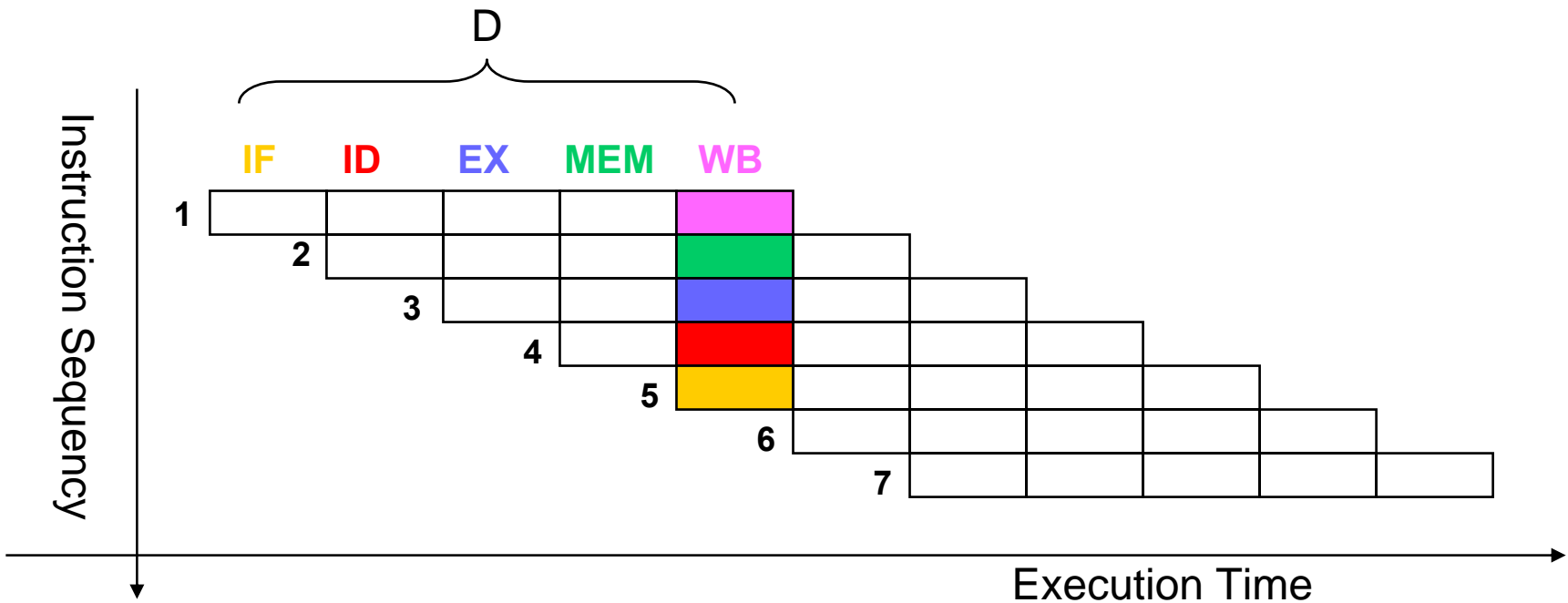

CS162

Instruction Level Parallelism

Instructor: Jun Yang

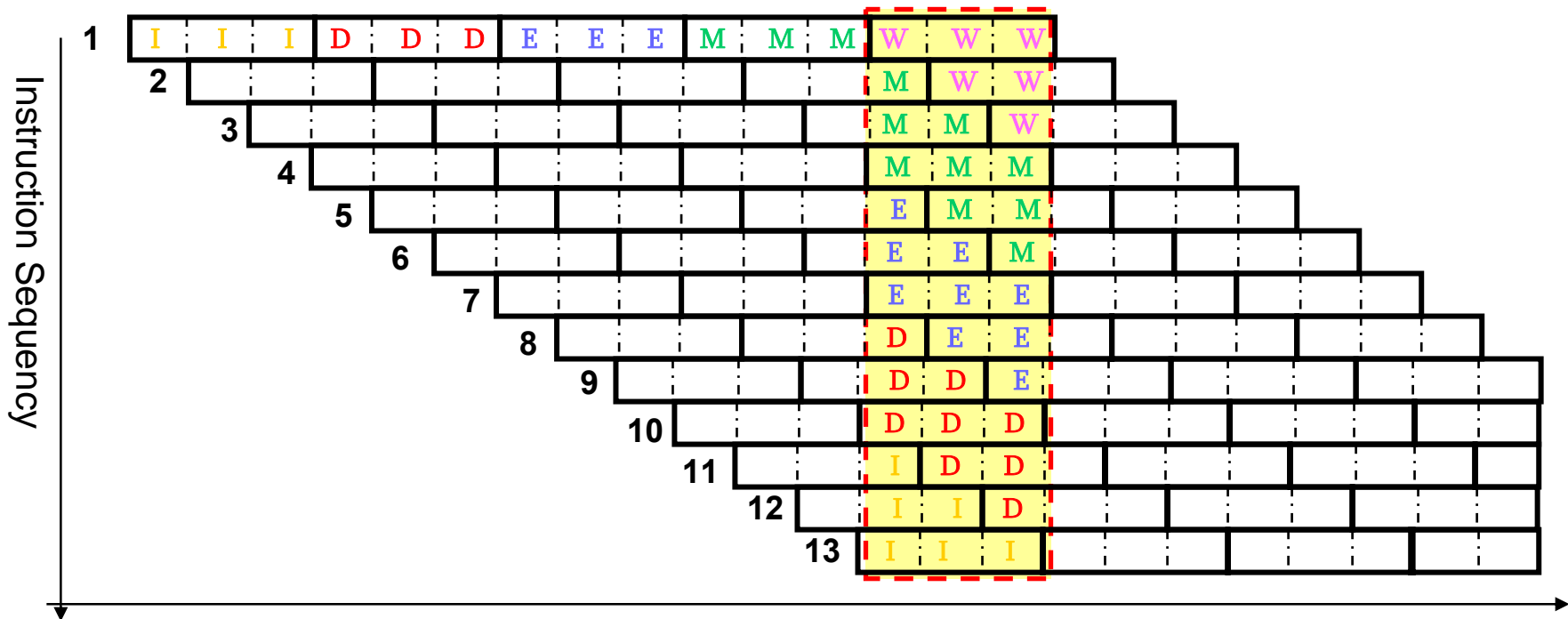
Scalar Pipeline (baseline)

