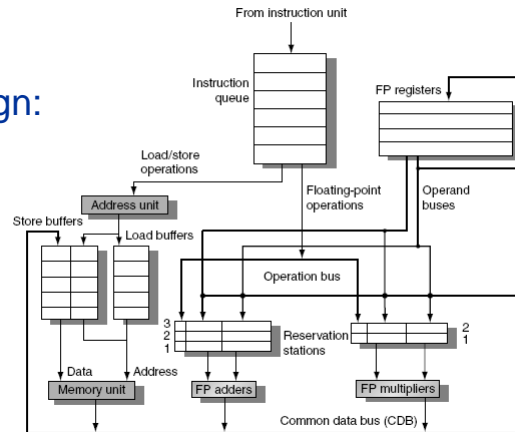


## Tomasulo's Algorithm

- Load and store buffers
  - Contain data and addresses, act like reservation stations
- Top-level design:



## Tomasulo's Algorithm

- Three Steps:
  - Issue
    - Get next instruction from FIFO queue
    - If available RS, issue the instruction to the RS with operand values if available
    - If operand values not available, stall the instruction issue
  - Execute
    - When operand becomes available, store it in any reservation stations waiting for it
    - When all operands are ready, issue the instruction
    - Loads and store maintained in program order through effective address
    - No instruction allowed to initiate execution until all branches that proceed it in program order have completed
  - Write result
    - Write result on CDB into reservation stations and store buffers
      - (Stores must wait until address and value are received)

# Tomasulo's Algorithm

- 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: Qj, Qk=0 → ready
  - Store buffers only have Qj for RS producing result
- A:** Used to hold info for the load store (initially immediate, then effective address)
- Busy:** Indicates reservation station or FU is busy
- Register result status— Qi indicates which functional unit will write each register, 0 means no write to this register

# Example

Instruction status							
Instruction	Issue	Execute	Write Result				
L.D F6,32(R2)	√	√	√				
L.D F2,44(R3)	√	√					
MUL.D F0,F2,F4	√						
SUB.D F8,F2,F6	√						
DIV.D F10,F0,F6	√						
ADD.D F6,F8,F2	√						

Reservation stations							
Name	Busy	Op	Vj	Vk	Qj	Qk	A
Load1	No						
Load2	Yes	Load					44 + Regs[R3]
Add1	Yes	SUB		Mem[32 + Regs[R2]]	Load2		
Add2	Yes	ADD			Add1	Load2	
Add3	No						
Mult1	Yes	MUL		Regs[F4]	Load2		
Mult2	Yes	DIV		Mem[32 + Regs[R2]]	Mult1		

Register status								
Field	F0	F2	F4	F6	F8	F10	F12	... F30
Qi	Mult1	Load2		Add2	Add1	Mult2		

# Dealing with WAR

Instruction			Instruction status			
L.D	F6, 32(R2)					
L.D	F2, 44(R3)					
MUL.D	F0, F2, F4					
SUB.D	F8, F2, F6					
DIV.D	F10, F0, F6					
ADD.D	F6, F8, F2					

Name	Busy	Op
Load1	No	
Load2	Yes	Load
Add1	Yes	SUB
Add2	Yes	ADD
Add3	No	
Mult1	Yes	MUL
Mult2	Yes	DIV

Register status							
Field	F0	F2	F4	F6	F8	F10	F12 ... F30
Qi	Mult1	Load2		Add2	Add1	Mult2	

The processor issues both DIV and ADD although there is a WAR hazard.

If F<sup>6</sup> is ready when DIV is issued, its value is read and stored in the RS (ADD may change it that is O.K.)

