

Tomasulo Algorithm

Review: Scoreboard Example Cycle 62

<u>Instruction status</u>				<i>Read</i>	<i>Execut</i>	<i>Write</i>	
<i>Instructi</i>	<i>j</i>	<i>k</i>	<i>Issu</i>	<i>operan</i>	<i>comple</i>	<i>Result</i>	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	62
ADD	F6	F8	F2	13	14	16	22

<u>Functional unit st</u>	<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for</i>	<i>FU for</i>	<i>Fj?</i>	<i>Fk?</i>
<i>Time Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>
Integer	No						
Mult1	No						
Mult2	No						
Add	No						
0 Divide	No						

<u>Register result status</u>	<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>...</i>	<i>F30</i>
Clock									
62									

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

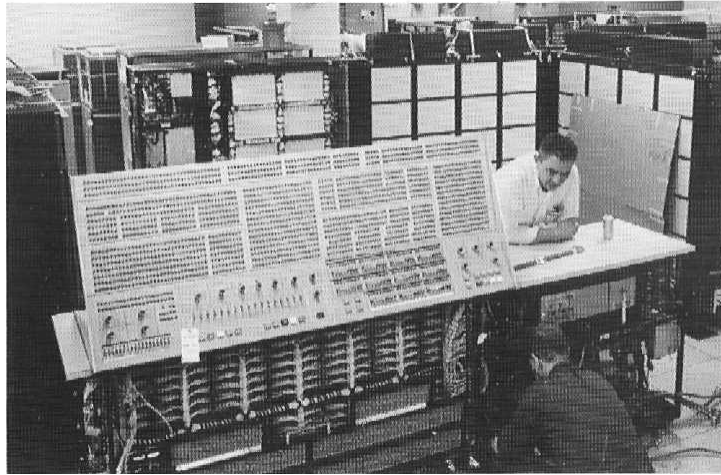
Review: Scoreboard Summary

- **Speedup 1.7 from compiler; 2.5 by hand BUT slow memory (no cache)**
- **Limitations of 6600 scoreboard**
 - No forwarding (First write register 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, ...**

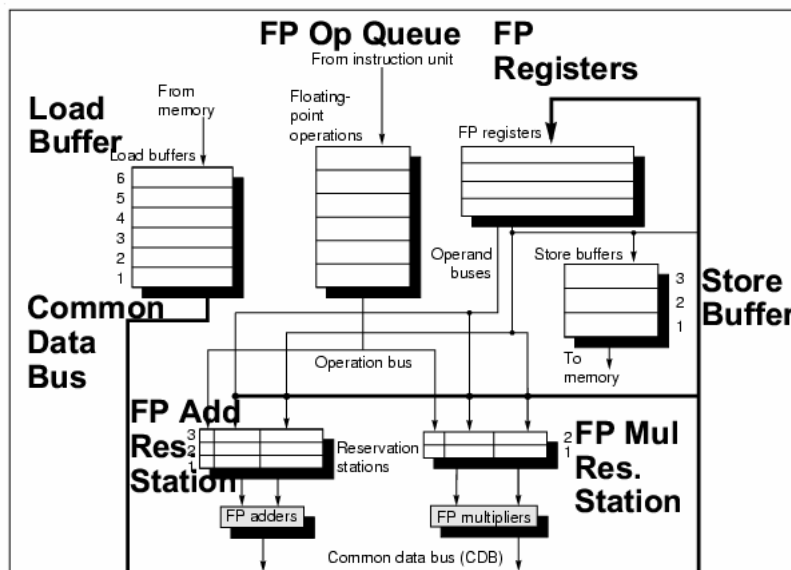
Installation of the IBM 360/91 in the Columbia Computer Center machine room in February or March 1969



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

Reservation Stations			<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>
Time	Name	Busy Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
0	Add1	No				
0	Add2	No				
0	Add3	No				
0	Mult1	No				
0	Mult2	No				

Register result status										
Clock			<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	... <i>F30</i>
0		FU								

Tomasulo Example Cycle 1

Instruction status				Execution		Write			Busy	Address
Instruction	<i>j</i>	<i>k</i>		Issue	complete	Result				
LD	F6	34+	R2		1			Load1	Yes	34+R2
LD	F2	45+	R3					Load2	No	
MULTI	F0	F2	F4					Load3	No	
SUBD	F8	F6	F2							
DIVD	F10	F0	F6							
ADDD	F6	F8	F2							

