DYNAMIC SCHEDULING

Why Dynamic Scheduling?

- All the static(complier) techniques discussed so far use in-order instruction issue.
- That means that if an instruction is stalled in the pipeline, no later instructions can proceed.
- With in-order issue, if two instructions have a hazard between them, the pipeline will stall, even if there are later instructions that are independent and would not stall.
- Compiler attempts to schedule the instructions, called static scheduling.
- Several early processors used another approach, called *dynamic scheduling*, whereby the hardware rearranges the instruction execution to reduce the stalls.

Example:

```
DIVD  F0,F2,F4
ADDD  F10,F0,F8
SUBD  F12,F8,F14
```

Advantages of Dynamic Scheduling

- Enables handling in compile time
- Simplifies compiler

- Out-of-order execution => out-of-order completion.
- When instructions execute out of order, it may arise WAR hazard and WAW hazard.
- Example:

```
DIVD  F0,F2,F4
ADDD  F10,F0,F8
SUBD  F8, F8, F14
```

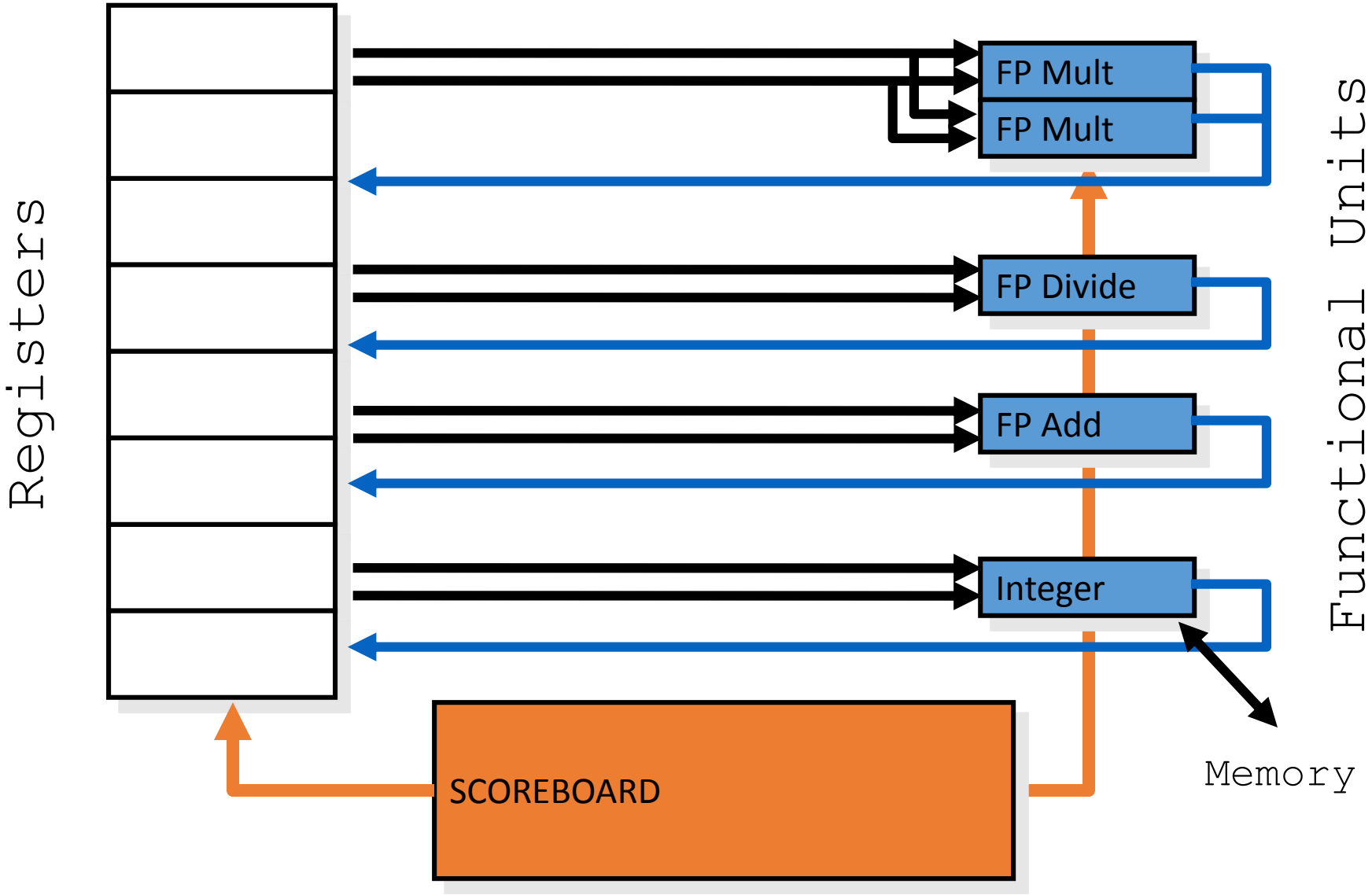
Instruction Parallelism by HW

- dynamically scheduled pipeline:: all instructions pass through issue stage in order (in-order issue)
- Enables out-of-order execution and allows out-of-order completion
- Will distinguish when an instruction *begins execution* and when it *completes execution*; in between instruction *in execution*

Dynamic Scheduling by Scoreboard:

- To implement out-of-order execution, ID stage must be split into two stages:
  1. Issue—decode instructions, check for structural hazards
  2. Read operands—wait until no data hazards, then read operands
- Scoreboards is first used in CDC6600 in 1963
- Scoreboard is a data structure contains set of registers used in instruction.
- Scoreboard decides
  - When to issue instruction
  - When to execute it
  - When to write registers (avoid WAR hazard)
- CDC 6600: In order issue, out-of-order execution (when there are no conflicts and the hardware is available). , out-of-order commit (or completion)
  - No forwarding

Scoreboard Architecture (CDC 6600)



- Integer- 1 clock cycle
- Add- 2 clock cycles
- Multi: 10 clock cycles
- Div: 40 clock cycles

Scoreboard Implications

- Out-of-order completion => WAR, WAW hazards?
- Solutions for WAR:
  - Stall write back until registers have been read
  - Read registers only during Read Operands stage
- Solution for WAW:
  - Detect hazard and stall issue of new instruction until other instruction completes
- Need to have multiple instructions in execution phase => multiple execution units or pipelined execution units
