

Lecture 5: Introduction to Advanced Pipelining: Out of Order Execution

Prof. John Kubiawicz
Computer Science 252
Fall 1998

JDK.F98
Slide 1

Review: Exceptions and Compiler Scheduling

- Exceptional control flow comes in three flavors:
 - Exceptions - relevant to current process
 - Interrupts - caused by external events
 - Machine checks - Extreme situations
- Such exceptional flow can also be classified as synchronous or asynchronous
- Precise exceptions or interrupts break the control flow at a well defined instruction such that:
 - All logically prior instructions have completed and committed state
 - Neither the instruction or any following instructions have committed state
- Careful compiler scheduling can remove stalls and speed up code. Dependencies must be maintained.
- Loop unrolling can offer additional parallelism.

JDK.F98
Slide 2

Can we use HW to get CPI closer to 1?

- Why in HW at run time?
 - Works when can't know real dependence at compile time
 - Compiler simpler
 - Code for one machine runs well on another
- Key idea: Allow instructions behind stall to proceed


```
DIVD  F0, F2, F4
ADDD  F10, F0, F8
SUBD  F12, F8, F14
```
- Out-of-order execution => out-of-order completion.

JDK.F98
Slide 3

Problems?

- How do we prevent WAR and WAW hazards?
- How do we deal with variable latency?
 - Forwarding for RAW hazards harder.

Instruction	Clock Cycle Number																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
LD F6,34(R2)	IF	ID	EX	MEM	WB												
LD F2,45(R3)		IF	ID	EX	MEM	WB											
MULTD F0,F2,F4			IF	ID	stall	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	MEM	WB
SUBD F8,F6,F2				IF	ID	A1	A2	MEM	WB								
DIVD F10,F0,F6					IF	ID	stall	stall	stall	stall	stall	stall	stall	stall	stall	D1	D2
ADDD F6,F8,F2						IF	ID	A1	A2	MEM	WB						

RAW hazard indicated by a red double-headed arrow between the MEM stage of the second LD instruction and the MEM stage of the MULTD instruction.

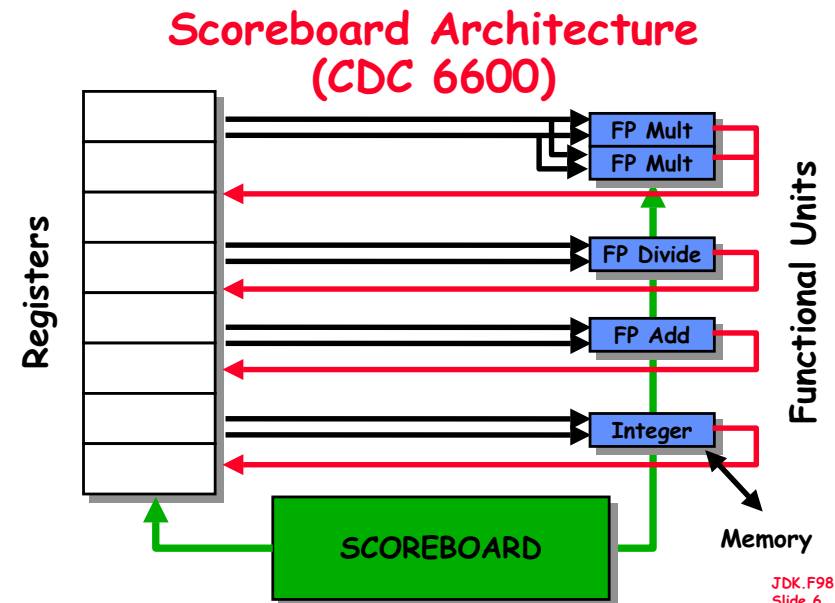
WAR hazard indicated by a red double-headed arrow between the MEM stage of the ADDD instruction and the MEM stage of the DIVD instruction.

