ILP and Tomasulo's Algorithm

David Andrews

Slides From D. Patterson

Inter-instruction Dependences

- *Data dependence*

  $r_3 \leftarrow r_1 \text{ op } r_2$
  $r_5 \leftarrow r_3 \text{ op } r_4$

  Read-after-Write (RAW)

- *Anti-dependence*

  $r_3 \leftarrow r_1 \text{ op } r_2$
  $r_1 \leftarrow r_4 \text{ op } r_5$

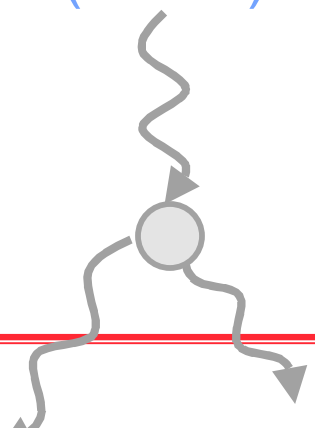
  Write-after-Read (WAR)

- *Output dependence*

  $r_3 \leftarrow r_1 \text{ op } r_2$
  $r_5 \leftarrow r_3 \text{ op } r_4$
  $r_3 \leftarrow r_6 \text{ op } r_7$

  Write-after-Write (WAW)

- Control dependence



ILP: Instruction-Level Parallelism

- ILP is a measure of the amount of inter-dependencies between instructions:
- Average ILP = no. instruction / no. cyc required

code1: ILP = 1

i.e. must execute serially

code2: ILP = 3

i.e. can execute at the same time

```
code1:   r1 ← r2 + 1
         r3 ← r1 / 17
         r4 ← r0 - r3
```

```
code2:   r1 ← r2 + 1
         r3 ← r9 / 17
         r4 ← r0 - r10
```

Scope of ILP Analysis

ILP=1

$$\left\{\begin{array}{l} r1 \Leftarrow r2 + 1 \\ r3 \Leftarrow r1 / 17 \\ r4 \Leftarrow r0 - r3 \end{array}\right.$$

$$\left.\begin{array}{l} r11 \Leftarrow r12 + 1 \\ r13 \Leftarrow r19 / 17 \\ r14 \Leftarrow r0 - r20 \end{array}\right\}$$