- Scoreboard keeps track of dependencies between instructions that have already issued.

Four Stages of Scoreboard Control

- Issue—decode instructions & check for structural hazards (ID1)
  - Instructions issued in program order (for hazard checking)
  - Don't issue if structural hazard
  - Don't issue if instruction is output dependent on previously issued but uncompleted instruction (no WAW hazards)
- Read operands—wait until no data hazards, then read operands (ID2)
  - All real dependencies (RAW hazards) resolved in this stage. Wait for instructions to write back data.
  - No data forwarding

- Execution—operate on operands (EX)
  - Functional unit begins execution upon receiving operands. When result is ready, scoreboard notified execute complete
- Write result—finish execution (WB)
  - Stall until no WAR hazards with previous instructions:

Example:

```
DIVD F0,F2,F4
ADDD F10,F0,F8
SUBD F8,F8,F14
```

CDC 6600 scoreboard would stall SUBD until ADDD reads operands

Dynamic Scheduling

Using A Scoreboard

Three Parts of the Scoreboard

1. Instruction status—Indicates which of 4 steps the instruction is in
2. Functional unit status—Indicates the state of the functional unit (FU). 9 fields for each functional unit

   Busy—Indicates whether the unit is busy or not

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

   Fi—Destination register

   Fj, Fk—Source-register numbers

   Qj, Qk—Functional units producing source registers Fj, Fk

   Rj, Rk—Flags indicating when Fj, Fk are ready and not yet read. Set to **No** after operands are read

3. Register result status—Indicates which functional unit will write each register, if one exists. Blank when no pending instructions will write that register

Scoreboard Example

Instruction status

Instruction		j	k	Issue	Read operanc	Executi complet	Write Result
LD	F6	34+	R2				
LD	F2	45+	R3				
MULT	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j Qj	FU for k Qk	Fj? Rj	Fk? Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Register result status

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

Dynamic Scheduling

Using A Scoreboard

Instruction status

Instruction		j	k	Issue	Read operand	Executic complet	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	62
ADDD	F6	F8	F2	13	14	16	22

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j Qj	FU for k Qk	Fj? Rj	Fk? Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
0	Divide	No								

Register result status

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

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 1

Issue LD #1

Shows in which cycle the operation occurred.

