## Advanced Computer Architectures

(High Performance Processors and Systems)

## Dynamic Scheduling: Tomasulo Algorithm

Politecnico di Milano

V1

Alessandro Verosimile < alessandro.verosimile@polimi.it> Marco D. Santambrogio <marco.santambrogio@polimi.it>

## EXE ON APRIL 23: ONLINE

# EXE ON APRIL 23: ONLINE ON APRIL 28, 2025 @6PM

# EXE ON APRIL 23: ONLINE ON APRIL 28, 2025 @6PM

DEIB GONFERENGE ROOM

# EXE ON APRIL 23: ONLINE ON APRIL 28, 2025 @6PM

DEIB GONFERENCE ROOM

## HPPS/AGA MIDTERM

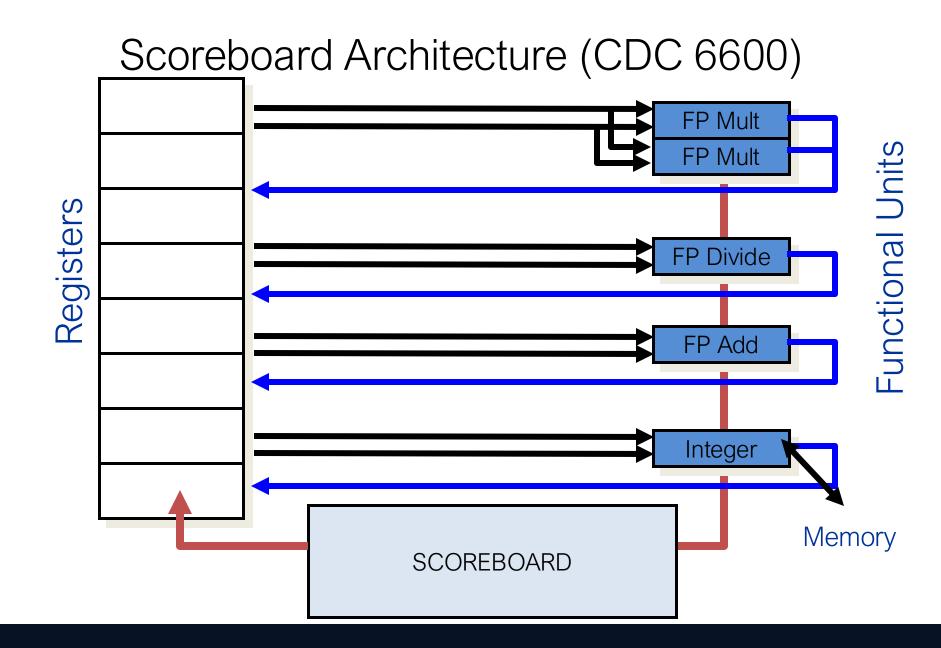
## Outline

- Tomasulo algorithm
- Comparison between Scoreboard and Tomasulo

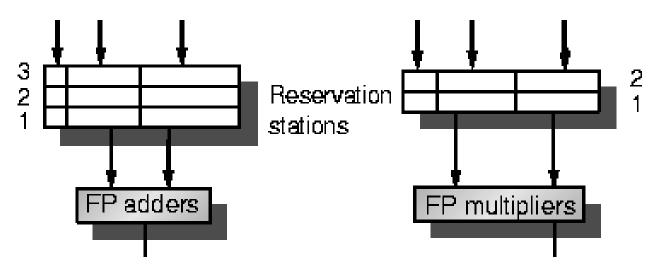
## Tomasulo Algorithm

- Another dynamic algorithm: allows execution to proceed in the presence of dependences
- Invented at IBM 3 years after CDC 6600 for the IBM 360/91
- Same Goal: high performance w/o special compilers
- Lead to:
  - Alpha 21264, HP 8000, MIPS 10000, Pentium II,
     PowerPC 604

- The control logic and the buffers are distributed with FUs (vs. centralized in scoreboard)
- Operand buffers are called Reservation Stations



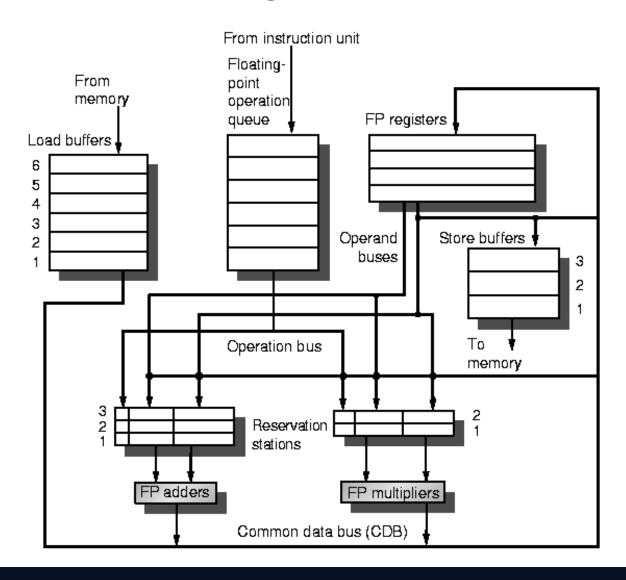
- The control logic and the buffers are distributed with FUs (vs. centralized in scoreboard)
- Operand buffers are called Reservation Stations
- Each instruction is an entry of a reservation station



