

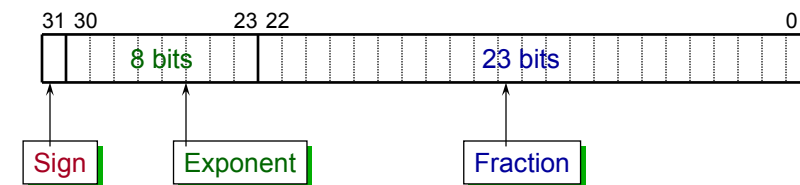
Instruction Level Parallelism (ILP)

- A necessary aside: floating point
- Incorporating floating point in the pipeline
- Scoreboarding
 - Multiple functional units (superscalar)
 - Multi-issue
 - Out of order execution / completion
 - In-order issue

1

(IEEE 754) Floating Point Representation

$$(-1)^S \times F \times 2^E$$



0: Positive
1: Negative

127 bias

Leading '1' is implied,
but not represented

Key Points:

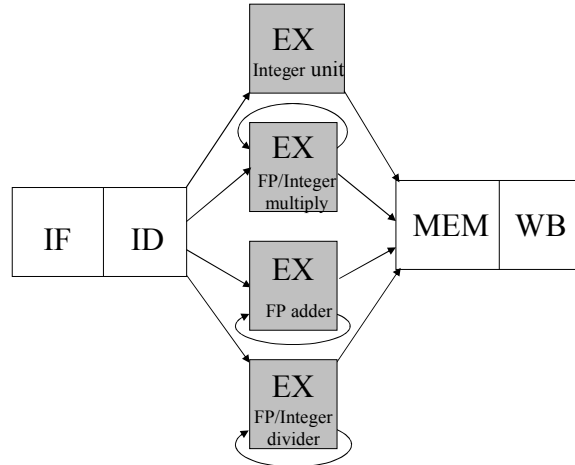
- Limited precision
- Limited range
- Distinct set of instructions
- Distinct set of registers
- Slow to operate on (relative to integers)

What does 0x00000000 represent?

How is 3.25 represented?

2

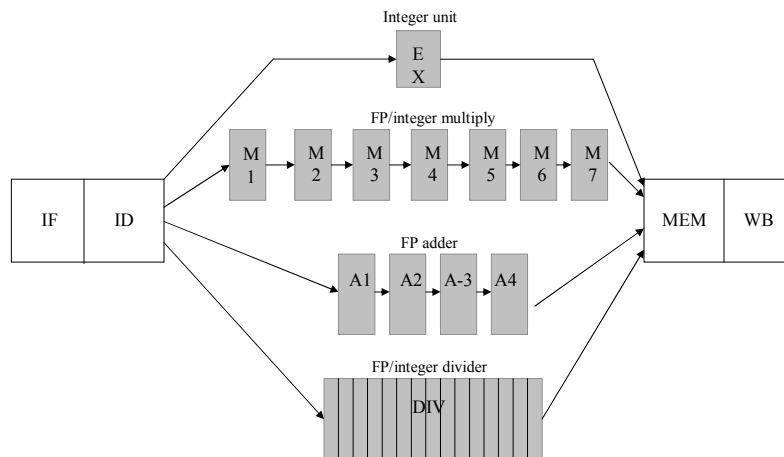
Floating Point and the Pipeline



The MIPS pipeline with floating-point functional units.

3

Pipelining the FP Units



4

New Problems

- **Structural Hazards**
 - FP divide takes many cycles, and is not pipelined
 - May need to write more than one register in a cycle
- **Instructions may complete out of order**
 - WAW (write-after-write) hazards are possible
 - “Precise exceptions” are more difficult to implement
- **Longer pipelines may result in more difficulty dealing with RAW (read-after-write) dependences**

Once we deal with those problems, a new problem is revealed:
can't keep all the functional units busy

5

Addressing the Problem

- Consider the following code fragment:
 DIVD F0, F2, F4
 ADDD F10, F0, F8
 MULTD F12, F8, F14
- What is the problem?
 - ADDD stalls due to RAW hazard
 - MULTD has no dependences, but is marooned behind ADDD
- In-order execution limits performance

6

Out of Order Execution

- Key idea: Allow **out-of-order execution**
DIVD F0, F2, F4
ADDD F10, F0, F8
SUBD F12, F8, F14
- Leads to **out-of-order completion**
- The technique we'll look at: scoreboarding
- (Dates to the CDC 6600 (1964))
- All instructions pass through a single issue stage for scheduling (**in-order issue**)