- ❑ Machine parallelism ($D=5$)
- ❑ Issue rate (1 instruction per cycle)
- ❑ Peak IPC (1 instruction per cycle)



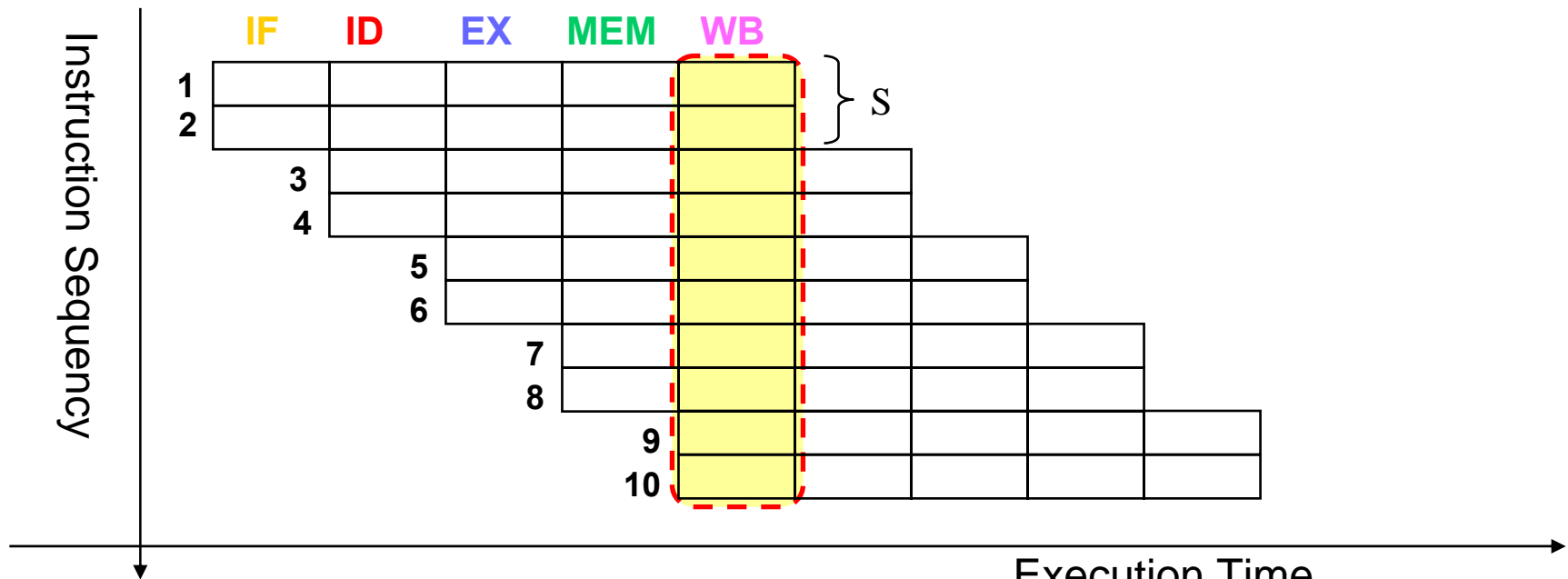
Superpipelined Machines

- ❑ 1 major cycle = M minor cycles
- ❑ Machine parallelism = M × D
- ❑ Issue rate = 1 minor cycle
- ❑ Peak IPC = 1 per minor cycle = M per baseline cycle
- ❑ Superpipelined machines are essentially deeper pieplined



Superscalar Machine

- ❑ Can issue > 1 instructions per cycle by hardware
- ❑ Replicate resources, e.g. multiple adders or multi-ported registers and caches
- ❑ Machine parallelism = $S \times D$ (10) where S is issue width (superscalar degree)
- ❑ Issue rate = 1 per cycle
- ❑ IPC = 2



What is Instruction-Level Parallelism (ILP)?

- ❑ **Fine-grained parallelism**
- ❑ **Enabled and improved by RISC**
 - More ILP of a RISC over CISC does not imply a better overall performance
 - CISC can be implemented like a RISC
- ❑ **A measure of inter-instruction dependency in a application**
 - ILP assumes a unit-cycle operation, infinite resources, prefect front-end
 - $ILP \neq IPC$
 - $IPC = \# \text{ instructions} / \# \text{ cycle}$
 - ILP is the number of instructions that can be executed in parallel; it is the upper bound of attainable IPC
- ❑ **Limited by**
 - Data dependency
 - Control dependency

Window in Search of ILP

ILP = 1

$$R5 = 8(R6)$$

$$R7 = R5 - R4$$

$$R9 = R7 \times R7$$

ILP = 1.5

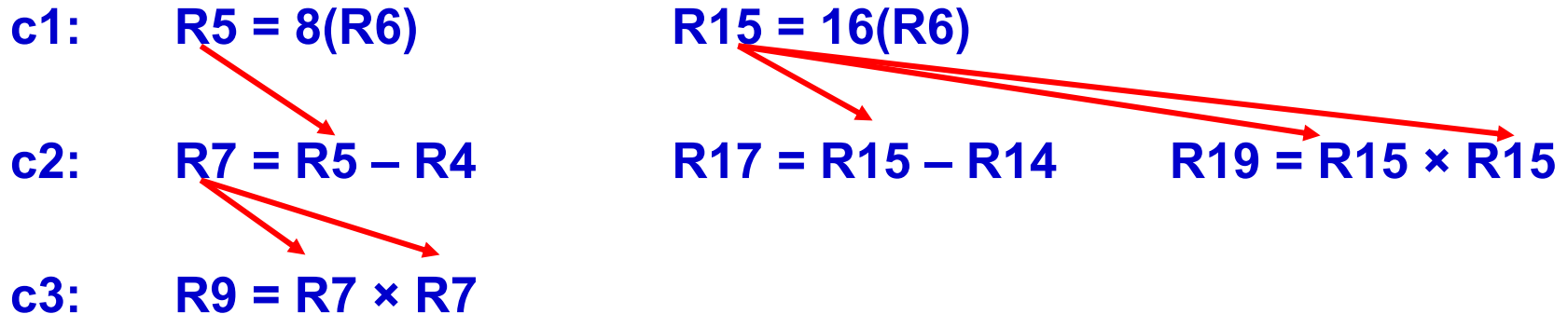
$$R15 = 16(R6)$$

$$R17 = R15 - R14$$

$$R19 = R15 \times R15$$

ILP = ?