If not ready, RS will read it from the FU producing it, again ADD may change F6 since we will read it from the FU not F6



## Instruction stream

**Instruction status:**

Instruction	j	k	Issue	Comp	Result
LD	F6	34+	R2		
LD	F2	45+	R3		
MULTD	F0	F2	F4		
SUBD	F8	F6	F2		
DIVD	F10	F0	F6		
ADD	F6	F8	F2		

Busy	Address
Load1	No
Load2	No
Load3	No

3 Load/Buffers

**Reservation Stations:**

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

FU count down

3 FP Adder R.S.  
2 FP Mult R.S.

**Register result status:**

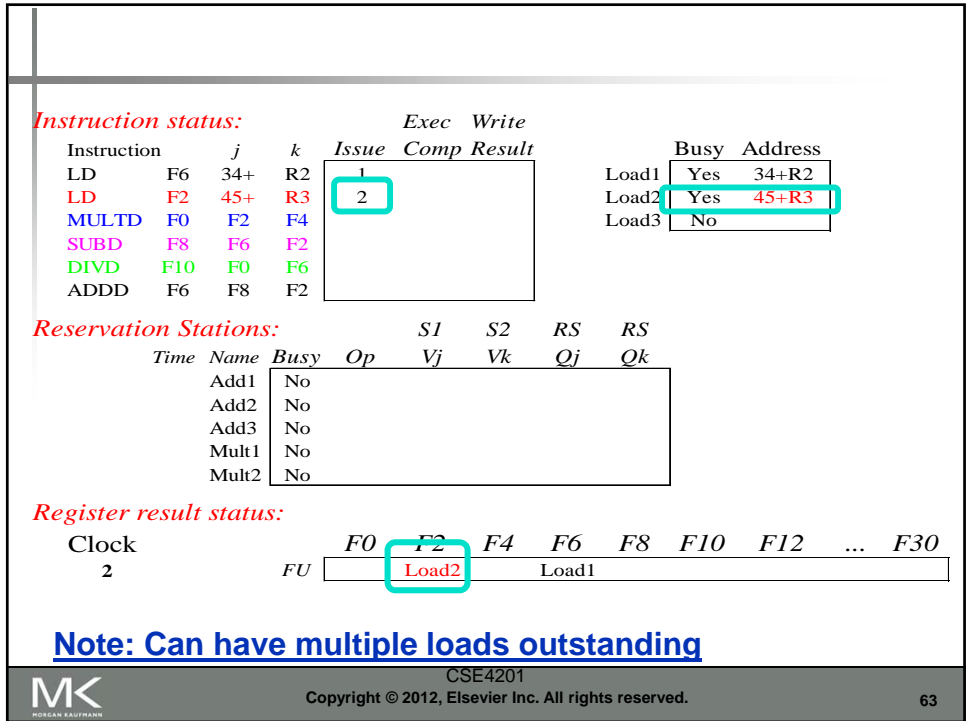
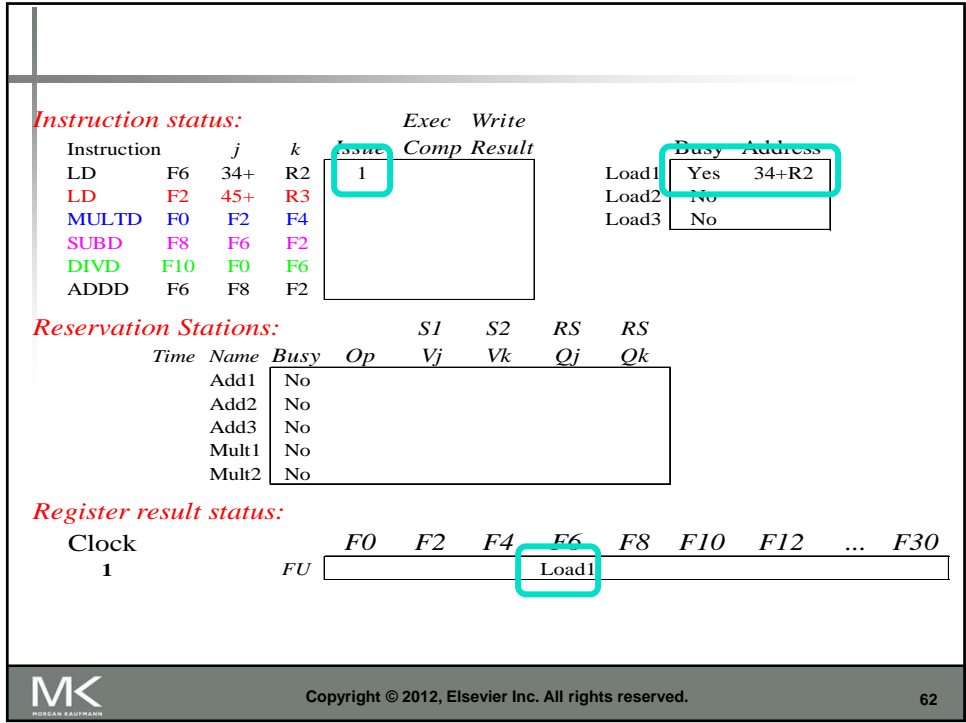
Register	F0	F2	F4	F6	F8	F10	F12 ... F30
FU							