Reservation Stations			<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>
Time	Name	Busy Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>
0	Add1	No				
0	Add2	No				
	Add3	No				
0	Mult1	No				
0	Mult2	No				

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

Tomasulo Example Cycle 2

Instruction status				Execution Write				
Instruction	j	k		Issue	complete	Result	Busy	Address
LD	F6	34+	R2	1			Load1	Yes 34+R2
LD	F2	45+	R3	2			Load2	Yes 45+R3
MULT	F0	F2	F4				Load3	No
SUBD	F8	F6	F2					
DIVD	F10	F0	F6					
ADD	F6	F8	F2					

Reservation Stations						
Time	Name	Busy 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				Execution Write				
Instruction	j	k		Issue	complete	Result	Busy	Address
LD	F6	34+	R2	1		3	Load1	Yes 34+R2
LD	F2	45+	R3	2			Load2	Yes 45+R3
MULT	F0	F2	F4	3			Load3	No
SUBD	F8	F6	F2					
DIVD	F10	F0	F6					
ADD	F6	F8	F2					

Reservation Stations						
Time	Name	Busy 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				Execution		Write				
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	3	4				
LD	F2	45+	R3	2	4		Load1	No		
MULT	F0	F2	F4	3			Load2	Yes	45+R3	
SUBD	F8	F6	F2	4			Load3	No		
DIVD	F10	F0	F6							
ADD	F6	F8	F2							
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>			
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>			
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			<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	... <i>F30</i>
4		FU	Mult1	Load2		M(34+R2)	Add1			

- Load2 completing; what is waiting for it?

Tomasulo Example Cycle 5

Instruction status				Execution		Write				
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	3	4				
LD	F2	45+	R3	2	4	5	Load1	No		
MULT	F0	F2	F4	3			Load2	No		
SUBD	F8	F6	F2	4			Load3	No		
DIVD	F10	F0	F6	5						
ADD	F6	F8	F2							
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>			
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>			
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			<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	... <i>F30</i>
5		FU	Mult1	M(45+R3)		M(34+R2)	Add1	Mult2		

Tomasulo Example Cycle 6

Instruction status				Execution		Write						
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address				
LD	F6	34+	R2	1	3	4						
LD	F2	45+	R3	2	4	5	Load1	No				
MULT	F0	F2	F4	3			Load2	No				
SUBD	F8	F6	F2	4			Load3	No				
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6								
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
1	Add1	Yes	SUBD	M(34+R2)	M(45+R3)							
0	Add2	Yes	ADD		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				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
6		FU		Mult1	M(45+R3)		Add2	Add1	Mult2			

- Issue ADDD here vs. scoreboard?

Tomasulo Example Cycle 7

Instruction status				Execution		Write						
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address				
LD	F6	34+	R2	1	3	4						
LD	F2	45+	R3	2	4	5	Load1	No				
MULT	F0	F2	F4	3			Load2	No				
SUBD	F8	F6	F2	4	7		Load3	No				
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6								
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
0	Add1	Yes	SUBD	M(34+R2)	M(45+R3)							
0	Add2	Yes	ADD		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				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
7		FU		Mult1	M(45+R3)		Add2	Add1	Mult2			

- Add1 completing; what is waiting for it?

Tomasulo Example Cycle 8

Instruction status				Execution			Write					
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	3	4			Load1	No		
LD	F2	45+	R3	2	4	5			Load2	No		
MULT	F0	F2	F4	3					Load3	No		
SUBDF	F8	F6	F2	4	7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6								
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
0	Add1	No										
2	Add2	Yes	ADDD	M()-M()	M(45+R3)							
0	Add3	No										
7	Mult1	Yes	MULTD	M(45+R3)	R(F4)							
0	Mult2	Yes	DIVD		M(34+R2)	Mult1						
Register result status:												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
8		FU		Mult1	M(45+R3)		Add2	M()-M()	Mult2			

Tomasulo Example Cycle 9

Instruction status				Execution			Write					
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	3	4			Load1	No		
LD	F2	45+	R3	2	4	5			Load2	No		
MULT	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				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
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				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
9		FU		Mult1	M(45+R3)		Add2	M()-M()	Mult2			

