#### CSCE 4213 Computer Architecture

# ILP and Tomasulo's Algorithm

David Andrews
Slides From D. Patterson



#### Inter-instruction Dependences

Data dependence

$$r_3 \leftarrow r_1$$
 op  $r_2$  Read-after-Write  $r_5 \leftarrow r_3$  op  $r_4$  (RAW)

Anti-dependence

$$r_3 \leftarrow r_1$$
 op  $r_2$  Write-after-Read  $r_1 \leftarrow r_4$  op  $r_5$  (WAR)

Output dependence

$$r_3 \leftarrow r_1 \text{ op } r_2$$
 Write-after-Write  $r_5 \leftarrow r_3 \text{ op } r_4$  (WAW)  $r_3 \leftarrow r_6 \text{ op } r_7$ 

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: 
$$r1 \leftarrow r2 + 1$$

$$r4 \leftarrow r0 - r3$$

code2: 
$$r1 \leftarrow r2 + 1$$

$$r3 \leftarrow r9 / 17$$

$$r4 \leftarrow r0 - r10$$

#### Scope of ILP Analysis

$$|LP=1| \left\{ \begin{array}{c} r1 \Leftarrow r2 + 1 \\ r3 \Leftarrow r1 / 17 \\ r4 \Leftarrow r0 - r3 \\ r11 \iff r12 + 1 \\ r13 \iff r19 / 17 \\ r14 \iff r0 - r20 \end{array} \right\}$$

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

INSTRUCTION PROCESSING CONSTRAINTS Code Dependences Resource Contention (Structural Dependences) Data Dependences **Control Dependences** (RAW) True Dependences **Storage Conflicts** (WAR)Anti-Dependences Output Dependences (WAW)

#### Resolving False Dependences

(1) R4 - R3 + 1

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

(2) R3 R5 + 1

(1) R3— R3 op R5

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

: **◄**- R3 :

(2) R3**◄-** R5 + 1

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

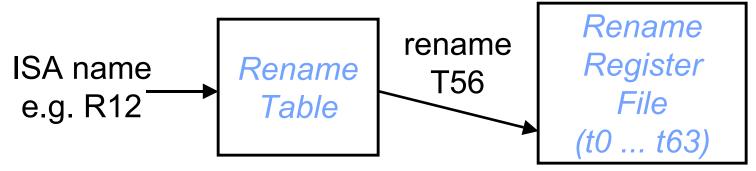
$$r1 \leftarrow r3 + r6$$

#### Renamed

$$r1 \leftarrow r2 / r3$$



#### 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?

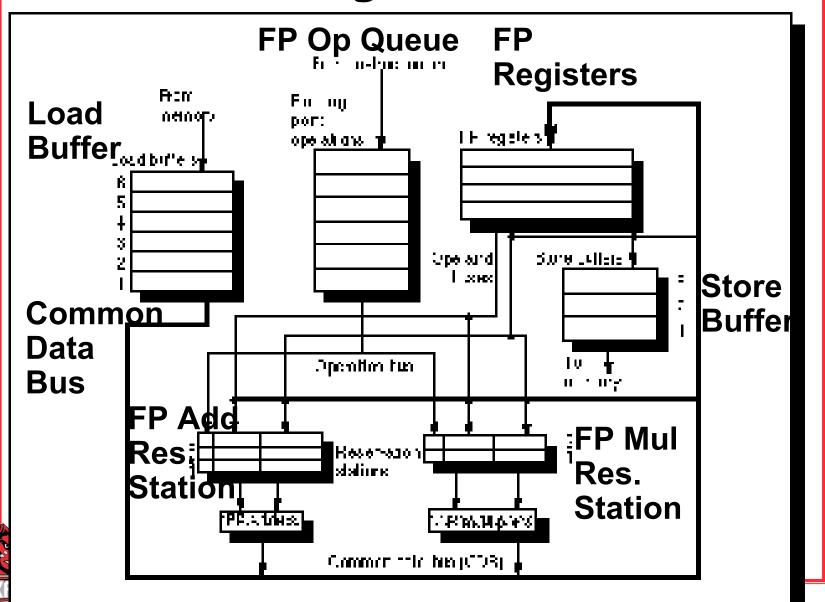
R1 
$$\leftarrow$$
 R2 / R3  
R4  $\leftarrow$  R1 \* R5  
R1  $\leftarrow$  R1 \* R5  
R1  $\leftarrow$  R3 + R6  
R1  $\leftarrow$  R3 + R6

#### 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

nstru	ction s	status_			Execution	Write						
nstru	ction	j	k	Issue	complete	Result			Busy	Addre	SS	
LD	F6	34+	R2					Load1	No			
LD	F2	45+	R3					Load2	No			
MULT	FO	F2	F4					Load3	No			
SUBD	F8	F6	F2									
DIVD	F10	FO	F6									
ADDD	)F6	F8	F2									
Reser	vation	Station	<u>1S</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for I	(			
	Time	Name	Busy	Ор	Vj	Vk	Qj	Qk				
	0	Add1	No									
	0	Add2	No									
	0	Add3	No									
	0	Mult1	No									
	0	Mult2	No									
Regis <sup>.</sup>	ter res	ult stat	tus									
Cloc	k			F0	F2	F4	F6	F8	F10	F12		F30
0			FU									

```
Execution
                                          Write
Instruction status
Instruction
                                                                                Address
                   k
                       Issue
                              complete
                                          Result
                                                                         Busy
     F6
           34+
                  R2
                                                                                34+R2
LD
                                                                 Load1
                                                                          Yes
           45+
                  R3
                                                                 Load2
LD
                                                                         No
MULIT FO
           F2
                  F4
                                                                 Load3
                                                                         No
SUBID F8
           F6
                  F2
                  F6
DIVID F10
           F<sub>0</sub>
ADDDF6
           F8
                  F2
                                          S2
                                                                 RS for k
Reservation Stations
                              S1
                                                     RS for j
      Time Name Busy Op
                              Vi
                                          Vk
                                                                 Qk
                                                      Qj
         0 Add1
                  No
         0 Add2
                  No
           Add3
                  No
         0 Mult1
                  No
         0 Mult2
                 No
Register result status
Clock
                                                                         F10 F12 ...
                       F0
                              F2
                                          F4
                                                      F6
                                                                 F8
                                                                                             F30
                  FU
                                                      Load1
```

```
Write
Instruction status
                              Execution
Instruction
                   k
                       Issue
                              complete
                                          Result
                                                                        Busy
                                                                                Address
     F6
           34 +
                  R2
                                                                                34+R2
LD
                                                                 Load1
                                                                         Yes
           45+
                  R3
                         2
                                                                 Load2
                                                                                45+R3
     F2
                                                                         Yes
LD
                  F4
                                                                 Load3
MULIT FO
           F2
                                                                        No
SUBID F8
           F6
                  F2
                  F6
DIVID F10
           FO
ADDDF6
           F8
                  F2
Reservation Stations
                                          S2
                              S1
                                                     RS for i
                                                                 RS for k
     Time Name Busy Op
                              Vi
                                          Vk
                                                                 Ok
                                                     Qi
         0 Add1
                  No
         0 Add2
                  No
           Add3
                  No
         0 Mult1
                  No
         0 Mult2
                 No
Register result status
                              F2
Clock
                       F0
                                                     F6
                                                                 F8
                                                                         F10 F12 ...
                                                                                             F30
                                          F4
                  FU
                              Load2
                                                     Load1
```