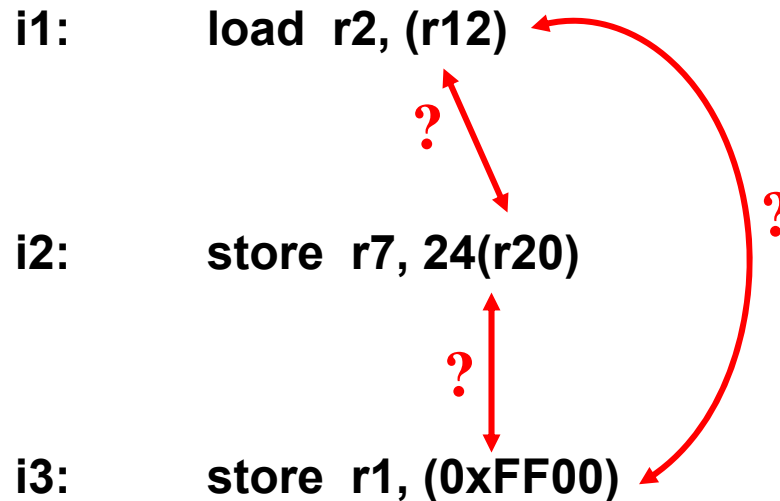
Window in Search of ILP



- ILP = $6/3 = 2$ better than 1 and 1.5
- Larger window gives more opportunities
- Who exploit the instruction window
- But what limits the window?

Memory Dependency

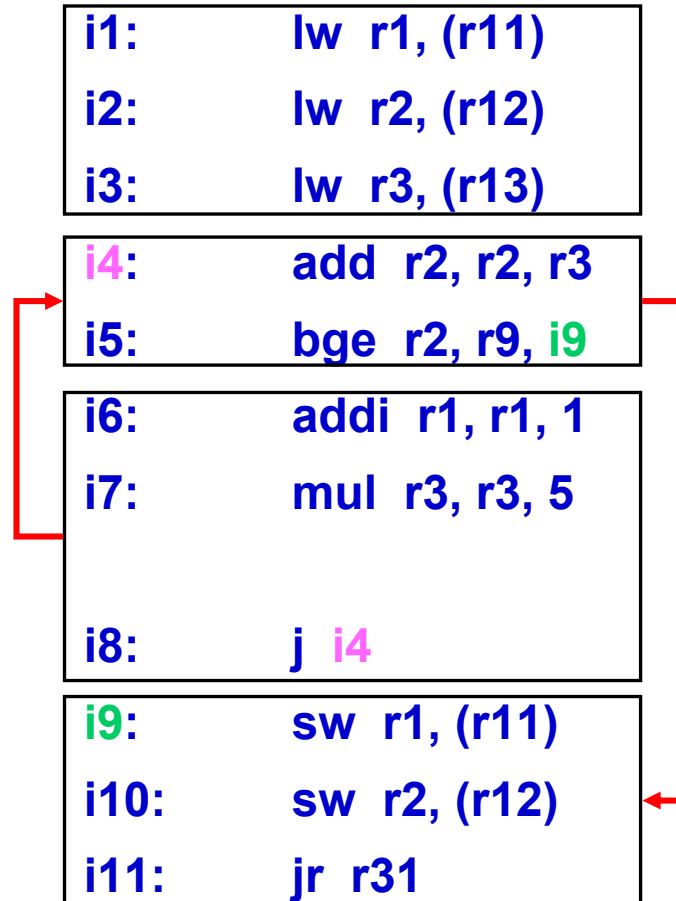
- ❑ Ambiguous dependency forces “sequentiality”
- ❑ To increase ILP, needs dynamic memory disambiguation mechanisms that are either safe or recoverable



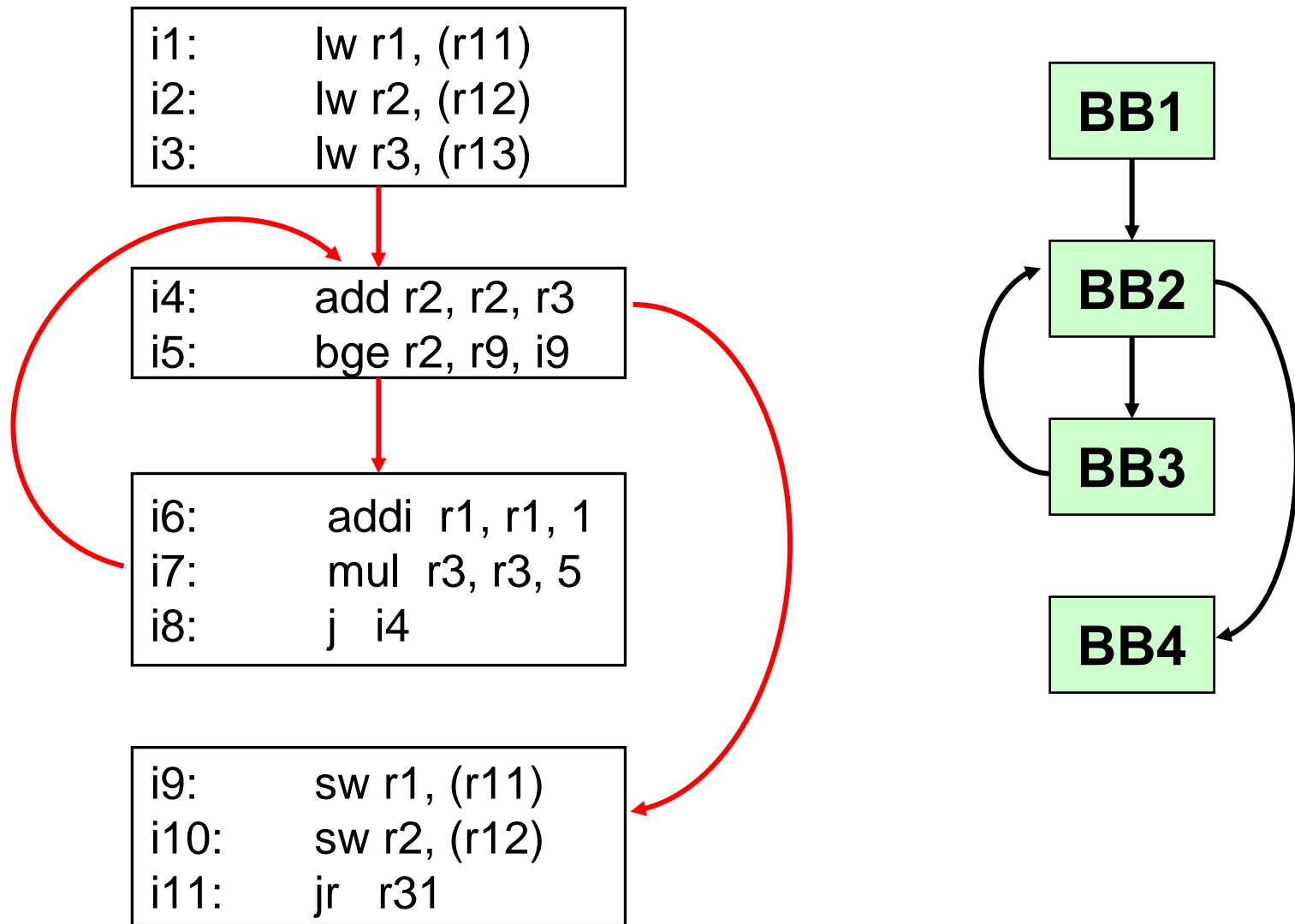
- ❑ ILP could be 1, could be 3, depending on the actual dependence