Instruction status

Instruction		j	k	Issue	Read operand	Executic complet	Write Result
LD	F6	34+	R2	1			
LD	F2	45+	R3				
MULT	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j Qj	FU for k Qk	Fj? Rj	Fk? Rk
	Integer	Yes	Load	F6		R2				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
1	FU				Integer					

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 2

LD #2 can't issue since integer unit is busy.
MULT can't issue because we require in-order issue.

Instruction status

Instructio	j	k	Issue	Read operai	Execu compl	Write Result
LD F6	34+	R2	1	2		
LD F2	45+	R3				
MUL F0	F2	F4				
SUB F8	F6	F2				
DIVI F10	F0	F6				
ADD F6	F8	F2				

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foi Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	Yes	Load	F6		R2				No
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Register result status

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

Scoreboard Example Cycle 3

Instruction status

Instructio	j	k	Issue	Read operar	Execu compl	Write Result
LD F6	34+	R2	1	2	3	
LD F2	45+	R3				
MUL F0	F2	F4				
SUB F8	F6	F2				
DIVI F10	F0	F6				
ADD F6	F8	F2				

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	Yes	Load	F6		R2				No
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
3	FU				Integer					

Scoreboard Example Cycle 4

Instruction status

Instructio	j	k	Issue	Read operai	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3				
MUL F0	F2	F4				
SUB F8	F6	F2				
DIVI F10	F0	F6				
ADD F6	F8	F2				

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	Yes	Load	F6		R2				No
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
4	FU				Integer					

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 5

Instruction status

Instruction		j	k	Issue	Read operand	Executi complet	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5			
MULT	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Issue LD #2 since integer unit is now free.

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j Qj	FU for k Qk	Fj? Rj	Fk? Rk
	Integer	Yes	Load	F2		R3				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
5	FU		Integer							

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 6

Issue MULT.

Instruction status

Instruction	j	k	Issue	Read operands	Execution complete	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6		
MUL F0	F2	F4	6			
SUB F8	F6	F2				
DIVD F10	F0	F6				
ADD F6	F8	F2				

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	Yes	Load	F2		R3				No
	Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	No								
	Divide	No								

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
6	FU	Mult1	Integer							

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 7

Instruction status

Instruction	j	k	Issue	Read operands	Execution complete	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	
MUL F0	F2	F4	6			
SUB F8	F6	F2	7			
DIVD F10	F0	F6				
ADD F6	F8	F2				

MULT can't read its operands (F2) because LD #2 hasn't finished.

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	Yes	Load	F2		R3				No
	Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	Yes	Sub	F8	F6	F2		Integer	Yes	No
	Divide	No								

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
7	FU	Mult1	Integer			Add				

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 8a

DIVD issues.
MULT and SUBD both waiting for F2.

Instruction status

Instruction	j	k	Issue	Read operands	Execution complete	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	
MUL F0	F2	F4	6			
SUB F8	F6	F2	7			
DIVD F10	F0	F6	8			
ADD F6	F8	F2				

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	Yes	Load	F2		R3				No
	Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	Yes	Sub	F8	F6	F2		Integer	Yes	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
8	FU	Mult1	Integer			Add	Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 8b

LD #2 writes F2.

Instruction status

Instruction		j	k	Issue	Read operand	Executio complet	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	F0	F2	F4	6			
SUBD	F8	F6	F2	7			
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j Qj	FU for k Qk	Fj? Rj	Fk? Rk
	Integer	No								
	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	Sub	F8	F6	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
8	*FU*	Mult1				Add	Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 9

Instruction status

Instructio	j	k	Issue	Read operai	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9		
SUB F8	F6	F2	7	9		
DIVI F10	F0	F6	8			
ADD F6	F8	F2				

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
10	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
2	Add	Yes	Sub	F8	F6	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
9	FU	Mult1				Add	Divide			

Now MULT and SUBD can both read F2. ADDD can't start because add unit is busy.

