
COSC4201

Chapter 4 Scoreboard

Prof. Mokhtar Aboelaze
York University

Overcoming Data Hazards with Dynamic Scheduling

- In the pipeline, if there is data dependency between an instruction already in the pipe and a fetched instruction that can not be hidden by forwarding, the pipeline stalls.
- That is known as static scheduling.
- In *dynamic scheduling* the hardware rearranges the instructions to reduce stalls. It simplifies the compiler and deals with dependences that were not known during the compilation.
- In dynamic scheduling, processor can not remove true data dependence, it tries to avoid stalls.

Dynamic Scheduling

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

- ADDD depends on DIVD, can not proceed, SUBD will be stalled also although it doesn't depend on any other instruction.
- The reason is structural and data hazards are checked in the ID stage, once we detected that ADDD depends on DIVD, the pipeline is stalled and no instructions will be fetched
- To proceed with SUBD, we must separate the issue process into 2 parts, checking the structural hazard, and waiting for the absence of data hazards
- We will check for structural hazard when we issue; thus we still use *in-order issue*. However we want the instructions to begin execution as soon as their data operands are ready

Dynamic Scheduling

- That of course creates out-of-order completion, which creates problems with exception handling, for the time being, we assume imprecise exception.
- We split the ID stage into two parts
 - **Issue** Decode and check structural hazards
 - **Read operands** Wait until no data hazard and read operands
- An instruction fetch stage proceeds the ID stage and may fetch in a single-entry latch, or a queue
- After that the EX stage

Dynamic Scheduling with scoreboard

- All instructions pass through the *issue* stage in order.
- Scoreboard was first used in CDC 6600
- Have to check for WAW and RAW hazards
- Assume we have one integer unit, two multipliers, one adder, and one divide unit.
- Scoreboard keep information about every instruction from fetch to execute
- The scoreboard controls when an instruction can read operands, start execution and when it can write its result.

Dynamic Scheduling with scoreboard

- There is no forwarding.
- Notice that there is no specific stage for write back, which means the operands can be written back in the cycle after completion.

Dynamic Scheduling with scoreboard

- Ignoring memory, 4 stages (NO forwarding)
- **Issue** If a FU is free, and no other active instruction has the same destination register, the instruction is issued. Otherwise, the instruction issue stalls, and no other instructions can be fetched (replaces a portion of ID stage in MIPS).
- **Read Operands** the scoreboard monitors the availability of the source operands, if available (no active instruction will write to the source registers, the instruction can read operands and execute (probably out-of-order))

Dynamic Scheduling with scoreboard

- **Execution** FU starts execution, after completion it notifies the scoreboard
- **Write Result** Scoreboard checks for WAR hazards, and stalls if necessary

DIV	F0,F2,F4
ADD	F10,F0,F8
SUB	F8,F8,F14
- **Scoreboard stalls SUB until ADD reads its operand (note that because DIV will take a long time, ADD stalls).**

Scoreboard

- 1. Instruction status** Indicates which of the 4 steps the instruction is in
- 2. Functional Unit Status** Indicates the state of the FU, there are nine fields
 - Busy** busy or not
 - OP** operation to be performed in the unit
 - F_i** Destination register
 - F_j, F_k** source register number
 - Q_j, Q_k** FU producing source registers F_j and F_k
 - R_j, R_k** Flags to indicates if F_j and F_k are ready or not
- 3. Register Result Status** Indicates which FU will write to each register

		Instruction status			
Instruction		Issue	Read operands	Execution complete	Write result
LD	F6, 34 (R2)	√	√	√	√
LD	F2, 45 (R3)	√	√	√	
MULTD	F0, F2, F4	√			
SUBD	F8, F6, F2	√			
DIVD	F10, F0, F6	√			
ADD	F6, F8, F2				

Functional unit status									
Name	Busy	Op	F _i	F _j	F _k	Q _j	Q _k	R _j	R _k
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									
	F0	F2	F4	F6	F8	F10	F12	...	F30
FU	Mult1	Integer			Sub	Divide			

Dynamic Scheduling with Scoreboard

◦ Consider the following example

◦ LD F6,34(R2)