Basic Blocks

```
a = array [i];
b = array [j];
c = array [k];
d = b + c;
while ( d < t ) {
    a++;
    c *= 5;
    d = b+c;
}
array [i] = a;
array [j] = d;
```



Control Flow Graph



Flynn's Bottleneck

□ ILP ≈ 1.86

- Programs on IBM 7090
- ILP exploited within basic blocks

□ [Riseman & Foster'72]

- Breaking control dependency
- A perfect machine model
- Benchmark includes numerical programs, assemblers and compilers

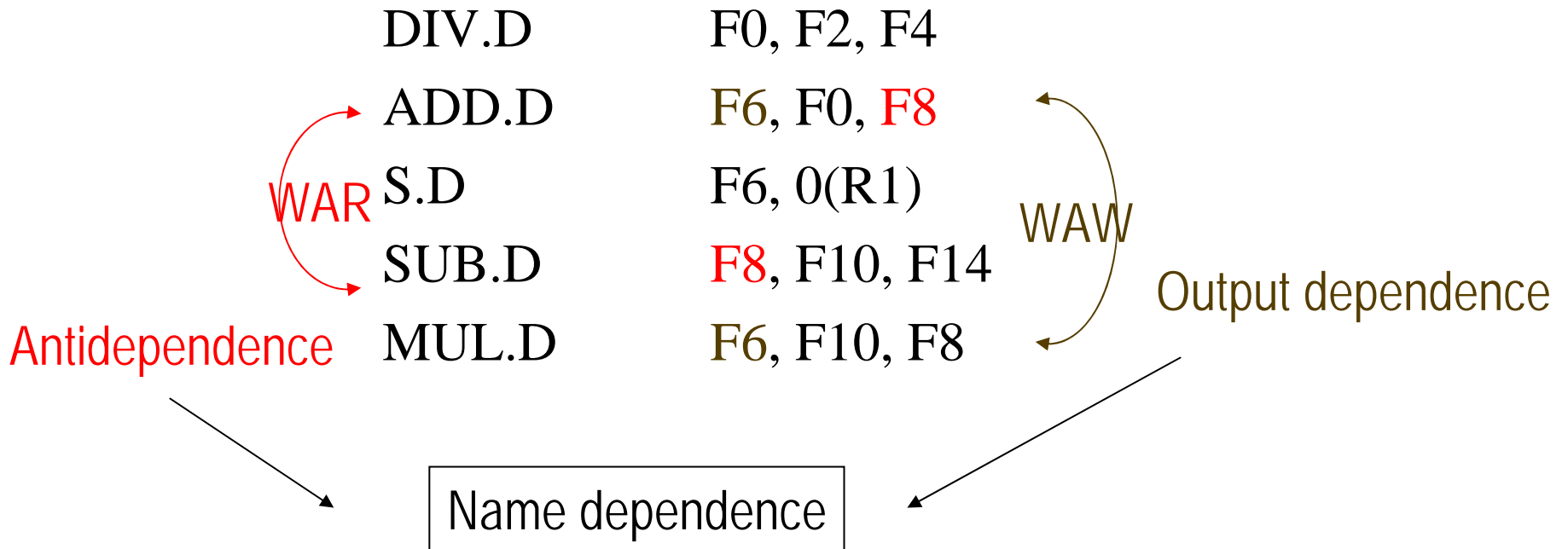
Passed jumps	0	1	2	8	32	128	∞
Average ILP	1.72	2.72	3.62	7.21	14.8	24.2	51.2

Attaining High ILP

- ❑ **Overcome Control Dependency**
 - **Branch prediction / Hint**
 - **Speculation**
 - **Predicated execution**
- ❑ **Overcome Data Dependency**
 - **Hardware techniques**
 - Dynamic scheduling
 - Register renaming
 - Memory disambiguation
 - **Software techniques**
 - Better register allocation
 - Memory disambiguation (compiler level)
- ❑ **Exploit Thread-Level Parallelism**
 - **Fine-grained or coarse-grained multithreading**
 - **Simultaneous Multithreading (SMT) or Intel's Hyperthreading**