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 11

ADD unit takes 2 cycle.

Instruction status

Instructio	j	k	Issue	Read operar	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9		
SUB F8	F6	F2	7	9	11	
DIVI F10	F0	F6	8			
ADD F6	F8	F2				

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
8	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
0	Add	Yes	Sub	F8	F6	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
11	*FU*	Mult1				Add	Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 12

Instruction status

Instructio	j	k	Issue	Read operan	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9		
SUB F8	F6	F2	7	9	11	12
DIVD F10	F0	F6	8			
ADD F6	F8	F2				

SUBD finishes.
DIVD waiting for F0.

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
7	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	No								
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
12	FU	Mult1					Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 13

ADDD issues.

Instruction status

Instructio	j	k	Issue	Read operai	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9		
SUB F8	F6	F2	7	9	11	12
DIVI F10	F0	F6	8			
ADD F6	F8	F2	13			

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
6	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
13	FU	Mult1			Add		Divide			

Scoreboard Example Cycle 14

Instruction status

Instructio	j	k	Issue	Read operai	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9		
SUB F8	F6	F2	7	9	11	12
DIVI F10	F0	F6	8			
ADD F6	F8	F2	13	14		

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
5	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
2	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
14	FU	Mult1			Add		Divide			

Scoreboard Example Cycle 15

Instruction status

Instructio	j	k	Issue	Read operar	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9		
SUB F8	F6	F2	7	9	11	12
DIVI F10	F0	F6	8			
ADD F6	F8	F2	13	14		

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
4	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
1	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
15	FU	Mult1			Add		Divide			

Scoreboard Example Cycle 16

Instruction status

Instructio	j	k	Issue	Read operar	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9		
SUB F8	F6	F2	7	9	11	12
DIVI F10	F0	F6	8			
ADD F6	F8	F2	13	14	16	

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
3	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
0	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
16	FU	Mult1			Add		Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 17

Instruction status

Instruction	j	k	Issue	Read operands	Execution complete	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9		
SUB F8	F6	F2	7	9	11	12
DIVD F10	F0	F6	8			
ADD F6	F8	F2	13	14	16	

ADDD can't write because of DIVD. RAW!

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
2	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
17	FU	Mult1			Add		Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 18

Nothing Happens!!

Instruction status

Instruction	j	k	Issue	Read operand	Execution compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9		
SUB F8	F6	F2	7	9	11	12
DIVD F10	F0	F6	8			
ADD F6	F8	F2	13	14	16	

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
1	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
18	FU	Mult1			Add		Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 19

Instruction status

Instructio	j	k	Issue	Read operai	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9	19	
SUB F8	F6	F2	7	9	11	12
DIVI F10	F0	F6	8			
ADD F6	F8	F2	13	14	16	

MULT completes execution.

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
0	Mult1	Yes	Mult	F0	F2	F4			No	No
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
19	FU	Mult1			Add		Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 20

MULT writes.

Instruction status

Instruction	j	k	Issue	Read operands	Execution complete	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9	19	20
SUB F8	F6	F2	7	9	11	12
DIVD F10	F0	F6	8			
ADD F6	F8	F2	13	14	16	

Functional unit status

Time Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for Qj	FU for Qk	Fj? Rj	Fk? Rk
Integer	No								
Mult1	No								
Mult2	No								
Add	Yes	Add	F6	F8	F2			No	No
Divide	Yes	Div	F10	F0	F6			Yes	Yes

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
20	FU				Add		Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 21

DIVD loads operands

Instruction status

Instruction	j	k	Issue	Read operands	Execution complete	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9	19	20
SUB F8	F6	F2	7	9	11	12
DIVD F10	F0	F6	8	21		
ADD F6	F8	F2	13	14	16	

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6			No	No

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
21	FU				Add		Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 22

Instruction status

Instructio	j	k	Issue	Read operan	Execu compl	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9	19	20
SUB F8	F6	F2	7	9	11	12
DIVI F10	F0	F6	8	21		
ADD F6	F8	F2	13	14	16	22

Now ADDD can write since WAR removed.

