

UNIVERSITY OF MASSACHUSETTS
Dept. of Electrical & Computer Engineering

Computer Architecture
ECE 568

Part 5

Dynamic scheduling with Scoreboards

Israel Koren
Fall 2011

ECE568/Koren Part.5.1

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Advantages of Dynamic Scheduling

- ◆ Handles cases when dependencies unknown at compile time
- ◆ It simplifies the compiler
- ◆ Allows code that compiled for one pipeline to run efficiently on a different pipeline
- ◆ Hardware speculation, a technique with significant performance advantages, builds on dynamic scheduling
- ◆ Key idea: Allow instructions behind stall to proceed
DIVD F0, F2, F4
ADDD F10, F0, F8
SUBD F12, F8, F14
- ◆ Enables out-of-order execution & out-of-order completion

ECE568/Koren Part.5.2

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

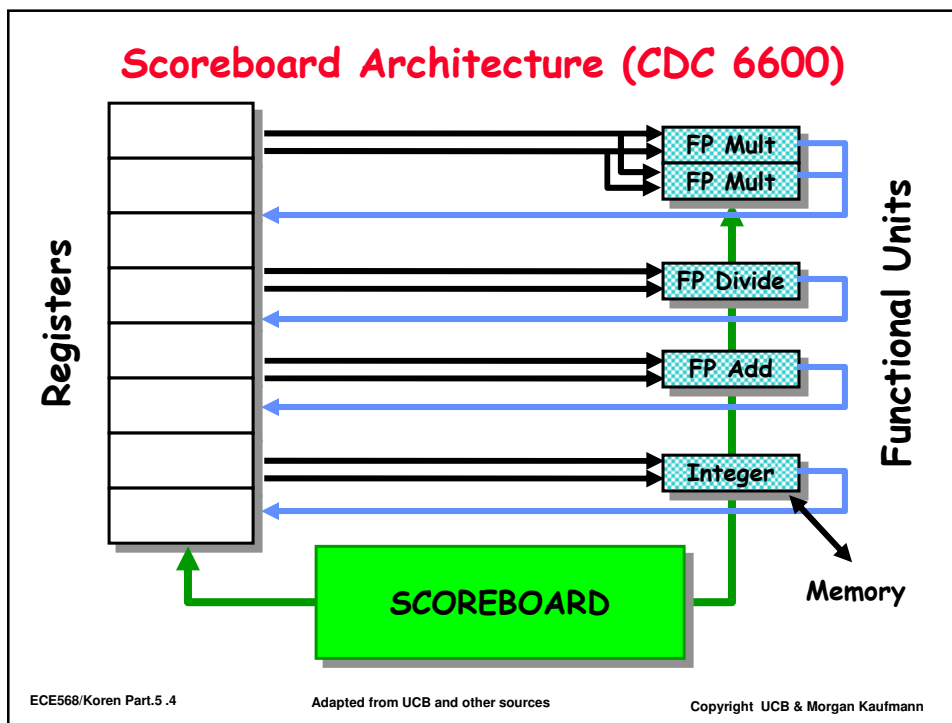
Scoreboard: a bookkeeping technique

- ◆ Scoreboards date to CDC 6600 in 1963
- ◆ Instructions execute whenever not dependent on previous instructions and no structural hazards
- ◆ CDC 6600: In order issue, out-of-order execution, out-of-order commit (or completion)
 - No forwarding
- ◆ Out-of-order execution divides ID stage:
 1. **Issue** — decode instructions, check for structural hazards
 2. **Read operands** — wait until no data hazards, then read operands

ECE568/Koren Part.5.3

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann



ECE568/Koren Part.5.4

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Implications

- ◆ Out-of-order completion => WAR, WAW hazards?
- ◆ Solutions for **WAR**:
 - Stall writeback until registers have been read
- ◆ 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 - 1

- ◆ Scoreboard replaces **ID, EX, WB** with 4 stages
- ◆ **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 any 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, since we wait for instructions to write back data
 - **No forwarding of data** in this model!