◦ LD F2,45(R3)

one integer unit, two multipliers, one adder, and one divide

◦ MULTD F0,F2,F4

◦ SUBD F8,F6,F2

◦ DIVD F10,F0,F6

◦ ADDD F6,F8,F2

- Add is 2 cycles, MULT is 10 and divide is 40

Scoreboard Example Cycle 1

<u>Instruction status</u>				<i>Read</i>	<i>Executic</i>	<i>Write</i>									
Instruction	<i>j</i>	<i>k</i>	<i>R</i>	<i>Issue</i>	<i>operand complet</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												
ADDD	F6	F8	F2												
<u>Functional unit status</u>				<i>Busy</i>	<i>Op</i>	<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</i>	<i>Fj?</i>	<i>Fk?</i>			
<i>Time</i>	<i>Name</i>														
	Integer			Yes	Load	F6		R2					Yes		
	Mult1			No											
	Mult2			No											
	Add			No											
	Divide			No											
<u>Register result status</u>															
Clock				F0	F2	F4	F6	F8	F10	F12	...	F30			
1	FU			Integer											

Scoreboard Example Cycle 2

<u>Instruction status</u>				<i>Read</i>		<i>Executic</i>		<i>Write</i>	
Instruction	<i>j</i>	<i>k</i>	<i>Rk</i>	<i>Issue</i>	<i>operand</i>	<i>complet</i>	<i>Result</i>		
LD	F6	34+	R2	1	2				
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>				<i>dest</i>		<i>S1</i>		<i>S2</i>		<i>FU for j</i>		<i>FU for k</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>																
<i>Clock</i>	<i>FU</i>	<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		Integer														

• Issue 2nd LD?

13

Fall 2008

Scoreboard Example Cycle 3

<u>Instruction status</u>				<i>Read</i>		<i>Executic</i>		<i>Write</i>	
Instruction	<i>j</i>	<i>k</i>	<i>Rk</i>	<i>Issue</i>	<i>operand</i>	<i>complet</i>	<i>Result</i>		
LD	F6	34+	R2	1	2	3			
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>				<i>dest</i>		<i>S1</i>		<i>S2</i>		<i>FU for j</i>		<i>FU for k</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>																
<i>Clock</i>	<i>FU</i>	<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>						
3		Integer														

• Issue MULT?

14

Fall 2008

Scoreboard Example Cycle 4

<u>Instruction status</u>				<i>Read Executic Write</i>			
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>operand complet</i>	<i>Result</i>	
LD	F6	34+	R2	1	2	3	4
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>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</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>F30</i>		
4	<i>FU</i>	Integer										

Scoreboard Example Cycle 5

<u>Instruction status</u>				<i>Read Executic Write</i>			
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>operand complet</i>	<i>Result</i>	
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				

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</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	F2		R3				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>F30</i>		
5	<i>FU</i>	Integer										

Scoreboard Example Cycle 6

<u>Instruction status</u>				<i>Read Executic Write</i>			
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operand complet</i>	<i>Result</i>		
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6		
MULT	F0	F2	F4	6			
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

<u>Functional unit status</u>										
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</i>	<i>Fj?</i>	<i>Fk?</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	Yes	Mult	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	No								
	Divide	No								

<u>Register result status</u>												
<i>Clock</i>		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	<i>...</i>	<i>F30</i>		
6	<i>FU</i>	Mult1	Integer									

Scoreboard Example Cycle 7

<u>Instruction status</u>				<i>Read Executic Write</i>			
Instruction	<i>j</i>	<i>k</i>	<i>Issue</i>	<i>operand complet</i>	<i>Result</i>		
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	
MULT	F0	F2	F4	6			
SUBD	F8	F6	F2	7			
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

<u>Functional unit status</u>										
<i>Time</i>	<i>Name</i>	<i>Busy</i>	<i>Op</i>	<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</i>	<i>Fj?</i>	<i>Fk?</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	Yes	Mult	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	Yes	Sub	F8	F6	F2		Integer	Yes	No
	Divide	No								

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

Scoreboard Example Cycle 8a

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

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</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	F2		R3				Yes
	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