Tomasulo Example Cycle 10

Instruction status				Execution	Write							
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>complete</i>	<i>Result</i>		Busy	Address			
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	4	5		Load2	No			
MULT	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2	4	7	8						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6	10							
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
	<i>Time</i>	<i>Name</i>	<i>Busy Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
	0	Add1	No									
	0	Add2	Yes	ADD	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				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
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	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>complete</i>	<i>Result</i>		Busy	Address			
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	4	5		Load2	No			
MULT	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2	4	7	8						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6	10	11						
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
	<i>Time</i>	<i>Name</i>	<i>Busy Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
	0	Add1	No									
	0	Add2	No									
	0	Add3	No									
	4	Mult1	Yes	MULTD	M(45+R3)	R(F4)						
	0	Mult2	Yes	DIVD		M(34+R2)	Mult1					
Register result status												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
11			FU	Mult1	M(45+R3)		(M-M)+M()	M()-M()	Mult2			

- Write result of ADD here vs. scoreboard?

Tomasulo Example Cycle 12

Instruction status				Execution		Write						
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address				
LD	F6	34+	R2	1	3	4						
LD	F2	45+	R3	2	4	5	Load1	No				
MULT	F0	F2	F4	3			Load2	No				
SUBD	F8	F6	F2	4	6	7	Load3	No				
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	10	11						
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
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				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
12			FU	Mult1	M(45+R3)		(M-M)+M()	M()-M()	Mult2			

- Note: all quick instructions complete already

Tomasulo Example Cycle 13

Instruction status				Execution		Write						
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address				
LD	F6	34+	R2	1	3	4						
LD	F2	45+	R3	2	4	5	Load1	No				
MULT	F0	F2	F4	3			Load2	No				
SUBD	F8	F6	F2	4	7	8	Load3	No				
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	10	11						
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
0	Add1	No										
0	Add2	No										
0	Add3	No										
2	Mult1	Yes	MULTD	M(45+R3)	R(F4)							
0	Mult2	Yes	DIVD	M(34+R2)	Mult1							
Register result status												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
13			FU	Mult1	M(45+R3)		(M-M)+M()	M()-M()	Mult2			

Tomasulo Example Cycle 14

Instruction status				Execution		Write					
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>				Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No		
LD	F2	45+	R3	2	4	5		Load2	No		
MULT	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				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
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				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	... <i>F30</i>
14		FU	Mult1	M(45+R3)			(M-M)+M()	M()-M()	Mult2		

Tomasulo Example Cycle 15

Instruction status				Execution		Write					
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>				Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No		
LD	F2	45+	R3	2	4	5		Load2	No		
MULT	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				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
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				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	... <i>F30</i>
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				Execution		Write						
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address				
LD	F6	34+	R2	1	3	4						
LD	F2	45+	R3	2	4	5	Load1	No				
MULT	F0	F2	F4	3	15	16	Load2	No				
SUBD	F8	F6	F2	4	7	8	Load3	No				
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	10	11						
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
0	Add1	No										
0	Add2	No										
	Add3	No										
0	Mult1	No										
40	Mult2	Yes	DIVD	M*F4	M(34+R2)							
Register result status												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
16		FU		M*F4	M(45+R3)		(M-M)+M()	M()-M()	Mult2			

- Note: Just waiting for divide

Tomasulo Example Cycle 55

Instruction status				Execution		Write						
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address				
LD	F6	34+	R2	1	3	4						
LD	F2	45+	R3	2	4	5	Load1	No				
MULT	F0	F2	F4	3	15	16	Load2	No				
SUBD	F8	F6	F2	4	7	8	Load3	No				
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	10	11						
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>					
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>					
0	Add1	No										
0	Add2	No										
	Add3	No										
0	Mult1	No										
1	Mult2	Yes	DIVD	M*F4	M(34+R2)							
Register result status												
Clock				<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>
55		FU		M*F4	M(45+R3)		(M-M)+M()	M()-M()	Mult2			