Clock

0

Clock cycle counter





**Instruction status:**

Instruction	j	k	R2	Exec		Write	Issue	Comp	Result	Load1	Busy	Address
				1	3							
LD	F6	34+	R2	1	3		1			Yes	34+R2	
LD	F2	45+	R3	2			2			Yes	45+R3	
MULTD	F0	F2	F4	3			3			No		
SUBD	F8	F6	F2									
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									

**Reservation Stations:**

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	MULTD		R(F4)	Load2	
	Mult2	No					

**Register result status:**

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

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

MK  
MORGAN KAUFMANN

Copyright © 2012, Elsevier Inc. All rights reserved.

64

**Instruction status:**

Instruction	j	k	R2	Exec		Write	Issue	Comp	Result	Load1	Busy	Address
				1	3							
LD	F6	34+	R2	1	3	4	1			No		
LD	F2	45+	R3	2	4		2			Yes	45+R3	
MULTD	F0	F2	F4	3			3			No		
SUBD	F8	F6	F2	4			4					
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									

**Reservation Stations:**

Time	Name	Busy	Op	S1	S2	RS	RS
				Vi	Vk	Qi	Qk
	Add1	Yes	SUBD	M(A1)		Load2	
	Add2	No					
	Add3	No					
	Mult1	Yes	MULTD		R(F4)	Load2	
	Mult2	No					

**Register result status:**

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
4	Mult1	Load2		M(A1)	Add1				

- Load2 completing; what is waiting for Load2?

MK  
MORGAN KAUFMANN

Copyright © 2012, Elsevier Inc. All rights reserved.

65

**Instruction status:**

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	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					

**Reservation Stations:**

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
2	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	No					
	Add3	No					
10	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD	M(A1)	Mult1		

**Register result status:**

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
5	FU	Mult1	M(A2)		M(A1)	Add1	Mult2		

- Timer starts down for Add1, Mult1

**Instruction status:**

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	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	6				

