Computer Architecture

计 算 机 体 系 结 构

Lecture 3. Instruction-Level Parallelism I

第三讲、指令级并行 I

Chao Li, PhD.

李超 博士

SJTU-SE346, Spring 2019

Review

- ISA, micro-architecture, physical design
- Evolution of ISA
- CISC vs RISC, IA32 and x86
- MIPS instruction fields
- Machine interface, user ISA and system ISA
- Good interface design
- Hardware elements
- Simple MIPS pipeline
- Pipeline speedup and pipeline design challenge

Outlines

- Pipeline Hazards
- Dynamic Scheduling: Scoreboarding
- Dynamic Scheduling: Tomasulo's Alg.

Problems with Simple Pipeline

View 1:

	time	**t0**	**t1**	**t2**	**t3**	**t4**	**t5**	…
I_1: r1 ← (r0) + 10		IF1	ID1	EX1	MA1	WB1		
I_2: r2 ← (r1) + 20			IF2	ID2	ID2	ID2	ID2	EX2
I_3: r3 ← r4+ r5				IF3	IF3	IF3	IF3	ID3

Stalled Stages

View 2:

Hardware Resource	*time*	**t0**	**t1**	**t2**	**t3**	**t4**	**t5**	**t6**	…
IF		I1	I2	I3	I3	I3	I3		
ID			I1	I2	I2	I2	I2	I3	
EX				I1	bubble	bubble	bubble	I2	
MA					I1	bubble	bubble	bubble	
WB						I1	bubble	bubble	

Pipeline **stalls** (add bubble) to avoid hazards

Dependency and Hazards

- Dependence:
  - Reflects original program order (which affects execution results)
  - Indicates the possibility for a hazard
  - Determines the degree of parallelism

- Dependences are a property of **programs**
  - May not cause hazard with well-designed pipeline

- Hazards are properties of the **hardware organization**
  - Cased by reordered instruction, overlapped execution, etc.

- Three types of dependences:
  - **data**, **name**, and **control**

Data Dependence

- (True) Data dependences
  - Instruction i produces a result that may be used by instruction j, or
  - Instruction j is data dependent on instruction k, and instruction k is data dependent on instruction i.
- There is data exchange for true data dependence

c = a + b

e = c + d

- Easy to determine for registers (at the decode stage)
- Hard for memory location (require effective address)

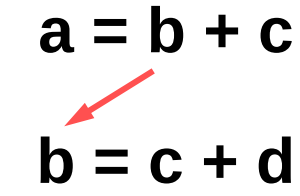
10(R1) == 20(R2) ? 10(R1) != 10(R1) ?

Q: pipeline hazards due to memory data dependences?

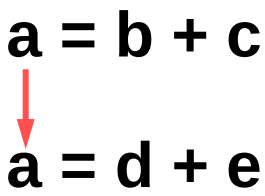
Name Dependence

- **Name dependence**: two instructions use the same register or memory location (called a name) but don't actually exchange data. **(but reuse storage locations)**

- Anti-dependence
  - Instruction j writes a register or memory location that instruction I reads from (instruction I is executed first)
- Output dependence
  - Instruction I and instruction j write the same register or memory location (must preserve program order)