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

Scoreboard Example Cycle 8b

<u>Instruction status</u>				<i>Read Executic Write</i>			
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>operand complet</i>	<i>Result</i>	
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				

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</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	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		<i>F0</i>	<i>F2</i>	<i>F4</i>	<i>F6</i>	<i>F8</i>	<i>F10</i>	<i>F12</i>	...	<i>F30</i>	
8	<i>FU</i>	Mult1				Add	Divide				

Scoreboard Example Cycle 9

<u>Instruction status</u>				<i>Read</i>		<i>Executic</i>		<i>Write</i>	
Instruction	<i>j</i>	<i>k</i>	<i>R</i>	<i>Issue</i>	<i>operand complet</i>	<i>Result</i>			
LD	F6	34+	R2	1	2	3	4		
LD	F2	45+	R3	5	6	7	8		
MULT	F0	F2	F4	6	9				
SUBD	F8	F6	F2	7	9				
DIVD	F10	F0	F6	8					
ADDD	F6	F8	F2						

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</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	No								
10	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
2	Add	Yes	Sub	F8	F6	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

<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										
9	<i>FU</i>	Mult1			Add		Divide			

• Read operands for MULT & SUBD? Issue ADDD?

Scoreboard Example Cycle 11

<u>Instruction status</u>				<i>Read</i>		<i>Executic</i>		<i>Write</i>	
Instruction	<i>j</i>	<i>k</i>	<i>R</i>	<i>Issue</i>	<i>operand complet</i>	<i>Result</i>			
LD	F6	34+	R2	1	2	3	4		
LD	F2	45+	R3	5	6	7	8		
MULT	F0	F2	F4	6	9				
SUBD	F8	F6	F2	7	9	11			
DIVD	F10	F0	F6	8					
ADDD	F6	F8	F2						

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</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	No								
8	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
0	Add	Yes	Sub	F8	F6	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

<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										
11	<i>FU</i>	Mult1			Add		Divide			

Scoreboard Example Cycle 12

<u>Instruction status</u>				<i>Read</i>		<i>Executic</i>		<i>Write</i>	
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>operand complet</i>	<i>Result</i>			
LD	F6	34+	R2	1	2	3	4		
LD	F2	45+	R3	5	6	7	8		
MULT	F0	F2	F4	6	9				
SUBD	F8	F6	F2	7	9	11	12		
DIVD	F10	F0	F6	8					
ADDD	F6	F8	F2						

<u>Functional unit status</u>				<i>dest</i>		<i>S1</i>		<i>S2</i>		<i>FU for j</i>		<i>FU for k</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	No															
7	Mult1	Yes	Mult	F0	F2	F4								Yes	Yes		
	Mult2	No															
	Add	No															
	Divide	Yes	Div	F10	F0	F6	Mult1							No	Yes		

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

• Read operands for DIVD?

Scoreboard Example Cycle 13

<u>Instruction status</u>				<i>Read</i>		<i>Executic</i>		<i>Write</i>	
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>operand complet</i>	<i>Result</i>			
LD	F6	34+	R2	1	2	3	4		
LD	F2	45+	R3	5	6	7	8		
MULT	F0	F2	F4	6	9				
SUBD	F8	F6	F2	7	9	11	12		
DIVD	F10	F0	F6	8					
ADDD	F6	F8	F2						
				13					

<u>Functional unit status</u>				<i>dest</i>		<i>S1</i>		<i>S2</i>		<i>FU for j</i>		<i>FU for k</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	No															
6	Mult1	Yes	Mult	F0	F2	F4								Yes	Yes		
	Mult2	No															
	Add	Yes	Add	F6	F8	F2								Yes	Yes		
	Divide	Yes	Div	F10	F0	F6	Mult1							No	Yes		

<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>	
13	<i>FU</i>	Mult1			Add		Divide				

Scoreboard Example Cycle 14

<u>Instruction status</u>				<i>Read</i>		<i>Executic</i>		<i>Write</i>	
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>operand complet</i>	<i>Result</i>			
LD	F6	34+	R2	1	2	3	4		
LD	F2	45+	R3	5	6	7	8		
MULT	F0	F2	F4	6	9				
SUBD	F8	F6	F2	7	9	11	12		
DIVD	F10	F0	F6	8					
ADDD	F6	F8	F2	13	14				