JDK.F98
Slide 4

Scoreboard: a bookkeeping technique

- 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
- Scoreboards date to CDC6600 in 1963
- Instructions execute whenever not dependent on previous instructions and no hazards.
- CDC 6600: In order issue, out-of-order execution, out-of-order commit (or completion)
 - No forwarding!
 - Imprecise interrupt/exception model for now

JDK.F98
Slide 5



JDK.F98
Slide 6

Scoreboard Implications

- Out-of-order completion => WAR, WAW hazards?
- Solutions for WAR:
 - Stall writeback 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
- No register renaming!
- 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.
- Scoreboard replaces ID, EX, WB with 4 stages

JDK.F98
Slide 7

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 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!

JDK.F98
Slide 8

Four Stages of Scoreboard Control

- **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, F8, F14
    
```

CDC 6600 scoreboard would stall SUBD until ADDD reads operands

JDK.F98
Slide 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., + 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
- **Register result status**—Indicates which functional unit will write each register, if one exists. Blank when no pending instructions will write that register

JDK.F98
Slide 10

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						
Integer	No									
Multi1	No									
Multi2	No									
Add	No									
Divide	No									

Register result status:

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

JDK.F98
Slide 11

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	$\forall f((Fj(f) \neq Fi(FU) \text{ or } Rj(f) = \text{No}) \& (Fk(f) \neq Fi(FU) \text{ or } Rk(f) = \text{No}))$	$\forall f(\text{if } Qj(f) = FU \text{ then } Rj(f) \leftarrow \text{Yes});$ $\forall f(\text{if } Qk(f) = FU \text{ then } Rj(f) \leftarrow \text{Yes});$ $Result(Fi(FU)) \leftarrow 0; Busy(FU) \leftarrow \text{No}$

JDK.F98
Slide 12

Scoreboard Example: Cycle 1

Instruction status:

Instruction	j	k	Read		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			
ADDD	F6	F8	F2			

Functional unit status:

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

JDK.F98
Slide 13

Scoreboard Example: Cycle 2

Instruction status:

Instruction	j	k	Read		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			
ADDD	F6	F8	F2			

Functional unit status:

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

JDK.F98
Slide 14

Scoreboard Example: Cycle 3

Instruction status:

Instruction	j	k	Read		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						
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				Integer					

• Issue MULT?

JDK.F98
Slide 15

Scoreboard Example: Cycle 4

Instruction status:

Instruction	j	k	Read		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						
Integer	No									
Mult1	No									
Mult2	No									
Add	No									
Divide	No									

Register result status:

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

JDK.F98
Slide 16

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						
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							

JDK.F98
Slide 17

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						
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							

JDK.F98
Slide 18

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						
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?

JDK.F98
Slide 19

Scoreboard Example: Cycle 8a (First half of clock 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						
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				

JDK.F98
Slide 20

Scoreboard Example: Cycle 8b (Second half of clock 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						
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			

JDK.F98
Slide 21

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						