Anti-dependence

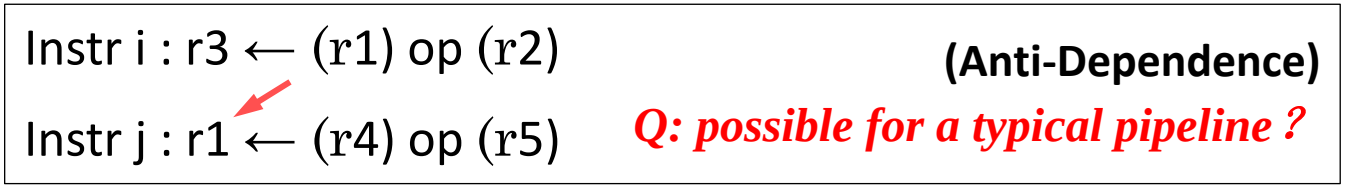


Output dependence

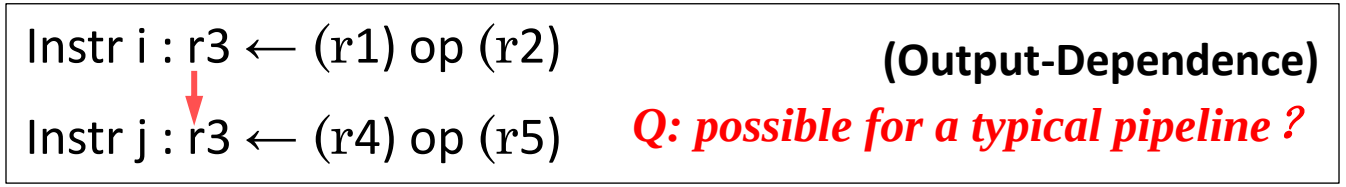
Summary of Possible Data Hazards

- RAW (read after write) Hazard: *Instr j gets the old value*

- WAR (write after read) Hazard: *Inst i gets the new value*



- WAW (write after write) Hazard: *produce wrong results*



Control Dependence

- Instructions are often controlled by some set of branches

```
if   p1 {
         S1;
}
S;
If   p2 {
         S2;
}
```

s1 is control dependent on p1

s2 is control dependent on p2

s is neither control dependent on p1 nor p2

- Can be viewed as a form of RAW hazard

Q: how do you understand this ?

Instruction Dependency and Pipeline Hazard

- Instruction *i* may need a resource being used by a later instruction *j*
  - May cause structural hazard (Q: how to avoid this?)

- Instruction *i* may produce a result that is needed by a later instruction *j*
  - May cause data hazard (a.k.a pipeline data hazard)

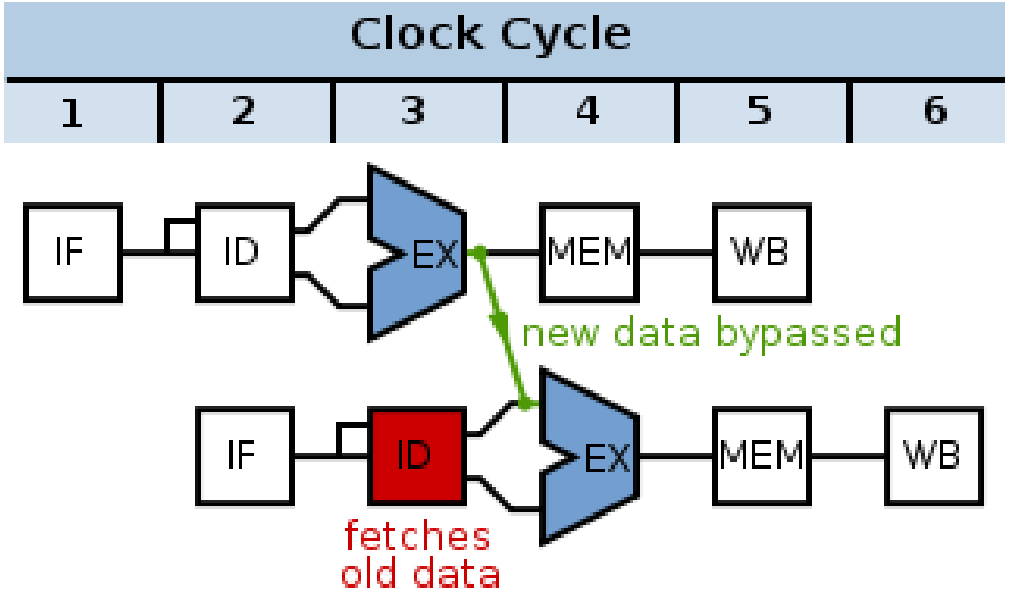
- Instruction *i* may determine the next instruction to be executed
  - May cause control hazard