- The control logic and the buffers are distributed with FUs (vs. centralized in scoreboard)
- Operand buffers are called Reservation Stations
- Each instruction is an entry of a reservation station
- Its operands are replaced by values or pointers (Register Renaming)

- Register Renaming allows to:
  - Avoid WAR and WAW hazards
  - Reservation stations are more than registers (so can do better optimizations than a compiler).
- Results are dispatched to other FUs through a Common Data Bus (CDB)
- Load/Stores treated as FUs

## Tomasulo Algorithm for an FPU



### Reservation Station Components

- Tag identifying the RS
- OP = the operation to perform on the component
- Vj, Vk = Value of the source operands
- Qj,Qk = Pointers to RS that produce Vj,Vk
   Zero value = Source op. is already available in Vj or Vk
- Busy = Indicates RS Busy
- Note: Only one of V-field or Q-field is valid for each operand

## **ISSUE**: 3-stage Tomasulo

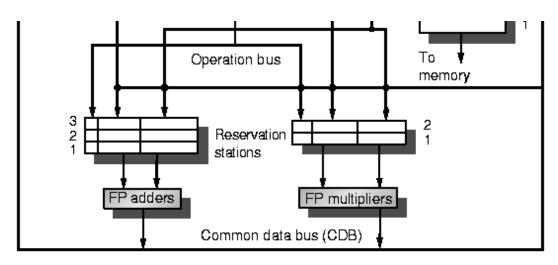
- Get an instruction I from the queue.
  - If it is an FP op. Check if a RS is empty (i.e., check for structural hazards)
- Rename registers
- WAR resolution
  - If I writes Rx, read by an instruction K already issued, K knows already the value of Rx or knows what instruction will write it. So the RF can be linked to I
- WAW resolution
  - Since we use in-order issue, the RF can be linked to I

## **EXECUTION**: 3-stage Tomasulo

- When both operands are ready then execute.
- If not ready, watch the common data bus for results.
  - By delaying execution until operands are available, RAW hazards are avoided.
     Notice that several instructions could become ready in the same clock cycle for the same FU.
- Load and stores: two-step process
  - First step: compute effective address, place it in load or store buffer.
  - Loads in Load Buffer execute as soon as memory unit is available
  - Stores in store buffer wait for the value to be stored before being sent to memory unit.

## **WRITE**: 3-stage Tomasulo

- When result is available: write on Common Data Bus and from there into RF and into all RSs (including store buffers) waiting for this result
- Stores write data to memory during this stage
- Mark reservation stations available.

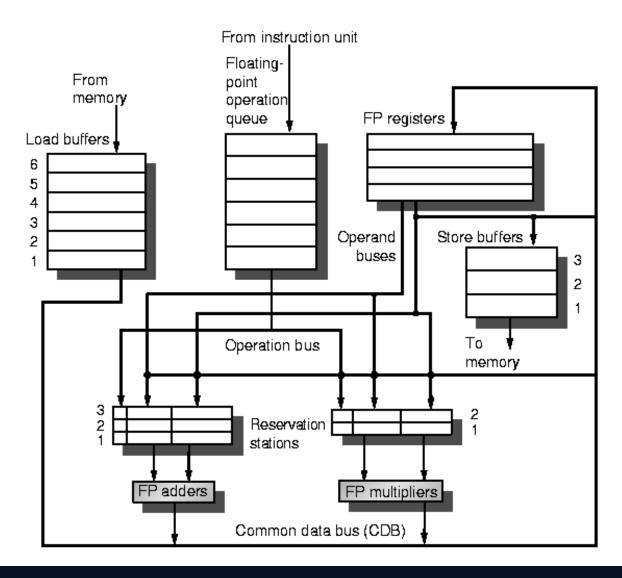


## Tomasulo Example

```
Instruction status:
                                          Write
                                    Exec
                                                               Busy Address
   Instruction
                              Issue
                                    Comp Result
                         k
   LD
             F6
                  34 +
                        R2
                                                        Load1
                                                                 No
   LD
             F2
                  45+
                        R3
                                                        Load2
                                                                 No
   MULTD
             F<sub>0</sub>
                   F2
                         F4
                                                        Load3
                                                                 No
   SUBD
             F8
                   F6
                         F2
   DIVD
            F10
                   F0
                         F6
                         F2
   ADDD
             F6
                   F8
Reservation Stations:
                                             S2
                                      SI
                                                   RS
                                                          RS
           Time Name Busy
                                      V_j
                                             Vk
                                                    Q_j
                                                          Qk
                               Op
                 Add1
                        No
                 Add2
                        No
                 Add3
                        No
                 Mult1
                        No
                 Mult2
                        No
```