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

Register result status:

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

• Read operands for MULT & SUB? Issue ADDD? JDK.F98
Slide 22

Scoreboard Example: Cycle 10

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						
Integer	No									
9 Mult1	Yes	Mult	F0	F2	F4				No	No
Mult2	No									
1 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
10	FU		Mult1		Add	Divide			

JDK.F98
Slide 23

Scoreboard Example: Cycle 11

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	
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						
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			

JDK.F98
Slide 24

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	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:

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

- Read operands for DIVD?

JDK.F98
Slide 25

Scoreboard Example: Cycle 13

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			

Functional unit status:

Time Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
			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			

JDK.F98
Slide 26

Scoreboard Example: Cycle 14

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		

Functional unit status:

Time Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
			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			Yes	Yes
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			

JDK.F98
Slide 27

Scoreboard Example: Cycle 15

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		

Functional unit status:

Time Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
			Fi	Fj	Fk	Qj	Qk	Rj	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			

JDK.F98
Slide 28

Scoreboard Example: Cycle 16

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			

Functional unit status:

Time Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
			Op	Fi	Fj	Fk	Qj	Qk	Rj
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	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
16		Mult1			Add		Divide			

JDK.F98
Slide 29

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?
			Op	Fi	Fj	Fk	Qj	Qk	Rj
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	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
17		Mult1			Add		Divide			

• Why not write result of ADD???

JDK.F98
Slide 30

Scoreboard Example: Cycle 18

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			

Functional unit status:

Time Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
			Op	Fi	Fj	Fk	Qj	Qk	Rj
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	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
18		Mult1			Add		Divide			

JDK.F98
Slide 31

Scoreboard Example: Cycle 19

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			
SUBD	F8	F6 F2	7	9	11	12		
DIVD	F10	F0 F6	8					
ADDD	F6	F8 F2	13	14	16			

Functional unit status:

Time Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
			Op	Fi	Fj	Fk	Qj	Qk	Rj
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	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
19		Mult1			Add		Divide			

JDK.F98
Slide 32

Scoreboard Example: Cycle 20

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				
ADDD	F6	F8	F2	13	14	16		

Functional unit status:

Time Name	Busy	Op	dest	S1	S2	FU	FU	Fj?	Fk?
			Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Multi1	No								
Multi2	No								
Add	Yes	Add	F6	F8	F2			No	No
Divide	Yes	Div	F10	F0	F6			Yes	Yes

Register result status:

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

JDK.F98
Slide 33

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	Qj	Qk	Rj	Rk
Integer	No								
Multi1	No								
Multi2	No								
Add	Yes	Add	F6	F8	F2			No	No
Divide	Yes	Div	F10	F0	F6			Yes	Yes

Register result status:

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

• WAR Hazard is now gone...

JDK.F98