Tomasulo's Algorithm

Review: False Dependencies



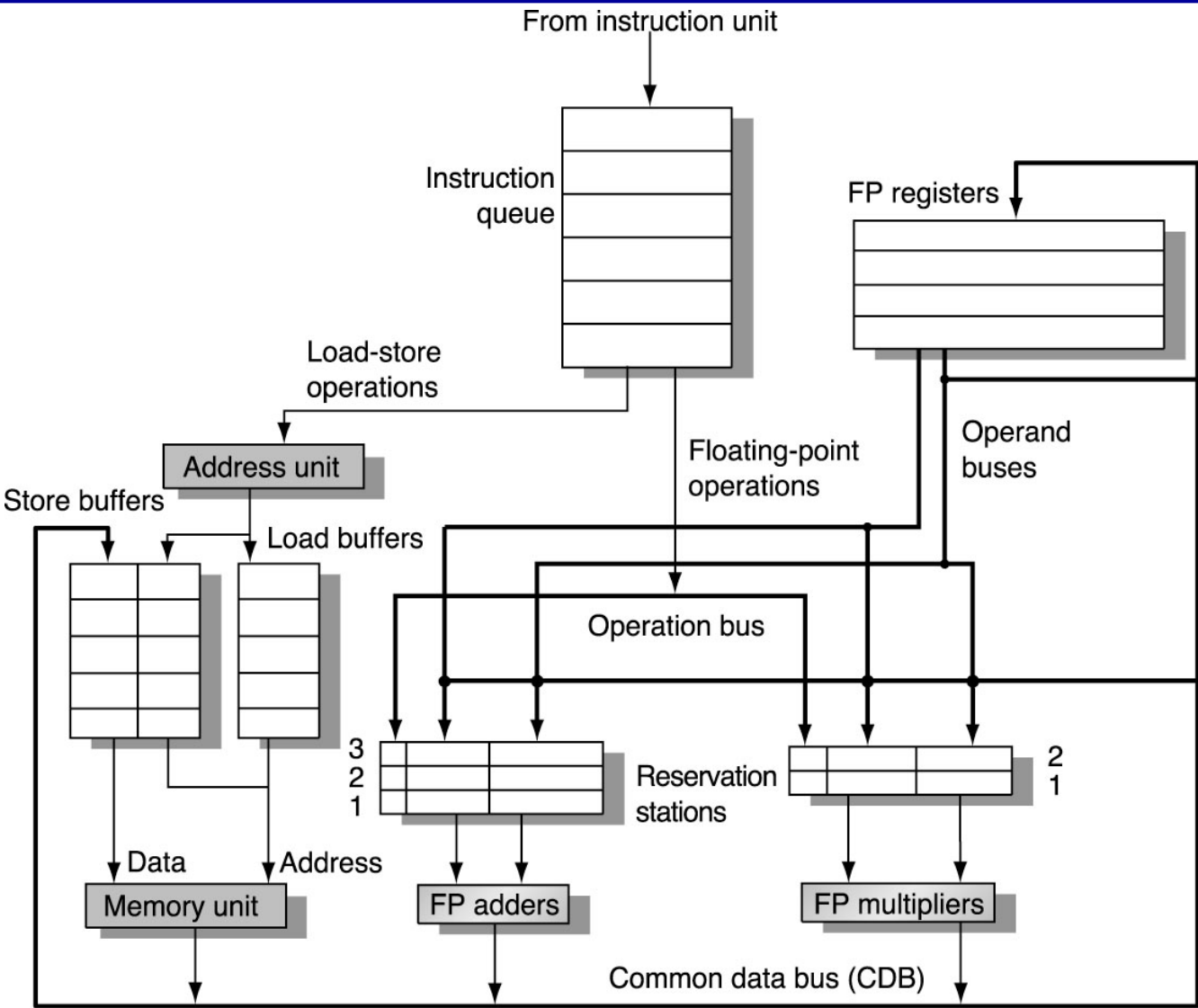
A Dynamic Algorithm Exploiting ILP: Tomasulo Algorithm

- ❑ **Implemented through reservation stations (rs) per functional unit**
 - **Buffers an operand as soon as it is available – avoids WAR hazards.**
 - **Pending instr. designate rs that will provide their inputs – avoids WAW hazards.**
 - **The last write in a sequence of same-register-writing actually updates the register**
 - **Decentralize hazard detection and execution control**
 - **Instruction results are passed directly to the FU from rs rather than from registers**
 - **Through common data bus (CDB)**

A Dynamic Algorithm Exploiting ILP : Tomasulo Algorithm

- ❑ For IBM 360/91 about 3 years after CDC 6600
- ❑ Goal: High Performance without special compilers
- ❑ Why study? Lead to Alpha 21264, HP 8000, MIPS 10000, Pentium II, Power PC 604 ...

FP unit and load-store unit using Tomasulo's alg.



Assuming the Following Latency

LD – 2 cycle

(compute address + data cache access)

ADDDs and SUBs are 2 cycles

Multiply is 10 cycles

Divide is 40 cycles

Reservation Station Components

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

V_j, V_k—**Value** of Source operands

- store buffers have **V** field, the value to be stored

Q_j, Q_k—Reservation stations producing source registers

- $Q_j, Q_k == 0 \Rightarrow$ ready

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 Major Stages of Tomasulo Algorithm

1. Issue—get instruction from FP Op Queue

Check for structure hazards. If reservation station (r) is free, the issue logic issues instr to r & read operands into r if they are ready. Perform register renaming – put r into the register status table for the destination register.

2. Execution—operate on operands (EX)

When both operands are ready then execute;
if not ready, watch CDB for result

3. Write result—finish execution (WB)

Write on Common Data Bus to all awaiting units;
mark reservation station available. Write result into dest. reg. if its status is r .

Normal data bus: data + destination (“go to” bus)

CDB: 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 broadcast

Tomasulo Example Cycle 0

<u>Instruction status</u>				<i>Execution</i>	<i>Write</i>							
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address			
LD	F6	34+	R2					Load1	No			
LD	F2	45+	R3					Load2	No			
MULTD	F0	F2	F4					Load3	No			
SUBD	F8	F6	F2									
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
<u>Reservation Stations</u>					<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	No									
	0	Mult2	No									
<u>Register result status</u>												
Clock				<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>
0			<i>FU</i>									

Tomasulo Example Cycle 1

<u>Instruction status</u>				<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1				Load1	Yes	34+R2		
LD	F2	45+	R3					Load2	No			
MULTD	F0	F2	F4					Load3	No			
SUBD	F8	F6	F2									
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
<u>Reservation Stations</u>					<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	No									
	0	Mult2	No									
<u>Register result status</u>												
Clock				<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>
1			<i>FU</i>				Load1					