Note: Can have multiple loads outstanding

```
Instruction status
                            Execution
                                       Write
Instruction
                  k
                            complete
                                       Result
                                                                    Busy
                                                                          Address
                     Issue
LD F6
                                                                          34+R2
          34+
                 R2
                                                            Load1
                                                                    Yes
        45+
                 R3
LD
                        2
                                                            Load2
                                                                    Yes
                                                                          45+R3
MULT FO
         F2
                F4
                                                            Load3
                                                                   No
SUBD F8
         F6
                F2
DIVID F10
         FO
                F6
ADDDF6
                F2
Reservation Stations
                            S1
                                       S2
                                                 RS for i
                                                            RS for k
     Time Name Busy Op
                                       Vk
                                                            Ok
         0 Add1
                No
        0 Add2 No
          Add3 No
        0 Mult1 Yes MULTD
                                       R(F4)
                                                 Load2
        0 Mult2 No
Register result status
Clock
                     FO.
                            F2
                                       F4
                                                 F6
                                                            F8
                                                                    F10 F12 ...
                                                                                      F30
                     Mult1 Load2
                                                 Load1
```

Note: registers names are removed ("renamed") in Reservation Stations; MULT issued vs. scoreboard

.oad1 completing; what is waiting for Load1?

Computer System Design Lab

<u>Instr</u>	uction s	<u>status</u>			Execution	Write							
Instr	uction	j	k	Issue	complete	Result			Busy	Addre	ss		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4			Load2	Yes	45+R3			
MUL		F2	F4	3				Load3	No				
SUBI	) F8	F6	F2	4									
DIVI	F10	FO	F6										
ADD	DF6	F8	F2										
Rese	<u>rvation</u>	Station	<u>1S</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for I	(				
	Time	Name	Busy	Ор	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										
<u>Regi</u>	ster res	<u>sult stat</u>	<u>us</u>										
Clo	ck			FO	F2	F4	F6	F8	F10	F12		F3	D
4	4		FU	Mult1	Load2		M(34+R2)	Add1					

Load2 completing; what is waiting for it?

<u>Instr</u>	uction s	status			Execution	Write							ĺ
Instr	uction	j	k	Issue	complete	Result			Busy	Addre	ss		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3				Load3	No				
SUBI	) F8	F6	F2	4									
DIVE	F10	FO	F6	5									
ADD	DF6	F8	F2										
Rese	<u>rvation</u>	Station	<u>1S</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for k					
	Time	Name	Busy	Ор	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						
Regi	ster res	ult stat	<u>us</u>										
Clo	ck			F0	F2	F4	F6	F8	F10	F12		F3	þ
	5		FU	Mult1	M(45+R3)		M(34+R2)	Add1	Mult2				

<u>Instr</u>	uction :	<u>status</u>			Execution	Write							
Instr	uction	j	k	Issue	complete	Result			Busy	Addre	SS		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	T FO	F2	F4	3				Load3	No				
SUBI	) F8	F6	F2	4									
DIV	F10	FO	F6	5									
ADD	DF6	F8	F2	6									
Rese	rvation	Station	<u>1S</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for k	(				
	Time	Name	Busy	Ор	Vj	Vk	Qj	Qk					
	1	Add1	Yes	SUBD	M(34+R2)	M(45+R3)							
	0	Add2	Yes	<b>ADDD</b>		M(45+R3)	Add1						
		Add3	No										
	9	Mult1	Yes	MULTD	M(45+R3)	R(F4)							
	0	Mult2	Yes	DIVD		M(34+R2)	Mult1						
Regi	ster res	sult stat	tus										
Clo	ck			FO	F2	F4	F6	F8	F10	F12		F3	þ
Ć	5		FU	Mult1	M(45+R3)		Add2	Add1	Mult2				I

<u>Instr</u>	uction s	<u>status</u>			Execution	Write							
Instr	uction	j	k	Issue	complete	Result			Busy	Addre	SS		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3				Load3	No				
SUBI	) F8	F6	F2	4	7								
DIVI	F10	FO	F6	5									
ADD	DF6	F8	F2	6									
Rese	<u>rvation</u>	Station	<u>าร</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for I	(				
	Time	Name	Busy	Ор	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						
<u>Regi</u>	ster res	ult stat	<u>tus</u>										
Clo	ck			FO	F2	F4	F6	F8	F10	F12		F3	p
7	7		FU	Mult1	M(45+R3)		Add2	Add1	Mult2				

Add1 completing; what is waiting for it?