Overcoming Data Hazards

- Hidden data hazards with bypassing or forwarding:
  - If data is available somewhere in the data path provide a bypass (forwarding path) to get it to the right stage

$$r3 \leftarrow (r1)\ \text{op}\ (r2)$$

$$r5 \leftarrow (r3)\ \text{op}\ (r4)$$

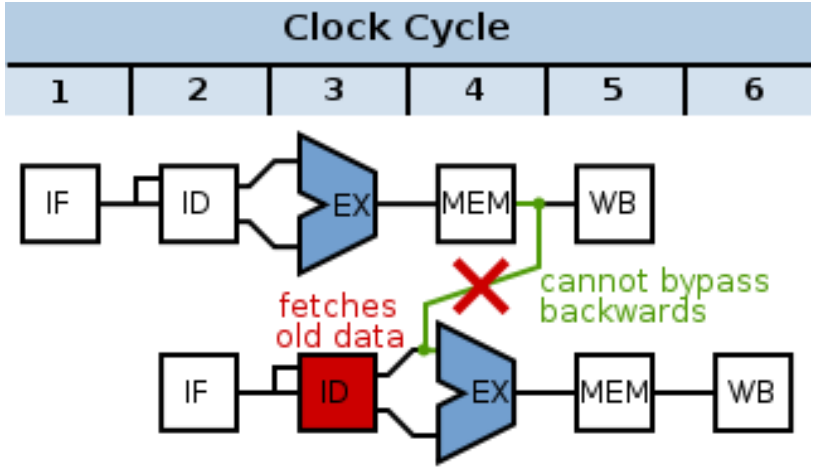


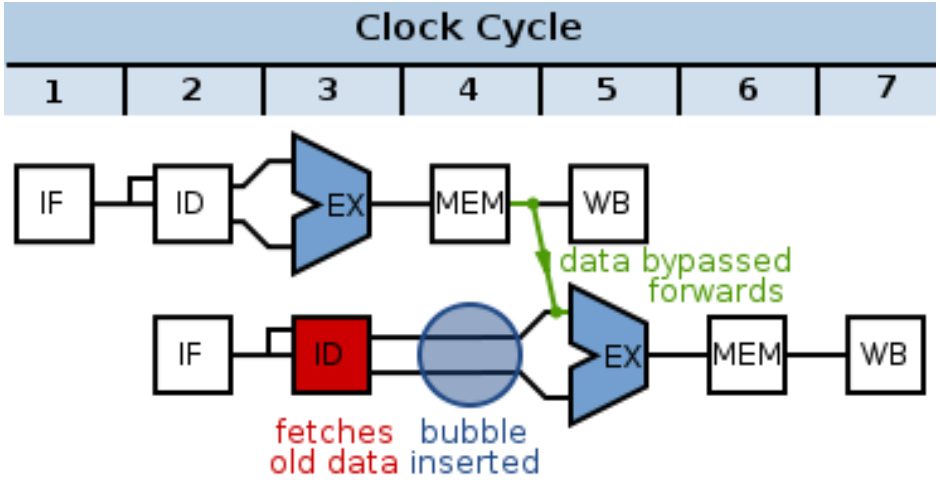
Overcoming Data Hazards

- Freeze earlier stages until the data becomes available:
  - The hardware mechanism to detect a data hazard and stall the pipeline is referred to as pipeline interlock

$$r2 \leftarrow M[(r1) + 10]$$

$$r4 \leftarrow (r2)\ op\ (r3)$$



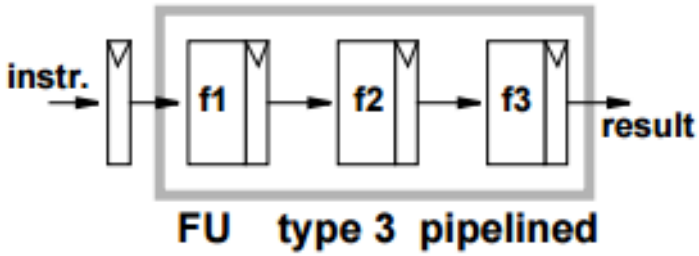


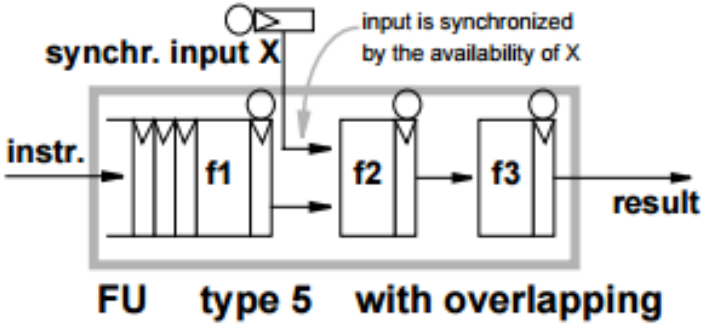
Outlines

- Pipeline Hazards
- Dynamic Scheduling: Scoreboarding
- Dynamic Scheduling: Tomasulo's Alg.