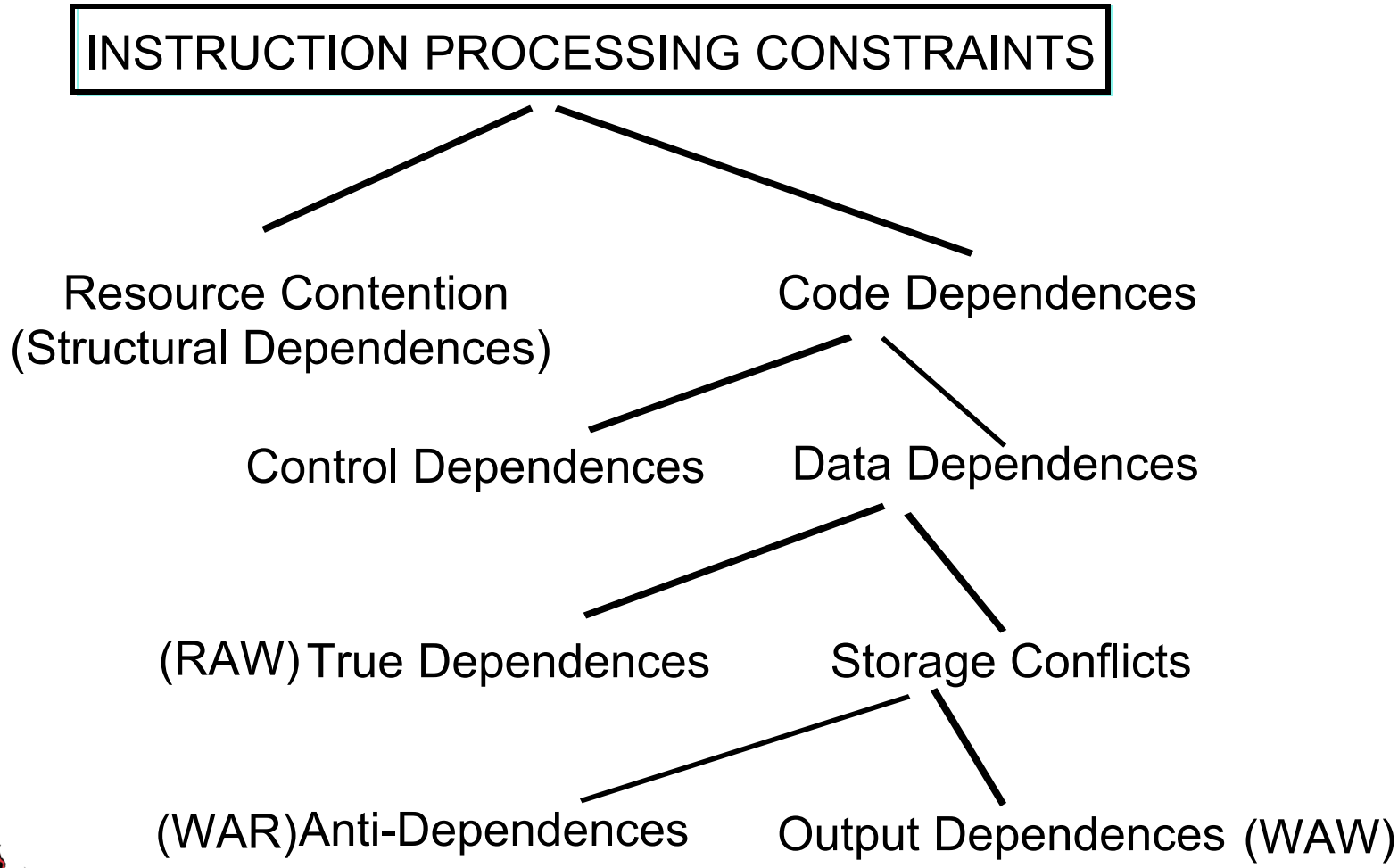
ILP=2

Out-of-order execution permits more ILP to be exploited

Purported Limits on ILP

Weiss and Smith [1984]	1.58
Sohi and Vajapeyam [1987]	1.81
Tjaden and Flynn [1970]	1.86
Tjaden and Flynn [1973]	1.96
Uht [1986]	2.00
Smith et al. [1989]	2.00
Jouppi and Wall [1988]	2.40
Johnson [1991]	2.50
Acosta et al. [1986]	2.79
Wedig [1982]	3.00
Butler et al. [1991]	5.8
Melvin and Patt [1991]	6
Wall [1991]	7
Kuck et al. [1972]	8
Riseman and Foster [1972]	51
Nicolau and Fisher [1984]	90

Superscalar Execution Check List



Resolving False Dependences

(1) R4 ← R3 + 1

(2) R3 ← R5 + 1

Must Prevent (2) from completing before (1) is dispatched

(1) R3 ← R3 op R5

(2) R3 ← R5 + 1

Must Prevent (2) from completing before (1) completes

Stalling: delay Dispatching (or write back) of the 2nd instruction

Copy Operands: Copy not-yet-used operand to prevent being overwritten (WAR)

Register Renaming: use a different register (WAW & WAR)

Register Renaming

- Anti and output dependencies are false dependencies

$$r_3 \leftarrow r_1 \text{ op } r_2$$
$$r_5 \leftarrow r_3 \text{ op } r_4$$
$$r_3 \leftarrow r_6 \text{ op } r_7$$

- The dependence is on name/location rather than data
- Given infinite number of registers, anti and output dependencies can always be eliminated