Tomasulo Example Cycle 56

Instruction status				Execution		Write				
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	3	4				
LD	F2	45+	R3	2	4	5	Load1	No		
MULT	F0	F2	F4	3	15	16	Load2	No		
SUBD	F8	F6	F2	4	7	8	Load3	No		
DIVD	F10	F0	F6	5	56					
ADDD	F6	F8	F2	6	10	11				
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>			
<i>Time</i>	<i>Name</i>	<i>Busy Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
0	Add1	No								
0	Add2	No								
	Add3	No								
0	Mult1	No								
0	Mult2	Yes	DIVD	M*F4	M(34+R2)					
Register result status										
Clock			<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	... <i>F30</i>
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				Execution		Write				
Instruction <i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	3	4				
LD	F2	45+	R3	2	4	5	Load1	No		
MULT	F0	F2	F4	3	15	16	Load2	No		
SUBD	F8	F6	F2	4	7	8	Load3	No		
DIVD	F10	F0	F6	5	56	57				
ADDD	F6	F8	F2	6	10	11				
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>			
<i>Time</i>	<i>Name</i>	<i>Busy Op</i>	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>				
0	Add1	No								
0	Add2	No								
	Add3	No								
0	Mult1	No								
0	Mult2	No								
Register result status										
Clock			<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	... <i>F30</i>
57		FU	M*F4	M(45+R3)		(M-M)+M()	M()-M()	M*F4/M		

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

Compare to Scoreboard Cycle 62

Instruction status				Read	Execut	Write	
Instructi	j	k		Issu	operan	comple	Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTF0	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

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

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

- Why takes longer on Scoreboard/6600?

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

Pipelined Functional Units (6 load, 3 store, 3 +, 2 x/÷)	Multiple Functional Units (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				ExecutiWrite			Busy Address		
Instructi	j	k	iteration	Issue	complet	Result	Load	Store	
LD	F0	0 R1	1				Load1	No	
MULTF4		F0 F2	1				Load2	No	
SD	F4	0 R1	1				Load3	No	Qi
LD	F0	0 R1	2				Store1	No	
MULTF4		F0 F2	2				Store2	No	
SD	F4	0 R1	2				Store3	No	

Reservation Stations			S1	S2	RS for	RS for	k
Time Name	Busy Op		Vj	Vk	Qj	Qk	Code:
0 Add1	No						LD F0 0 R1
0 Add2	No						MULTF4 F0 F2
0 Add3	No						SD F4 0 R1
0 Mult1	No						SUBI R1 R1 #8
0 Mult2	No						BNEZ R1 Loop

Register result status											
Clock	R1	F0	F2	F4	F6	F8	F10	F12	..	F30	
0	80	Qi									

Loop Example Cycle 1

Instruction status				ExecutiWrite			Busy Address		
Instructi	j	k	iteration	Issue	complet	Result	Load	Store	
LD	F0	0 R1	1	1			Load1	Yes	80
MULTF4		F0 F2	1				Load2	No	
SD	F4	0 R1	1				Load3	No	Qi
LD	F0	0 R1	2				Store1	No	
MULTF4		F0 F2	2				Store2	No	
SD	F4	0 R1	2				Store3	No	

Reservation Stations			S1	S2	RS for	RS for	k
Time Name	Busy Op		Vj	Vk	Qj	Qk	Code:
0 Add1	No						LD F0 0 R1
0 Add2	No						MULTF4 F0 F2
0 Add3	No						SD F4 0 R1
0 Mult1	No						SUBI R1 R1 #8
0 Mult2	No						BNEZ R1 Loop

Register result status											
Clock	R1	F0	F2	F4	F6	F8	F10	F12	..	F30	
1	80	Qi	Load1								

Loop Example Cycle 2

Instruction status				ExecutiWrite			Busy Address	
Instructi	j	k	iteration	Issue	complet	Result	Load	Store
LD	F0	0 R1	1	1			Load1	Yes 80
MULTF4	F0 F2		1	2			Load2	No
SD	F4	0 R1	1				Load3	No Qi
LD	F0	0 R1	2				Store1	No
MULTF4	F0 F2		2				Store2	No
SD	F4	0 R1	2				Store3	No

Reservation Stations		S1	S2	RS for	RS for	k
Time Name	Busy Op	Vj	Vk	Qj	Qk	Code:
0 Add1	No					LD F0 0 R1
0 Add2	No					MULTF4 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		F0	F2	F4	F6	F8	F10	F12 .. F30
Clock	R1							
2	80 Qi	Load1		Mult1				

Loop Example Cycle 3

Instruction status				ExecutiWrite			Busy Address	
Instructi	j	k	iteration	Issue	complet	Result	Load	Store
LD	F0	0 R1	1	1			Load1	Yes 80
MULTF4	F0 F2		1	2			Load2	No
SD	F4	0 R1	1	3			Load3	No Qi
LD	F0	0 R1	2				Store1	Yes 80 Mult1
MULTF4	F0 F2		2				Store2	No
SD	F4	0 R1	2				Store3	No