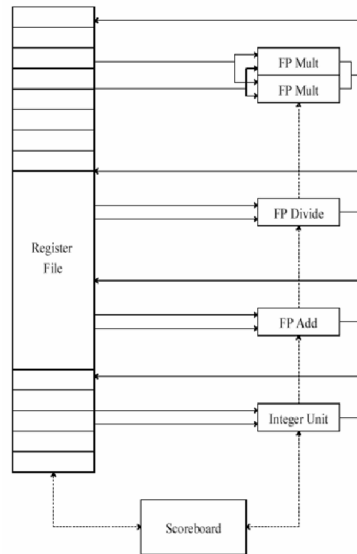
7

The MIPS Pipeline and Scoreboarding

- Ignore memory interface components for simplicity
- IF fetches an instruction each cycle
 - There is a small window (buffer) of already fetched instructions
 - If issue stalls, the buffer fills
 - When instructions complete, they leave the buffer
- Scoreboarding: 4-stage execution
 - Issue – check structural WAW hazards, stall until clear
 - Read ops – check RAW. Wait till operands ready, read regs
 - Execute – execute operation. Notify scoreboard when done
 - Write – check for WAR. Stall write until clear

8

Dynamic MIPS Datapath



9

Scoreboard Implications

- **Out-of-order completion => WAR, WAW hazards?**
- **Solutions for WAR**
 - Queue both the operation and copies of its operands
 - Read registers only during Read Operands stage
- **For WAW, must detect hazard: stall until other completes**
- **Need to have multiple instructions in execution phase => multiple execution units or pipelined execution units**
- **Scoreboard keeps track of dependencies, state or operations**
- **Scoreboard replaces ID, EX, WB with 4 stages**

10

Dealing with Hazards

- **Structural hazards**
 - Obvious – desired EX unit must be free
- **Data Hazards**
 - More complex due to parallel multi-cycle EX units
 - **RAW** – delay operand fetch until producing instruction finishes register write back
 - **WAW** – issue only one instruction at a time for each destination register
 - **WAR** – delay write back until reading inst has fetched the previous value

11

Four Stages of Scoreboard Control

- 1. Issue**—decode instructions & check for structural hazards (ID1)

If a functional unit for the instruction is free and no other active instruction has the same destination register (WAW), then issue the instruction. Otherwise, stall this and all following instructions.
- 2. Read operands**—wait until no flow dependences (RAW), then read operands from registers

No forwarding done. Why? Not as useful in this situation, as each functional unit writes a register as soon as the value is available.
- 3. Execution**—operate on operands (EX)
- 4. Write result**—finish execution (WB)

Must wait for earlier instructions to have read result register (WAR)

12

Scoreboard: 3 Tables + Rules

- 1. Instruction status**—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 if Fj, Fk are ready
- 3. Register result status**—Indicates which functional unit will write each register.

13

Detailed Scoreboard Pipeline Control

Instruction status	Wait until	Bookkeeping
Issue	Not busy (FU) and not result(D)	$Busy(FU) \leftarrow \text{yes}; Op(FU) \leftarrow op;$ $Fi(FU) \leftarrow 'D'; Fj(FU) \leftarrow 'S1';$ $Fk(FU) \leftarrow 'S2'; Qj \leftarrow Result('S1');$ $Qk \leftarrow Result('S2'); Rj \leftarrow \text{not } Qj;$ $Rk \leftarrow \text{not } Qk; Result('D') \leftarrow FU;$
Read operands	Rj and Rk	$Rj \leftarrow \text{No}; Rk \leftarrow \text{No}$
Execution complete	Functional unit done	
Write result	$\forall f((Fj(f) \neq Fi(FU) \text{ or } Rj(f) = \text{No}) \&$ $(Fk(f) \neq Fi(FU) \text{ or } Rk(f) = \text{No}))$	$\forall f(\text{if } Qj(f) = FU \text{ then } Rj(f) \leftarrow \text{Yes});$ $\forall f(\text{if } Qk(f) = FU \text{ then } Rj(f) \leftarrow \text{Yes});$ $Result(Fi(FU)) \leftarrow 0; Busy(FU) \leftarrow \text{No}$

14

Scoreboard Example

- Assume 2 MULT, 1 DIV, 1 ADD/SUB FP units, plus 1 integer ALU
- EX stage timings:
 - Int ALU takes 1 cycle
 - FP adder takes 2 cycles
 - FP mult takes 10 cycles
 - FP div takes 40 cycles