#### Register result status:

## Tomasulo Algorithm for an FPU



## Reservation Station Components

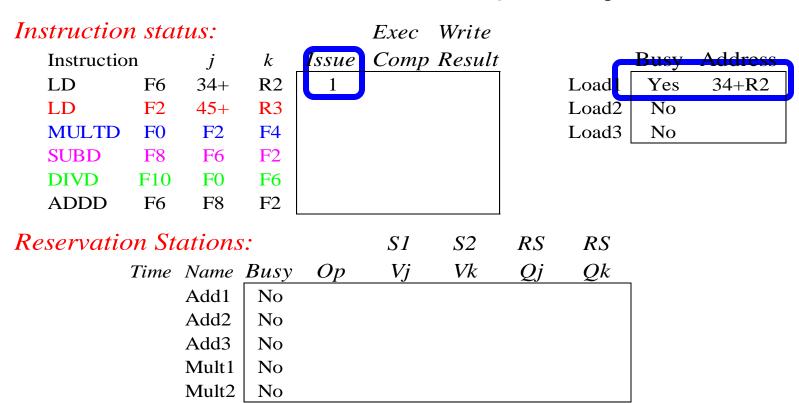
- Tag identifying the RS
- OP = the operation to perform on the component.
- Vj, Vk = Value of the source operands
- Qj,Qk = Pointers to RS that produce Vj,Vk
   Zero value = Source op. is already available in Vj or Vk
- Busy = Indicates RS Busy
- Note: Only one of V-field or Q-field is valid for each operand

## Tomasulo Example

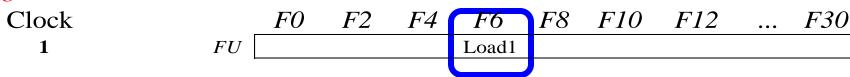
```
Instruction status:
                                          Write
                                    Exec
                                                               Busy Address
   Instruction
                              Issue
                                    Comp Result
                         k
   LD
             F6
                  34 +
                        R2
                                                        Load1
                                                                 No
   LD
             F2
                  45+
                        R3
                                                        Load2
                                                                 No
   MULTD
             F<sub>0</sub>
                   F2
                         F4
                                                        Load3
                                                                 No
   SUBD
             F8
                   F6
                         F2
   DIVD
            F10
                   F0
                         F6
                         F2
   ADDD
             F6
                   F8
Reservation Stations:
                                             S2
                                      SI
                                                   RS
                                                          RS
           Time Name Busy
                                      V_j
                                             Vk
                                                    Q_j
                                                          Qk
                               Op
                 Add1
                        No
                 Add2
                        No
                 Add3
                        No
                 Mult1
                        No
                 Mult2
                        No
```

#### Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F30

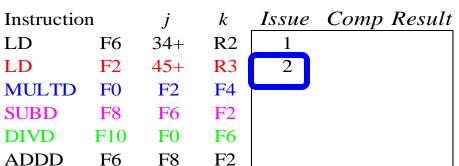


#### Register result status:



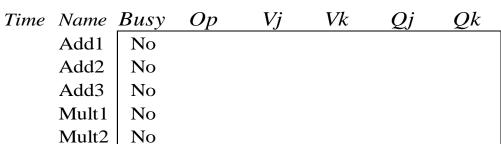
Exec Write

#### Instruction status:



	Busy	Address
Load	Yes	34+R2
Load <sup>2</sup>	Yes	45+R3
Load3	No	

#### Reservation Stations:

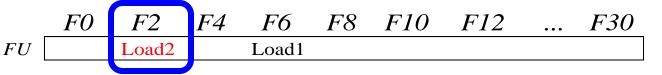


SI

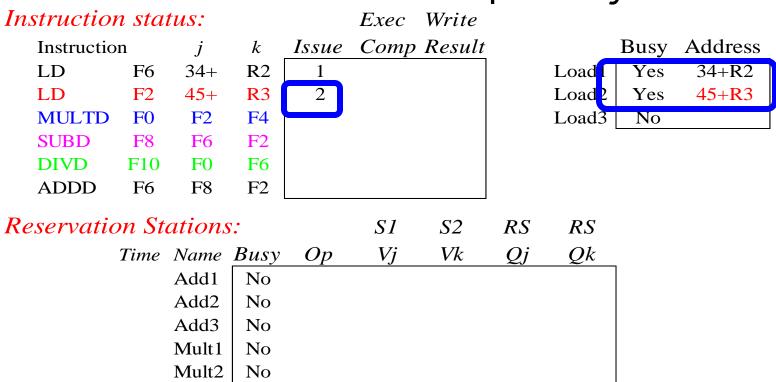
*S*2

RS

#### Register result status:



RS



Register result status:

Clock F0 F2 F4 F6 F8 F10 F12 ... F30
2 FU Load2 Load1

Note: Unlike 6600, can have multiple loads outstanding

*S*2

RS

RS

Exec Write

#### Instruction status:

Instructio	n	j	k	Issue	Comp	Result
LD	F6	34+	R2	1	3	
LD	F2	45+	<b>R</b> 3	2		
<b>MULTD</b>	FO	F2	F4	3		
SUBD	F8	F6	F2			
DIVD	F10	F0	<b>F6</b>			
ADDD	F6	F8	F2			

	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

#### Reservation Stations:

Time Name Busy Op $V_j$ Vk $Q_{j}$ QkAdd1 No Add2 No Add3 No Mult1 Yes MULTD R(F4) Load2 Mult2

SI

#### Register result status:

Clock 3

F0 F2 F4 F6 F8 F10 F12 ... F30

FU Mult1 Load2 Load1

Exec Write

#### Instruction status:

SUBD

Load1 Yes 34+R	2
Load2 Yes 45+R	3
Load3 No	

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

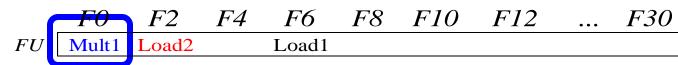
Roadlacompleting: what is waiting for Load1?

```
Time Name Busy Op Vj Vk Qj Qk

Add1 No
Add2 No
Add3 No
Mult1 Yes MULTD R(F4) Load2

Mult2 No
```

#### Register result status:



F30

## Tomasulo Example Cycle 4

Write

*S*2

Exec

#### Instruction status:

Instruction Comp Result kIssue LDF6 34 +R2 3 4 LD 45 +**R3** 4 F2 **MULTD** F0 F2 F4 **SUBD** F8 F6 F2 4 DIVD F10 F<sub>0</sub> F6

F2

F8

	Busy	Address
Load1	No	
Load2	Yes	45+R3
Load3	No	

#### Reservation Stations:

F6

 $V_i$ VkTime Name Busy Op $Q_{j}$ QkLoad2 Add1 **SUBD** M(A1)Yes Add2 No Add3 No Mult1 Yes MULTD R(F4) Load2 Mult2 No

SI

#### Register result status:

Clock 4

**ADDD** 

 F0
 F2
 F4
 F6
 F8
 F10
 F12

 FU
 Mult1
 Load2
 M(A1)
 Add1

RS

RS

```
Instruction status:
                                     Exec
                                          Write
                                     Comp Result
   Instruction
                              Issue
                         k
   LD
             F6
                  34 +
                         R2
                                       3
                                              4
                  45 +
                         R3
                                       4
   LD
             F2
   MULTD
                   F2
                         F4
             FO
   SUBD
             F8
                         F2
                   F6
   DIVD
            F10
                   FO
                         F6
   ADDD
                         F2
             F6
                   F8
```

	Busy	Address
Load1		
Load2	Yes	45+R3
Load3		

#### Reservation Stations:

 $V_i$ Time Name Busy Vk $Q_j$ QkOp**SUBD** M(A1)Load2 Add1 Yes Add2 No Add3 No Mult1 Yes MULTD R(F4) Load2 Mult2 No

SI

#### Register result status:

*S*2

RS

RS

Load2 completing; what is waiting for Load2?

#### Instruction status:

Instructio	n	j	k	Issue	Comp	Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	<b>F</b> 4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	FO	F6	5		
ADDD	F6	F8	F2			

