

CS152 – Computer Architecture and Engineering

Lecture 17 – Advanced Pipelining: Tomasulo Algorithm

2003-10-23

Dave Patterson
(www.cs.berkeley.edu/~patterson)

www-inst.eecs.berkeley.edu/~cs152/



Scoreboard Review

- HW exploiting ILP (Instruction Level Parallelism)
 - 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 instruction & read operands)
 - Enables out-of-order execution => out-of-order completion (but in order execution)
 - ID stage checked both for structural & data dependencies
 - Original version didn't handle forwarding.
 - No automatic register renaming = WAW, WAR stalls



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 Pro, Pentium 4, PowerPC 604, ...

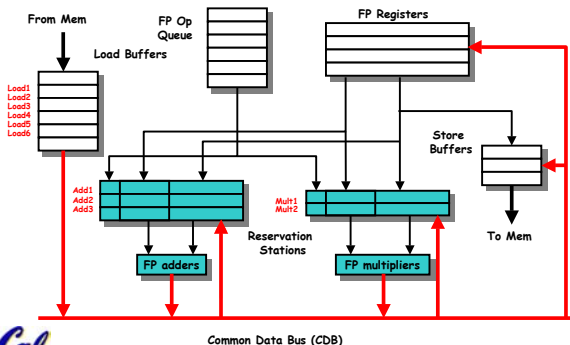


Tomasulo Algorithm vs. Scoreboard

- Control & buffers **distributed** with Function Units (FU) vs. centralized in scoreboard;
 - FU buffers called "**reservation stations**" (RS); 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



Tomasulo Organization



Reservation Station Components

- Busy:** Indicates reservation station or FU is busy (*like CDC*)
- Op:** Operation to perform in the unit (+, -, ...) (*like CDC*)
- Vj, Vk:** **V**alue of Source operands
 - Store buffers has V field, result to be stored
 - (CDC scoreboard had 3 register numbers, not their values)
- Qj, Qk:** Reservation stations producing source registers (value to be written) (*like CDC*)
 - Note: No register ready flags as in Scoreboard; Qj, Qk=0 => **ready** (nothing is writing them)
 - Store buffers only have Qi for RS producing result
- Register result status**—Indicates which functional unit will write each register, if one exists. Blank when no pending instructions that will write that register. (*like CDC*)



Three Stages of Tomasulo Algorithm

- Issue**—get instruction from FP Op Queue
If reservation station free (no structural hazard), control issues instr & sends operands (renames registers).
 - Execution**—operate on operands (EX)
When both operands ready then execute; if not ready, watch Common Data Bus for result
 - 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

Detailed Tomasulo Pipeline Control

Instruction status	Wait until	Action or bookkeeping
Issue	Station or buffer empty (No structural hazard)	$RS[r].Busy \leftarrow \text{yes}; Register[D].Q1 \leftarrow r;$ if (Register[S1].Q1 = 0) $RS[r].Q1 \leftarrow Register[S1].Q1$ else $RS[r].Vj \leftarrow S1; RS[r].Qj \leftarrow 0;$ if (Register[S2].Q1 = 0) $RS[r].Qk \leftarrow Register[S2].Q1$ else $RS[r].Vk \leftarrow S2; RS[r].Qk \leftarrow 0;$ (Mark RS busy and D to be written by RS if RAW on S1 or S2, copy RS into Q, else copy data into V and set Q to 0)
Execute	$RS[r].Qj=0$ and $RS[r].Qk=0$ (No RAW hazard)	None (operands are in Vj and Vk)
Write result	Execution completed at r and CDB available (No structural hazard on CDB)	$RS[r].Busy \leftarrow \text{No}$ $Vx \leftarrow (Register[x].Q1=r)$ $Fx \leftarrow \text{result}; Register[x].Q1 \leftarrow 0;$ $Vk \leftarrow (RS[r].Qj=r)$ $RS[x].Vj \leftarrow \text{result}; RS[x].Qj \leftarrow 0;$ $Vx \leftarrow (RS[r].Qk=r)$ $RS[x].Vk \leftarrow \text{result}; RS[x].Qk \leftarrow 0;$ $Vx \leftarrow (Store[x].Q1=r)$ $(Store[x].V \leftarrow \text{result}; Store[x].Q1 \leftarrow 0);$ (Mark RS free. For all other RS, if waiting for result, write and reset Q)

D = destination, S1 and S2 = source register numbers, and r is the reservation station or buffer that D is assigned to, RS is the reservation-station data structure. The value returned by a reservation station or by the load unit is called result. Register is the register data structure (control, not the register file), while Store is the store-buffer data structure.