15

Scoreboard Example

<u>Instruction status</u>		Read	Execu	Write
Instruction	<i>k</i>	Issue	oper	compl
				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	

<u>Functional unit status</u>		dest	S1	S2	FU for	FU for	Fj?	Fk?
Time	Name	Busy	Op	Fi	Fj	Fk	Qj	Qk
							Rj	Rk
	Integer	No						
	Mult1	No						
	Mult2	No						
	Add	No						
	Divide	No						

<u>Register result status</u>		F0	F2	F4	F6	F8	F10	F12	...	F30
Clock	FU									

16

Scoreboard Example Cycle 1

<u>Instruction status</u>			<i>Read</i>		<i>Execu</i>		<i>Write</i>					
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operan</i>	<i>compl</i>	<i>Result</i>						
LD	F6	34+	R2	1								
LD	F2	45+	R3									
MULT	F0	F2	F4									
SUBD	F8	F6	F2									
DIVD	F10	F0	F6									
ADD	F6	F8	F2									

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for</i>	<i>FU for</i>	<i>Fj?</i>	<i>Fk?</i>
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
	Integer	Yes	Load	F6		R2				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	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>	Intege									

17

Scoreboard Example Cycle 2

<u>Instruction status</u>			<i>Read</i>		<i>Execu</i>		<i>Write</i>					
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operan</i>	<i>compl</i>	<i>Result</i>						
LD	F6	34+	R2	1 2								
LD	F2	45+	R3									
MUL	F0	F2	F4									
SUB	F8	F6	F2									
DIVC	F10	F0	F6									
ADD	F6	F8	F2									

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for</i>	<i>FU for</i>	<i>Fj?</i>	<i>Fk?</i>
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
	Integer	Yes	Load	F6		R2				No
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	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>	Integer									

- Issue 2nd LD?

18

Scoreboard Example Cycle 3

Instruction status				Read	Execu	Write
Instruction	<i>j</i>	<i>k</i>		Issue	operat	compl Result
LD	F6	34+	R2	1	2	3
LD	F2	45+	R3			
MUL	F0	F2	F4			
SUB	F8	F6	F2			
DIV	F10	F0	F6			
ADD	F6	F8	F2			

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

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

- Issue MULT?

19

Scoreboard Example Cycle 4

Instruction status				Read	Execu	Write	
Instruction	<i>j</i>	<i>k</i>		Issue	operat	compl Result	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3				
MUL	F0	F2	F4				
SUB	F8	F6	F2				
DIV	F10	F0	F6				
ADD	F6	F8	F2				

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

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

20

Scoreboard Example Cycle 5

<u>Instruction status</u>			<i>Read ExecuWrite</i>			
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operat</i>	<i>compl</i>	<i>Result</i>
LD	F6	34+ R2	1	2	3	4
LD	F2	45+ R3	5			
MUL	F0	F2 F4				
SUB	F8	F6 F2				
DIV	F10	F0 F6				
ADD	F6	F8 F2				

<u>Functional unit status</u>		<i>dest S1 S2 FU for FU for Fj? Fk?</i>								
<i>Tim</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
	Integer	Yes	Load	F2		R3				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

<u>Register result status</u>		<i>F0 F2 F4 F6 F8 F10 F12 ... F30</i>										
<i>Clock</i>	<i>FU</i>											
5		Integer										

21

Scoreboard Example Cycle 6

<u>Instruction status</u>			<i>Read ExecuWrite</i>			
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operat</i>	<i>compl</i>	<i>Result</i>
LD	F6	34+ R2	1	2	3	4
LD	F2	45+ R3	5	6		
MUL	F0	F2 F4	6			
SUB	F8	F6 F2				
DIV	F10	F0 F6				
ADD	F6	F8 F2				

<u>Functional unit status</u>		<i>dest S1 S2 FU for FU for Fj? Fk?</i>								
<i>Tim</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
	Integer	Yes	Load	F2		R3				No
	Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	No								
	Divide	No								

<u>Register result status</u>		<i>F0 F2 F4 F6 F8 F10 F12 ... F30</i>										
<i>Clock</i>	<i>FU</i>											
6		Mult Integer										

22

Scoreboard Example Cycle 7

Instruction status			Read	Execu	Write
Instruction	j	k	Issue	operai	compl 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		
DIV	F10	F0 F6			
ADD	F6	F8 F2			

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

Register result status		F0	F2	F4	F6	F8	F10	F12	...	F30	
Clock	7	FU									
		Mult	Integer		Add						

- Read multiply operands?