Instru	iction st	atus_			Execution	Write						
Instr	ıction	j	k	Issue	complete	Result			Busy	Addres	s	
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	4	5		Load2	No			
MUL	ΓF0	F2	F4	3				Load3	No			
SUB	D F8	F6	F2	4	7	8						
DIV	F10	F0	F6	5								
ADD		F8	F2	6								
Rese	rvation :	<u>Stations</u>			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					
Regi	<u>ster resu</u>	<u>ılt status</u>										
Clo	ck			F0	F2	F4	F6	F8	F10	F12		F30
8	3		FU	Mult1	M(45+R3)		Add2	M()-M()	Mult2			

<u>Instr</u>	uction :	<u>status</u>			Execution	Write							
Instr	uction	j	k	Issue	complete	Result			Busy	Addre	SS		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3				Load3	No				
SUBI	) F8	F6	F2	4	7	8							
DIV	F10	FO	F6	5									
ADD	DF6	F8	F2	6									
Rese	rvation	Station	<u>1S</u>		<i>S1</i>	<i>S2</i>	RS for j	RS for k	(				
	Time	Name	Busy	Ор	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						
Regi	ter res	sult stat	tus										
Clo	ck			F0	F2	F4	F6	F8	F10	F12		F3	C
9	)		FU	Mult1	M(45+R3)		Add2	M()-M()	Mult2				I

<u>Instr</u>	uction s	status_			Execution	Write							
Instr	uction	j	k	Issue	complete	Result			Busy	Addre	SS		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3				Load3	No				
SUBI	) F8	F6	F2	4	7	8							
DIVE	F10	FO	F6	5									
ADD	DF6	F8	F2	6	10								
Rese	<u>rvation</u>	Station	<u>1S</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for k	(				
	Time	Name	Busy	Ор	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						
Regi	ster res	ult stat	us										
Clo	ck			F0	F2	F4	F6	F8	F10	F12		F3	C
1 (	)		FU	Mult1	M(45+R3)		Add2	M()-M()	Mult2				Ī

Add2 completing; what is waiting for it?

Instr	iction st	atus_			Execution	Write							
Instr	ection	j	k	Issue	complete	Result			Busy	Addres	S		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL		F2	F4	3				Load3	No				
SUB		F6	F2	4	7	8							
DIV		F0	F6	5									
ADD	DF6	F8	F2	6	10	11							
Rese	rvation	<u>Stations</u>			S1	S2	RS for j	RS for k					
	Time	Name	Busy	Ор	Vj	Vk	Qj	Qk					
	0	Add1	No										
	0	Add2	No										
	0	Add3	No										
	4	Mult1	Yes	MULTD	M(45+R3)	R(F4)							
	0	Mult2		DIVD		M(34+R2)	Mult1						
Regi	ster resu												
Clo				F0	F2	F4	F6	F8	F10	F12		F30	)
1	1		FU	Mult1	M(45+R3)		(M-M)+M()	M()–M()	Mult2				
					,		, ,						

#### Write result of ADDD here

<u>Instr</u>	uction s	status_			Execution	Write							
Instr	uction	j	k	Issue	complete	Result			Busy	Addre	SS		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3				Load3	No				
SUB	) F8	F6	F2	4	6	7							
DIV	F10	FO	F6	5									
ADD	DF6	F8	F2	6	10	11							
Rese	rvation	Station	<u>1S</u>		<i>S1</i>	<i>S2</i>	RS for j	RS for k	-				
	Time	Name	Busy	Ор	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						
Regi	ster res	ult stat	<u>us</u>										
Clo	ck			FO	F2	F4	F6	F8	F10	F12		F3	D
1 2	2		FU	Mult1	M(45+R3)		(M-M)+M()	M()-M()	Mult2				

Note: all quick instructions complete already

uction s	status_			Execution	Write							
uction	j	k	Issue	complete	Result			Busy	Addre	SS		
F6	34+	R2	1	3	4		Load1	No				
F2	45+	R3	2	4	5		Load2	No				
ΓFO	F2	F4	3				Load3	No				
) F8	F6	F2	4	7	8							
F10	FO	F6	5									
DF6	F8	F2	6	10	11							
rvation	Station	<u>1S</u>		<i>S1</i>	<i>S2</i>	RS for j	RS for k	•				
Time	Name	Busy	Ор	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						
ter res	ult stat	<u>us</u>										
ck			F0	F2	F4	F6	F8	F10	F12		F3	Ø
3		FU	Mult1	M(45+R3)		(M-M)+M()	M()-M()	Mult2				Ī
	F6 F2 F0 F8 F10 DF6 rvation Time 0 0 ster res	F6 34+ F2 45+ F0 F2 F8 F6 F10 F0 DF6 F8 rvation Station Time Name 0 Add1 0 Add2 Add3 2 Mult1 0 Mult2 ster result stat	R	Rection   j   k   Issue   F6   34+ R2   1   F2   45+ R3   2   F0   F2   F4   3   3   5   F8   F6   F2   4   5   5   5   5   5   5   5   5   5	Rection   j   k   Issue   complete   F6   34+   R2   1   3   3   52   4   4   54   54   54   54   54	Result   F6   34+   R2   1   3   4   5   5   7   7   8   7   7   8   7   7   8   7   7	Station   j   k   Issue   complete   Result	Ster result status   Complete   Result   Resul	Ster result status	Ster result status   Complete   Result   Resul	Ster result status   Ster   Ster	Success   Succ

<u>Instr</u>	uction s	<u>status</u>			Execution	Write							
Instr	uction	j	k	Issue	complete	Result			Busy	Addre	SS		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3				Load3	No				
SUBI	) F8	F6	F2	4	7	8							
DIVE	F10	FO	F6	5									
ADD	DF6	F8	F2	6	10	11							
Rese	Reservation Station		<u>1S</u>		<i>S1</i>	<i>S2</i>	RS for j	RS for k	(				
	Time	Name	Busy	Ор	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						
Regi	ster res	ult stat	tus										
Clo	ck			FO	F2	F4	F6	F8	F10	F12		F3	} (
1 4	4		FU	Mult1	M(45+R3)		(M-M)+M()	M()-M()	Mult2				1

Instr	uction s	status			Execution	Write							
Instr	uction	j	k	Issue	complete	Result			Busy	Addres	SS		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3	15			Load3	No				
SUBI	) F8	F6	F2	4	7	8							
DIVE	F10	FO	F6	5									
ADD	DF6	F8	F2	6	10	11							
Rese	eservation Statio		<u>1S</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for k					
	Time	Name	Busy	Ор	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						
Regi	ster res	ult stat	us										
Clo	ck			FO	F2	F4	F6	F8	F10	F12		F3(	þ
1 .	5		FU	Mult1	M(45+R3)		(M-M)+M()	M()-M()	Mult2				

Mult1 completing; what is waiting for it?