Administrivia

- Design full cache, but only simulation on Friday 10/24; demo board Friday 10/31
- Thur 11/6: Design Doc for Final Project due
 - Deep pipeline? Superscalar? Out-of-order?
 - Read section 4.2 from CA:AQA 2/e
- Fri 11/14: Demo Project modules
- Wed 11/19: 5:30 PM Midterm 2 in 1 LeConte
- Tues 11/22: Field trip to Xilinx
- CS 152 Project week: 12/1 to 12/5
 - Mon: TA Project demo, Tue: 30 min Presentation, Wed: Processor racing, Fri: Written report



Tomasulo Example

Instruction status:

Instruction	j	k	Issue	Comp	Result	Busy	Address
LD	F6	34+	R2			Load1	No
LD	F2	45+	R3			Load2	No
MULTD	F0	F2	F4			Load3	No
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Reservation Stations:

Time	Name	Busy	Op	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
Add1	No						
Add2	No						
Add3	No						
Multi1	No						
Multi2	No						

Register result status:

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



Tomasulo Example Cycle 1

Instruction status:

Instruction	j	k	Issue	Comp	Result	Busy	Address
LD	F6	34+	R2			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						
Multi1	No						
Multi2	No						

Register result status:

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



Tomasulo Example Cycle 2

Instruction status:

Instruction	j	k	Issue	Comp	Result	Busy	Address
LD	F6	34+	R2			Load1	Yes 34+R2
LD	F2	45+	R3			Load2	Yes 45+R3
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						
Multi1	No						
Multi2	No						

Register result status:

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

Note: Unlike 6600, can have multiple loads outstanding (Assume latency for loads is 2 clock cycles)



Tomasulo Example Cycle 3

Instruction status:

Instruction	j	k	Exec Write			Busy	Address
			Issue	Comp	Result		
LD	F6	34+	R2	1	3	Load1	Yes 34+R2
LD	F2	45+	R3	2	4	Load2	Yes 45+R3
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
Add1	No						
Add2	No						
Add3	No						
Mult1	Yes	MULTD	M(A2)	R(F4)	Load2		
Mult2	No						

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
3		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	Load1	No
LD	F2	45+	R3	2	4	Load2	Yes 45+R3
MULTD	F0	F2	F4	3	4	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
Add1	Yes	SUBD	M(A1)	M(A1)	Load2		
Add2	No						
Add3	No						
Mult1	Yes	MULTD	M(A2)	R(F4)	Load2		
Mult2	No						

Register result status:

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

- Load2 completing; what is waiting for Load2?

Tomasulo Example Cycle 5

Instruction status:

Instruction	j	k	Exec Write			Busy	Address
			Issue	Comp	Result		
LD	F6	34+	R2	1	3	Load1	No
LD	F2	45+	R3	2	4	Load2	No
MULTD	F0	F2	F4	3	4	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
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		Mult1	M(A2)	M(A1)	Add1	Mult2			

- Issue ADDD here vs. scoreboard?

Tomasulo Example Cycle 6

Instruction status:

Instruction	j	k	Exec Write			Busy	Address
			Issue	Comp	Result		
LD	F6	34+	R2	1	3	Load1	No
LD	F2	45+	R3	2	4	Load2	No
MULTD	F0	F2	F4	3	4	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)			
Add2	Yes	ADDD	M(A2)	Add1			
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		Mult1	M(A2)	Add2	Add1	Mult2			

- Issue ADDD here vs. scoreboard?

Tomasulo Example Cycle 7

Instruction status:

Instruction	j	k	Exec Write			Busy	Address
			Issue	Comp	Result		
LD	F6	34+	R2	1	3	Load1	No
LD	F2	45+	R3	2	4	Load2	No
MULTD	F0	F2	F4	3	4	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)			
Add2	Yes	ADDD	M(A2)	Add1			
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		Mult1	M(A2)	Add2	Add1	Mult2			

- Add1 completing; what is waiting for it?

Tomasulo Example Cycle 8

Instruction status:

Instruction	j	k	Exec Write			Busy	Address
			Issue	Comp	Result		
LD	F6	34+	R2	1	3	Load1	No
LD	F2	45+	R3	2	4	Load2	No
MULTD	F0	F2	F4	3	4	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
Add1	No						
2 Add2	Yes	ADDD	(M-M)	M(A2)			
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		Mult1	M(A2)	Add2	(M-M)	Mult2			

-

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)		
	Add3	No					
6	Multi1	Yes	MULTD	M(A2)	R(F4)		
	Multi2	Yes	DIVD		M(A1)	Multi1	

Register result status:

Clock	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
9		Multi1	M(A2)		Add2	(M-M)	Multi2			

Tomasulo Example Cycle 10

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			

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
0	Add2	Yes	ADDD	(M-M)	M(A2)		
	Add3	No					
5	Multi1	Yes	MULTD	M(A2)	R(F4)		
	Multi2	Yes	DIVD		M(A1)	Multi1	

Register result status:

Clock	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
10		Multi1	M(A2)		Add2	(M-M)	Multi2			

Add2 completing; what is waiting for it?