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:

```
Time Name Busy
                              Vk
                                          Ok
                 Op
                                    Oi
   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 5

F

F0	F2
N /T14 1	TA /T / A

Mult1 M(A2) M(A1) Add1 Mult2

#### Instruction status:

Instructio	n	$\dot{J}$	$\boldsymbol{k}$	Issue	Comp	Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	<b>R</b> 3	2	4	5
MULTD	F0	F2	F4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	FO	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:

Time Name	Busy	Op	Vj	Vk	Qj	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 6

$$FO$$
  $F$ 
 $U$  Mult1  $M$ 

*F6* 

Mult2

FU Mult1 M(A2) Add2 Add1

Issue ADDD here vs. scoreboard

Exec Write

#### Instruction status:

Instructio	n	j	k	Issue	Comp	Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	FO	F2	F4	3		
SUBD	F8	F6	F2	4	7	
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2.	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

RS

#### Reservation Stations:

Time Name	Busy	Op	Vj	Vk	Qj	Qk
0 Add1	Yes	SUBD	M(A1)	M(A2)		
Add2	Yes	ADDD		M(A2)	Add1	
Add3	No					
8 Mult1	Yes	MULTD	M(A2)	R(F4)		
Mult2	Yes	DIVD		M(A1)	Mult1	

SI

#### Register result status:

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

Add1 completing; what is waiting for it?

Write

*S*2

RS

RS

Exec

#### Instruction status:

Comp Result Issue Instruction kR2 LD F6 34+1 3 4 LD 45 +**R**3 5 F2. MULTD F0 F2 F4 3 **SUBD** F2 4 7 8 F8 F6 **DIVD** F10 5 FO F6

F2

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:

F6

F8

VkTime Name Busy Vi $Q_{j}$ OkOpAdd1 No 2 Add2 Yes ADDD (M-M) M(A2) Add3 No 7 Mult1 Yes MULTD M(A2) R(F4)Mult2 Yes DIVD M(A1) Mult1

SI

#### Register result status:

Clock 8

**ADDD** 

FU

F0 F2

F6 F8

F10

*F12* 

... F30

Mult1 M(A2) Add2 (M-M) Mult2

F4

Write

Exec

#### Instruction status:

Instructio	n	$\dot{J}$	$\boldsymbol{k}$	Issue	Comp	Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	<b>R</b> 3	2	4	5
MULTD	FO	F2	<b>F</b> 4	3		
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	<b>F6</b>	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:

VkTime Name Busy OpQj OkAdd1 No 1 Add2 Yes ADDD (M-M) M(A2) Add3 No 6 Mult1 Yes MULTD M(A2) R(F4)Mult2 **DIVD** Yes M(A1) Mult1

SI

#### Register result status:

Clock *F*2 *F*6 F8 *F10* F12 F0*F4* F30 9 FUMult1 M(A2)Add2 (M-M)Mult2

*S*2

RS

RS

```
Instruction status:
                                     Exec
                                           Write
                                                                 Busy Address
                                     Comp Result
   Instruction
                              Issue
                          k
   LD
             F6
                   34+
                         R2
                                        3
                                               4
                                                                  No
                                                          Load1
                  45 +
   LD
             F2
                         R3
                                        4
                                               5
                                                          Load2
                                                                  No
   MULTD
             F0
                   F2
                         F4
                                                          Load3
                                                                  No
   SUBD
                   F6
                         F2
                                               8
   DIVD
             F10
                   F<sub>0</sub>
                         F6
   ADDD
                   F8
                         F2
             F6
                                       10
Reservation Stations:
                                              S2
                                       SI
                                                    RS
                                                           RS
            Time Name Busy
                               Op
                                       Vi
                                              Vk
                                                     Q_{j}
                                                           Qk
                 Add1
                         No
                0 \text{ Add} 2
                        Yes ADDD (M-M) M(A2)
                 Add3
                         No
                5 Mult1
                         Yes MULTD M(A2) R(F4)
```

#### Register result status:

F10 F12 Clock F0F2F4 *F6* F8 F30 10 FUMult1 M(A2)Add2 (M-M)Mult2

M(A1) Mult1

Add2 completing; what is waiting for it?

Yes

**DIVD** 

Mult2

#### Instruction status:

Instructio	n	j	k	Issue	Comp	Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	FO	F2	F4	3		
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	<b>F6</b>	5		
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:

```
        Time
        Name
        Busy
        Op
        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:

FU

F30

$$(M-M+M)$$
  $(M-M)$   $Mult2$ 

11

*S*2

RS

RS

#### Instruction status: Write Exec Comp Result Instruction kIssue R2 LD F6 34 +3 4 LD F2 45 +**R**3 4 5 **MULTD** F0 F2 F4 **SUBD** 8 F8 F2 7

F6

F2

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:

F10

F6

F0

F8

**DIVD** 

**ADDD** 

10

SI

5

6

#### Register result status:

Clock *F*2 F8 F10 *F12* F30 F0*F4 F6* 12 FUMult1 M(A2)(M-M)Mult2 (M-M+M)

Write

11

*S*2

RS

RS

Exec

10

SI

#### Instruction status:

F2

F8

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:

F6

**ADDD** 

#### Register result status:

Clock F8 F10 F12 F0F2*F4* F6 F30 **13** FUMult1 M(A2)(M-M)Mult2 (M-M+M)

```
Instruction status:
                                         Write
                                   Exec
                                   Comp Result
                                                                   Address
                             Issue
                                                             Busy
   Instruction
                        k
                        R2
   LD
            F6
                  34 +
                                      3
                                            4
                                                       Load1
                                                               No
                               1
   LD
                  45+
                        R3
            F2
                                      4
                                            5
                                                       Load2
                                                               No
                               3
   MULTD
            F0
                  F2
                        F4
                                                       Load3
                                                               No
   SUBD
                               4
                                      7
            F8
                  F6
                        F2
                                            8
            F10
                               5
   DIVD
                  F0
                        F6
   ADDD
            F6
                  F8
                        F2
                               6
                                     10
                                            11
Reservation Stations:
                                           S2
                                     SI
                                                  RS
                                                        RS
                                     V_i
                                           Vk
                                                        Qk
           Time Name Busy
                              Op
                                                  Q_{j}
                 Add1
                        No
                 Add2
                        No
                 Add3
                        No
               1 Mult1
                       Yes MULTD M(A2) R(F4)
                                          M(A1) Mult1
```

**DIVD** 

### Register result status:

Mult2

Yes

Clock		F0	<i>F2</i>	<i>F4</i>	<i>F6</i>	F8	F10	<i>F12</i>	•••	F30
14	FU	Mult1	M(A2)	(	(M-M+M)	(M-M)	Mult2			

#### Instruction status: ExecWrite Comp Result Busy Address Instruction kIssue LD 34 +3 No F6 R2 4 Load1 1 LD 45 +4 5 F2 **R**3 Load2 No **MULTD** F0 F2 F4 Load3 15 No 8 **SUBD** F8 F6 F2 4 **DIVD** F10 F0 F6 **ADDD** F6 F8 F2 6 10 11 Reservation Stations: SIS2RS RS ViVk $Q_j$ QkTime Name Busy OpAdd1 No Add2 No Add3 No 0 Mult1 Yes MULTD M(A2) R(F4)Mult2 **DIVD** Yes M(A1) Mult1

### Register result status:

Clock		FO	F2	<i>F4</i>	<i>F6</i>	F8	F10	<i>F12</i>	•••	F30
15	FU	Mult1	M(A2)		(M-M+M)	(M-M)	Mult2			

```
Instruction status:
                                         Write
                                   Exec
                                   Comp Result
                                                                   Address
   Instruction
                                                             Busy
                            Issue
                        k
   LD
                 34 +
                        R2
                                     3
                                            4
                                                               No
            F6
                                                      Load1
                               1
   LD
                 45 +
                        R3
                                            5
            F2
                                                      Load2
                                                              No
   MULTD
                  F2
                        F4
                                     15
                                                      Load3
            F0
                                           16
                                                              No
   SUBD
            F8
                  F6
                        F2
                              4
   DIVD
            F10
                  F0
                        F6
   ADDD
            F6
                  F8
                        F2
                              6
                                     10
                                           11
Reservation Stations:
                                    SI
                                           S2
                                                 RS
                                                        RS
                                     V_i
                                           Vk
                                                  Qi
                                                        Qk
           Time Name Busy
                             Op
                        No
                Add1
                Add2
                        No
                Add3
                       No
                Mult1
                        No
              40 Mult2
                       Yes
                            DIVD
                                   M*F4 M(A1)
```

### Register result status:

Clock		FO	F2	<i>F4</i>	<i>F6</i>	F8	F10	<i>F12</i>	•••	F30
16	FU	M*F4	M(A2)		(M-M+M)	(M-M)	Mult2			

Faster than light computation (skip a couple of cycles)



Write

*S*2

RS

RS

Exec

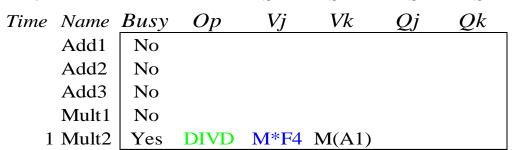
SI

#### Instruction status:

Instructio	n	j	k	Issue	Comp	Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MULTD	F0	F2	F4	3	15	16
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	<b>F6</b>	5		
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:



### Register result status:

Clock F0F2*F4* F6 *F*8 *F10 F12* F30 55 FUM\*F4 M(A2)(M-M)Mult2 (M-M+M)

```
Instruction status:
                                           Write
                                     Exec
                                     Comp Result
                                                                Busy
                                                                       Address
   Instruction
                              Issue
                          k
   LD
             F6
                   34 +
                         R2
                                        3
                                              4
                                                         Load1
                                                                  No
   LD
                   45 +
                         R3
                                              5
                                                         Load2
                                                                  No
   MULTD
             F<sub>0</sub>
                   F2
                         F4
                                       15
                                              16
                                                         Load3
                                                                  No
   SUBD
             F8
                         F2
                                              8
                   F6
                                5
   DIVD
             F10
                                       56
                   F0
                         F6
   ADDD
             F6
                   F8
                         F2
                                       10
                                              11
Reservation Stations:
                                       SI
                                             S2
                                                    RS
                                                           RS
            Time Name Busy
                                       V_i
                                              Vk
                                                     Oi
                                                           Ok
                               Op
                 Add1
                         No
                 Add2
                         No
                 Add3
                         No
                 Mult1
                         No
                0 Mult2
                         Yes
                              DIVD
                                     M*F4 M(A1)
```

Register result status:

*F10* F8 Clock F0F2F4 *F6* F12 *F30* **56** FUM\*F4 M(A2)(M-M)Mult2 (M-M+M)

Mult2 is completing; what is waiting for it?