Slide 34

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	Qj	Qk	Rj	Rk
Integer	No								
Multi1	No								
Multi2	No								
Add	No								
39 Divide	Yes	Div	F10	F0	F6			No	No

Register result status:

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

JDK.F98
Slide 35

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	Qj	Qk	Rj	Rk
Integer	No								
Multi1	No								
Multi2	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			

JDK.F98
Slide 36

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	Qj	Qk	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										

JDK.F98
Slide 37

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	Qj	Qk	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

JDK.F98
Slide 38

CDC 6600 Scoreboard

- Speedup 1.7 from compiler; 2.5 by hand BUT slow memory (no cache) limits benefit
- Limitations of 6600 scoreboard:
 - No forwarding hardware
 - Limited to instructions in basic block (small *window*)
 - Small number of functional units (structural hazards), especially integer/load store units
 - Do not issue on structural hazards
 - Wait for WAR hazards
 - Prevent WAW hazards

JDK.F98
Slide 39

CS 252 Administrivia

- Today is last day for TeleBears.
- Get your photo taken by Aaron!
- Textbook Reading for Lectures 6 to 8
 - Computer Architecture: A Quantitative Approach,
 - » Rest of Chapter 4, Appendix B
- Wednesday papers:
 - Fisher, "Very Long Instruction Word Architectures and the ELI-512"
 - Colwell, et al, "A VLIW Architecture for a Trace Scheduling Compiler"
 - » Joint summary on these
 - Smith, et al, "Efficient Superscalar Performance Through Boosting"
 - » Summary on this
- Paper summaries:
 - Will hand out first three sets next Wednesday
 - you can skip two paper summaries (paragraphs) this term.
 - you can be late on one days worth of summaries. Arrange this with Aaron.

JDK.F98
Slide 40

CS 252 Administrivia

- Exercises for Lectures 3 to 8
 - Due Friday Sept 18 in class
 - 4.2, 4.10, 4.19, 4.24
 - 4.14 parts c) and d) only
 - B.2
 - Done in pairs, but both need to understand whole assignment
 - Study groups encouraged, but pairs do own work
- Comment on Syllabus:
 - Some chaos should be expected
 - Next couple of topics:
 - » Superscalar execution
 - » VLIWs
 - » Data and Control Prediction

JDK.F98
Slide 41

Another Dynamic Algorithm: Tomasulo Algorithm

- For IBM 360/91 about 3 years after CDC 6600 (1966)
- Goal: High Performance without special compilers
- Differences between IBM 360 & CDC 6600 ISA
 - IBM has only 2 register specifiers/instr vs. 3 in CDC 6600
 - IBM has 4 FP registers vs. 8 in CDC 6600
 - IBM has memory-register ops
- Why Study? lead to Alpha 21264, HP 8000, MIPS 10000, Pentium II, PowerPC 604, ...

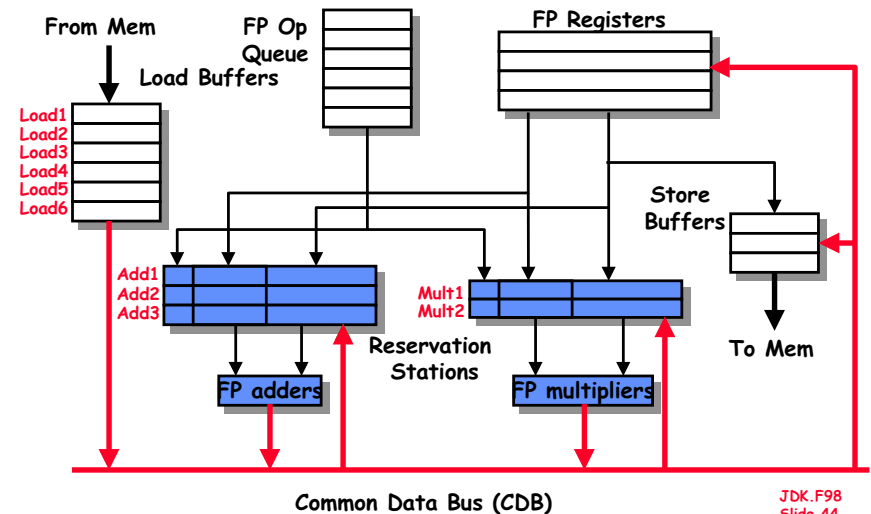
JDK.F98
Slide 42

Tomasulo Algorithm vs. Scoreboard

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

JDK.F98
Slide 43

Tomasulo Organization



JDK.F98
Slide 44

Reservation Station Components

Op: Operation to perform in the unit (e.g., + or -)

Vj, Vk: Value of Source operands

- Store buffers has V field, result to be stored

Qj, Qk: Reservation stations producing source registers (value to be written)

- Note: No ready flags as in Scoreboard; Qj, Qk=0 => ready
- Store buffers only have Qi for RS producing result

Busy: Indicates reservation station or FU is busy

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

JDK.F98
Slide 45

Three Stages of Tomasulo Algorithm

1. Issue—get instruction from FP Op Queue

If reservation station free (no structural hazard), control issues instr & sends operands (renames registers).

2. Execution—operate on operands (EX)

When both operands ready then execute; if not ready, watch Common Data Bus for result

3. Write result—finish execution (WB)

Write on Common Data Bus to all awaiting units; mark reservation station available