Tomasulo Example Cycle 11

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					
4	Multi1	Yes	MULTD	M(A2)	R(F4)		
	Multi2	Yes	DIVD		M(A1)	Multi1	

Register result status:

Clock	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
11		Multi1	M(A2)		(M-M+N)	(M-M)	Multi2			

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

Tomasulo Example Cycle 12

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					
3	Multi1	Yes	MULTD	M(A2)	R(F4)		
	Multi2	Yes	DIVD		M(A1)	Multi1	

Register result status:

Clock	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
12		Multi1	M(A2)		(M-M+N)	(M-M)	Multi2			

Tomasulo Example Cycle 13

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					
2	Multi1	Yes	MULTD	M(A2)	R(F4)		
	Multi2	Yes	DIVD		M(A1)	Multi1	

Register result status:

Clock	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
13		Multi1	M(A2)		(M-M+N)	(M-M)	Multi2			

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	Multi1	Yes	MULTD	M(A2)	R(F4)		
	Multi2	Yes	DIVD		M(A1)	Multi1	

Register result status:

Clock	FU	F0	F2	F4	F6	F8	F10	F12	...	F30
14		Multi1	M(A2)		(M-M+N)	(M-M)	Multi2			

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	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					
0	Multi	Yes	MULTD	M(A2)	R(F4)		
	Multi2	Yes	DIVD	M(A1)	Multi1		

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
15		Multi1	M(A2)		(M-M+N)(M-M)		Multi2		

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					
	Multi1	No					
40	Multi2	Yes	DIVD	M*F4	M(A1)		

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
16		M*F4	M(A2)		(M-M+N)(M-M)		Multi2		

Faster than light computation
(skip a couple of cycles)

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					
	Multi1	No					
1	Multi2	Yes	DIVD	M*F4	M(A1)		

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
55		M*F4	M(A2)		(M-M+N)(M-M)		Multi2		

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	56			
ADDD	F6	F8	F2	6	10	11		

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Multi1	No					
0	Multi2	Yes	DIVD	M*F4	M(A1)		

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
56		M*F4	M(A2)		(M-M+N)(M-M)		Multi2		

• Multi2 is completing; what is waiting for it?

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					
	Multi1	No					
0	Multi2	Yes	DIVD	M*F4	M(A1)		

Register result status:

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
56		M*F4	M(A2)		(M-M+N)(M-M)		Multi2		

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

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 (multiple load)
- WAW, WAR control hazards vs. renaming registers
- Lack of forwarding (must write and read register)
- 4 steps vs. 3 steps of control

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

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

window size: ~ 14 instructions ~ 5 instructions

No issue on structural hazard same

WAR: renaming avoids stall completion

WAW: renaming avoids stall issue

Broadcast results from FU Write/read registers

Control: reservation stations central scoreboard



Tomasulo Analysis

- Complexity
 - delays of 360/91, MIPS 10000, IBM 620, Alpha 21264, ...
- Many associative stores (CDB) at high speed
- Performance limited by Common Data Bus
 - Multiple CDBs => more FU logic for parallel associative stores



PRS: State Example 2

What is Instruction Status at end of Clock 5?

Instruction status:

Instruction	j	k	1)	2)	3)	4)	5)	6)
LD	F2	0	R1	Exec. Comp	Exec. Comp	Write Result	Write Result	Write Result
MULTD	F4	F2	F0	Issued	Read Oper	Read Oper	Read Oper	Read Oper
LD	F6	0	R2	Not Issued	Issued	Issued	Exec. Comp	Exec. Comp
ADDD	F6	F4	F6	Not Issued	Not Issued	Not Issued	Issued	Issued
SD		F6	R2	Not Issued	Not Issued	Not Issued	Not Issued	Issued



PRS: Tomasulo Example 2

Instruction status:

Instruction	j	k	Issue	Exec Comp	Write Result	Busy	Address	RS?
LD	F2	0	R1	1		No		
MULTD	F4	F2	F0			No		
LD	F6	0	R2			No		
ADDD	F6	F4	F6			No		
SD		F6	R2			No		

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	RS	RS
	Add1	No				Oj	Ok
	Add2	No					
	Add3	No					
	Multi1	No					
	Multi2	No					

Register result status:

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



PRS: State Example 2

What is Instruction Status at end of Clock 10?

Instruction status:

Instruction	j	k	1)	2)	3)	4)	5)
LD	F2	0	R1	Write Result	Write Result	Write Result	Write Result
MULTD	F4	F2	F0	Read Oper	Read Oper	Read Oper	Read Oper
LD	F6	0	R2	Exec. Comp	Write Result	Write Result	Write Result
ADDD	F6	F4	F6	Not Issued	Issued	Issued	Read Oper
SD		F6	R2	Not Issued	Not Issued	Issued	Read Oper



PRS: State Example 2

What is Instruction Status at end of Clock 17?

Instruction status:

Instruction	<i>j</i>	<i>k</i>	1)	2)	3)	4)	5)
LD	F2	0 R1	Wrote Result	Wrote Result	Wrote Result	Wrote Result	Wrote Result
MULTD	F4	F0 F0	Exec. Complete	Wrote Result	Wrote Result	Wrote Result	Wrote Result
LD	F6	0 R2	Wrote Result	Wrote Result	Wrote Result	Wrote Result	Wrote Result
ADD	F6	F4 F6	Issued	Issued	Read Operands	Exec. Complete	Wrote Result
SD	F6	R2	Issued	Issued	Issued	Issued	Read Operands

Cal

Summary

- 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 4; PowerPC 604; MIPS R10000; HP-PA 8000; Alpha 21264

Cal