy	Address	
LD	F0	0	R1	1	1		Load1	Yes	80	
	F4	F0	F2	1	2		Load2	Yes	72	
SD	F4	0	R1	1	3		Load3	No		Qi
LD	F0	0	R1	2	6		Store1	Yes	80	Mult1
	F4	F0	F2	2			Store2	No		
SD	F4	0	R1	2			Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	RS for k Qj	RS for k Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
0	Mult1	Yes	MULTD	R(F2)	Load1		SUBI	R1	R1 #8
0	Mult2	No					BNEZ	R1	Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
6	72	Qi	Load2		Mult1			

- **Note: F0 never sees Load1 result**

Loop Example Cycle 7

Instruction status

Instruction		j	k	iteration	Issue	Write Result		Busy	Address	
LD	F0	0	R1	1	1		Load1	Yes	80	
	F4	F0	F2	1	2		Load2	Yes	72	
SD	F4	0	R1	1	3		Load3	No		Qi
LD	F0	0	R1	2	6		Store1	Yes	80	Mult1
	F4	F0	F2	2	7		Store2	No		
SD	F4	0	R1	2			Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	RS for k Qj	RS for k Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
0	Mult1	Yes	MULTD	R(F2)	Load1		SUBI	R1	R1 #8
0	Mult2	Yes	MULTD	R(F2)	Load2		BNEZ	R1	Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
7	72	Qi	Load2		Mult2			

- **Note: MULT2 has no registers names in RS**

Loop Example Cycle 8

Instruction status

Instruction		j	k	iteration	Issue	Write Result		Busy	Address	
LD	F0	0	R1	1	1		Load1	Yes	80	
	F4	F0	F2	1	2		Load2	Yes	72	
SD	F4	0	R1	1	3		Load3	No		Qi
LD	F0	0	R1	2	6		Store1	Yes	80	Mult1
	F4	F0	F2	2	7		Store2	Yes	72	Mult2
SD	F4	0	R1	2	8		Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	RS for k Qj	RS for k Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
0	Mult1	Yes	MULTD	R(F2)	Load1		SUBI	R1	R1 #8
0	Mult2	Yes	MULTD	R(F2)	Load2		BNEZ	R1	Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
8	72	Qi	Load2		Mult2			

Loop Example Cycle 9

Instruction status

Instruction		j	k	iteration	Issue		Write Result		Busy	Address	
LD	F0	0	R1	1	1	9		Load1	Yes	80	
	F4	F0	F2	1	2			Load2	Yes	72	
SD	F4	0	R1	1	3			Load3	No		Qi
LD	F0	0	R1	2	6			Store1	Yes	80	Mult1
	F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD	F4	0	R1	2	8			Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	RS for k Qj	RS for k Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
0	Mult1	Yes	MULTD		R(F2)	Load1	SUBI	R1	R1 #8
0	Mult2	Yes	MULTD		R(F2)	Load2	BNEZ	R1	Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
9	64	Qi	Load2		Mult2			

- **Load1 completing; what is waiting for it?**

Loop Example Cycle 10

Instruction status

Instruction		j	k	iteration	Issue		Write Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
	F4	F0	F2	1	2			Load2	Yes	72	
SD	F4	0	R1	1	3			Load3	No		Qi
LD	F0	0	R1	2	6	10		Store1	Yes	80	Mult1
	F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD	F4	0	R1	2	8			Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	RS for k Qj	RS for k Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
4	Mult1	Yes	MULTD	M(80)	R(F2)		SUBI	R1	R1 #8
0	Mult2	Yes	MULTD		R(F2)	Load2	BNEZ	R1	Loop

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
10	64	Qi	Load2		Mult2			

- **Load2 completing; what is waiting for it?**

Loop Example Cycle 11

Instruction status

Instruction		j	k	iteration	Issue		Write Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
	F4	F0	F2	1	2			Load2	No		
SD	F4	0	R1	1	3			Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1
	F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD	F4	0	R1	2	8			Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	RS for k Qj	RS for k Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
3	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	F12 ...
11	64	Qi	Load3		Mult2			

Loop Example Cycle 12

Instruction status

Instruction		j	k	iteration	Issue		Write Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
	F4	F0	F2	1	2			Load2	No		
SD	F4	0	R1	1	3			Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1
	F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD	F4	0	R1	2	8			Store3	No		

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	RS for k Qj	RS for k Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
2	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	F12 ...
12	64	Qi	Load3		Mult2			

Loop Example Cycle 13

Instruction status						Write				
Instruction	j	k	iteration	Issue		Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
F4	F0	F2	1	2			Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1
F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD F4	0	R1	2	8			Store3	No		

Reservation Stations			S1	S2		RS for k			
Name		Op	Vj	Vk	Qj	Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
1	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	F12 ...
13	64	Qi	Load3		Mult2		

Loop Example Cycle 14

Instruction status						Write				
Instruction	j	k	iteration	Issue		Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
F4	F0	F2	1	2	14		Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1
F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD F4	0	R1	2	8			Store3	No		

Reservation Stations			S1	S2		RS for k			
Name		Op	Vj	Vk	Qj	Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
0	Mult1	Yes	MULTD	M(80)	R(F2)		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	F12 ...
14	64	Qi	Load3		Mult2		