Exec Write

#### Instruction status:

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:

SI

### Register result status:

Clock F0*F*2 F4F6 F8 F10 F12 F30 **56** M\*F4FUM(A2)(M-M)Result (M-M+M)

*S*2

RS

RS

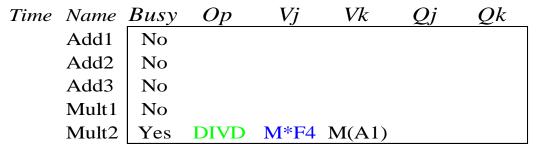
Exec Write

### Instruction status:

Instructio	n	j	$\boldsymbol{k}$	Issue	Comp	Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	<b>R</b> 3	2	4	5
MULTD	FO	F2	F4	3	15	16
SUBD	F8	F6	F2	4	7	8
DIVD	F10	F0	F6	5	56	57
ADDD	F6	F8	F2	6	10	11

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

#### Reservation Stations:



SI

### Register result status:

Clock 56

FU M\*F4 M(A2

*F4* 

*S*2

F8

RS

F10

*F12* 

. F30

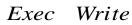
M(A2) (M-M+M) (M-M) Result

*F6* 

RS

#### Instruction status:

Instruction kLD 34 +R2 F6 LD F2. 45 +**R**3 **MULTD** F0 F2 F4 **SUBD** F8 F6 F2 **DIVD** F10 F0 F6 **ADDD** F2 F6 F8



		_		,,,,,,,
_	Issue		Comp	n Result
2	1		3	4
3	2		4	5
.	3		15	16
,	4		7	8
,	5		56	57
, [	6		10	11

	Busy	Address
Load1	No	
Load2	No	

No

#### Reservation Stations:

Op $V_j$ VkTime Name Busy  $Q_{j}$ QkAdd1 No Add2 No Add3 No Mult1 No Mult2 M\*F4 M(A1) Yes DIVD

SI

### Register result status:

Clock 56

FO M\*F

*F*2

*F4* 

*S*2

*F6* 

RS

F8 .

F10 F12

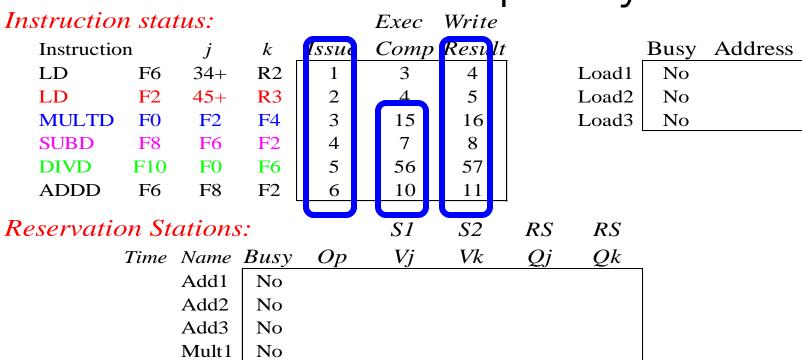
2 ... *F30* 

FU M\*F4 M(A2)

(M-M+M) (M-M) Result

Load3

RS



### Register result status:

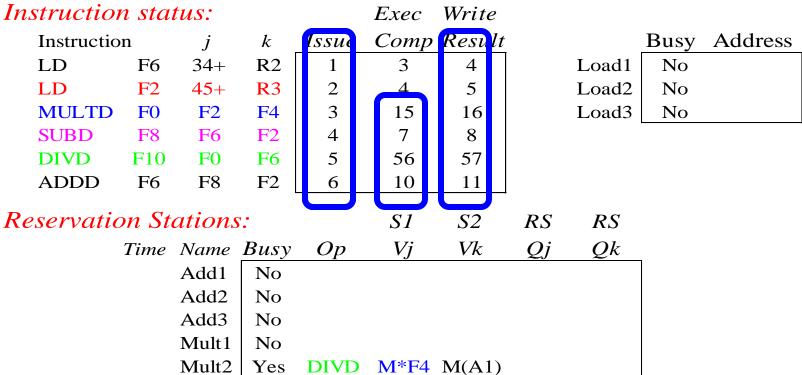
Mult2

Yes

DIVD

Clock F0*F*2 *F4 F6* F8 *F10* F12 F30 **56** M\*F4 FUM(A2)(M-M)Result (M-M+M)

M\*F4 M(A1)



### Register result status:

Clock F0F2*F4 F6* F8 F10F12 F30 **56** FUM\*F4 M(A2)Result (M-M+M)(M-M)

Once again: In-order issue, out-of-order execution and completion.