<u>Instr</u>	uction s	<u>status</u>			Execution	Write							
Instr	uction	j	k	Issue	complete	Result			Busy	Addre	SS		
LD	F6	34+	R2	1	3	3 4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3	15	16		Load3	No				
SUB	) F8	F6	F2	4	7	8							
DIVE	F10	FO	F6	5									
ADD	DF6	F8	F2	6	10	11							
Reservation S		Station	<u>1S</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for k					
	Time Name Busy (		Ор	Vj	Vk	Qj	Qk						
	0	Add1	No										
	0	Add2	No										
		Add3	No										
	0	Mult1	No										
	40	Mult2	Yes	DIVD	M*F4	M(34+R2)							
Regi	ster res	ult stat	tus										
Clo	ck			F0	F2	F4	F6	F8	F10	F12		F3	; (
1 (	5		FU	M*F4	M(45+R3)		(M-M)+M()	M()-M()	Mult2				Ī

Note: Just waiting for divide

<u>Instr</u>	uction :	<u>status</u>			Execution	Write							
Instr	str <mark>uction <i>j k</i></mark>		Issue	complete	Result			Busy	Addre	SS			
LD	F6	34+	R2	1	3	3 4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3	15	16		Load3	No				
SUBI	) F8	F6	F2	4	7	8							
DIVE	F10	FO	F6	5									
ADD	DF6	F8	F2	6	10	11							
Rese	rvation	Station	<u>1S</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for k					
	Time	Time Name Busy		Ор	Vj	Vk	Qj	Qk					
	0	Add1	No										
	0	Add2	No										
		Add3	No										
	0	Mult1	No										
	1	Mult2	Yes	DIVD	M*F4	M(34+R2)							
Regi	ster res	sult stat	<u>us</u>										
Clo	ck			FO	F2	F4	F6	F8	F10	F12		<i>F3</i>	p
5	5		FU	M*F4	M(45+R3)		(M-M)+M()	M()-M()	Mult2				Ī

<u>Instr</u>	uction s	status			Execution	Write							
Instr	uction			Issue	complete	Result			Busy	Addres	SS		
LD	F6	34+	R2	1	3	4		Load1	No				
LD	F2	45+	R3	2	4	5		Load2	No				
MUL	ΓFO	F2	F4	3	15	16		Load3	No				
SUBI	) F8	F6	F2	4	7	7 8							
DIVE	F10	FO	F6	5	56								
ADD	DF6	F8	F2	6	10	11							
Rese	rvation	Station	<u>1S</u>		<i>S</i> 1	<i>S2</i>	RS for j	RS for k	-				
	Time	Time Name Busy		Ор	Vj	Vk	Qj	Qk					
	0	Add1	No										
	0	Add2	No										
		Add3	No										
	0	Mult1	No										
	0	Mult2	Yes	DIVD	M*F4	M(34+R2)							
Regi	ster res	ult stat	<u>us</u>										
Clo	ck			FO	F2	F4	F6	F8	F10	F12		F3	þ
5 (	5		FU	M*F4	M(45+R3)		(M-M)+M()	M()-M()	Mult2				

Mult 2 completing; what is waiting for it?

<u>Instr</u>	uction s	status			Exe	cutio	7	Wri	te							
Instr	uction	j	k	Issue	con	nplete	è	Res	sult			Busy	Addre	SS		
LD	F6	34+	R2	1		3		4			Load1	No				
LD	F2	45+	R3	2		4			5		Load2	No				
MUL	ΓFO	F2	F4	3		15			16		Load3	No				
SUBI	) F8	F6	F2	4		7			8							
DIVE	F10	FO	F6	5		56			57							
ADD	DF6	F8	F2	6		10			<u> 11</u>							
Rese	Reservation Station		<u>1S</u>		<i>S1</i>			<i>S2</i>		RS for j	RS for k					
	Time	Name	Busy	Ор	)p Vj		Vk		Qj	Qk						
	0	Add1	No													
	0	Add2	No													
		Add3	No													
	0	Mult1	No													
	0	Mult2	No													
<u>Regi</u>	ster res	<u>ult stat</u>	<u>us</u>													
Clo	ck			FO	<i>F2</i>			F4	_	F6	F8	F10	F12		F3	0
5	7		FU	M*F4	M(4	5+R3	)			(M-M)+M()	M()-M()	M*F4/N	M			
Ï													_			

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

Computer System Design Lab

#### Cycle 62

```
Instruction status
                          Read
                                 Executi Write
Instruction
                    Issue operanceomple Result
                k
     F6
          34+ R2
                                    3
                                          4
LD
     F2
          45+ R3
LD
                      5
                                          8
                                         20
          F2
                      6
                                    19
MULT FO
               F4
                             9
          F6
               F2
SUBD F8
                                    11
                                         12
               F6
                            21
                                   61
                                         62
DIVD F10 F0
                      8
               F2
                      13
                                         22
ADDDF6
          F8
                                    16
                            14
                                 dest
                                         S1
Functional unit status
                                              S2
                                                  FU for | FU for k Fj?
                                                                       Fk?
     Time Name
                    Busy Op
                                 Fi
                                         Fi
                                             Fk
                                                                        Rk
                                                   Qi
                                                          Qk
                    No
          Integer
          Mult1
                    No
          Mult2
                    No
          Add
                    No
        0 Divide
                    No
Register result status
                          F2
                                        F6 F8 F10 F12
                                                                        F30
Clock
                                  F4
                    F0
               FU
```