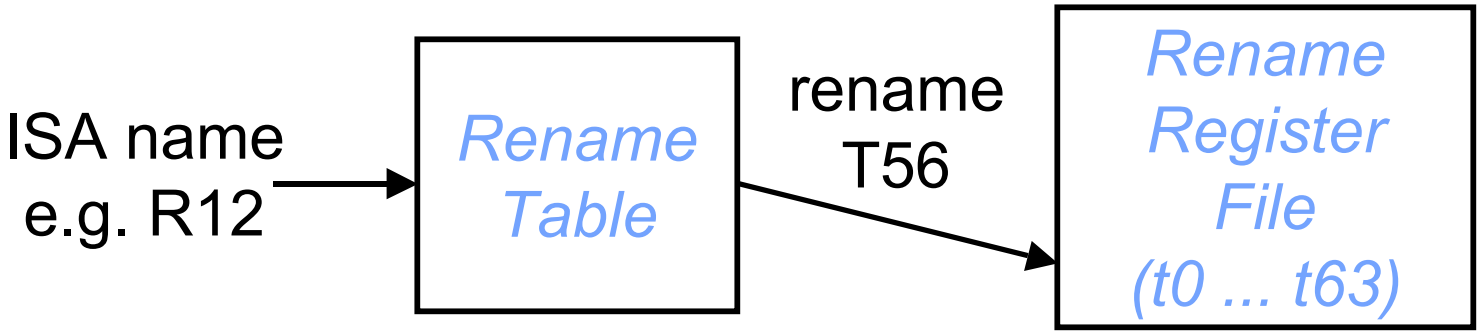
Original

r1 ← r2 / r3
r4 ← r1 * r5
r1 ← r3 + r6
r3 ← r1 - r4

Renamed

r1 ← r2 / r3
r4 ← r1 * r5
r8 ← r3 + r6
r9 ← r8 - r4

Hardware Register Renaming



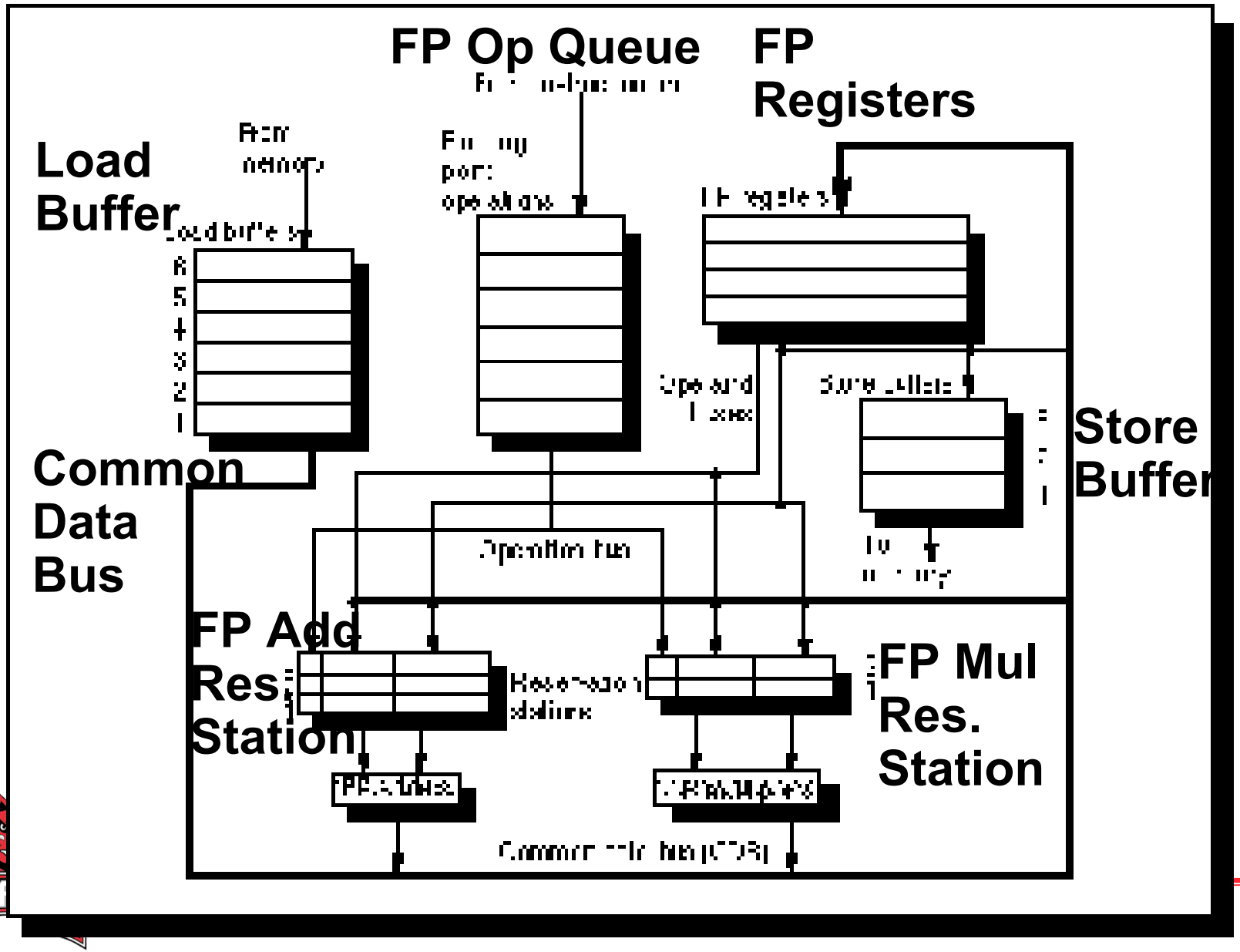
- maintain bindings from ISA reg. names to rename registers
- When issuing an instruction that updates 'RD':
  - allocate an unused rename register TX
  - recording binding from 'RD' to TX
- When to remove a binding? When to de-allocate a rename register?