- Normal data bus: data + destination ("go to" bus)
- Common data bus: data + source ("come from" bus)
 - 64 bits of data + 4 bits of Functional Unit source address
 - Write if matches expected Functional Unit (produces result)
 - Does the broadcast

JDK.F98
Slide 46

Tomasulo Example

Instruction status:

Instruction	j	k	Exec Write		Busy	Address
			Issue	Comp Result		
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:

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
Add1	No						
Add2	No						
Add3	No						
Mult1	No						
Mult2	No						

Register result status:

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

JDK.F98
Slide 47

Tomasulo Example Cycle 1

Instruction status:

Instruction	j	k	Exec Write		Busy	Address
			Issue	Comp Result		
LD	F6	34+	R2	I	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:

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
Add1	No						
Add2	No						
Add3	No						
Mult1	No						
Mult2	No						

Register result status:

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

JDK.F98
Slide 48

Tomasulo Example Cycle 2

Instruction status:

Instruction	j	k	Exec Write		Busy	Address
			Issue	Comp Result		
LD	F6	34+	R2	1	No	34+R2
LD	F2	45+	R3	2	Yes	45+R3
MULTD	F0	F2	F4		No	
SUBD	F8	F6	F2		No	
DIVD	F10	F0	F6		No	
ADDD	F6	F8	F2		No	

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
Add1		No					
Add2		No					
Add3		No					
Mult1		No					
Mult2		No					

Register result status:

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

Note: Unlike 6600, can have multiple loads outstanding

Tomasulo Example Cycle 3

Instruction status:

Instruction	j	k	Exec Write		Busy	Address
			Issue	Comp Result		
LD	F6	34+	R2	1	No	34+R2
LD	F2	45+	R3	2	Yes	45+R3
MULTD	F0	F2	F4	3	No	
SUBD	F8	F6	F2		No	
DIVD	F10	F0	F6		No	
ADDD	F6	F8	F2		No	

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
Add1		No					
Add2		No					
Add3		No					
Mult1	Yes	MULTD		R(F4)		Load2	
Mult2	No						

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
3	FU	Mult1	Load2		Load1				

- Note: registers names are removed ("renamed") in Reservation Stations; MULT issued vs. scoreboard
- Load1 completing; what is waiting for Load1?

Tomasulo Example Cycle 4

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	No	
LD	F2	45+	R3	2	4		Yes	45+R3
MULTD	F0	F2	F4	3			No	
SUBD	F8	F6	F2	4			No	
DIVD	F10	F0	F6				No	
ADDD	F6	F8	F2				No	

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
Add1	Yes	SUBD		M(A1)			Load2
Add2	No						
Add3	No						
Mult1	Yes	MULTD		R(F4)		Load2	
Mult2	No						

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
4	FU	Mult1	Load2		M(A1)	Add1			

- Load2 completing; what is waiting for Load1?

Tomasulo Example Cycle 5

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	No	
LD	F2	45+	R3	2	4	5	No	
MULTD	F0	F2	F4	3			No	
SUBD	F8	F6	F2	4			No	
DIVD	F10	F0	F6	5			No	
ADDD	F6	F8	F2				No	

Reservation Stations:

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

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
5	FU	Mult1	M(A2)		M(A1)	Add1	Mult2		

Tomasulo Example Cycle 6

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	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	6				

Reservation Stations:

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

Register result status:

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

• Issue ADDD here vs. scoreboard?

JDK.F98
Slide 53

Tomasulo Example Cycle 7

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	F0	F2	F4	3			Load3	No
SUBD	F8	F6	F2	4	7			
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6				

Reservation Stations:

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

Register result status:

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

• Add1 completing; what is waiting for it?

JDK.F98
Slide 54

Tomasulo Example Cycle 8

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	F0	F2	F4	3			Load3	No