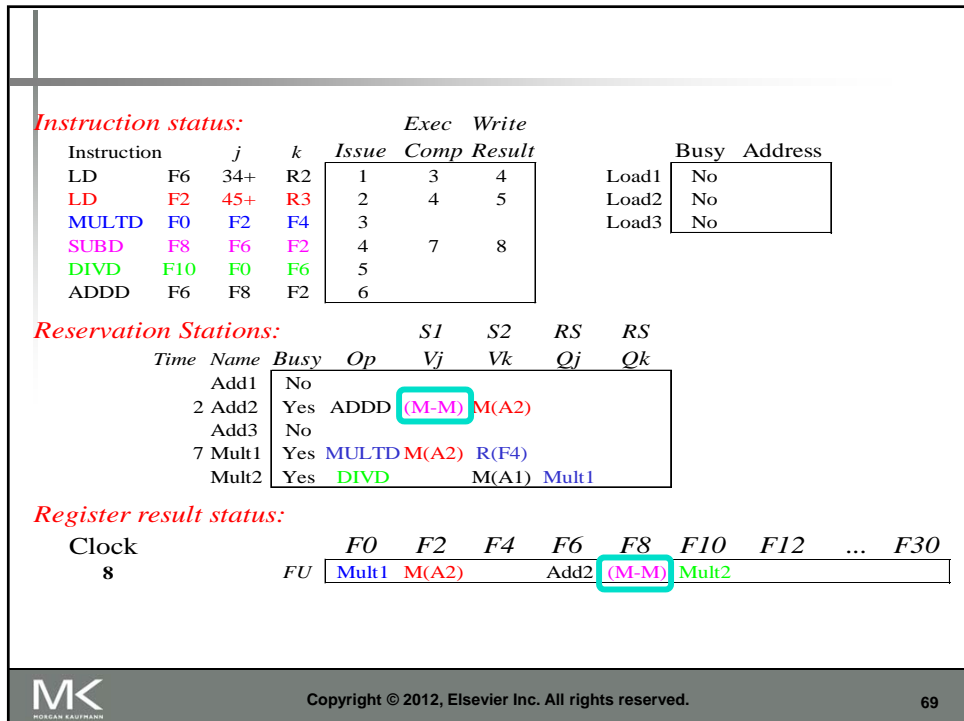
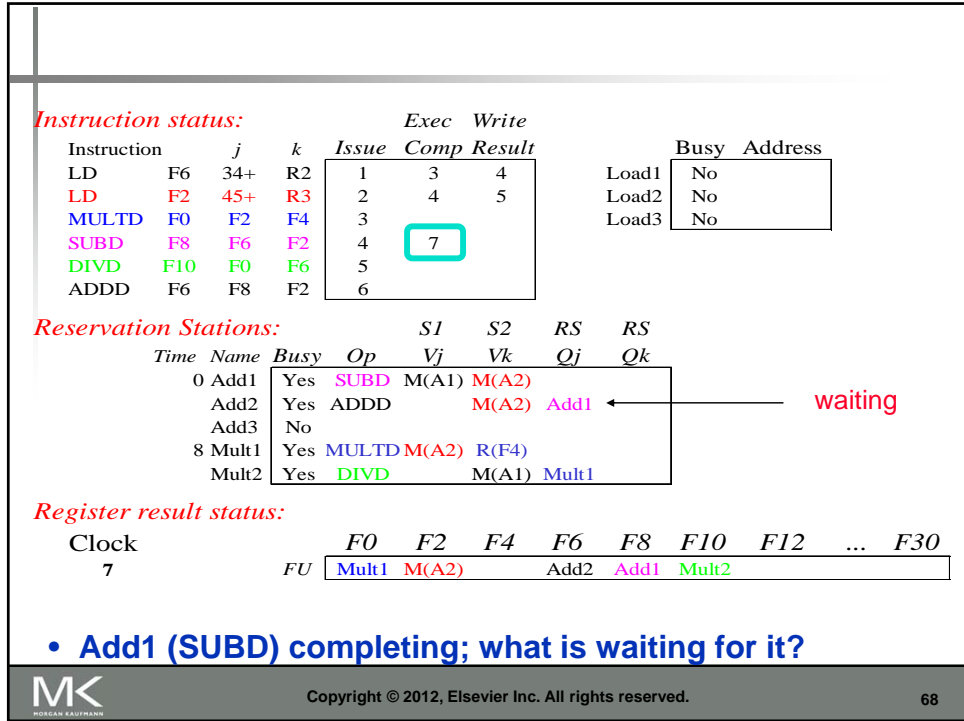
**Reservation Stations:**

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
1	Add1	Yes	SUBD	M(A1)	M(A2)		
	Add2	Yes	ADDD		M(A2)	Add1	
	Add3	No					
9	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD	M(A1)	Mult1		

**Register result status:**

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
6	FU	Mult1	M(A2)		Add2	Add1	Mult2		

- Issue ADDD here despite name dependency on F6?