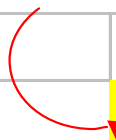
Tomasulo Example Cycle 2

<u>Instruction status</u>				<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2-			Load1	Yes	34+R2		
LD	F2	45+	R3	2				Load2	Yes	45+R3		
MULTD	F0	F2	F4					Load3	No			
SUBD	F8	F6	F2									
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
<u>Reservation Stations</u>					<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	No									
	0	Mult2	No									
<u>Register result status</u>												
Clock				<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>
2			<i>FU</i>		Load2		Load1					

Tomasulo Example Cycle 3

<u>Instruction status</u>				<i>Execution</i>		<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>		Busy	Address			
LD	F6	34+	R2	1	2--3			Load1	Yes	34+R2		
LD	F2	45+	R3	2	3-			Load2	Yes	45+R3		
MULTD	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2									
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
<u>Reservation Stations</u>						<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	Mult			R(F4)	Load2				
	0	Mult2	No									
<u>Register result status</u>												
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>
3			<i>FU</i>	Mult1	Load2		Load1					

read value



Tomasulo Example Cycle 4

<u>Instruction status</u>				<i>Execution</i>		<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>		Busy	Address			
LD	F6	34+	R2	1	2--3	4	Load1	No				
LD	F2	45+	R3	2	3--4		Load2	Yes	45+R3			
MULTD	F0	F2	F4	3			Load3	No				
SUBD	F8	F6	F2	4								
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
<u>Reservation Stations</u>						<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	Yes	Sub	M(A1)			Load2				
	0	Add2	No									
		Add3	No									
	0	Mult1	Yes	Mult		R(F4)	Load2					
	0	Mult2	No									
<u>Register result status</u>												
Clock				<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>
4				FU	Mult1	Load2	M(A1)	Add1				

Tomasulo Example Cycle 5

<u>Instruction status</u>				<i>Execution</i>		<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>		Busy	Address			
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3				Load3	No			
SUBD	F8	F6	F2	4								
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2									
<u>Reservation Stations</u>						<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>				
	2	Add1	Yes	Sub	M(A1)	M(A2)						
	0	Add2	No									
		Add3	No									
	10	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<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>
5				FU	Mult1	M(A2)	M(A1)	Add1	Mult2			

Tomasulo Example Cycle 6

<u>Instruction status</u>				<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 --							
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6								
<u>Reservation Stations</u>					<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>				
	1	Add1	Yes	Sub	M(A1)	M(A2)						
	0	Add2	Yes	Add		M(A2)	Add1					
		Add3	No									
	9	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<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>
6			<i>FU</i>	Mult1	M(A2)		Add2	Add1	Mult2			

Tomasulo Example Cycle 7

<u>Instruction status</u>				<i>Execution</i>		<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>		Busy	Address			
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7							
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6								
<u>Reservation Stations</u>						<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	Yes	Sub	M(A1)	M(A2)						
	0	Add2	Yes	Add		M(A2)	Add1					
		Add3	No									
	8	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<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>
7			<i>FU</i>	Mult1	M(A2)		Add2	Add1	Mult2			

Tomasulo Example Cycle 8

<u>Instruction status</u>				<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6								
<u>Reservation Stations</u>					<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	Add	M1-M2	M(A2)						
		Add3	No									
	7	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
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(A2)		Add2	M1-M2	Mult2			

Tomasulo Example Cycle 9

<u>Instruction status</u>				<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 --							
<u>Reservation Stations</u>					<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	Add	M1-M2	M(A2)						
		Add3	No									
	6	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<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>
9			FU	Mult1	M(A2)		Add2	M1-M2	Mult2			

Tomasulo Example Cycle 10

<u>Instruction status</u>				<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 -- 10							
<u>Reservation Stations</u>					<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	Yes	Add	M1-M2	M(A2)						
		Add3	No									
	5	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<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>
10			FU	Mult1	M(A2)		Add2	M1-M2	Mult2			

Tomasulo Example Cycle 11

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>					<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									
		Add2	No									
		Add3	No									
	4	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
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(A2)		M1-M2+M(j)	M1-M2	Mult2			

Tomasulo Example Cycle 12

<u>Instruction status</u>						<i>Execution</i>	<i>Write</i>					
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 --			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>						<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									
		Add2	No									
		Add3	No									
	4	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
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(A2)		M1-M2+M(M1-M2	Mult2			

Tomasulo Example Cycle 15

<u>Instruction status</u>					<i>Execution</i>	<i>Write</i>						
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 -- 15			Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>					<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									
		Add2	No									
		Add3	No									
	0	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
<u>Register result status</u>												
Clock				<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>
15			<i>FU</i>	Mult1	M(A2)	M1-M2+M(A1)		M1-M2	Mult2			

Tomasulo Example Cycle 16

<u>Instruction status</u>						<i>Execution</i>	<i>Write</i>					
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 -- 15	16		Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5								
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>						<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									
		Add2	No									
		Add3	No									
		Mult1	No									
	40	Mult2	Yes	Div	M*F4	M(A1)						
<u>Register result status</u>												
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(A2)	M1-M2+M(A1)		M1-M2	Mult2			

Tomasulo Example Cycle 56

<u>Instruction status</u>				<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 -- 15	16		Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5	17 -- 56							
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>					<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									
		Add2	No									
		Add3	No									
		Mult1	No									
	0	Mult2	Yes	Div	M*F4	M(A1)						
<u>Register result status</u>												
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(A2)	M1-M2+M(A1)		M1-M2	Mult2			

Tomasulo Example Cycle 57

<u>Instruction status</u>				<i>Execution</i>	<i>Write</i>							
Instruction		<i>j</i>	<i>k</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>			Busy	Address		
LD	F6	34+	R2	1	2--3	4		Load1	No			
LD	F2	45+	R3	2	3--4	5		Load2	No			
MULTD	F0	F2	F4	3	6 -- 15	16		Load3	No			
SUBD	F8	F6	F2	4	6 -- 7	8						
DIVD	F10	F0	F6	5	17 -- 56	57						
ADDD	F6	F8	F2	6	9 -- 10	11						
<u>Reservation Stations</u>					<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									
		Add2	No									
		Add3	No									
		Mult1	No									
	0	Mult2	No									
<u>Register result status</u>												
Clock				<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>
57			<i>FU</i>	<i>M*F4</i>	<i>M(A2)</i>		<i>M1-M2+M(</i>	<i>M1-M2</i>	result			

