
COSC4201

Instruction Level Parallelism

Dynamic Scheduling

Prof. Mokhtar Aboelaze

Parts of these slides are taken from Notes by
Prof. David Patterson (UCB)

Fall 08

CSE4201

Outline

- **Data dependence and hazards**
- **Exposing parallelism (loop unrolling and scheduling)**
- **Reducing branch costs (prediction)**
- **Dynamic scheduling**
- **Speculation**
- **Multiple issue and static scheduling**
- **Advanced techniques**
- **Example**

Fall 08

CSE4201

Introduction

- In dynamic scheduling, the hardware rearranges the instruction execution to reduce stalls.
- Can handle cases where dependence is not known during compile time (memory reference).
- Simplifies the compiler
- Can tolerate unpredictable cases (cache miss).
- Speculation is based on dynamic scheduling.
- Can allow code compiled and for another pipeline to run efficiently on any pipeline

Fall 08

CSE4201

Dynamic Scheduling – The idea

- DIVD F0, F2, F4
ADDD F10, F0, F8
SUBD F12, F8, F14
- SUB can not be issued because of the dependence of ADD on DIV, although there is no data dependence to prevent issuing it.
- In a classical pipeline, structural and data hazard are checked in the ID stage.
- Here we must separate between checking for structural hazards, and waiting for the absence of data hazard.
- In-order issue, but instructions starts executions soon as its data operands are available.

Fall 08

CSE4201

Dynamic Scheduling – The Idea

◦ Will distinguish when an instruction *begins execution* and when it *completes execution*; between 2 times, the instruction is *in execution*

◦ May create WAR and WAW hazards

DIV.D F0, F2, F4

ADD.D F6, F0, F8

SUB.D F8, F10, F16

MUL.D F6, F10, F8

◦ Exceptions?

Tomasulo's Algorithm – The Idea

◦ Developed by Robert Tomasulo for the IBM 360/91.

◦ Main objective is to get good performance without the need to a very specialized compilers.

◦ It tracks when operands for instructions are available to minimize RAW hazard.

◦ It uses register renaming to avoid WAR and WAW hazards.

◦ It is employed in many modern processors although with some variations.

Tomasulo's Algorithm – The Idea

	◦ DIV.D F0,F2,F4		
Anti	◦ ADD.D F6,F0,F8		
	◦ S.D F6,0(R1)		
	◦ SUB.D F8,F10,F14		
	◦ MUL.D F6,F10,F8	◦ DIV.D F0,F2,F4	
		◦ ADD.D S,F0,F8	

This could be done by the compiler, what if intervening branches.

	◦ S.D S,0(R1)		
	◦ SUB.D T,F10,F14		
	◦ MUL.D F6,F10,T		

In Tomasulo's this is done by using reservation stations

Fall 08

CSE4201

Tomasulo's Algorithm

- Control & buffers distributed with Function Units (FU)
 - FU buffers called "reservation stations"; have pending operands
- Registers in instructions replaced by values or pointers to reservation stations (RS); called register renaming ;
 - Renaming avoids WAR, WAW hazards
 - More reservation stations than registers at least for 360/91, 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
 - Avoids RAW hazards by executing an instruction only when its operands are available