**Instruction status:**

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

**Reservation Stations:**

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
1	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
6	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD	M(A1)	Mult1		

**Register result status:**

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
9	FU	Mult1	M(A2)		Add2	(M-M)	Mult2		



**Instruction status:**

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

**Reservation Stations:**

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
0	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
5	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD	M(A1)	Mult1		

**Register result status:**

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
10	FU	Mult1	M(A2)		Add2	(M-M)	Mult2		

- Add2 (ADDD) completing; what is waiting for it?





**Instruction status:**

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	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:**

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
4	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

**Register result status:**

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
11	FU	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2			

- Write result of ADDD here?
- All quick instructions complete in this cycle!

**Instruction status:**

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	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:**

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
3	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD		M(A1)	Mult1	

**Register result status:**

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
12	FU	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2			

**Instruction status:**

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

**Reservation Stations:**

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
2	Mult1	Yes	MUL TD	M(A2)	R(F4)		
	Mult2	Yes	DIVD	M(A1)	Mult1		

**Register result status:**

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
13	FU	Mult1	M(A2)	(M-M+M)	(M-M)	Mult2			

**Instruction status:**

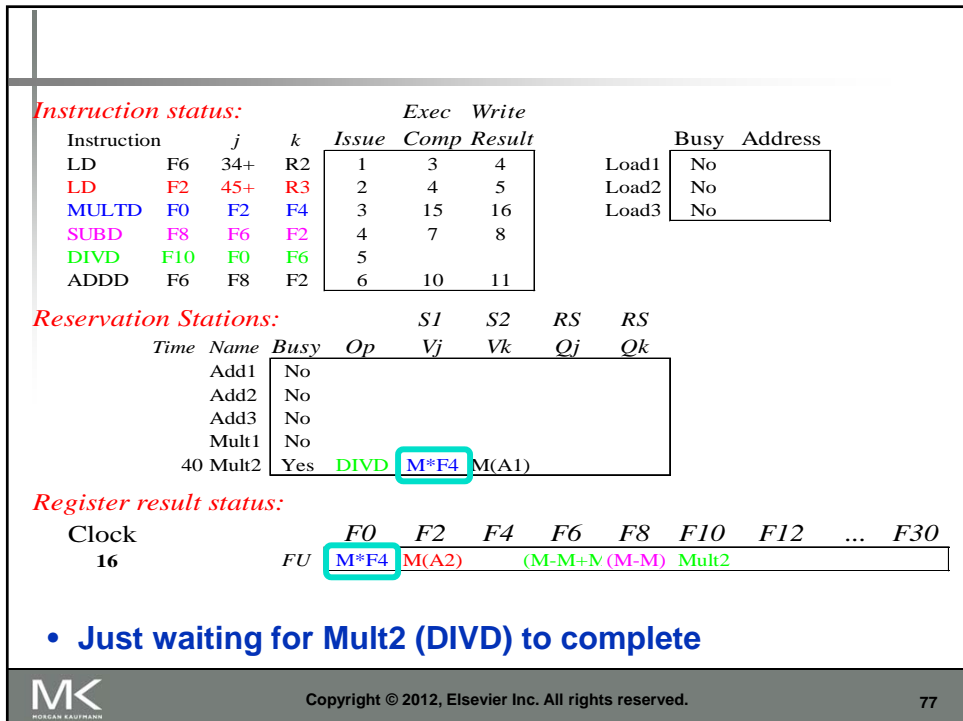
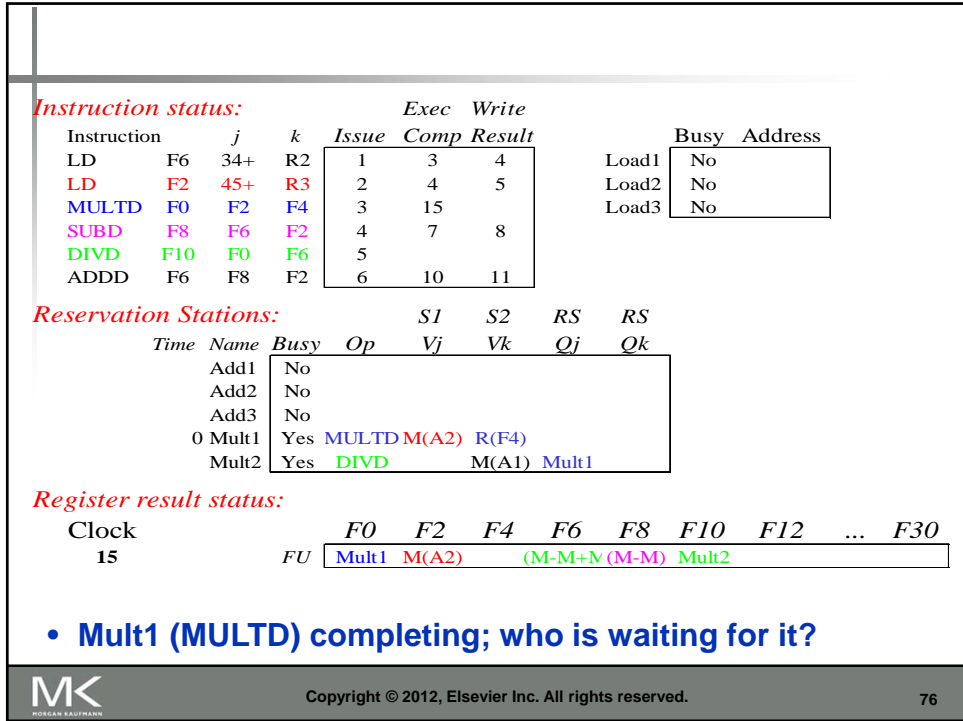
Instruction	j	k	Exec Write			Load1	Load2	Load3	Busy	Address
			Issue	Comp	Result					
LD	F6	34+	R2	1	3	4		No		
LD	F2	45+	R3	2	4	5		No		
MUL TD	F0	F2	F4	3				No		
SUBD	F8	F6	F2	4	7	8				
DIVD	F10	F0	F6	5						
ADDD	F6	F8	F2	6	10	11				