Functional unit status

Tim	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU foı Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
40	Divide	Yes	Div	F10	F0	F6			No	No

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
22	FU						Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 61

DIVD completes execution

Instruction status

Instruction	j	k	Issue	Read operands	Execution complete	Write Result
LD F6	34+	R2	1	2	3	4
LD F2	45+	R3	5	6	7	8
MUL F0	F2	F4	6	9	19	20
SUB F8	F6	F2	7	9	11	12
DIVD F10	F0	F6	8	21	61	
ADD F6	F8	F2	13	14	16	22

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for Qj	FU for Qk	Fj? Rj	Fk? Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
0	Divide	Yes	Div	F10	F0	F6			No	No

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
61	FU						Divide			

Dynamic Scheduling

Using A Scoreboard

Scoreboard Example Cycle 62

DONE ! !

Instruction status

Instruction		j	k	Issue	Read operand	Executi complet	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	F0	F2	F4	6	9	19	20
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8	21	61	62
ADDD	F6	F8	F2	13	14	16	22

Functional unit status

Time	Name	Busy	Op	dest Fi	S1 Fj	S2 Fk	FU for j Qj	FU for k Qk	Fj? Rj	Fk? Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
0	Divide	No								

Register result status

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

Review: Scoreboard

- Limitations of 6600 scoreboard
  - No forwarding
  - Limited to instructions in basic block (small *window*)
  - Large number of functional units (structural hazards)
  - Stall on WAR hazards
  - Stall on WAW hazards

```
DIV.D   F0, F2, F4
ADD.D   F6, F0, F8
S.D     F6, 0(R1)
SUB.D   F8, F10, F14
MUL.D   F6, F10, F8
```

WAR

WAW

Output dependence

Antidependence

Name dependence

Another Dynamic Algorithm: Tomasulo Algorithm

- For IBM 360/91 about 3 years after CDC 6600
- Goal: High Performance without special compilers
- RAW hazards are avoided by executing an instruction only when its operands are available.
- Out of order write
- Uses register renaming to minimize WAW and WAR hazards.
- Uses temporary registers to remove name dependency.
- Register renaming is provided by the reservation station.

Another Dynamic Algorithm: Tomasulo Algorithm

WAW Hazard

WAR Hazard

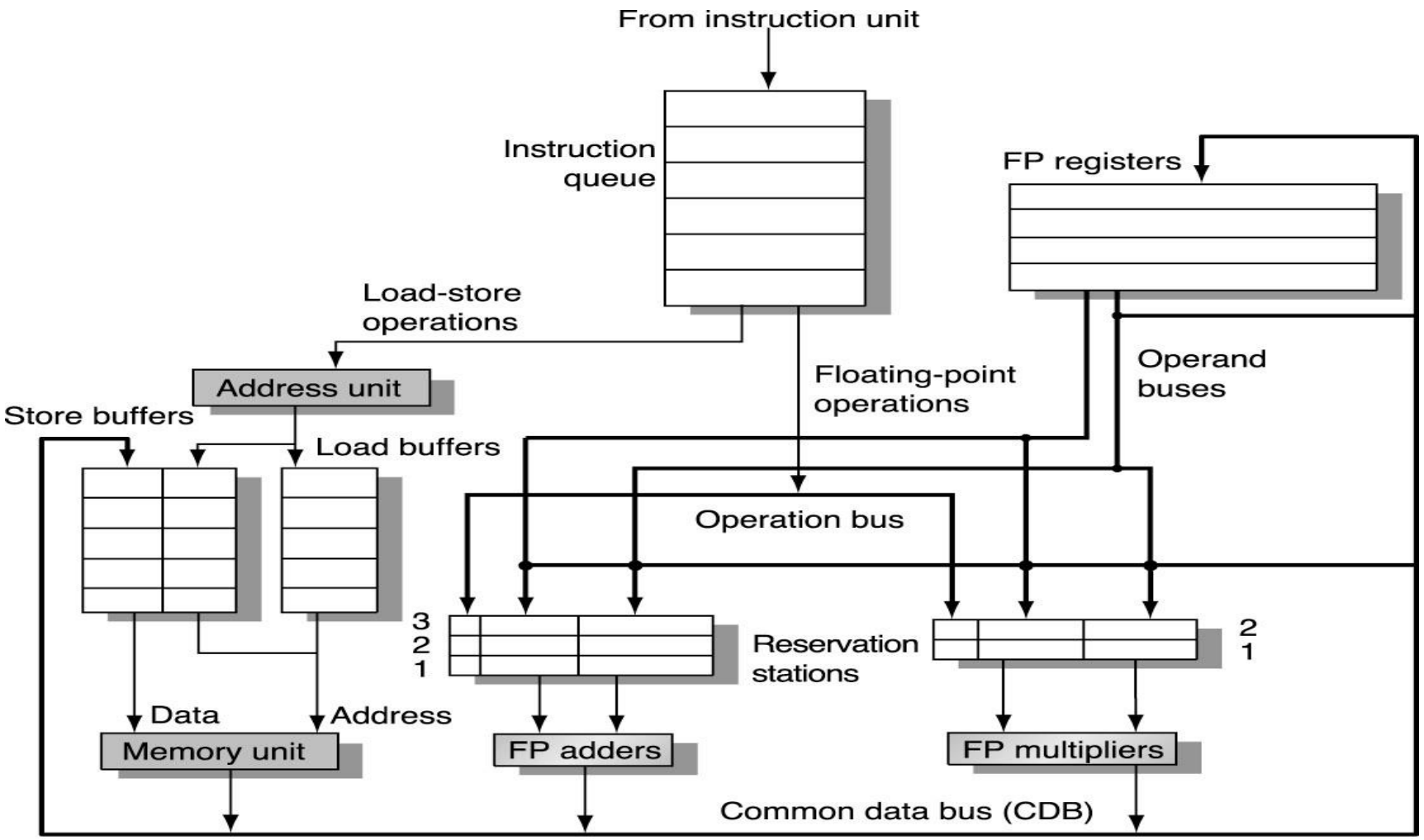
```
DIV.D   F0, F2, F4
ADD.D   F6, F0, F8
S.D     F6, 0(R1)
SUB.D   F8, F10, F14
MUL.D   F6, F10, F8
```