Reservation Stations		S1	S2	RS for	RS for	k
Time Name	Busy Op	Vj	Vk	Qj	Qk	Code:
0 Add1	No					LD F0 0 R1
0 Add2	No					MULTF4 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		F0	F2	F4	F6	F8	F10	F12 .. F30
Clock	R1							
3	80 Qi	Load1		Mult1				

• Note: MULT1 has no registers names in RS

Loop Example Cycle 4

Instruction status				ExecutiWrite			Busy Address	
Instructi	j	k	iteration	Issue	complet	Result	Load	Address
LD	F0	0 R1	1	1			Load1	Yes 80
MULTF4	F0 F2		1	2			Load2	No
SD	F4	0 R1	1	3			Load3	No Qi
LD	F0	0 R1	2				Store1	Yes 80 Mult1
MULTF4	F0 F2		2				Store2	No
SD	F4	0 R1	2				Store3	No

Reservation Stations		S1	S2	RS for	RS for	k
Time Name	Busy Op	Vj	Vk	Qj	Qk	Code:
0 Add1	No					LD F0 0 R1
0 Add2	No					MULTF4 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		F0	F2	F4	F6	F8	F10	F12 .. F30
Clock	R1							
4	72 Qi	Load1	Mult1					

Loop Example Cycle 5

Instruction status				ExecutiWrite			Busy Address	
Instructi	j	k	iteration	Issue	complet	Result	Load	Address
LD	F0	0 R1	1	1			Load1	Yes 80
MULTF4	F0 F2		1	2			Load2	No
SD	F4	0 R1	1	3			Load3	No Qi
LD	F0	0 R1	2				Store1	Yes 80 Mult1
MULTF4	F0 F2		2				Store2	No
SD	F4	0 R1	2				Store3	No

Reservation Stations		S1	S2	RS for	RS for	k
Time Name	Busy Op	Vj	Vk	Qj	Qk	Code:
0 Add1	No					LD F0 0 R1
0 Add2	No					MULTF4 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		F0	F2	F4	F6	F8	F10	F12 .. F30
Clock	R1							
5	72 Qi	Load1	Mult1					

Loop Example Cycle 6

Instruction status				Execution			Write		
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address	
LD	F0	0 R1	1	1			Load1	Yes 80	
MULT	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	
MULT	F4	F0 F2	2				Store2	No	
SD	F4	0 R1	2				Store3	No	

Reservation Stations			S1	S2	RS for j	RS for k
Time	Name	Busy Op	Vj	Vk	Qj	Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
0	Mult1	Yes	MULTD	R(F2)	Load1	
0	Mult2	No				

Register result status									
Clock	R1		F0	F2	F4	F6	F8	F10	F12 .. F30
6	72	Qi	Load2		Mult1				

• Note: F0 never sees Load1 result

Loop Example Cycle 7

Instruction status				Execution			Write		
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address	
LD	F0	0 R1	1	1			Load1	Yes 80	
MULT	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	
MULT	F4	F0 F2	2	7			Store2	No	
SD	F4	0 R1	2				Store3	No	

Reservation Stations			S1	S2	RS for j	RS for k
Time	Name	Busy Op	Vj	Vk	Qj	Qk
0	Add1	No				
0	Add2	No				
0	Add3	No				
0	Mult1	Yes	MULTD	R(F2)	Load1	
0	Mult2	Yes	MULTD	R(F2)	Load2	

Register result status									
Clock	R1		F0	F2	F4	F6	F8	F10	F12 .. F30
7	72	Qi	Load2		Mult2				

• Note: MULT2 has no registers names in RS

Loop Example Cycle 8

Instruction status				Executi Write			Busy Address	
Instructi	j	k	iteration	Issue	complet	Result	Load1	Load2
LD	F0	0 R1	1	1			Yes	80
MULTF4	F0 F2		1	2			Yes	72
SD	F4	0 R1	1	3			No	Qi
LD	F0	0 R1	2	6			Yes	80 Mult1
MULTF4	F0 F2		2	7			Yes	72 Mult2
SD	F4	0 R1	2	8			No	