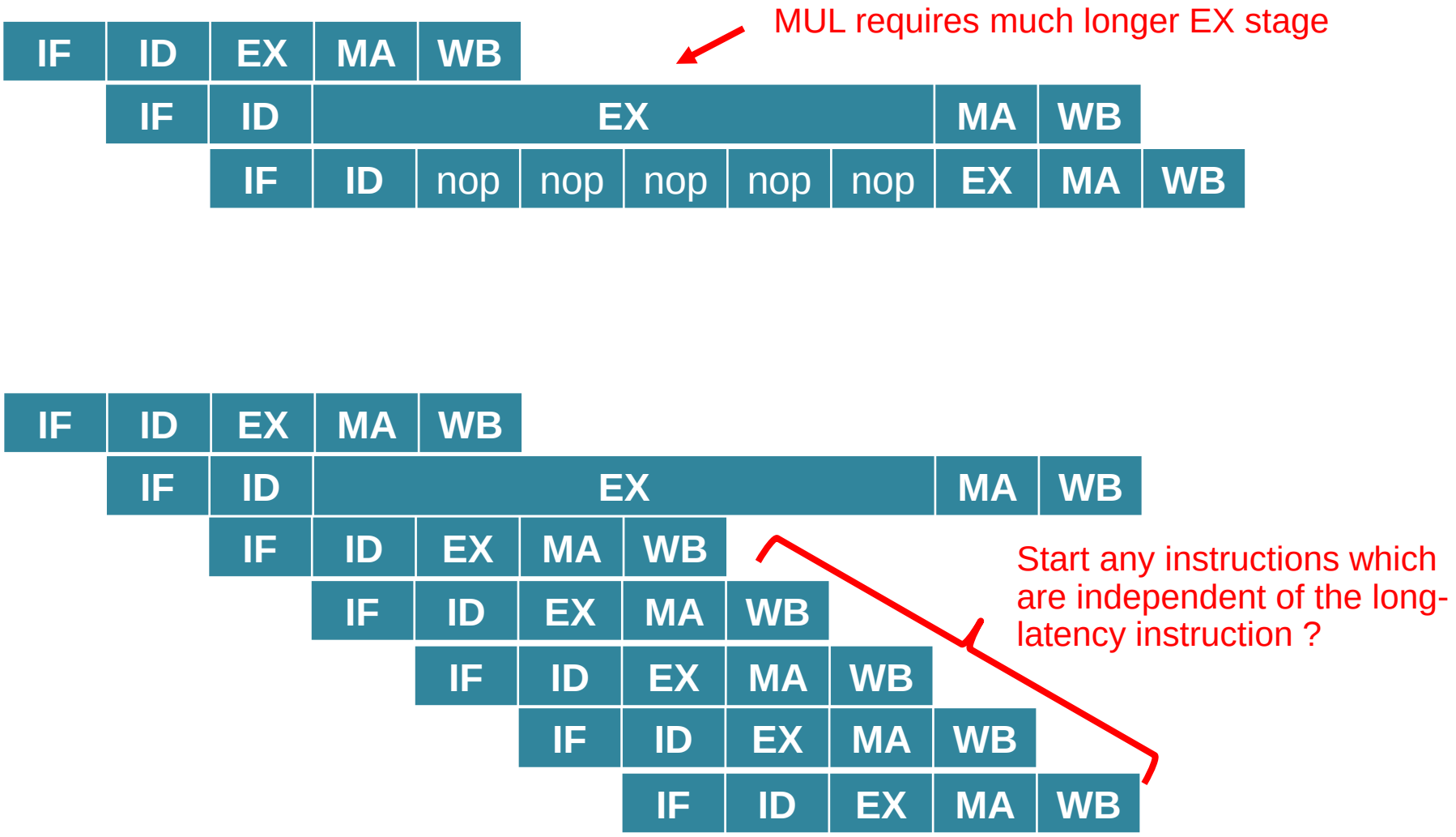
FU and Structure Hazards

- A functional unit (FU) is a basic processing element that computes some results based on its inputs.
  - Adders, multipliers, ALUs, register files, load/store units, etc.
- Different types of FUs:
  - 1. FU with a single clock tick of execution time
  - 2. FU with n clock ticks of execution time, non-pipelined
  - 3. FU with n clock ticks of execution time, pipelined
  - 4. FU with variable execution time, non-overlapped
  - 5. FU with variable execution time, overlapped