Register renaming (Remove WAW and WAR hazard)

```
DIV.D   F0, F2, F4
ADD.D   S, F0, F8
S.D     S, 0(R1)
SUB.D   T, F10, F14
MUL.D   F6, F10, T
```

(Subsequent use of F8 as T)

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



Major components of Tomasualo Structure

1. **Instruction Queue:** Fetch unit keeps the instructions in the instruction queue where they are issued in FIFO(maintaining in order).
2. **Reservation station:** Buffers the instruction and operands.

   Operands → value(already computed/ available)

   available, instruction is dispatched to functional unit for execution

   →pending (Name of RST/ load & store buffer).

   Not available, RST tracks CDB. When available, RST buffers, instruction is dispatched to functional unit for execution

Continued..

3. **Common Data Bus(CDB)**: Result has passed to various components like

   Other RST(Waiting the result)

   Load & Store buffer(waiting)

   Registers

4. **FP Registers :** keeping the operands
5. **Load & Store Buffers**
6. **Multiple FP Functional Units**
7. **Address Unit :** E.A. Calculation for Load & Store instruction.

Reservation Station

- Each Functional Unit (FU) has one or more reservation station.
- Instructions are issued if there is an empty reservation station.
- Scoreboard -> issued an instruction only when the FU is free.
- Operands are read from the register file if they are available.
- The reservation station holds
  - Instruction that have been issued and are awaiting execution at a functional unit.
  - The operands for that instruction (if they have already been computed or source of the operands otherwise)
  - The information needed to control the instruction once it has begun execution
- Renaming to larger set of register + buffering source operands
  - Prevents registers as bottleneck
  - WAR hazards are avoided because an operand is already stored in reservation station even when a write to the same register is performed out of order.
  - WAW hazards are avoided because of the user of pointers to reservation stations instead of the register pointers as tags on the CDB.

Three Stages of Tomasulo Algorithm

1. Issue—get instruction from instruction Queue
   - Stall if structural hazard, i.e. no space in the reservation station (RS).
   - If RS is free, the issue logic issues instruction to *RS* & read operands into *rs if ready*
   - *(Register renaming => Solves WAR, WAW).*
2. Execution—operate on operands (EX)

   When both operands are ready then execute; if not ready, watch CDB for result – Solves RAW
3. Write result—finish execution (WB)
   - Write on Common Data Bus to all awaiting units;
   - mark reservation station available.
   - Write result into destination register if its status is *rs*. => Solves WAW.