Fall 08

Load and Stores treated as FUs with RSs as well

CSE4201

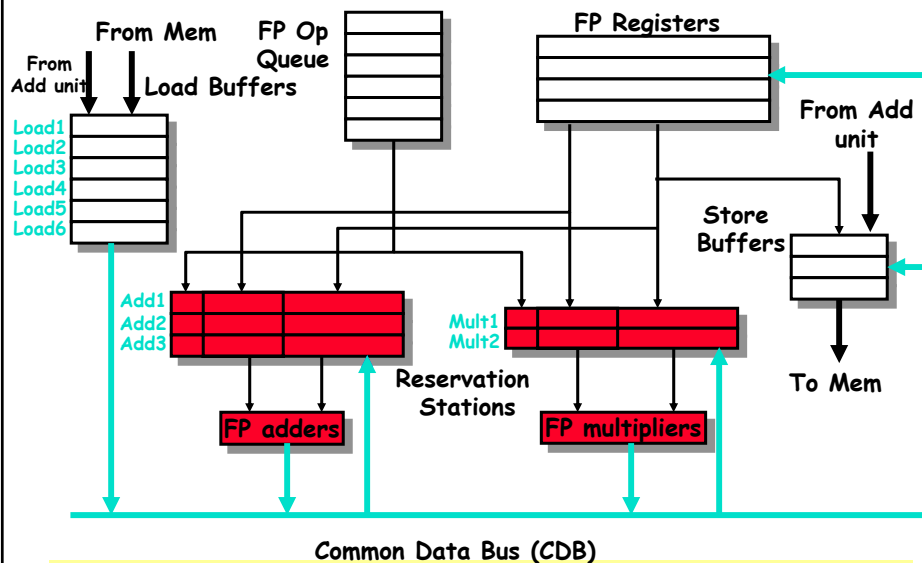
Tomasulo's Algorithm

- RS fetches operands as soon as they are available (no need to read them from a register).
- Pending instructions designate the RS's that will provide their inputs.
- When successive writes to the same register, only the last one is actually written to the register.
- Common data bus CDB is used to bypass registers in operand fetching.

Fall 08

CSE4201

Tomasulo's algorithm in MIPS



Fall 08

CSE4201

Tomasulo's Algorithm

Stages

1. **Issue**— AKA **dispatch** get instruction from FP Instruction Queue (FIFO)
 - If no empty reservation station, stall.
 - If reservation station free (no structural hazard), send the instruction (issue) to the station.
 - If operands ready (in register), send them else keep track of the reservation station that will produce them (renaming).

Fall 08

CSE4201

Tomasulo's Algorithm

2. **Execute**— AKA **execute** operate on operands (EX)

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

What if 2 instructions became ready in the same cycle?

Load/store require 2-step execution, calculating effective address and accessing memory.

Effective address is calculated and store in the load/store queue, maintained in a program order through the effective address calculation.

To preserve exception behavior, no instruction is allowed to execute until all previous branches are resolved. If a processor records the exception not raise, instruction can execute but no write-back

Fall 08

CSE4201

Tomasulo's Algorithm

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)

Fall 08

CSE4201

Reservation Station

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

V_j, V_k: Value of Source operands

- Store buffers has V field, result to be stored

Q_j, Q_k: Reservation stations producing source registers (value to be written)

- Note: Q_j, Q_k=0 => ready
- Store buffers only have Q_j for RS producing result

A: Used to hold info for the load store (initially immediate, then effective address)

Busy: Indicates reservation station or FU is busy

Register result status— Q_i indicates which functional unit will write each register, 0 means no write to this register

Fall 08

CSE4201

Tomasulo Example

Instruction stream

Instruction status:

Instruction	j	k	Exec		Write
			Issue	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		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

3 Load/Buffers

Reservation Stations:

FU count down

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

3 FP Adder R.S.
2 FP Mult R.S.

Register result status:

Clock
0
Clock cycle counter

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

Tomasulo Example Cycle 1

Instruction status:

Instruction	j	k	Exec		Write
			Issue	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		

	Busy	Address
Load1	Yes	34+R2
Load2	No	
Load3	No	

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
1

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

Tomasulo Example Cycle 2

Instruction status:

Instruction	j	k	Exec Write		Busy	Address
			Issue	Comp Result		
LD	F6	34+	R2	1	Yes	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	S1 Vj	S2 Vk	RS Qj	RS 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: Can have multiple loads outstanding

Fall 08

CSE4201

Tomasulo Example Cycle 3

Instruction status:

Instruction	j	k	Exec Write		Busy	Address
			Issue	Comp Result		
LD	F6	34+	R2	1	Yes	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	S1 Vj	S2 Vk	RS Qj	RS 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

- Load1 completing; what is waiting for Load1?

Fall 08

CSE4201

Tomasulo Example Cycle 4

Instruction status:

Instruction	j	k	Issue	Exec		Write	Busy	Address
				Comp	Result			
LD	F6	34+	R2	1	3	4	Load1	No
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					
ADDD	F6	F8	F2					

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
2	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 Load2?