23

Scoreboard Example Cycle 8a

Instruction status			Read	Execu	Write
Instruction	j	k	Issue	operai	compl 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		
DIV	F10	F0 F6	8		
ADD	F6	F8 F2			

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

Register result status		F0	F2	F4	F6	F8	F10	F12	...	F30	
Clock	8	FU									
		Mult	Integer		Add	Divide					

24

Scoreboard Example Cycle 8b

<u>Instruction status</u>				Read		Execu		Write	
Instructio	j	k	Issue	operan	compl	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					
ADD	F6	F8	F2						

<u>Functional unit status</u>				dest	S1	S2	FU for	FU for	Fj?	Fk?
Time	Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	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

<u>Register result status</u>											
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30	
8	FU	Mult1				Add	Divide				

25

Scoreboard Example Cycle 9

<u>Instruction status</u>				Read		Execu		Write	
Instructio	j	k	Issue	operan	compl	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				
DIV	F10	F0	F6	8					
ADD	F6	F8	F2						

<u>Functional unit status</u>				dest	S1	S2	FU for	FU for	Fj?	Fk?
Time	Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	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

<u>Register result status</u>											
Clock		F0	F2	F4	F6	F8	F10	F12	...	F30	
9	FU	Mult1				Add	Divide				

- Read operands for MULT & SUBD? Issue ADDD?

26

Scoreboard Example Cycle 11

Instruction status			Read	Execu	Write					
Instruction	j	k	Issue	operat	compl	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					
DIV	F10	F0 F6	8							
ADD	F6	F8 F2								

Functional unit status		dest	S1	S2	FU for	FU for Fj?	Fk?		
Tim	Name	Busy Op	Fi	Fj	Fk	Qj	Qk	Rj	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		F0	F2	F4	F6	F8	F10	F12	...	F30
Clock										
11	FU	Mult1			Add	Divide				

27

Scoreboard Example Cycle 12

Instruction status			Read	Execu	Write					
Instruction	j	k	Issue	operat	compl	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				
DIV	F10	F0 F6	8							
ADD	F6	F8 F2								

Functional unit status		dest	S1	S2	FU for	FU for Fj?	Fk?		
Tim	Name	Busy Op	Fi	Fj	Fk	Qj	Qk	Rj	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		F0	F2	F4	F6	F8	F10	F12	...	F30
Clock										
12	FU	Mult1				Divide				

- Read operands for DIVD?

28

Scoreboard Example Cycle 13

Instruction status				Read	Execu	Write					
Instruction	j	k		Issue	operat	compl	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				
DIV	F10	F0	F6	8							
ADD	F6	F8	F2	13							

Functional unit status				dest	S1	S2	FU for	FU for Fj?	Fk?	
Tim	Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	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			

29

Scoreboard Example Cycle 14

Instruction status				Read	Execu	Write					
Instruction	j	k		Issue	operat	compl	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				
DIV	F10	F0	F6	8							
ADD	F6	F8	F2	13	14						

Functional unit status				dest	S1	S2	FU for	FU for Fj?	Fk?	
Tim	Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	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			

30

Scoreboard Example Cycle 15

<u>Instruction status</u>				<i>Read ExecuWrite</i>			
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operat</i>	<i>compl</i>	<i>Result</i>	
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
DIV	F10	F0	F6	8			
ADD	F6	F8	F2	13	14		

<u>Functional unit status</u>		<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for</i>	<i>FU for Fj?</i>	<i>Fk?</i>
<i>Tim</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>
	Integer	No					
4	Mult1	Yes	Mult	F0	F2	F4	No
	Mult2	No					No
1	Add	Yes	Add	F6	F8	F2	No
	Divide	Yes	Div	F10	F0	F6	Mult1

<u>Register result status</u>		<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>
Clock										
15	<i>FU</i>	Mult1			Add		Divide			

31

Scoreboard Example Cycle 16

<u>Instruction status</u>				<i>Read ExecuWrite</i>			
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operat</i>	<i>compl</i>	<i>Result</i>	
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
DIV	F10	F0	F6	8			
ADD	F6	F8	F2	13	14	16	

<u>Functional unit status</u>		<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for</i>	<i>FU for Fj?</i>	<i>Fk?</i>
<i>Tim</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>
	Integer	No					
3	Mult1	Yes	Mult	F0	F2	F4	No
	Mult2	No					No
0	Add	Yes	Add	F6	F8	F2	No
	Divide	Yes	Div	F10	F0	F6	Mult1