Long-Latency Operations



Dynamic Scheduling and Out-of-order Execution

- Idea: Dynamic HW control of hazard and issue
- Implementation: two classical approaches
  - Control-centric: Scoreboarding
  - Data-centric: Tomasulo's algorithm
- Variants of these schemes are also seen today

In-order execution:
(statically scheduled)

1. lw $3, 100($4)	in execution, cache miss
2. add $2, $3, $4	waits until the miss is satisfied
3. sub $5, $6, $7	waits for the add

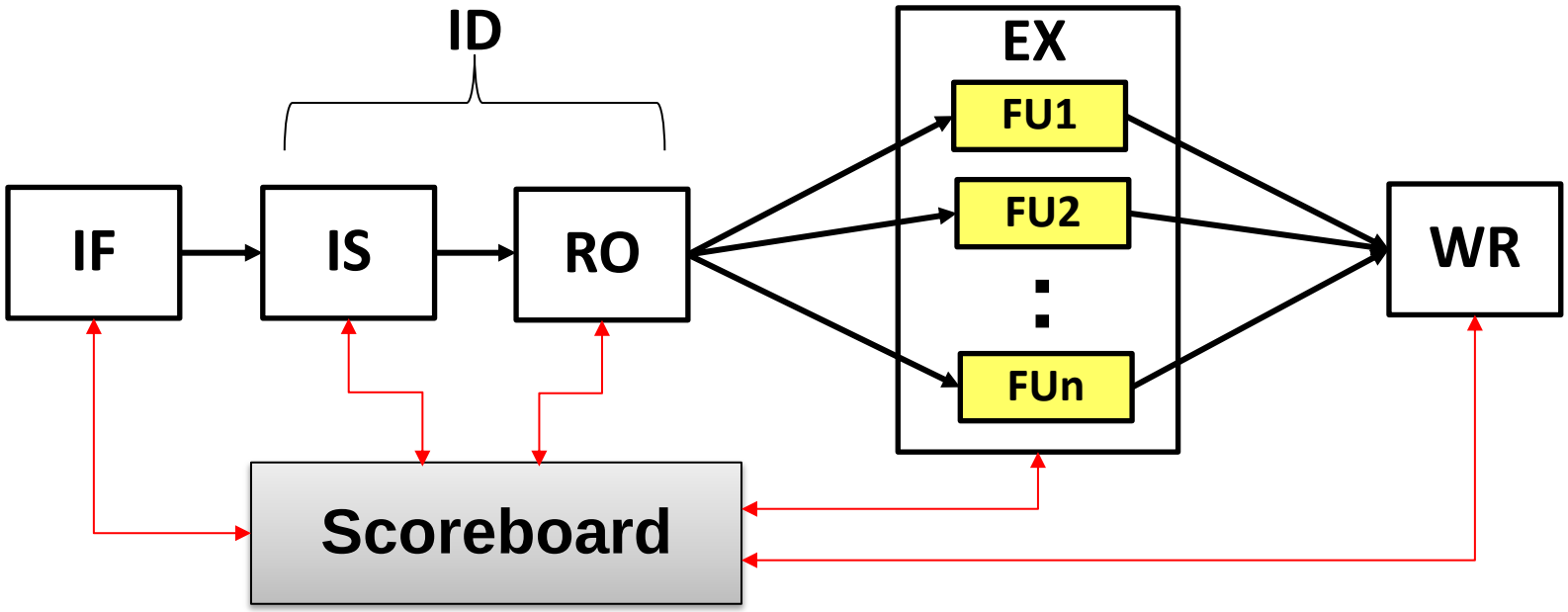
Out-of-order execution:
(dynamically scheduled)