Tomasulo Loop Example

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

- Multiply takes 4 clocks
- First load takes 8 clocks (cache misses), second load takes 1 clock (hit)
- To be clear, will show clocks for SUBI, BNEZ (integer instr.)
- Reality: integer instructions are ahead

Loop Example Cycle 0

Instruction status				Execution				Write			
Instruction	j	k	iteration	Issue	complete	Result	Busy	Address			
LD	F0	0 R1	1				Load1	No			
MULTD	F4	F0 F2	1				Load2	No			
SD	F4	0 R1	1				Load3	No	Qi	V	
LD	F0	0 R1	2				Store1	No			
MULTD	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	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	No						BNEZ	R1	Loop
Register result status											
Clock	R1	$F0$	$F2$	$F4$	$F6$	$F8$	$F10$	$F12$...	$F30$	
0											

Loop Example Cycle 1

Instruction status				Execution		Write					
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>		Busy	Address		
LD	F0	0 R1	1	1			Load1	Yes	80		
MULTD	F4	F0 F2	1				Load2	No			
SD	F4	0 R1	1				Load3	No		Qi	V
LD	F0	0 R1	2				Store1	No			
MULTD	F4	F0 F2	2				Store2	No			
SD	F4	0 R1	2				Store3	No			
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>	<i>Code:</i>		
	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	No						BNEZ	R1	Loop
Register result status											
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30	
1	80	Load1									

Loop Example Cycle 2

Instruction status				Execution		Write						
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>		Busy	Address			
LD	F0	0	R1	1	1	2 --	Load1	Yes	80			
MULTD	F4	F0	F2	1	2		Load2	No				
SD	F4	0	R1	1			Load3	No		Qi	V	
LD	F0	0	R1	2			Store1	No				
MULTD	F4	F0	F2	2			Store2	No				
SD	F4	0	R1	2			Store3	No				
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>	<i>Code:</i>			
	0	Add1	No						LD	F0	0	R1
	0	Add2	No						MULT	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	Yes	Mult		R(F2)	Load1		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>
	2			Load1		Mult1						

Loop Example Cycle 3

Instruction status				Execution		Write						
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address				
LD	F0	0	R1	1	1	2 --	Load1	Yes	80			
MULTD	F4	F0	F2	1	2		Load2	No				
SD	F4	0	R1	1	3		Load3	No		Q	V	
LD	F0	0	R1	2			Store1	Yes	80	Mult1		
MULTD	F4	F0	F2	2			Store2	No				
SD	F4	0	R1	2			Store3	No				
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>	<i>Code:</i>			
	0	Add1	No						LD	F0	0	R1
	0	Add2	No						MULT	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	Yes	Mult		R(F2)	Load1		SUBI	R1	R1	#8
	0	Mult2	No						BNEZ	R1	Loop	
Register result status												
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30		
		Load1									Mult1	

Loop Example Cycle 4

Instruction status				Execution		Write							
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address					
LD	F0	0	R1	1	1	2 --	Load1	Yes	80				
MULTD	F4	F0	F2	1	2		Load2	No					
SD	F4	0	R1	1	3		Load3	No		Q	V		
LD	F0	0	R1	2			Store1	Yes	80	Mult1			
MULTD	F4	F0	F2	2			Store2	No					
SD	F4	0	R1	2			Store3	No					
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>	<i>Code:</i>				
	0	Add1	No						LD	F0	0	R1	
	0	Add2	No						MULT	F4	F0	F2	
	0	Add3	No						SD	F4	0	R1	
	0	Mult1	Yes	Mult		R(F2)	Load1		SUBI	R1	R1	#8	
	0	Mult2	No						BNEZ	R1	Loop		
Register result status													
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30			
		Load1									Mult1		

Loop Example Cycle 5

Instruction status				Execution				Write					
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address					
LD	F0	0	R1	1	1	2 --	Load1	Yes	80				
MULTD	F4	F0	F2	1	2		Load2	No					
SD	F4	0	R1	1	3		Load3	No	Q		V		
LD	F0	0	R1	2			Store1	Yes	80		Mult1		
MULTD	F4	F0	F2	2			Store2	No					
SD	F4	0	R1	2			Store3	No					
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>	<i>Code:</i>				
	0	Add1	No						LD	F0	0	R1	
	0	Add2	No						MULT	F4	F0	F2	
	0	Add3	No						SD	F4	0	R1	
	0	Mult1	Yes	Mult		R(F2)	Load1		SUBI	R1	R1	#8	
	0	Mult2	No						BNEZ	R1	Loop		
Register result status													
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30			
	72	Load1									Mult1		

Loop Example Cycle 6: F0 never sees load from location 80

Instruction status				Execution		Write							
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address					
LD	F0	0	R1	1	1	2 --		Load1	Yes	80			
MULTD	F4	F0	F2	1	2			Load2	Yes	72			
SD	F4	0	R1	1	3			Load3	No		Q	V	
LD	F0	0	R1	2	6			Store1	Yes	80	Mult1		
MULTD	F4	F0	F2	2				Store2	No				
SD	F4	0	R1	2				Store3	No				
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>	<i>Code:</i>				
	0	Add1	No						LD	F0	0	R1	
	0	Add2	No						MULT	F4	F0	F2	
	0	Add3	No						SD	F4	0	R1	
	0	Mult1	Yes	Mult		R(F2)	Load1		SUBI	R1	R1	#8	
	0	Mult2	No						BNEZ	R1	Loop		
Register result status													
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30			
		Load2		Mult1									

Loop Example Cycle 7:

Register file completely detached from iteration 1

Instruction status				Execution				Write				
Instruction	<i>j</i>	<i>k</i>	iteration	Issue	complete	Result	Busy	Address				
LD	F0	0	R1	1	1	2 --	Load1	Yes	80			
MULTD	F4	F0	F2	1	2		Load2	Yes	72			
SD	F4	0	R1	1	3		Load3	No		Q	V	
LD	F0	0	R1	2	6		Store1	Yes	80	Mult1		
MULTD	F4	F0	F2	2	7		Store2	No				
SD	F4	0	R1	2			Store3	No				
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>	<i>Code:</i>			
	0	Add1	No						LD	F0	0	R1
	0	Add2	No						MULT	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	Yes	Mult		R(F2)	Load1		SUBI	R1	R1	#8
	0	Mult2	Yes	Mult		R(F2)	Load2		BNEZ	R1	Loop	
Register result status												
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30		
			Load2		Mult2							