<u>Register result status</u>		<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>
Clock										
16	<i>FU</i>	Mult1			Add		Divide			

32

Scoreboard Example Cycle 17

Instruction status			Read Execu Write			
Instruction	j	k	Issue	operat	compl	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
DIV	F10	F0 F6	8			
ADD	F6	F8 F2	13	14	16	

Functional unit status		dest	S1	S2	FU for	FU for Fj?	Fk?
Tim. Name	Busy Op	Fi	Fj	Fk	Qj	Qk	Rj 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		F0	F2	F4	F6	F8	F10	F12	...	F30
Clock										
17	FU	Mult1			Add		Divide			

- Write result of ADDD?

33

Scoreboard Example Cycle 18

Instruction status			Read Execu Write			
Instruction	j	k	Issue	operat	compl	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
DIV	F10	F0 F6	8			
ADD	F6	F8 F2	13	14	16	

Functional unit status		dest	S1	S2	FU for	FU for Fj?	Fk?
Tim. Name	Busy Op	Fi	Fj	Fk	Qj	Qk	Rj 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		F0	F2	F4	F6	F8	F10	F12	...	F30
Clock										
18	FU	Mult1			Add		Divide			

34

Scoreboard Example Cycle 19

<u>Instruction status</u>				Read	Execu	Write					
Instruction	j	k	Issue	operai	compl	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				
DIVC	F10	F0	F6	8							
ADD	F6	F8	F2	13	14	16					

<u>Functional unit status</u>		dest	S1	S2	FU for	FU for Fj?	Fk?		
Tim	Name	Busy Op	Fi	Fj	Fk	Qj	Qk	Rj	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

<u>Register result status</u>		F0	F2	F4	F6	F8	F10	F12	...	F30
Clock										
19	FU	Mult1			Add		Divide			

35

Scoreboard Example Cycle 20

<u>Instruction status</u>				Read	Execu	Write					
Instruction	j	k	Issue	operai	compl	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				
DIVC	F10	F0	F6	8							
ADD	F6	F8	F2	13	14	16					

<u>Functional unit status</u>		dest	S1	S2	FU for	FU for Fj?	Fk?		
Tim	Name	Busy Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
	Integer	No							
	Mult1	No							
	Mult2	No							
	Add	Yes Add	F6	F8	F2			No	No
	Divide	Yes Div	F10	F0	F6			Yes	Yes

<u>Register result status</u>		F0	F2	F4	F6	F8	F10	F12	...	F30
Clock										
20	FU				Add		Divide			

36

Scoreboard Example Cycle 21

<u>Instruction status</u>			<i>Read Execu Write</i>			
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operat</i>	<i>compl</i>	<i>Result</i>
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
DIV	F10	F0 F6	8	21		
ADD	F6	F8 F2	13	14	16	

<u>Functional unit status</u>		<i>dest</i>		<i>S1</i>	<i>S2</i>	<i>FU for FU for Fj?</i>		<i>Fk?</i>		
<i>Tim</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			No	No
	Divide	Yes	Div	F10	F0	F6			No	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>	
21	FU				Add		Divide				37

Scoreboard Example Cycle 22

<u>Instruction status</u>			<i>Read Execu Write</i>			
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operat</i>	<i>compl</i>	<i>Result</i>
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
DIV	F10	F0 F6	8	21		
ADD	F6	F8 F2	13	14	16	22

<u>Functional unit status</u>		<i>dest</i>		<i>S1</i>	<i>S2</i>	<i>FU for FU for Fj?</i>		<i>Fk?</i>		
<i>Tim</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>Fi</i>	<i>Fj</i>	<i>Fk</i>	<i>Qj</i>	<i>Qk</i>	<i>Rj</i>	<i>Rk</i>
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
40	Divide	Yes	Div	F10	F0	F6			No	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>	
22	FU						Divide				

38

Scoreboard Example Cycle 61

Instruction status		Read Execut Write			
Instruction	j k	Issue	operat	compl	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
DIV	F10 F0 F6	8	21	61	
ADD	F6 F8 F2	13	14	16	22

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

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

39

Scoreboard Example Cycle 62

Instruction status		Read Execut Write			
Instruction	j k	Issue	operan	comple	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
ADD	F6 F8 F2	13	14	16	22

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

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

40