Reservation Station Components

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

Vj, Vk— Value of the source operand.

Qj, Qk— Name of the RS that would provide the source operands.

A- used to hold information for the memory address calculation for the load and store.

Busy—Indicates reservation station or FU is busy

Register File Status—Indicates which functional unit will write each register, if one exists. Blank when no pending instructions that will write that register meaning that the value is already available.

Tomasulo Example Cycle 0

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2					Load1	No			
LD	F2	45+	R3					Load2	No			
MULTD	F0	F2	F4					Load3	No			
SUBD	F8	F6	F2									
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk	A			
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	0	Mult1	No									
	0	Mult2	No									
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
0			FU									

Tomasulo Example Cycle 1

Instruction status					Execution	Write						
Instruction		*j*	*k*	*Issue*	*complete*	*Result*			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									
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*	*A*			
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	0	Mult1	No									
	0	Mult2	No									
Register result status												
Clock				*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	*...*	*F30*
1			*FU*				Load1					

Tomasulo Example Cycle 2

Instruction status					Execution	Write						
Instruction		*j*	*k*	*Issue*	*complete*	*Result*			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			Assume Load takes 2 cycles						
DIVD	F10	F0	F6									
ADDD	F6	F8	F2									
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	0	Mult1	No									
	0	Mult2	No									
Register result status												
Clock				*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	...	*F30*
2			*FU*		Load2		Load1					

Tomasulo Example Cycle 3

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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									
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	0	Add1	No									
	0	Add2	No		read value							
		Add3	No									
	0	Mult1	Yes	Mult		R(F4)	Load2					
	0	Mult2	No									
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
3			FU	Mult1	Load2		Load1					

Tomasulo Example Cycle 4

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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									
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	0	Add1	Yes	Sub	M(34+R2)			Load2				
	0	Add2	No									
		Add3	No									
	0	Mult1	Yes	Mult		R(F4)	Load2					
	0	Mult2	No									
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
4			FU	Mult1	Load2		M(34+R2	Add1				

Tomasulo Example Cycle 5

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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									
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	2	Add1	Yes	Sub	M(34+R2)	M(45+R3)						
	0	Add2	No									
		Add3	No									
	10	Mult1	Yes	Mult	M(45+R3)	R(F4)						
	0	Mult2	Yes	Div		M(34+R2)	Mult1					
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
5			FU	Mult1	M(45+R3)		M(34+R2	Add1	Mult2			

Tomasulo Example Cycle 6

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2	1	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								
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	1	Add1	Yes	Sub	M(34+R2)	M(45+R3)						
	0	Add2	Yes	Add		M(45+R3)	Add1					
		Add3	No									
	9	Mult1	Yes	Mult	M(45+R3)	R(F4)						
	0	Mult2	Yes	Div		M(34+R2)	Mult1					
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
6			FU	Mult1	M(45+R3)		Add2	Add1	Mult2			

Tomasulo Example Cycle 7

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2	1	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								
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	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					
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
7			FU	Mult1	M(A2)		Add2	Add1	Mult2			

Tomasulo Example Cycle 8

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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								
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	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					
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
8			FU	Mult1	M(A2)		Add2	M1-M2	Mult2			

Tomasulo Example Cycle 9

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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 --							
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	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					
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
9			FU	Mult1	M(A2)		Add2	M1-M2	Mult2			

Tomasulo Example Cycle 10

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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							
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	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					
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
10			FU	Mult1	M(A2)		Add2	M1-M2	Mult2			

Tomasulo Example Cycle 11

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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						
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	0	Add1	No									
		Add2	No									
		Add3	No									
	4	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
11			FU	Mult1	M(A2)	M1-M2+M(		M1-M2	Mult2			

Tomasulo Example Cycle 12

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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						
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	0	Add1	No									
		Add2	No									
		Add3	No									
	4	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
12			FU	Mult1	M(A2)	M1-M2+M(		M1-M2	Mult2			