1. lw $3, 100($4)	in execution, cache miss
3. sub $5, $6, $7	can execute during the cache miss
2. add $2, $3, $4	waits until the miss is satisfied

CDC 6600

- Mainframe supercomputer in 1964
- 10 parallel functional units (FUs) that are not pipelined
  - 4 floating-point units: 2 Multipliers, 1 adder, 1 divider

Scoreboarding



- **Scoreboard**: A central control to prevent hazard
  - **Issue**: check for structural/WAW hazard; stall issue until clear
  - **Read operands**: read operands if no RAW hazards
  - **Execution**: followed by notification to scoreboard
  - **Write result**: checks for WAR; stall write until clear

Scoreboard Components

- Instruction status – which of the 4 steps the instruction is in
- Functional unit status – 9 fields
  - *Op* : Operation to perform in the unit
  - *Fi* : Destination register number
  - *Fj*, *Fk* : Source register number
  - *Qj*, *Qk* : Functional units producing Fj, Fk.
  - *Rj*, *Rk* : Flags indicating when Fj, Fk are ready
  - *Busy* : Indicates whether the unit is busy or not
- Register results status
  - Indicates which functional unit will write each register, if one exists. Blank when no pending instructions that will write that register.

Scoreboard Example

Instruction Status

		j	*k*
1. LD	F6	34	R2
2. LD	F2	45	R3
3. MUL.D	F0	F2	F4
4. SUB.D	F8	F6	F2
5. DIV. D	F10	F0	F6
6. ADD.D	F6	F8	F2

IS	RO	EX	WR

Execution complete

Register Results Status

F0	
F2	
F4	
F6	
F8	
F10	

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
		Dest.	S1	S2	FU for j	FU for k	Fj ok?	Fk ok?		
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Assumptions: Load (2 cycles), Add (2 cycles), Mult(10 cycles), Divi (40 cycles)

Scoreboard Example: Cycle 1

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1			
2. LD	F2	45	R3				
3. MUL.D	F0	F2	F4				
4. SUB.D	F8	F6	F2				
5. DIV. D	F10	F0	F6				
6. ADD.D	F6	F8	F2				

Register Results Status

F0	
F2	
F4	
F6	Integer
F8	
F10	

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi* (Dest.)	*Fj* (S1)	*Fk* (S2)	*Qj* (FU for j)	*Qk* (FU for k)	*Rj* (Fj ok?)	*Rk* (Fk ok?)
	Integer	Yes	LD	F6		R2				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Q: can we issue MUL.D?

Scoreboard Example: Cycle 2

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2		
2. LD	F2	45	R3				
3. MUL.D	F0	F2	F4				
4. SUB.D	F8	F6	F2				
5. DIV. D	F10	F0	F6				
6. ADD.D	F6	F8	F2				

Register Results Status

F0	
F2	
F4	
F6	Integer
F8	
F10	

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
		Dest.	S1	S2	FU for j	FU for k	Fj ok?	Fk ok?		
	Integer	Yes	LD	F6		R2				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

MUL.D can't issue due to in-order issue!

Scoreboard Example: Cycle 3

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	
2. LD	F2	45	R3				
3. MUL.D	F0	F2	F4				
4. SUB.D	F8	F6	F2				
5. DIV. D	F10	F0	F6				
6. ADD.D	F6	F8	F2				

Register Results Status

F0	
F2	
F4	
F6	Integer
F8	
F10	

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi* (Dest.)	*Fj* (S1)	*Fk* (S2)	*Qj* (FU for j)	*Qk* (FU for k)	*Rj* (Fj ok?)	*Rk* (Fk ok?)
	Integer	Yes	LD	F6		R2				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Scoreboard Example: Cycle 4

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3				
3. MUL.D	F0	F2	F4				
4. SUB.D	F8	F6	F2				
5. DIV. D	F10	F0	F6				
6. ADD.D	F6	F8	F2				

Register Results Status

F0	
F2	
F4	
F6	~~**Integer**~~
F8	
F10	

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
				Dest.	S1	S2	FU for j	FU for k	Fj ok?	Fk ok?
	Integer	Yes	LD	F6		R2				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Scoreboard Example: Cycle 5