#### Tomasulo IBM 360 Summary

Pipelined Functional Units

(6 load, 3 store,  $3 + 2 \times + 2 \times + 3 \times + 2 \times + 3 \times + 2 \times + 3 \times +$ 

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 Computer System Design Lab

<u>In</u>	struction	status				Execu	tioi Write					
In	struction	j	k	iteration	Issue	compl	et (Result	_	Busy	Add	ress	
L	D FO	0	R1	1				Load1	No			
М	ULT F4	FO	F2	1				Load2	No			
S	) F4	0	R1	1				Load3	No		Qi	
L	D FO	0	R1	2				Store1	No			
М	ULT F4	FO	F2	2				Store2	No			
S	D F4	0	R1	2				Store3	No			
<u>R</u>	<u>eservatio</u> i	<u>n Statio</u>	<u>ns</u>		<i>S1</i>	<i>S2</i>	RS for	RS for k				
	Time	<i>Name</i>	Busy	<sup>,</sup> Ор	Vj	Vk	Qj	Qk	Code:	•		
	0	Add1	No						LD	FO	0	R1
	0	Add2	No						MULT	F4	FO	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	No						SUBI	R1	R1	#8
	0	Mult2	No						BNEZ	R1	Loo	p
<u>R</u>	<u>egister re</u>	sult sta	<u>itus</u>									
	lock	R 1		F0	F2	F4	F6	F8	F10	F1.	2	F30
	0	80	Qi									

<u>In</u>	struction	status				Execution	o Write					
In	struction	j	k	iteration	Issue	complet	: Result	_	Busy	Addı	ress	
L	D FO	0	R1	1	1			Load1	Yes	80		
Μ	ULT F4	FO	F2	1				Load2	No			
S	) F4	0	R1	1				Load3	No		Qi	
L	D FO	0	R1	2				Store1	No			
Μ	ULT F4	FO	F2	2				Store2	No			
S	) F4	0	R1	2				Store3	No			
<u>R</u>	eservatio	<u>n Statio</u>	<u>ns</u>		<i>S1</i>	<i>S2</i>	RS for	RS for k	•			
	Time	<i>Name</i>	Busy	<sup>,</sup> Ор	Vj	Vk	Qj	Qk	Code:	•		
	0	Add1	No						LD	FO	0	R1
	0	Add2	No						MULT	F4	FO	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	No						SUBI	R1	R1	#8
	0	Mult2	No						BNEZ	R1	Loo	p
<u>R</u>	<u>egister re</u>	<u>sult sta</u>	<u>tus</u>									
C	lock	R 1		<i>F0</i>	F2	F4	F6	F8	F10	F12	2	F30
	1	80	Qi	Load1								
_												

Instruction s	status				Execution	o Write					
Instruction	j	k	iteration	Issue	complet	te Result		Busy	Add	ress	
LD FO	0	R1	1	1			Load1	Yes	80		
MULT F4	FO	F2	1	2			Load2	No			
SD F4	0	R1	1				Load3	No		Qi	
L <mark>D FO</mark>	0	R1	2				Store1	No			
MULT F4	FO	F2	2				Store2	No			
SD F4	0	R1	2				Store3	No			
<b>Reservation</b>	<u>Statio</u>	<u>ns</u>		<i>S1</i>	<i>S2</i>	RS for	,RS for k	<b>(</b>			
Time	Name	Busy	<sup>,</sup> Ор	Vj	Vk	Qj	Qk	Code	•		
0 /	Add1	No						LD	FO	0	R1
0 /	Add2	No						MULT	F4	FO	F2
0 /	Add3	No						SD	F4	0	R1
0 1	Mult1	Yes	MULTD		R(F2)	Load1		SUBI	R1	R1	#8
0 1	Mult2	No						BNEZ	R1	Loo	p
R <mark>egister res</mark>	<u>ult sta</u>	<u>itus</u>									
Clock	R1		F0	F2	F4	F6	F8	F10	F12	2	<i>F30</i>
2	80	Qi	Load1		Mult1	-	-	•			

Instruction	status				Execution	oı Write					
Instruction	j	k	iteration	Issue	complet	te Result	_	Busy	Addı	ress	
LD FO	0	R1	1	1			Load1	Yes	80		
MULT F4	FO	F2	1	2			Load2	No			
SD F4	0	R1	1	3			Load3	No		Qi	
LD FO	0	R1	2				Store1	Yes	80	Mult	1
MULT F4	FO	F2	2				Store2	No			
SD F4	0	R1	2				Store3	No			
<u>Reservatio</u>	<u>n Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	,RS for k	(			
Time	e Name	Busy	<sup>,</sup> Ор	Vj	Vk	Qj	Qk	Code:	•		
0	Add1	No						LD	FO	0	R1
0	Add2	No						<b>MULT</b>	F4	FO	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	0
R <mark>egister re</mark>	<u>sult sta</u>	itus									
Clock	R 1		FO	F2	F4	F6	F8	F10	F12	2	F30
3	80	Qi	Load1		Mult1						

Note: MULT1 has no registers names in RS

<u>Instruction s</u>	status				Execution	o Write					
Instruction	j	k	iteration	Issue	complet	te Result		Busy	Addı	ress	
LD FO	0	R1	1	1			Load1	Yes	80		
MULT F4	FO	F2	1	2			Load2	No			
SD F4	0	R1	1	3			Load3	No		Qi	
L <mark>D FO</mark>	0	R1	2				Store1	Yes	80	Mult	:1
MULT F4	FO	F2	2				Store2	No			
SD F4	0	R1	2				Store3	No			
<b>Reservation</b>	Statio	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k				
Time	Name	Busy	<sup>,</sup> Ор	Vj	Vk	Qj	Qk	Code.	•		
0 .	Add1	No						LD	FO	0	R1
0 .	Add2	No						MULT	F4	FO	F2
0 .	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI	R1	R1	#8
0	Mult2	No						BNEZ	R1	Loo	p
R <mark>egister res</mark>	<u>ult sta</u>	itus									
Clock	R1		F0	F2	F4	F6	F8	F10	F12	<u> </u>	F30
4	72	Qi	Load1		Mult1						