Tomasulo Example Cycle 15

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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						
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	0	Add1	No									
		Add2	No									
		Add3	No									
	0	Mult1	Yes	Mult	M(A2)	R(F4)						
	0	Mult2	Yes	Div		M(A1)	Mult1					
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
15			FU	Mult1	M(A2)	M1-M2+M(		M1-M2	Mult2			

Tomasulo Example Cycle 16

Instruction status					Execution	Write						
Instruction		*j*	*k*	*Issue*	*complete*	*Result*			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						
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	0	Add1	No									
		Add2	No									
		Add3	No									
		Mult1	No									
	40	Mult2	Yes	Div	M*F4	M(A1)						
Register result status												
Clock				*F0*	*F2*	*F4*	*F6*	*F8*	*F10*	*F12*	*...*	*F30*
16			*FU*	M*F4	M(A2)	M1-M2+M(		M1-M2	Mult2			

Tomasulo Example Cycle 56

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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						
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	0	Add1	No									
		Add2	No									
		Add3	No									
		Mult1	No									
	0	Mult2	Yes	Div	M*F4	M(A1)						
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
56			FU	M*F4	M(A2)	M1-M2+M(		M1-M2	Mult2			

Tomasulo Example Cycle 57

Instruction status					Execution	Write						
Instruction		j	k	Issue	complete	Result			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						
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	Qk				
	0	Add1	No									
		Add2	No									
		Add3	No									
		Mult1	No									
	0	Mult2	No									
Register result status												
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30
57			FU	M*F4	M(A2)	M1-M2+M(		M1-M2	result			

Tomasulo Algorithm Vs Scoreboard

Tomasulo Algorithm	Scoreboard
Control and buffers distributed with FU.	Centralized in scoreboard
FU buffers called reservation station (RS) have rending operands	Operands in register
Registers in instruction replaced by values or pointers RS (Register renaming) Avoids WAW hazard…… Avoids WAR hazard……	No register renaming Stall issue Stall completion
No issue on structural hazards	No issue on structural hazard
Results to FU from RS not through registers over CDB that broadcasts results to all FUs	Write/read register

Tomasulo Drawback

- Many associative stores (CDB) at high speed.
- Performance limited by common data bus
- Each CDB must go to multiple FU
- Number of FU that can complete per cycle limited to one.

Example:Scoreboard tables before MUL.D writes results

			Instruction Status						
Instruction		Issue	Read Operands	Execution Complete			Write Result		
L.D	**F6,34(R2)**	X	X		X			X	
L.D	**F2,45(R3)**	X	X		X			X	
MUL.D	**F0,F2,F4**	X	X		X				
SUB.D	**F8,F6,F2**	X	X		X			X	
DIV.D	**F10,F0,F6**	X							
ADD.D	**F6,F8,F2**	X	X		X				
		Functional unit status							
Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			No	No
Mult2	No								
Add	Yes	Add	F6	F8	F2			No	No
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes
		Register result status							
	F0	F2	F4	F6	F8	F10	F12	……	
unit	Mult1	Integer			Add	Divide			

Tomasulo Exercise (Scenario is given)

Consider a set of instructions

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

And there are 2 load/store, 3 Add and 2 Mult.

and the scenario is

(i) 1st load has finished it's write result stage.

(ii) 2nd load has completed it's execution stage.

(iii) Remaining four instructions have finished their issue stage.

Show the status of each table.

Step-1 (Instruction status)

Draw the table and put a tick(√) mark.

Instruction	issue	Execution Complete	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	√		

Step-2(Reservation Station Status)

Name of RST	Busy	OP	Vj	Vk	Qj	Qk	A
Load1	No						
Load2	Yes	Load					44+Reg[R3]
Add1	Yes	Sub		Mem[32+Reg[R2]]	Load2		
Add2	Yes	Add			Add1	Load2	
Add3	No						
Mult1	Yes	Mult		Reg[F4]	Load2		
Mult2	Yes	Div		Mem[32+Reg[R2]]	Mult1		

Step-3(Register Result Status)

F0	F2	F4	F6	F8	F10	F12	…	F30	F31
Mult1	Load2		Add2	Add1	Mult2				