Four Stages of Scoreboard Control - 2

♦ **Execution** — operate on operands (**EX**)

- The functional unit begins execution upon receiving operands. When the result is ready, it notifies the scoreboard that it has completed execution

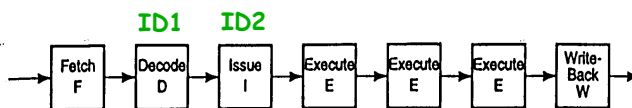
♦ **Write result** — finish execution (**WB**)

- Stall until no WAR hazards with previous instructions:

Example: DIVD F0, F2, F4
 ADDD F10, F0, F8
 SUBD F8, F6, F14

CDC 6600 scoreboard would stall SUBD until ADDD reads operands

Example: $X = Y + Z$ & $A = B * C$



In-order instruction issuing	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
$R_1 \leftarrow (Y)$	F	D	I	E	E	E	W															
$R_2 \leftarrow (Z)$		F	D	I	E	E	E	W														
$R_3 \leftarrow (R_1) + (R_2)$			F	D	*	*	I	E	E	E	W											
$X \leftarrow (R_3)$				F	*	*	D	*	*	I	E	E	E	W								
$R_4 \leftarrow (B)$						F	*	*	D	I	E	E	E	W								
$R_5 \leftarrow (C)$									F	D	I	E	E	E	W							
$R_6 \leftarrow (R_4) \times (R_5)$										F	D	*	*	I	E	E	E	W				
$A \leftarrow (R_6)$											F	*	*	D	*	*	I	E	E	E	W	

Software static scheduling reduced the # of cycles to 16

Dynamic Scheduling - Scoreboarding

Assume: Two LOADs can overlap & data forwarding

	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8
$R_1 \leftarrow (Y)$	F	D	I	E	E	E	W											
$R_2 \leftarrow (Z)$		F	D	I	E	E	E	W										
$R_3 \leftarrow (R_1) + (R_2)$			F	D	I	*	*	E	E	E	W							
$X \leftarrow (R_3)$				F	D	I	*	*	*	*	E	E	E	W				
$R_4 \leftarrow (B)$					F	D	I	E	E	E	W							
$R_5 \leftarrow (C)$						F	D	I	E	E	E	W						
$R_6 \leftarrow (R_4) \times (R_5)$							F	D	I	*	*	E	E	E	W			
$A \leftarrow (R_6)$								F	D	I	*	*	*	*	E	E	E	W

- 18 vs. 22 cycles
- No structural hazards, only data hazards

Source: J. Smith, IEEE Computer, July 1989.

ECE568/Koren Part.5.9

Three Parts of the Scoreboard

- ◆ **Instruction status:**
Which of 4 steps the instruction is in
- ◆ **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., ADD or SUB)
 - Fi:** Destination register
 - Fj, Fk:** Source-register numbers
 - Rj, Rk:** Flags indicating when Fj, Fk are ready
 - Qj, Qk:** Functional units producing source registers Fj, Fk
- ◆ **Register result status** — Indicates which functional unit will write each register, if such FU exists. Blank when no pending instructions will write that register

ECE568/Koren Part.5.10

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example

Instruction status:

Instruction	j	k	Read Exec Write		
			Issue	Oper	Comp Result
LD	F6	34+	R2		
LD	F2	45+	R3		
MULTD	F0	F2	F4		
SUBD	F8	F6	F2		
DIVD	F10	F0	F6		
ADDD	F6	F8	F2		

Functional unit status:

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

Register result status:

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

ECE568/Koren Part.5 .11

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Detailed Scoreboard Pipeline Control

Op D, S1, S2

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

ECE568/Koren Part.5 .12

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 1

Instruction status:

Instruction	j	k	Read		Exec	Write
			Issue	Oper	Comp	Result
LD	F6	34+	R2	1		
LD	F2	45+	R3			
MULTD	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADD	F6	F8	F2			

Functional unit status:

Time Name	Busy	Op	dest		S1	S2	FU	FU	Fj?	Fk?
			Fi	Fj	Fk	Oj	Ok	Rj	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				Integer					

ECE568/Koren Part.5.13

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 2

Instruction status:

Instruction	j	k	Read		Exec	Write
			Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	
LD	F2	45+	R3			
MULTD	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADD	F6	F8	F2			

Functional unit status:

Time Name	Busy	Op	dest		S1	S2	FU	FU	Fj?	Fk?
			Fi	Fj	Fk	Oj	Ok	Rj	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
2				Integer					

• Issue 2nd LD?