<u>In</u>	struction	<u>status</u>				Execution	o Write					
In	struction	j	k	iteration	Issue	complet	te Result	_	Busy	Add	ress	
L	) F0	0	R1	1	1			Load1	Yes	80		
Μ	ULT F4	FO	F2	1	2			Load2	No			
S	) F4	0	R1	1	3			Load3	No		Qi	
L	) F0	0	R1	2				Store1	Yes	80	Mult	1
Μ	ULT F4	FO	F2	2				Store2	No			
S	) F4	0	R1	2				Store3	No			
<u>R</u>	eservation	<u>Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k	•			
	Time	Name	Busy	′ Ор	Vj	Vk	Qj	Qk	Code:	•		
	0	Add1	No						LD	FO	0	R1
	0	Add2	No						MULT	F4	FO	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	)
R	<u>egister re</u> s	<u>sult sta</u>	tus									
C	lock	R 1		FO	F2	F4	F6	F8	F10	F12	2	F30
	5	72	Qi	Load1		Mult1						

<u>In</u>	stru	ction	<u>status</u>				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	D	FO	0	R1	1	1			Load1	Yes	80		
M	ULT	F4	FO	F2	1	2			Load2	Yes	72		
S	þ	F4	0	R1	1	3			Load3	No		Qi	
L	D	FO	0	R1	2	6			Store1	Yes	80	Mult	:1
M	ULT	F4	FO	F2	2				Store2	No			
S	D	F4	0	R1	2				Store3	No			
R	eser	vatior	<u> Statio</u>	<u>ns</u>		<i>S1</i>	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code.			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						MULT	F4	FO	F2
		0	Add3	No						SD	F4	0	R1
		0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI	R1	R1	#8
		0	Mult2	No						BNEZ	R1	Loo	p
R	egist	ter re	<u>sult sta</u>	tus									
C	loc	k	R 1		FO	F2	F4	F6	F8	F10	F12		F30
	6		72	Qi	Load2		Mult1						

Note: F0 never sees Load1 result

<u>In</u>	<u>stru</u>	ction	status				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	D	FO	0	R1	1	1			Load1	Yes	80		
M	ULT	F4	FO	F2	1	2			Load2	Yes	72		
S	D	F4	0	R1	1	3			Load3	No		Qi	
L	D	F0	0	R1	2	6			Store1	Yes	80	Mult	:1
M	ULT	F4	FO	F2	2	7			Store2	No			
S	D	F4	0	R1	2				Store3	No			
<u>R</u>	eser	vatior	<u>Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k	•			
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code:			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						MULT	F4	FO	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	Loo	р
R	egist	ter re	<u>sult sta</u>	<u>tus</u>									
C	loc	k	R 1		F0	F2	F4	F6	F8	F10	<i>F</i> 12		F30
	7		72	Qi	Load2		Mult2						

Note: MULT2 has no registers names in RS

<u>Instruction</u> s	status				Execution	o Write					
Instruction	j	k	iteration	Issue	complet	te Result		Busy	Addr	ess	
LD FO	0	R1	1	1			Load1	Yes	80		
MULT F4	FO	F2	1	2			Load2	Yes	72		
SD F4	0	R1	1	3			Load3	No		Qi	
L <mark>D FO</mark>	0	R1	2	6			Store1	Yes	80	Mult	1
MULT F4	FO	F2	2	7			Store2	Yes	72	Mult	.2
SD F4	0	R1	2	8			Store3	No			
<b>Reservation</b>	<u>Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k	<b>(</b>			
Time	Name	Busy	<sup>,</sup> Ор	Vj	Vk	Qj	Qk	Code	•		
0	Add1	No						LD	FO	0	R1
0	Add2	No						MULT	F4	FO	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	Loo	р
R <mark>egister res</mark>	<u>sult sta</u>	<u>itus</u>									
Clock	R1		FO	F2	F4	F6	F8	F10	F12		F30
8	72	Qi	Load2		Mult2	-	-	•	•		

<u>In</u>	struction	<u>status</u>				Execution	oı Write					
In	struction	j	k	iteration	Issue	comple	te Result	_	Busy	Addr	ess	
L	) FO	0	R1	1	1	9		Load1	Yes	80		
Μ	ULT F4	FO	F2	1	2			Load2	Yes	72		
S	) F4	0	R1	1	3			Load3	No		Qi	
L	) F0	0	R1	2	6			Store1	Yes	80	Mult1	
Μ	ULT F4	FO	F2	2	7			Store2	Yes	72	Mult2	2
S	) F4	0	R1	2	8			Store3	No			
<u>R</u>	eservation	<u> Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k			•	
	Time	Name	Busy	⁄ Ор	Vj	Vk	Qj	Qk	Code.	•		
	0	Add1	No						LD	FO	OR	R1
	0	Add2	No						MULT	F4	FO F	2
	0	Add3	No						SD	F4	OR	R1
	0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI	R1	R1 #	<del>/</del> 8
	0	Mult2	Yes	MULTD		R(F2)	Load2		<b>BNEZ</b>	R1	Loop	
<u>R</u>	<u>egister re</u>	<u>sult sta</u>	tus						-			
C	lock	R1		F0	F2	F4	F6	F8	F10	F12	E F	F30
	9	64	Qi	Load2		Mult2						
							•		•			

Load1 completing; what is waiting for it?