SUBD	F8	F6	F2	4	7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6				

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
2	Add2	Yes	ADDD	(M-M)	M(A2)		
3	Add3	No					
7	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD	M(A1)	Mult1		

Register result status:

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

JDK.F98
Slide 55

Tomasulo Example Cycle 9

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	F0	F2	F4	3			Load3	No
SUBD	F8	F6	F2	4	7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6				

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
1	Add2	Yes	ADDD	(M-M)	M(A2)		
2	Add3	No					
6	Mult1	Yes	MULTD	M(A2)	R(F4)		
	Mult2	Yes	DIVD	M(A1)	Mult1		

Register result status:

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

JDK.F98
Slide 56

Tomasulo Example Cycle 10

Instruction status:

Instruction	j	k	Exec Write			Issue	Comp	Result	Busy	Address
			S1	S2	RS					
LD	F6	34+	R2			1	3	4	Load1	No
LD	F2	45+	R3			2	4	5	Load2	No
MULTD	F0	F2	F4			3			Load3	No
SUBD	F8	F6	F2			4	7	8		
DIVD	F10	F0	F6			5				
ADDD	F6	F8	F2			6	10			

Reservation Stations:

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

Register result status:

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

- Add2 completing; what is waiting for it?

JDK.F98
Slide 57

Tomasulo Example Cycle 11

Instruction status:

Instruction	j	k	Exec Write			Issue	Comp	Result	Busy	Address
			S1	S2	RS					
LD	F6	34+	R2			1	3	4	Load1	No
LD	F2	45+	R3			2	4	5	Load2	No
MULTD	F0	F2	F4			3			Load3	No
SUBD	F8	F6	F2			4	7	8		
DIVD	F10	F0	F6			5				
ADDD	F6	F8	F2			6	10	11		

Reservation Stations:

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

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
11	FU		Mult1	M(A2)	(M-M+N(M-M))	Mult2			

- Write result of ADDD here vs. scoreboard?
- All quick instructions complete in this cycle!

JDK.F98
Slide 58

Tomasulo Example Cycle 12

Instruction status:

Instruction	j	k	Exec Write			Issue	Comp	Result	Busy	Address
			S1	S2	RS					
LD	F6	34+	R2			1	3	4	Load1	No
LD	F2	45+	R3			2	4	5	Load2	No
MULTD	F0	F2	F4			3			Load3	No
SUBD	F8	F6	F2			4	7	8		
DIVD	F10	F0	F6			5				
ADDD	F6	F8	F2			6	10	11		

Reservation Stations:

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

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
12	FU		Mult1	M(A2)	(M-M+N(M-M))	Mult2			

JDK.F98
Slide 59

Tomasulo Example Cycle 13

Instruction status:

Instruction	j	k	Exec Write			Issue	Comp	Result	Busy	Address
			S1	S2	RS					
LD	F6	34+	R2			1	3	4	Load1	No
LD	F2	45+	R3			2	4	5	Load2	No
MULTD	F0	F2	F4			3			Load3	No
SUBD	F8	F6	F2			4	7	8		
DIVD	F10	F0	F6			5				
ADDD	F6	F8	F2			6	10	11		

Reservation Stations:

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

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
13	FU		Mult1	M(A2)	(M-M+N(M-M))	Mult2			

JDK.F98
Slide 60

Tomasulo Example Cycle 14

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	F0	F2	F4	3			Load3	No
SUBD	F8	F6	F2	4	7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	10	11		

Reservation Stations:

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

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
14	FU	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2		

JDK.F98
Slide 61

Tomasulo Example Cycle 15

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	F0	F2	F4	3	15		Load3	No
SUBD	F8	F6	F2	4	7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	10	11		

Reservation Stations:

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

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
15	FU	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2		

JDK.F98
Slide 62

Tomasulo Example Cycle 16

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	F0	F2	F4	3	15	16	Load3	No
SUBD	F8	F6	F2	4	7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	10	11		

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
Add1		No					