ECE568/Koren Part.5.14

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 3

Instruction status:

Instruction	j	k	Read		Exec	Write
			Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3
LD	F2	45+	R3			
MULTD	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

Functional unit status:

Time	Name	Busy	Op	dest		S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	Rj	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								

• Issue MULT?

ECE568/Koren Part.5.15

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 4

Instruction status:

Instruction	j	k	Read		Exec	Write
			Issue	Oper	Comp	Result
LD	F6	34+	R2	1	2	3
LD	F2	45+	R3			4
MULTD	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

Functional unit status:

Time	Name	Busy	Op	dest		S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	Rj	Rk	
Integer		No									
Mult1		No									
Mult2		No									
Add		No									
Divide		No									

Register result status:

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

ECE568/Koren Part.5.16

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 5

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5			
MULTD	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest		S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	Rj	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								

ECE568/Koren Part.5.17

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 6

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6		
MULTD	F0	F2	F4	6			
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest		S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	Rj	Rk	
Integer		Yes	Load	F2			R3				Yes
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								

ECE568/Koren Part.5.18

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 7

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	
MULTD	F0	F2	F4	6			
SUBD	F8	F6	F2	7			
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	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		Integer	Yes	No
Divide		No								

Register result status:

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

- Read multiply operands?

ECE568/Koren Part.5 .19

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 8a (1st half of cycle)

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	
MULTD	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	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	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		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			

ECE568/Koren Part.5 .20

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 8b (2nd half of cycle)

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	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	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	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

Register result status:

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

ECE568/Koren Part.5 .21

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 9

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9		
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	Rj	Rk
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

Note →
remaining
cycles

Register result status:

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

- Read operands for MULT & SUB. Issue ADDD?

ECE568/Koren Part.5 .22

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 12

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	12
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

Functional unit status:

Time	Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	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:

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

- Read operands for DIVD? Issue ADDD?

ECE568/Koren Part.5 .23

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 17

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULTD	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	

WAR Hazard!

Functional unit status:

Time	Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	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:

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

- Why not write result of ADD???

ECE568/Koren Part.5 .24

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Scoreboard Example: Cycle 21

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
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	9	11	12
DIVD	F10	F0	F6	8	21		
ADDD	F6	F8	F2	13	14	16	

Functional unit status:

Time	Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	Rj	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
21	FU			Add	Divide				

- WAR Hazard is now gone...

Scoreboard Example: Cycle 22

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
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	9	11	12
DIVD	F10	F0	F6	8	21		
ADDD	F6	F8	F2	13	14	16	22

Functional unit status:

Time	Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	Rj	Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
39	Divide	Yes	Div	F10	F0	F6			No	No

Register result status:

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

Scoreboard Example: Cycle 61

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
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	9	11	12
DIVD	F10	F0	F6	8	21	61	
ADDD	F6	F8	F2	13	14	16	22

Functional unit status:

Time	Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	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			

ECE568/Koren Part.5 .27

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

Review: Scoreboard Example: Cycle 62

Instruction status:

Instruction	j	k	Read		Exec	Write	
			Issue	Oper	Comp	Result	
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	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	S1	S2	FU	FU	Fj?	Fk?
				Fi	Fj	Fk	Oj	Ok	Rj	Rk
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Register result status:

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

- In-order issue; out-of-order execute & commit

ECE568/Koren Part.5 .28

Adapted from UCB and other sources

Copyright UCB & Morgan Kaufmann

CDC 6600 Scoreboard - Summary

- ◆ **Key idea of Scoreboard: Allow instructions behind stall to proceed (Decode \Rightarrow Issue instruction & read operands)**
 - Enables out-of-order execution \Rightarrow out-of-order completion
 - ID stage checked both for structural & data dependencies
 - Original version didn't handle forwarding
 - Limited to instructions in basic block (small *window*)
 - Small number of functional units (structural hazards), especially integer/load store units
 - Prevent WAW hazards, but
 - Wait for WAR hazards