Computer System Design Lab

<u>In</u>	struct	ion stat	:us				Executio	ı Write					
In	struct	ion <i>j</i>		k	iteration	Issue	complet	€ Result		Busy	Addr	ress	
L	D F	C	0	R1	1	1	9	10	Load1	No			
Μ	ULT F	4	F0	F2	1	2			Load2	Yes	72		
S	D F	4	0	R1	1	3			Load3	No		Qi	
L	D F	C	0	R1	2	6	10		Store1	Yes	80	Mult	:1
Μ	ULT F	4	F0	F2	2	7			Store2	Yes	72	Mult	:2
S	D F	4	0	R1	2	8			Store3	No			
R	eserva	tion St	<u>atio</u>	<u>ns</u>		<i>S1</i>	<i>S2</i>	RS for	RS for k	•			
	T.	ime Nai	ne	Busy	Ор	Vj	Vk	Qj	Qk	Code:			
		0 Add	11	No						LD	FO	0	R1
		0 Add	12	No						<b>MULT</b>	F4	FO	F2
		0 Add	£k	No						SD	F4	0	R1
		4 Mul	t1	Yes	MULTD	M(80)	R(F2)			SUBI	R1	R1	#8
		0 Mul	t2	Yes	MULTD		R(F2)	Load2		BNEZ	R1	Loo	р
<u>R</u>	<u>egiste</u> ı	<u>r result</u>	sta	tus									
C	lock	R	1		F0	F2	F4	F6	F8	F10	F12	<u> </u>	F30
	10	6	4	Qi	Load2		Mult2						
	<i>y</i>												

Load2 completing; what is waiting for it?

<u>In</u>	<u>stru</u>	ction	<u>status</u>				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complet	<i>Result</i>		Busy	Addr	ess	
L	þ	F0	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2			Load2	No			
S	þ	F4	0	R1	1	3			Load3	Yes	64	Qi	
L	þ	FO	0	R1	2	6	10	11	Store1	Yes	80	Mult	t1
M	ULT	F4	FO	F2	2	7			Store2	Yes	72	Mult	t2
S	b	F4	0	R1	2	8			Store3	No			
R	eser	vatior	Statio	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code.			
			<i>Name</i> Add1	<i>Busy</i> No	Ор	Vj	Vk	Qj	Qk	Code.	FO	0	R1
		0			Ор	Vj	Vk	Qj	Qk	1	FO		R1 F2
		0	Add1	No	Ор	Vj	Vk	Qj	Qk	LD	FO		
		0 0 0	Add1 Add2	No No	<i>Op</i> MULTD	<i>Vj</i> M(80)		Qj	Qk	LD MULT	F0 F4	F0 0	F2
		0 0 0 3	Add1 Add2 Add3	No No No	,		R(F2)	Qj	Qk	LD MULT SD	F0 F4 F4	F0 0	F2 R1 #8
Re	egist	0 0 0 3 4	Add1 Add2 Add3 Mult1	No No No Yes Yes	MULTD	M(80)	R(F2)	Qj	Qk	LD MULT SD SUBI	F0 F4 F4 R1	FO O R1	F2 R1 #8
	egist	0 0 0 3 4 ter re	Add1 Add2 Add3 Mult1 Mult2	No No No Yes Yes	MULTD	M(80)	R(F2)	Qj F6	Qk F8	LD MULT SD SUBI	F0 F4 F4 R1 R1	FO O R1 Loo	F2 R1 #8 p
		0 0 0 3 4 ter re	Add1 Add2 Add3 Mult1 Mult2 sult sta	No No No Yes Yes	MULTD MULTD	M(80) M(72)	R(F2) R(F2)			LD MULT SD SUBI BNEZ	F0 F4 F4 R1 R1	FO O R1 Loo	F2 R1 #8 p

<u>In</u>	struction status struction <i>j k</i>						Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	D	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2			Load2	No			
S	D	F4	0	R1	1	3			Load3	Yes	64	Qi	
L	D	FO	0	R1	2	6	10	11	Store1	Yes	80	Mult	1
M	ULT	F4	FO	F2	2	7			Store2	Yes	72	Mult	2
S	D	F4	0	R1	2	8			Store3	No			
<u>R</u>	eser	vatior	<u>Statio</u>	<u>ns</u>		<i>S1</i>	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code:			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						MULT	F4	FO	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	Loo	р
<u>R</u>	egist	ter re	<u>sult sta</u>	<u>tus</u>									
C	loc	k	R 1		F0	F2	F4	F6	F8	F10	<i>F1</i> 2		F30
	12		64	Qi	Load3		Mult2						

<u>In</u>	stru	ction	<u>status</u>				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	D	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2			Load2	No			
S	D	F4	0	R1	1	3			Load3	Yes	64	Qi	
L	D	F0	0	R1	2	6	10	11	Store1	Yes	80	Mult	:1
М	ULT	F4	FO	F2	2	7			Store2	Yes	72	Mult	2
S	D	F4	0	R1	2	8			Store3	No			
<u>R</u>	eser	<u>vatior</u>	<u>Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code.			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						MULT	F4	FO	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	Loo	р
<u>R</u>	egist	ter re	<u>sult sta</u>	<u>tus</u>									
C	loc	k	R1		F0	<i>F2</i>	F4	F6	F8	F10	F12		F30
	13		64	Qi	Load3		Mult2						

<u>In</u>	struction status struction j k						Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	D	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2	14		Load2	No			
S	D	F4	0	R1	1	3			Load3	Yes	64	Qi	
L	)	FO	0	R1	2	6	10	11	Store1	Yes	80	Mult	:1
M	ULT	F4	FO	F2	2	7			Store2	Yes	72	Mult	:2
S	<b>D</b>	F4	0	R1	2	8			Store3	No			
<u>R</u>	eser	vatior	<u>Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k	•			
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code:			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						MULT	F4	FO	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	Loo	р
<u>R</u>	egist	ter re	<u>sult sta</u>	tus									
C	loc	k	R 1		F0	F2	F4	F6	F8	F10	F12		F30
	14		64	Qi	Load3		Mult2						

Mult1 completing; what is waiting for it?

<u>In</u>	stru	ction	<u>status</u>				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	D	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2	14	15	Load2	No			
S	þ	F4	0	R1	1	3			Load3	Yes	64	Qi	
L	þ	FO	0	R1	2	6	10	11	Store1	Yes	80	M(8	0)*R(
M	ULT	F4	FO	F2	2	7	15		Store2	Yes	72	Mult	2
S	þ	F4	0	R1	2	8			Store3	No			
<u>R</u>	eser	vatior	<u>Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code:			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						MULT	F4	FO	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	Loo	р
<u>R</u>	egist	ter re	<u>sult sta</u>	tus									
C	loc	k	R 1		FO	<i>F2</i>	F4	F6	F8	F10	<i>F1</i> 2		F30
	15		64	Qi	Load3		Mult2						

Mult2 completing; what is waiting for it?