Add2		No					
Add3		No					
	Mult1	No					
40	Mult2	Yes	DIVD	M*F4		M(A1)	

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
16	FU	M*F4	M(A2)		(M-M+N)	(M-M)	Mult2		

JDK.F98
Slide 63

Faster than light computation
(skip a couple of cycles)

JDK.F98
Slide 64

Tomasulo Example Cycle 55

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	F0	F2	F4	3	15	16	Load3	No
SUBD	F8	F6	F2	4	7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	10	11		

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
Add1		No					
Add2		No					
Add3		No					
Mult1		No					
1 Mult2	Yes	DIVD	M*F4	M(A1)			

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
55	FU	M*F4	M(A2)		(M-M+N (M-M))	Mult2			

JDK.F98
Slide 65

Tomasulo Example Cycle 56

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	F0	F2	F4	3	15	16	Load3	No
SUBD	F8	F6	F2	4	7	8		
DIVD	F10	F0	F6	5				
ADDD	F6	F8	F2	6	10	11		

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
Add1		No					
Add2		No					
Add3		No					
Mult1		No					
0 Mult2	Yes	DIVD	M*F4	M(A1)			

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
56	FU	M*F4	M(A2)		(M-M+N (M-M))	Mult2			

- Mult2 is completing; what is waiting for it?

JDK.F98
Slide 66

Tomasulo Example Cycle 57

Instruction status:

Instruction	j	k	Exec Write			Busy	Address	
			Issue	Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
LD	F2	45+	R3	2	4	5	Load2	No
MULTD	F0	F2	F4	3	15	16	Load3	No
SUBD	F8	F6	F2	4	7	8		
DIVD	F10	F0	F6	5	56	57		
ADDD	F6	F8	F2	6	10	11		

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
Add1		No					
Add2		No					
Add3		No					
Mult1		No					
0 Mult2	Yes	DIVD	M*F4	M(A1)			

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
56	FU	M*F4	M(A2)		(M-M+N (M-M))	Mult2			

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

JDK.F98
Slide 67

Compare to Scoreboard Cycle 62

Instruction status:

Instruction	j	k	Read Exec Write			Exec Write				
			Issue	Oper	Comp	Result	Issue	Comp	Result	
LD	F6	34+	R2	1	2	3	4	1	3	4
LD	F2	45+	R3	5	6	7	8	2	4	5
MULTD	F0	F2	F4	6	9	19	20	3	15	16
SUBD	F8	F6	F2	7	9	11	12	4	7	8
DIVD	F10	F0	F6	8	21	61	62	5	56	57
ADDD	F6	F8	F2	13	14	16	22	6	10	11

- Why take longer on scoreboard/6600?
 - Structural Hazards
 - Lack of forwarding

JDK.F98
Slide 68

Tomasulo v. Scoreboard (IBM 360/91 v. CDC 6600)

Pipelined Functional Units (6 load, 3 store, 3 +, 2 x/÷)
Multiple Functional Units (1 load/store, 1 +, 2 x, 1 ÷)

window size: ≤ 14 instructions

No issue on structural hazard

WAR: renaming avoids

WAW: renaming avoids

Broadcast results from FU

Control: reservation stations

≤ 5 instructions

same

stall completion

stall issue

Write/read registers

central scoreboard

JDK.F98
Slide 69

Tomasulo Drawbacks

- Complexity
 - delays of 360/91, MIPS 10000, IBM 620?
- Many associative stores (CDB) at high speed
- Performance limited by Common Data Bus
 - Multiple CDBs => more FU logic for parallel assoc stores

JDK.F98
Slide 70

Summary #1

- HW exploiting ILP
 - Works when can't know dependence at compile time.
 - Code for one machine runs well on another
- Key idea of Scoreboard: Allow instructions behind stall to proceed (Decode => Issue instr & read operands)
 - Enables out-of-order execution => out-of-order completion
 - ID stage checked both for structural & data dependencies
 - Original version didn't handle forwarding.
 - No automatic register renaming

JDK.F98
Slide 71

Summary #2

- Reservations stations: renaming to larger set of registers + buffering source operands
 - Prevents registers as bottleneck
 - Avoids WAR, WAW hazards of Scoreboard
 - Allows loop unrolling in HW
- Not limited to basic blocks (integer units gets ahead, beyond branches)
- Helps cache misses as well
- Lasting Contributions
 - Dynamic scheduling
 - Register renaming
 - Load/store disambiguation
- 360/91 descendants are Pentium II; PowerPC 604; MIPS R10000; HP-PA 8000; Alpha 21264

JDK.F98
Slide 72