- **Mult1 completing; what is waiting for it?**

Loop Example Cycle 15

Instruction status						Write				
Instruction	j	k	iteration	Issue		Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(
MULTI F4	F0	F2	2	7	15		Store2	Yes	72	Mult2
SD F4	0	R1	2	8			Store3	No		

Reservation Stations			S1	S2		RS for k			
Name		Op	Vj	Vk	Qj	Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
0	Mult1	No					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	F12 ...
15	64	Qi	Load3		Mult2		

- **Mult2 completing; what is waiting for it?**

Loop Example Cycle 16

Instruction status						Write				
Instruction	j	k	iteration	Issue		Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(
MULTI F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(
SD F4	0	R1	2	8			Store3	No		

Reservation Stations			S1	S2		RS for k			
Name		Op	Vj	Vk	Qj	Qk	Code:		
0	Add1	No					LD	F0	0 R1
0	Add2	No						F4	F0 F2
0	Add3	No					SD	F4	0 R1
0	Mult1	Yes	MULTD		R(F2)	Load3	SUBI	R1	R1 #8
0	Mult2	No					BNEZ	R1	Loop

Register result status

Clock	R1	F0	F2	F4	F6	F8	F12 ...
16	64	Qi	Load3		Mult1		

Loop Example Cycle 17

Instruction status						Write				
Instruction	j	k	iteration	Issue		Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(
MULTI F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(
SD F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations		S1	S2		RS for k			
Name	Op	Vj	Vk	Qj	Qk	Code:		
0 Add1	No					LD	F0	0 R1
0 Add2	No						F4	F0 F2
0 Add3	No					SD	F4	0 R1
0 Mult1	Yes	MULTD		R(F2)	Load3	SUBI	R1	R1 #8
0 Mult2	No					BNEZ	R1	Loop

Register result status

Clock	R1	F0	F2	F4	F6	F8	F12 ...
17	64	Qi	Load3		Mult1		

Loop Example Cycle 18

Instruction status						Write				
Instruction	j	k	iteration	Issue		Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3	18		Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(
MULTI F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(
SD F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations		S1	S2		RS for k			
Name	Op	Vj	Vk	Qj	Qk	Code:		
0 Add1	No					LD	F0	0 R1
0 Add2	No						F4	F0 F2
0 Add3	No					SD	F4	0 R1
0 Mult1	Yes	MULTD		R(F2)	Load3	SUBI	R1	R1 #8
0 Mult2	No					BNEZ	R1	Loop

Register result status

Clock	R1	F0	F2	F4	F6	F8	F12 ...
18	56	Qi	Load3		Mult1		

Loop Example Cycle 19

Instruction status						Write				
Instruction	j	k	iteration	Issue		Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3	18	19	Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	No		
F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(
SD F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations		S1	S2		RS for k			
Name	Op	Vj	Vk	Qj	Qk	Code:		
0 Add1	No					LD	F0	0 R1
0 Add2	No						F4	F0 F2
0 Add3	No					SD	F4	0 R1
0 Mult1	Yes	MULTD		R(F2)	Load3	SUBI	R1	R1 #8
0 Mult2	No					BNEZ	R1	Loop

Register result status

Clock	R1	F0	F2	F4	F6	F8	F12 ...
19	56	Qi	Load3		Mult1		

Loop Example Cycle 20

Instruction status						Write				
Instruction	j	k	iteration	Issue		Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3	18	19	Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	No		
F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(
SD F4	0	R1	2	8	20		Store3	Yes	64	Mult1

Reservation Stations		S1	S2		RS for k			
Name	Op	Vj	Vk	Qj	Qk	Code:		
0 Add1	No					LD	F0	0 R1
0 Add2	No						F4	F0 F2
0 Add3	No					SD	F4	0 R1
0 Mult1	Yes	MULTD		R(F2)	Load3	SUBI	R1	R1 #8
0 Mult2	No					BNEZ	R1	Loop

Register result status

Clock	R1	F0	F2	F4	F6	F8	F12 ...
20	56	Qi	Load3		Mult1		

Loop Example Cycle 21

Instruction status

Instruction		j	k	iteration	Issue		Write Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
	F4	F0	F2	1	2	14	15	Load2	No		
SD	F4	0	R1	1	3	18	19	Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	No		
	F4	F0	F2	2	7	15	16	Store2	No		
SD	F4	0	R1	2	8	20	21	Store3	Yes	64	Mult1

Reservation Stations

	Name	Op	S1 Vj	S2 Vk	Qj	Qk	RS for k Code:			
0	Add1	No					LD	F0	0	R1
0	Add2	No						F4	F0	F2
0	Add3	No					SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load3	SUBI	R1	R1	#8
0	Mult2	No					BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F12 ...
21	56	Qi	Load3		Mult1			

Tomasulo Summary

- **Reservations stations: renaming to larger set of registers + buffering source operands**
  - **Prevents registers as bottleneck**
  - **Avoids WAR, WAW hazards of Scoreboard**
  - **Allows loop unrolling in HW**
- **Not limited to basic blocks (integer units gets ahead, beyond branches)**
- **Helps cache misses as well**
- **Lasting Contributions**
  - **Dynamic scheduling**
  - **Register renaming**
  - **Load/store disambiguation**
- **360/91 descendants are Pentium II; PowerPC 604; MIPS R10000; HP-PA 8000; Alpha 21264**