<u>In</u>	stru	ction	<u>status</u>				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	)	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2	14	15	Load2	No			
S	)	F4	0	R1	1	3			Load3	Yes	64	Qi	
L	)	FO	0	R1	2	6	10	11	Store1	Yes	80	M(8	0)*R(
M	ULT	F4	FO	F2	2	7	15	16	Store2	Yes	72	M(7	2)*R(
S	)	F4	0	R1	2	8			Store3	No			
<u>R</u>	eser	vatior	<u>Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k	•			
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code:			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						MULT	F4	FO	F2
		0	Add3	No						SD	F4	0	R1
		0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
		0	Mult2	No						BNEZ	R1	Loo	р
R	egist	er re	<u>sult sta</u>	tus									
C	loc	k	R1		F0	<i>F2</i>	F4	F6	F8	F10	F12		F30
	16		64	Qi	Load3		Mult1						

<u>In</u>	stru	ction	status				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	D	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2	14	15	Load2	No			
S	D	F4	0	R1	1	3			Load3	Yes	64	Qi	
L	D	FO	0	R1	2	6	10	11	Store1	Yes	80	M(8	0)*R(
M	ULT	F4	FO	F2	2	7	15	16	Store2	Yes	72	M(7	2)*R(
S	D	F4	0	R1	2	8			Store3	Yes	64	Mult	:1
<u>R</u>	eser	vatior	<u> Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code:			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						<b>MULT</b>	F4	FO	F2
		0	Add3	No						SD	F4	0	R1
		0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
		0	Mult2	No						BNEZ	R1	Loo	р
<u>R</u>	egist	ter re	<u>sult sta</u>	tus									
C	loc	k	R 1		FO	<i>F2</i>	F4	F6	F8	F10	F12		F30
	17		64	Qi	Load3		Mult1						

<u>In</u>	stru	ction	<u>status</u>				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	)	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2	14	15	Load2	No			
S	)	F4	0	R1	1	3	18		Load3	Yes	64	Qi	
L	)	FO	0	R1	2	6	10	11	Store1	Yes	80	M(8	0)*R(
M	ULT	F4	FO	F2	2	7	15	16	Store2	Yes	72	M(7	2)*R(
S	)	F4	0	R1	2	8			Store3	Yes	64	Mult	:1
<u>R</u>	eser	vatior	<u>Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code:			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						MULT	F4	FO	F2
		0	Add3	No						SD	F4	0	R1
		0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
		0	Mult2	No						BNEZ	R1	Loo	р
R	egist	er re	<u>sult sta</u>	tus									
C	loc	k	R1		FO	<i>F2</i>	F4	F6	F8	F10	F12		F30
	18		56	Qi	Load3		Mult1						

<u>In</u>	stru	ction	<u>status</u>				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	)	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2	14	15	Load2	No			
S	)	F4	0	R1	1	3	18	19	Load3	Yes	64	Qi	
L	)	FO	0	R1	2	6	10	11	Store1	No			
М	ULT	F4	FO	F2	2	7	15	16	Store2	Yes	72	M(7	2)*R(
S	)	F4	0	R1	2	8			Store3	Yes	64	Mult	:1
R	eser	vatior	<u>Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code.			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						<b>MULT</b>	F4	FO	F2
		0	Add3	No						SD	F4	0	R1
		0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
		0	Mult2	No						BNEZ	R1	Loo	р
Re	egist	ter re	<u>sult sta</u>	tus									
C	loc	k	R1		FO	<i>F2</i>	F4	F6	F8	F10	F12		<i>F30</i>
	19		56	Qi	Load3		Mult1						

<u>In</u>	struction status struction <i>j k</i>						Executio	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	D	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2	14	15	Load2	No			
S	D	F4	0	R1	1	3	18	19	Load3	Yes	64	Qi	
L	D	FO	0	R1	2	6	10	11	Store1	No			
M	ULT	F4	FO	F2	2	7	15	16	Store2	Yes	72	M(7	2)*R(
S	D	F4	0	R1	2	8	20		Store3	Yes	64	Mult	t1
R	eser	vatior	<u>1 Statio</u>	<u>ns</u>		<i>S</i> 1	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code.			
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						<b>MULT</b>	F4	FO	F2
		0	Add3	No						SD	F4	0	R1
		0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
		0	Mult2	No						BNEZ	R1	Loo	р
R	egist	<u>er re</u>	<u>sult sta</u>	tus									
C	loc	k	R1		FO	<i>F2</i>	F4	F6	F8	F10	F12		F30
	20		56	Qi	Load3		Mult1						

<u>In</u>	stru	ction	<u>status</u>				Execution	Write					
In	stru	ction	j	k	iteration	Issue	complete	Result		Busy	Addr	ess	
L	D	FO	0	R1	1	1	9	10	Load1	No			
M	ULT	F4	FO	F2	1	2	14	15	Load2	No			
S	þ	F4	0	R1	1	3	18	19	Load3	Yes	64	Qi	
L	b	FO	0	R1	2	6	10	11	Store1	No			
M	ULT	F4	FO	F2	2	7	15	16	Store2	No			
S	b	F4	0	R1	2	8	20	21	Store3	Yes	64	Mult	:1
R	eser	vatior	<u>Statio</u>	<u>ns</u>		<i>S1</i>	<i>S2</i>	RS for	RS for k				
		Time	Name	Busy	Ор	Vj	Vk	Qj	Qk	Code.	•		
		0	Add1	No						LD	FO	0	R1
		0	Add2	No						<b>MULT</b>	F4	FO	F2
		0	Add3	No						SD	F4	0	R1
		0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
		0	Mult2	No						<b>BNEZ</b>	R1	Loo	р
R	egist	ter re	sult sta	tus									
C	loc	k	R1		FO	<i>F2</i>	F4	F6	F8	F10	F12	•••	<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