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5			
3. MUL.D	F0	F2	F4				
4. SUB.D	F8	F6	F2				
5. DIV. D	F10	F0	F6				
6. ADD.D	F6	F8	F2				

Register Results Status

F0	
F2	Integer
F4	
F6	
F8	
F10	

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
		Dest.	S1	S2	FU for j	FU for k	Fj ok?	Fk ok?		
	Integer	Yes	LD	F2		R3				Yes
	Mult1	No								
	Mult2	No								
	Add	No								
	Divide	No								

Scoreboard Example: Cycle 6

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6		
3. MUL.D	F0	F2	F4	C6			
4. SUB.D	F8	F6	F2				
5. DIV. D	F10	F0	F6				
6. ADD.D	F6	F8	F2				

Register Results Status

F0	**Mult1**
F2	**Integer**
F4	
F6	
F8	
F10	

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
				Dest.	S1	S2	FU for j	FU for k	Fj ok?	Fk ok?
	Integer	Yes	LD	F2		R3				Yes
	Mult1	Yes	ML	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	No								
	Divide	No								

Scoreboard Example: Cycle 7

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	
3. MUL.D	F0	F2	F4	C6			
4. SUB.D	F8	F6	F2	C7			
5. DIV. D	F10	F0	F6				
6. ADD.D	F6	F8	F2				

Register Results Status

F0	**Mult1**
F2	**Integer**
F4	
F6	
F8	**Add**
F10	

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
				Dest.	S1	S2	FU for j	FU for k	Fj ok?	Fk ok?
	Integer	Yes	LD	F2		R3				Yes
	Mult1	Yes	ML	F0	F2	F4	Integer		No	Yes
	Mult2	No								
	Add	Yes	SU	F8	F6	F2		Integer	Yes	No
	Divide	No								

Scoreboard Example: Cycle 8 - 1

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6			
4. SUB.D	F8	F6	F2	C7			
5. DIV. D	F10	F0	F6				
6. ADD.D	F6	F8	F2				

Register Results Status

F0	**Mult1**
F2	~~**Integer**~~
F4	
F6	
F8	**Add**
F10	

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
		Dest.	S1	S2	FU for j	FU for k	Fj ok?	Fk ok?		
	Integer	No								
	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	SU	F8	F6	F2			Yes	Yes
	Divide	No								

Scoreboard Example: Cycle 8 - 2

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6			
4. SUB.D	F8	F6	F2	C7			
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2				

Register Results Status

F0	**Mult1**
F2	~~**Integer**~~
F4	
F6	
F8	**Add**
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
		Dest.	S1	S2	FU for j	FU for k	Fj ok?	Fk ok?		
	Integer	No								
	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	SU	F8	F6	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Scoreboard Example: Cycle 9

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9		
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2				

Register Results Status

F0	**Mult1**
F2	
F4	
F6	
F8	**Add**
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
10	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
2	Add	Yes	SU	F8	F6	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 10

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9		
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2				

Register Results Status

F0	**Mult1**
F2	
F4	
F6	
F8	**Add**
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
9	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
1	Add	Yes	SU	F8	F6	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 11

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9	C11	
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2				

Register Results Status

F0	**Mult1**
F2	
F4	
F6	
F8	**Add**
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
8	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
0	Add	Yes	SU	F8	F6	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 12

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2				

Register Results Status

F0	**Mult1**
F2	
F4	
F6	
F8	**~~Add~~**
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
7	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	No								
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 13

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2	C13			

Register Results Status

F0	**Mult1**
F2	
F4	
F6	**Add**
F8	
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi* (Dest.)	*Fj* (S1)	*Fk* (S2)	*Qj* (FU for j)	*Qk* (FU for k)	*Rj* (Fj ok?)	*Rk* (Fk ok?)
	Integer	No								
6	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	AD	F6	F8	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Scoreboard Example: Cycle 14

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2	C13	C14		

Register Results Status

F0	**Mult1**
F2	
F4	
F6	**Add**
F8	
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi* (Dest.)	*Fj* (S1)	*Fk* (S2)	*Qj* (FU for j)	*Qk* (FU for k)	*Rj* (Fj ok?)	*Rk* (Fk ok?)
	Integer	No								
5	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
2	Add	Yes	AD	F6	F8	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Scoreboard Example: Cycle 15