Loop Example Cycle 8:

First and second iteration completely overlapped

Instruction status				Execution		Write						
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address				
LD	F0	0	R1	1	1	2 --		Load1	Yes	80		
MULTD	F4	F0	F2	1	2			Load2	Yes	72		
SD	F4	0	R1	1	3			Load3	No		Q	V
LD	F0	0	R1	2	6			Store1	Yes	80	Mult1	
MULTD	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>					
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>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						MULT	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	Yes	Mult		R(F2)	Load1		SUBI	R1	R1	#8
	0	Mult2	Yes	Mult		R(F2)	Load2		BNEZ	R1	Loop	
Register result status												
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30		
			Load2			Mult2						

Loop Example Cycle 9

Instruction status				Execution		Write						
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address				
LD	F0	0	R1	1	1	2 -- 9	Load1	Yes	80			
MULTD	F4	F0	F2	1	2		Load2	Yes	72			
SD	F4	0	R1	1	3		Load3	No		Q	V	
LD	F0	0	R1	2	6		Store1	Yes	80	Mult1		
MULTD	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>					
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>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						MULT	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	Yes	Mult		R(F2)	Load1		SUBI	R1	R1	#8
	0	Mult2	Yes	Mult		R(F2)	Load2		BNEZ	R1	Loop	
Register result status												
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30		
			Load2			Mult2						

Loop Example Cycle 10

Instruction status				Execution				Write				
Instruction		<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy		Address		
LD	F0	0	R1	1	1	2 -- 9	10	Load1	No			
MULTD	F4	F0	F2	1	2			Load2	Yes	72		
SD	F4	0	R1	1	3			Load3	No		Q V	
LD	F0	0	R1	2	6	10		Store1	Yes	80	Mult1	
MULTD	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>					
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>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						MULT	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	4	Mult1	Yes	Mult	M[80]	R(F2)			SUBI	R1	R1	#8
	0	Mult2	Yes	Mult		R(F2)	Load2		BNEZ	R1	Loop	
Register result status												
Clock		R1		F0	F2	F4	F6	F8	F10	F12	...	F30
	10	64	Load2			Mult2						

Loop Example Cycle 11

Instruction status				Execution				Write				
Instruction		<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address			
LD	F0	0	R1	1	1	2 -- 9	10	Load1	No			
MULTD	F4	F0	F2	1	2			Load2	No			
SD	F4	0	R1	1	3			Load3	Yes	64	Q	V
LD	F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1	
MULTD	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>					
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>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						MULT	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	3	Mult1	Yes	Mult	M[80]	R(F2)			SUBI	R1	R1	#8
	4	Mult2	Yes	Mult	M[72]	R(F2)			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>
				Load3		Mult2						

Loop Example Cycle 12:

Why not issue 3rd multiply

Instruction status				Execution		Write							
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address					
LD	F0	0	R1	1	1	2 -- 9	10	Load1	No				
MULTD	F4	F0	F2	1	2			Load2	No				
SD	F4	0	R1	1	3			Load3	Yes	64	Q	V	
LD	F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1		
MULTD	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>						
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>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						MULT	F4	F0	F2	
	0	Add3	No						SD	F4	0	R1	
	2	Mult1	Yes	Mult	M[80]	R(F2)			SUBI	R1	R1	#8	
	3	Mult2	Yes	Mult	M[72]	R(F2)			BNEZ	R1	Loop		
Register result status													
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30			
			Load3			Mult2							

Loop Example Cycle 14

Instruction status				Execution				Write					
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address					
LD	F0	0	R1	1	1	2 -- 9	10	Load1	No				
MULTD	F4	F0	F2	1	2	14		Load2	No				
SD	F4	0	R1	1	3			Load3	Yes	64	Q	V	
LD	F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1		
MULTD	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>						
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>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						MULT	F4	F0	F2	
	0	Add3	No						SD	F4	0	R1	
	0	Mult1	Yes	Mult	M[80]	R(F2)			SUBI	R1	R1	#8	
	1	Mult2	Yes	Mult	M[72]	R(F2)			BNEZ	R1	Loop		
Register result status													
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30			
			Load3			Mult2							

Loop Example Cycle 15

Instruction status				Execution				Write					
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address					
LD	F0	0	R1	1	1	2 -- 9	10	Load1	No				
MULTD	F4	F0	F2	1	2	14	15	Load2	No				
SD	F4	0	R1	1	3			Load3	Yes	64	Q	V	
LD	F0	0	R1	2	6	10	11	Store1	Yes	80		80*R2	
MULTD	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 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>	<i>Code:</i>				
	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	Mult	M[72]	R(F2)			BNEZ	R1	Loop		
Register result status													
Clock	R1	F0	F2	F4	F6	F8	F10	F12	...	F30			
			Load3			Mult3							

Loop Example Cycle 16 ...

Instruction status				Execution		Write							
Instruction	<i>j</i>	<i>k</i>	<i>iteration</i>	<i>Issue</i>	<i>complete</i>	<i>Result</i>	Busy	Address					
LD	F0	0	R1	1	1	2 -- 9	10	Load1	No				
MULTD	F4	F0	F2	1	2	14	15	Load2	No				
SD	F4	0	R1	1	3			Load3	Yes	64	Q	V	
LD	F0	0	R1	2	6	10	11	Store1	Yes	80			80*R2
MULTD	F4	F0	F2	2	7	15	16	Store2	Yes	72	Mult2		70*R2
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>						
	<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>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						MULT	F4	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			
			Load3			Mult3							

Tomasulo Summary

- ❑ Prevents Register as bottleneck
- ❑ Avoids WAR, WAW hazards
- ❑ Allows loop unrolling in HW
- ❑ Not limited to basic blocks (provided branch prediction)
- ❑ Lasting Contributions
 - Dynamic scheduling
 - Register renaming
 - Load/store disambiguation