### Tomasulo (IBM) versus Scoreboard (CDC)

- Issue window size=14
- No issue on structural hazards
- WAR, WAW avoided with renaming
- Broadcast results from FU
- Control distributed on RS
- Allows loop unrolling in HW

- Issue window size=5
- No issue on structural hazards
- Stall the completion for WAW and WAR hazards
- Results written back on registers.
- Control centralized through the Scoreboard.

- Control & buffers distributed with Function Units (FU) vs. centralized in scoreboard;
  - FU buffers called "reservation stations"; have pending operands

- Control & buffers distributed with Function Units (FU) vs. centralized in scoreboard;
  - FU buffers called "reservation stations"; have pending operands
- Registers in instructions replaced by values or pointers to reservation stations(RS); called register renaming;
  - avoids WAR, WAW hazards
  - More reservation stations than registers, so can do optimizations compilers can't

- Control & buffers distributed with Function Units (FU) vs. centralized in scoreboard;
  - FU buffers called "reservation stations"; have pending operands
- Registers in instructions replaced by values or pointers to reservation stations(RS); called register renaming;
  - avoids WAR, WAW hazards
  - More reservation stations than registers, so can do optimizations compilers can't
- Results to FU from RS, not through registers, over Common Data Bus that broadcasts results to all FUs

- Control & buffers distributed with Function Units (FU) vs. centralized in scoreboard;
  - FU buffers called "reservation stations"; have pending operands
- Registers in instructions replaced by values or pointers to reservation stations(RS); called register renaming;
  - avoids WAR, WAW hazards
  - More reservation stations than registers, so can do optimizations compilers can't
- Results to FU from RS, not through registers, over Common Data Bus that broadcasts results to all FUs
- Load and Stores treated as FUs with RSs as well

- Control & buffers distributed with Function Units (FU) vs. centralized in scoreboard;
  - FU buffers called "reservation stations"; have pending operands
- Registers in instructions replaced by values or pointers to reservation stations(RS); called register renaming;
  - avoids WAR, WAW hazards
  - More reservation stations than registers, so can do optimizations compilers can't
- Results to FU from RS, not through registers, over Common Data Bus that broadcasts results to all FUs
- Load and Stores treated as FUs with RSs as well
- Integer instructions can go past branches, allowing FP ops beyond basic block in FP queue

# Compare to Scoreboard Cycle 62

Instruction	n sta	tus:				Read	Exec	Write	?
Instructio	n	j	k	1	ssue	Oper	Comp	Resul	lt
LD	F6	34+	R2	П	1	2	3	4	
LD	F2	45+	R3		5	6	7	8	
MULTD	F0	F2	F4		6	9	19	20	
SUBD	F8	F6	F2		7	9	11	12	
DIVD	F10	F0	F6		8	21	61	62	
ADDD	F6	F8	F2	I	13	14	16	22	

	Exec Write						
Issue Comp Result							
1		3		4			
2		4		5			
3		15		16			
4		7		8			
5		56		57			
6		10		11			

# Compare to Scoreboard Cycle 62

Instruction status:					R	ead	Exec		Write	
Instructio	n	j	$\boldsymbol{k}$	Issue	0	per	Com	p.	Resul	t
LD	F6	34+	R2	1		2	3		4	
LD	F2	45+	R3	5		6	7		8	
MULTD	F0	F2	F4	6		9	19		20	
SUBD	F8	F6	F2	7	Ш	9	11		12	
DIVD	F10	F0	F6	8		21	61		62	
ADDD	F6	F8	F2	13	Ц	14	16		22	

Exec Write						
Issue Comp Result						
1	3	4				
2	4	5				
3	15	16				
4	7	8				
5	56	57				
6	10	11				

Why take longer on scoreboard/6600?

Structural Hazards

Lack of forwarding

### Tomasulo Drawbacks

- Complexity
  - Large amount of hardware
  - -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

# Summary

- HW exploiting ILP
  - Works when can't know dependence at compile time.
  - Code for one machine runs well on another
- 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)
- Lasting Contributions
  - Dynamic scheduling
  - Register renaming
  - Load/store disambiguation

### Advanced Computer Architectures

(High Performance Processors and Systems)

# Dynamic Scheduling: Tomasulo Algorithm

Politecnico di Milano