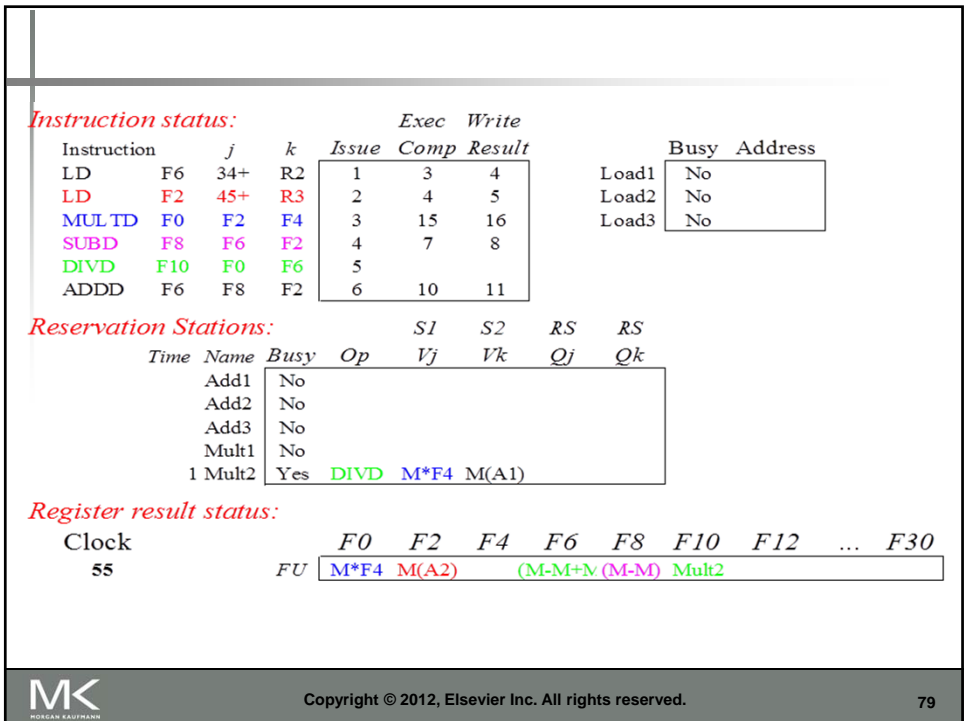
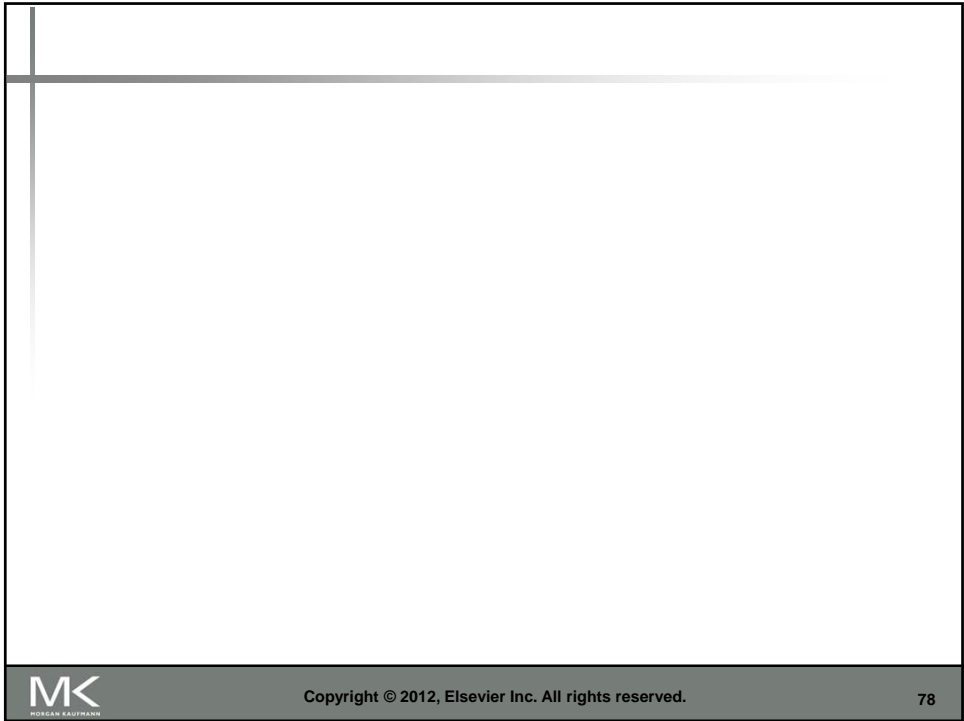
**Reservation Stations:**