Reservation Stations		S1	S2	RS for	RS for	k
Time Name	Busy Op	Vj	Vk	Qj	Qk	Code:
0 Add1	No					LD F0 0 R1
0 Add2	No					MULTF4 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		F0	F2	F4	F6	F8	F10	F12 .. F30
Clock	R1							
8	72 Qi	Load2	Mult2					

Loop Example Cycle 9

Instruction status				Executi Write			Busy Address	
Instructi	j	k	iteration	Issue	complet	Result	Load1	Load2
LD	F0	0 R1	1	1	9		Yes	80
MULTF4	F0 F2		1	2			Yes	72
SD	F4	0 R1	1	3			No	Qi
LD	F0	0 R1	2	6			Yes	80 Mult1
MULTF4	F0 F2		2	7			Yes	72 Mult2
SD	F4	0 R1	2	8			No	

Reservation Stations		S1	S2	RS for	RS for	k
Time Name	Busy Op	Vj	Vk	Qj	Qk	Code:
0 Add1	No					LD F0 0 R1
0 Add2	No					MULTF4 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		F0	F2	F4	F6	F8	F10	F12 .. F30
Clock	R1							
9	64 Qi	Load2	Mult2					

• Load1 completing; what is waiting for it?

Loop Example Cycle 10

Instruction status				Execution			Write			Busy Address	
Instruction	j	k	iteration	Issue	complet	Result					
LD F0	0	R1	1	1	9	10	Load1	No			
MULT 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	
MULT F4	F0	F2	2	7			Store2	Yes	72	Mult2	
SD F4	0	R1	2	8			Store3	No			

Reservation Stations			S1	S2	RS for	RS for	k
Time	Name	Busy Op	Vj	Vk	Qj	Qk	Code:
0	Add1	No					LD F0 0 R1
0	Add2	No					MULT 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		F0	F2	F4	F6	F8	F10	F12	..	F30
Clock	R1									
10	64	Qi	Load2	Mult2						

- Load2 completing; what is waiting for it?

Loop Example Cycle 11

Instruction status				Execution			Write			Busy Address	
Instruction	j	k	iteration	Issue	complet	Result					
LD F0	0	R1	1	1	9	10	Load1	No			
MULT 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	
MULT F4	F0	F2	2	7			Store2	Yes	72	Mult2	
SD F4	0	R1	2	8			Store3	No			

Reservation Stations			S1	S2	RS for	RS for	k
Time	Name	Busy Op	Vj	Vk	Qj	Qk	Code:
0	Add1	No					LD F0 0 R1
0	Add2	No					MULT 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		F0	F2	F4	F6	F8	F10	F12	..	F30
Clock	R1									
11	64	Qi	Load3	Mult2						

Loop Example Cycle 12

Instruction status				Execution				Write Back			
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complete	Result	Busy	Address			
LD	F0	0 R1	1	1	9	10	Load1	No			
MULT	F4	F0 F2	1	2			Load2	No			
SD	F4	0 R1	1	3			Load3	Yes	64	Q1	
LD	F0	0 R1	2	6	10	11	Store1	Yes	80	Mult1	
MULT	F4	F0 F2	2	7			Store2	Yes	72	Mult2	
SD	F4	0 R1	2	8			Store3	No			
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	Code:			
0	Add1	No						LD	F0	0 R1	
0	Add2	No						MULT	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	F10	F12	..	F30
12	64	Q1	Load3		Mult2						

Loop Example Cycle 13

Instruction status				Execution				Write Back			
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complete	Result	Busy	Address			
LD	F0	0 R1	1	1	9	10	Load1	No			
MULT	F4	F0 F2	1	2			Load2	No			
SD	F4	0 R1	1	3			Load3	Yes	64	Q1	
LD	F0	0 R1	2	6	10	11	Store1	Yes	80	Mult1	
MULT	F4	F0 F2	2	7			Store2	Yes	72	Mult2	
SD	F4	0 R1	2	8			Store3	No			
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for j</i>	<i>RS for k</i>				
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	Code:			
0	Add1	No						LD	F0	0 R1	
0	Add2	No						MULT	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	F10	F12	..	F30
13	64	Q1	Load3		Mult2						

Loop Example Cycle 14

Instruction status				Execution			Write					
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complet	Result	Busy	Address				
LD	F0	0 R1	1	1	9	10	Load1	No				
MULT	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		
MULT	F4	F0 F2	2	7			Store2	Yes	72	Mult2		
SD	F4	0 R1	2	8			Store3	No				
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for</i>		<i>RS for</i>		<i>k</i>		
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	Code:				
0	Add1	No						LD	F0	0 R1		
0	Add2	No						MULT	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	F10	F12	..	F30
14	64	Qi		Load3		Mult2						

- Mult1 completing; what is waiting for it?