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</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	No								
5	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
2	Add	Yes	Add	F6	F8	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

<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	14	FU		Mult1	Add	Divide				

Scoreboard Example Cycle 15

<u>Instruction status</u>				<i>Read</i>		<i>Executic</i>		<i>Write</i>	
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>operand complet</i>	<i>Result</i>			
LD	F6	34+	R2	1	2	3	4		
LD	F2	45+	R3	5	6	7	8		
MULT	F0	F2	F4	6	9				
SUBD	F8	F6	F2	7	9	11	12		
DIVD	F10	F0	F6	8					
ADDD	F6	F8	F2	13	14				

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</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	No								
4	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
1	Add	Yes	Add	F6	F8	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

<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	FU		Mult1	Add	Divide				

Scoreboard Example Cycle 16

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

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

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

Scoreboard Example Cycle 17

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

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

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

• Write result of ADDD (DIV did not read F6?)

Scoreboard Example Cycle 18

<u>Instruction status</u>				<i>Read</i>	<i>Executic</i>	<i>Write</i>					
Instruction	<i>j</i>	<i>k</i>	<i>R</i>	<i>Issue</i>	<i>operand</i>	<i>completi</i>	<i>Result</i>				
LD	F6	34+	R2	1	2	3	4				
LD	F2	45+	R3	5	6	7	8				
MULT	F0	F2	F4	6	9						
SUBD	F8	F6	F2	7	9	11	12				
DIVD	F10	F0	F6	8							
ADDD	F6	F8	F2	13	14	16					

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</i>	<i>Fj?</i>	<i>Fk?</i>
Time	Name	<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								
1	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

<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>	
18	<i>FU</i>	Mult1			Add		Divide				

Scoreboard Example Cycle 19

<u>Instruction status</u>				<i>Read</i>	<i>Executic</i>	<i>Write</i>					
Instruction	<i>j</i>	<i>k</i>	<i>R</i>	<i>Issue</i>	<i>operand</i>	<i>completi</i>	<i>Result</i>				
LD	F6	34+	R2	1	2	3	4				
LD	F2	45+	R3	5	6	7	8				
MULT	F0	F2	F4	6	9	19					
SUBD	F8	F6	F2	7	9	11	12				
DIVD	F10	F0	F6	8							
ADDD	F6	F8	F2	13	14	16					

<u>Functional unit status</u>				<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</i>	<i>Fj?</i>	<i>Fk?</i>
Time	Name	<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								
0	Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	Add	F6	F8	F2			Yes	Yes
	Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

<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>	
19	<i>FU</i>	Mult1			Add		Divide				

Scoreboard Example Cycle 22

<u>Instruction status</u>				Read		Executic Write	
Instruction	<i>j</i>	<i>k</i>		Issue	operand complet	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		
ADDD	F6	F8	F2	13	14	16	22

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

<u>Register result status</u>		F0	F2	F4	F6	F8	F10	F12	...	F30
Clock	22	FU Divide								

Scoreboard Example Cycle 61

<u>Instruction status</u>				Read		Executic Write	
Instruction	<i>j</i>	<i>k</i>		Issue	operand complet	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	
ADDD	F6	F8	F2	13	14	16	22

<u>Functional unit status</u>				dest	S1	S2	FU for j	FU for k	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			Yes	Yes

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

Scoreboard Example Cycle 62

<u>Instruction status</u>				<i>Read</i>	<i>Executic</i>	<i>Write</i>	
Instruction	<i>j</i>	<i>k</i>		<i>Issue</i>	<i>operand complet</i>	<i>Result</i>	
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

<u>Functional unit status</u>		<i>dest</i>	<i>S1</i>	<i>S2</i>	<i>FU for j</i>	<i>FU for k</i>	<i>Fj?</i>	<i>Fk?</i>	
<i>Time</i>	<i>Name</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								
	Mult1								
	Mult2								
	Add								
	0 Divide								

<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>
62	<i>FU</i>									