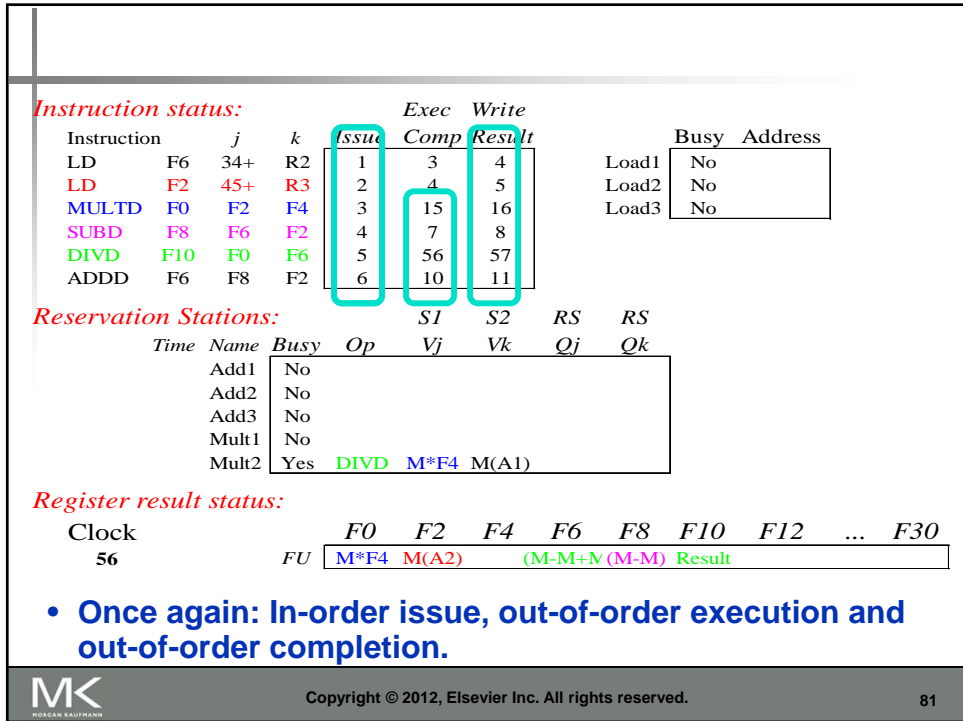
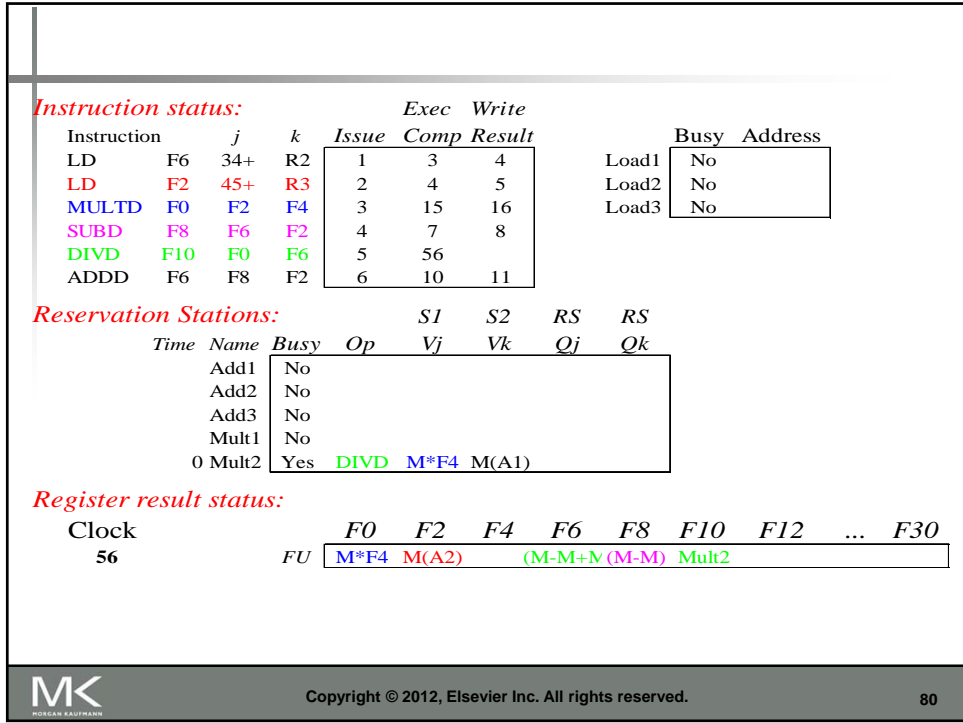
Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
1	Mult1	Yes	MUL TD	M(A2)	R(F4)		
	Mult2	Yes	DIVD	M(A1)	Mult1		

**Register result status:**

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
14	FU	Mult1	M(A2)	(M-M+M)	(M-M)	Mult2			







## Tomasulo's Algorithm

- Load and stores could be done out of order provided they access different memory locations.
- If they access same location, must preserve order (WAR, RAW, or WAW).
- If address calculation is done in program order, load/store can check if any uncompleted load/store share the same address
- Either wait or forward if possible.