R1 ← R2 / R3		R1 ← R2 / R3
R4 ← R1 * R5	→	R4 ← R1 * R5
R1 ← R3 + R6		R11 ← R3 + R6

History of Tomasulo's Algorithm

- For IBM 360/91 in 1966
- Goal: High Performance without special compilers
  - IBM has only 2 register specifiers/instr
  - IBM has 4 FP registers
- Why Study? lead to Alpha 21264, HP 8000, MIPS 10000, Pentium II, PowerPC 604, …

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 bus

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

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	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
LD	F6	34+	R2	1		
LD	F2	45+	R3			
MULT	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 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					

Tomasulo Example Cycle 2

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1		
LD	F2	45+	R3	2		
MULT	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 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: 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		
MULT 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 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		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	4			Load2	Yes	45+R3		
MULT	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2	4								
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									

Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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					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			
MULT	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2	4								
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2									
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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					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			
MULT	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2	4								
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6								
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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			

Tomasulo Example Cycle 7

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			
MULT	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2	4	7							
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6								
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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												
Instruction		*j*	*k*	*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			
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					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	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				*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			
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					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*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			
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							
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*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			
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					*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									
	4	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*
11			*FU*	Mult1	M(45+R3)		(M-M)+M()	M()–M()	Mult2			

- **Write result of ADDD here**

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			
MULT	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*	*Busy*	*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					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			
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					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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					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			
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					*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									
	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					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			
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					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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					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			
MULT	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					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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					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			
MULT	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					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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					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			
MULT	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					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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					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			
MULT	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					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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**

Cycle 62

Instruction status

Instruction		j	k	Issue	Read operanc	Executi comple	Write Result
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
ADDD	F6	F8	F2	13	14	16	22

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j 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									

Tomasulo IBM 360 Summary

Pipelined Functional Units

(6 load, 3 store, 3 +, 2 x/÷)

Window size: ≤ 14 instructions

No issue on structural hazard

WAR: renaming avoids

WAW: renaming avoids

Broadcast results from FU

Control: reservation stations

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

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1			
MULT F4	F0	F2	1			
SD F4	0	R1	1			
LD F0	0	R1	2			
MULT 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

Time	Name	Busy	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					

Code:

```
LD    F0   0  R1
MULT  F4   F0 F2
SD    F4   0  R1
SUBI  R1   R1 #8
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

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1			
SD F4	0	R1	1			
LD F0	0	R1	2			
MULT 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

Time	Name	Busy	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					

Code:

```
LD    F0   0  R1
MULT  F4   F0 F2
SD    F4   0  R1
SUBI  R1   R1 #8
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

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1	2		
SD F4	0	R1	1			
LD F0	0	R1	2			
MULT 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

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

Code:

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

Register result status

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

Loop Example Cycle 3

Instruction status

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

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

Code:

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

Register result status

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

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

Loop Example Cycle 4

Instruction status

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

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

Code:

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

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
4	72	Qi	Load1		Mult1					

Loop Example Cycle 5

Instruction status

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

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

Code:

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

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
5	72	Qi	Load1		Mult1					

Loop Example Cycle 6

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write 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

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

Code:

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

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

Instruction		j	k	iteration	Issue	Execution complete	Write 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

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

Code:

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

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

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1		
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6		
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	Qi
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

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

Code:

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

Register result status

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

Loop Example Cycle 9

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6		
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	Qi
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

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

Code:

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

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
9	64	Qi	Load2		Mult2					

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

Loop Example Cycle 10

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
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

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
4	Mult1	Yes	MULTD	M(80)	R(F2)		
0	Mult2	Yes	MULTD		R(F2)	Load2	

Code:

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

Register result status

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

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

Loop Example Cycle 11

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write 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	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

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
3	Mult1	Yes	MULTD	M(80)	R(F2)		
4	Mult2	Yes	MULTD	M(72)	R(F2)		

Code:

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

Register result status

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

Loop Example Cycle 12

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write 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	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

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
2	Mult1	Yes	MULTD	M(80)	R(F2)		
3	Mult2	Yes	MULTD	M(72)	R(F2)		

Code:

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

Register result status

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

Loop Example Cycle 13

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write 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	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

Time	Name	Busy	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(80)	R(F2)		
2	Mult2	Yes	MULTD	M(72)	R(F2)		

Code:

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

Register result status

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

Loop Example Cycle 14

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write 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

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD	M(80)	R(F2)		
1	Mult2	Yes	MULTD	M(72)	R(F2)		

Code:

LD	F0	0	R1
MULT	F4	F0	F2
SD	F4	0	R1
SUBI	R1	R1	#8
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

Instruction		j	k	iteration	Issue	Execution complete	Write 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

Time	Name	Busy	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	Yes	MULTD	M(72)	R(F2)		

Code:

LD	F0	0	R1
MULT	F4	F0	F2
SD	F4	0	R1
SUBI	R1	R1	#8
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

Instruction		j	k	iteration	Issue	Execution complete	Write 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	16	Store2	Yes	72	M(72)*R(
SD	F4	0	R1	2	8			Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load3	
0	Mult2	No					

Code:

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

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12	...	F30
16	64	Qi	Load3		Mult1						

Loop Example Cycle 17

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write 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	16	Store2	Yes	72	M(72)*R(
SD	F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

	Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
	0	Add1	No					
	0	Add2	No					
	0	Add3	No					
	0	Mult1	Yes	MULTD		R(F2)	Load3	
	0	Mult2	No					

Code:

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

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12	...	F30
17	64	Qi	Load3		Mult1						

Loop Example Cycle 18

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write 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	18		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	16	Store2	Yes	72	M(72)*R(
SD	F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

	Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
	0	Add1	No					
	0	Add2	No					
	0	Add3	No					
	0	Mult1	Yes	MULTD		R(F2)	Load3	
	0	Mult2	No					

Code:

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

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12	...	F30
18	56	Qi	Load3		Mult1						

Loop Example Cycle 19

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write 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	18	19	Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	No		
MULT	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

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load3	
0	Mult2	No					

Code:

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

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12	...	F30
19	56	Qi	Load3		Mult1						

Loop Example Cycle 20

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write 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	18	19	Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	No		
MULT	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

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load3	
0	Mult2	No					

Code:

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

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12	...	F30
20	56	Qi	Load3		Mult1						

Loop Example Cycle 21

Instruction status												
Instruction		*j*	*k*	*iteration*	*Issue*	*Execution complete*	*Write 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	18	19	Load3	Yes	64	Qi	
LD	F0	0	R1	2	6	10	11	Store1	No			
MULT	F4	F0	F2	2	7	15	16	Store2	No			
SD	F4	0	R1	2	8	20	21	Store3	Yes	64	Mult1	

Reservation Stations				*S1*	*S2*	*RS for j*	*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
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*	*F10*	*F12*	...	*F30*
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