Loop Example Cycle 15

Instruction status				Execution			Write					
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complet	Result	Busy	Address				
LD	F0	0 R1	1	1	9	10	Load1	No				
MULT	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(
MULT	F4	F0 F2	2	7	15		Store2	Yes	72	Mult2		
SD	F4	0 R1	2	8			Store3	No				
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for</i>		<i>RS for</i>		<i>k</i>		
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	Code:				
0	Add1	No						LD	F0	0 R1		
0	Add2	No						MULT	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	F10	F12	..	F30
15	64	Qi		Load3		Mult2						

- Mult2 completing; what is waiting for it?

Loop Example Cycle 16

Instruction status				Execution			Write					
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complet	Result	Busy	Address				
LD	F0	0 R1	1	1	9	10	Load1	No				
MULTI	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				<i>S1</i>	<i>S2</i>	<i>RS for</i>		<i>RS for</i>		<i>k</i>		
	<i>Time Name</i>	<i>Busy Op</i>		<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	<i>Code:</i>				
	0 Add1	No						LD	F0	0 R1		
	0 Add2	No						MULTI	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		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>..F30</i>		
16	64	Qi	Load3		Mult1							

Loop Example Cycle 17

Instruction status				Execution			Write					
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complet	Result	Busy	Address				
LD	F0	0 R1	1	1	9	10	Load1	No				
MULTI	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				<i>S1</i>	<i>S2</i>	<i>RS for</i>		<i>RS for</i>		<i>k</i>		
	<i>Time Name</i>	<i>Busy Op</i>		<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	<i>Code:</i>				
	0 Add1	No						LD	F0	0 R1		
	0 Add2	No						MULTI	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		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>..F30</i>		
17	64	Qi	Load3		Mult1							

Loop Example Cycle 18

Instruction status				Execution			Write					
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complet	Result	Busy	Address				
LD	F0	0 R1	1	1	9	10	Load1	No				
MULTI	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				<i>S1</i>	<i>S2</i>	<i>RS for</i>		<i>RS for</i>		<i>k</i>		
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	Code:				
0	Add1	No						LD	F0	0 R1		
0	Add2	No						MULTI	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		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>.. F30</i>		
18	56	Qi	Load3		Mult1							

Loop Example Cycle 19

Instruction status				Execution			Write					
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complet	Result	Busy	Address				
LD	F0	0 R1	1	1	9	10	Load1	No				
MULTI	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				
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				<i>S1</i>	<i>S2</i>	<i>RS for</i>		<i>RS for</i>		<i>k</i>		
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	Code:				
0	Add1	No						LD	F0	0 R1		
0	Add2	No						MULTI	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		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>.. F30</i>		
19	56	Qi	Load3		Mult1							

Loop Example Cycle 20

Instruction status				Execution				Write			
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complet	Result	Busy	Address			
LD	F0	0 R1	1	1	9	10	Load1	No			
MULTI	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			
MULTI	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				<i>S1</i>	<i>S2</i>	<i>RS for</i>		<i>RS for</i>		<i>k</i>	
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	Code:			
0	Add1	No						LD	F0	0 R1	
0	Add2	No						MULTI	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		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	.. <i>F30</i>	
20	56	Qi	Load3		Mult1						

Loop Example Cycle 21

Instruction status				Execution				Write			
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complet	Result	Busy	Address			
LD	F0	0 R1	1	1	9	10	Load1	No			
MULTI	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			
MULTI	F4	F0 F2	2	7	15	16	Store2	No			
SD	F4	0 R1	2	8	20	21	Store3	Yes	64	Mult1	
Reservation Stations				<i>S1</i>	<i>S2</i>	<i>RS for</i>		<i>RS for</i>		<i>k</i>	
Time	Name	Busy	Op	<i>Vj</i>	<i>Vk</i>	<i>Qj</i>	<i>Qk</i>	Code:			
0	Add1	No						LD	F0	0 R1	
0	Add2	No						MULTI	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		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	.. <i>F30</i>	
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**