Fall 08

CSE4201

Tomasulo Example Cycle 5

Instruction status:

Instruction	j	k	Issue	Exec		Write	Busy	Address
				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					

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS 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		

- Timer starts down for Add1, Mult1

Fall 08

CSE4201

Tomasulo Example Cycle 6

Instruction status:

Instruction	j	k	Issue	Exec Write		Busy	Address	
				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	S1	S2	RS	RS
				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	FU	Mult1	M(A2)		Add2	Add1	Mult2		

- Issue ADDD here despite name dependency on F6?

Fall 08

CSE4201

Tomasulo Example Cycle 7

Instruction status:

Instruction	j	k	Issue	Exec Write		Busy	Address	
				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	S1	S2	RS	RS
				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	

← waiting

Register result status:

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

- Add1 (SUBD) completing; what is waiting for it?

Fall 08

CSE4201

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	S1	S2	RS	RS
				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	FU	Mult1	M(A2)		Add2	(M-M)	Mult2		

Fall 08

CSE4201

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	S1	S2	RS	RS
				Vj	Vk	Qj	Qk
	Add1	No					
1	Add2	Yes	ADDD	(M-M)	M(A2)		
	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		

Fall 08

CSE4201

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	S1	S2	RS	RS
				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 (ADDD) completing; what is waiting for it?

Fall 08

CSE4201

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	S1	S2	RS	RS
				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+M)	(M-M)	Mult2			

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

Fall 08

CSE4201

Tomasulo Example Cycle 12

Instruction status:

Instruction	j	k	Issue	Exec Write		Busy	Address
				Comp	Result		
LD	F6	34+	R2	1	3	4	Load1
LD	F2	45+	R3	2	4	5	Load2
MULTD	F0	F2	F4	3			Load3
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	S1	S2	RS	RS
				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	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2			

Fall 08

CSE4201

Tomasulo Example Cycle 13

Instruction status:

Instruction	j	k	Issue	Exec Write		Busy	Address
				Comp	Result		
LD	F6	34+	R2	1	3	4	Load1
LD	F2	45+	R3	2	4	5	Load2
MULTD	F0	F2	F4	3			Load3
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	S1	S2	RS	RS
				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	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2			

Fall 08

CSE4201

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	S1	S2	RS	RS
				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	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2			

Fall 08

CSE4201

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	S1	S2	RS	RS
				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	Mult1	M(A2)		(M-M+N)	(M-M)	Mult2			

- Mult1 (MULTD) completing; what is waiting for it?

Fall 08

CSE4201

Tomasulo Example Cycle 16

Instruction status:

Instruction	j	k	Exec Write			Load1	Load2	Load3	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	15	16		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	S1		S2		RS		RS	
				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		

- Just waiting for Mult2 (DIVD) to complete

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	S1	S2	RS	RS
				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		

Fall 08

CSE4201

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	S1	S2	RS	RS
				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 (DIVD) is completing; what is waiting for it?

Fall 08

CSE4201

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	S1		S2		RS	RS
				Vj	Vk	Qj	Qk		
Add1		No							
Add2		No							
Add3		No							
Mult1		No							
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)	Result		

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

Tomasulo overlap iterations of loops

- Register renaming
 - Multiple iterations use different physical destinations for registers (dynamic loop unrolling).
- Reservation stations
 - Permit instruction issue to advance past integer control flow operations
 - Also buffer old values of registers - totally avoiding the WAR stall
- Other perspective: Tomasulo building data flow dependency graph on the fly

Tomasulo's 2 major advantages

1. Distribution of the hazard detection logic
 - **distributed reservation stations and the CDB**
 - **If multiple instructions waiting on single result, & each instruction has other operand, then instructions can be released simultaneously by broadcast on CDB**
 - **If a centralized register file were used, the units would have to read their results from the registers when register buses are available**
2. Elimination of stalls for WAW and WAR hazards
3. Load/store can be done out of program order if they do not address the same memory location (effective address).