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2	C13	C14		

Register Results Status

F0	**Mult1**
F2	
F4	
F6	**Add**
F8	
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
4	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
1	Add	Yes	AD	F6	F8	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 16

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2	C13	C14	C16	

Register Results Status

F0	**Mult1**
F2	
F4	
F6	**Add**
F8	
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
3	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
0	Add	Yes	AD	F6	F8	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 17

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2	C13	C14	C16	

Register Results Status

F0	Mult1
F2	
F4	
F6	Add
F8	
F10	Divide

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi* (Dest.)	*Fj* (S1)	*Fk* (S2)	*Qj* (FU for j)	*Qk* (FU for k)	*Rj* (Fj ok?)	*Rk* (Fk ok?)
	Integer	No								
2	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	AD	F6	F8	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	~~Mult1~~		~~No~~	Yes

ADDD can't write because of DIV.D WAR!

Scoreboard Example: Cycle 18

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9		
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2	C13	C14	C16	

Register Results Status

F0	**Mult1**
F2	
F4	
F6	**Add**
F8	
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
1	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	AD	F6	F8	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 19

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9	C19	
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2	C13	C14	C16	

Register Results Status

F0	**Mult1**
F2	
F4	
F6	**Add**
F8	
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
0	Mult1	Yes	ML	F0	F2	F4			Yes	Yes
	Mult2	No								
	Add	Yes	AD	F6	F8	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6	Mult1		No	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 20

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9	C19	C20
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8			
6. ADD.D	F6	F8	F2	C13	C14	C16	

Register Results Status

F0	~~Mult1~~
F2	
F4	
F6	Add
F8	
F10	Divide

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi* (Dest.)	*Fj* (S1)	*Fk* (S2)	*Qj* (FU for j)	*Qk* (FU for k)	*Rj* (Fj ok?)	*Rk* (Fk ok?)
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	Yes	AD	F6	F8	F2			Yes	Yes
	Divide	Yes	DI	F10	F0	F6			Yes	Yes

Scoreboard Example: Cycle 21

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9	C19	C20
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8	C21		
6. ADD.D	F6	F8	F2	C13	C14	C16	

Register Results Status

F0	
F2	
F4	
F6	Add
F8	
F10	Divide

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi* (Dest.)	*Fj* (S1)	*Fk* (S2)	*Qj* (FU for j)	*Qk* (FU for k)	*Rj* (Fj ok?)	*Rk* (Fk ok?)
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	Yes	AD	F6	F8	F2			Yes	Yes
40	Divide	Yes	DI	F10	F0	F6			Yes	Yes

Scoreboard Example: Cycle 22

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9	C19	C20
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8	C21		
6. ADD.D	F6	F8	F2	C13	C14	C16	C22

Register Results Status

F0	
F2	
F4	
F6	~~Add~~
F8	
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
		Dest.	S1	S2	FU for j	FU for k	Fj ok?	Fk ok?		
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
39	Divide	Yes	DI	F10	F0	F6			Yes	Yes

Scoreboard Example: Cycle 23

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9	C19	C20
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8	C21		
6. ADD.D	F6	F8	F2	C13	C14	C16	C22

Register Results Status

F0	
F2	
F4	
F6	
F8	
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
39	Divide	Yes	DI	F10	F0	F6			Yes	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 61

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9	C19	C20
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8	C21	C61	
6. ADD.D	F6	F8	F2	C13	C14	C16	C22

Register Results Status

F0	
F2	
F4	
F6	
F8	
F10	**Divide**

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
0	Divide	Yes	DI	F10	F0	F6			Yes	Yes

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

Scoreboard Example: Cycle 62

Instruction Status

		j	*k*	IS	RO	EX	WR
1. LD	F6	34	R2	C1	C2	C3	C4
2. LD	F2	45	R3	C5	C6	C7	C8
3. MUL.D	F0	F2	F4	C6	C9	C19	C20
4. SUB.D	F8	F6	F2	C7	C9	C11	C12
5. DIV. D	F10	F0	F6	C8	C21	C61	C62
6. ADD.D	F6	F8	F2	C13	C14	C16	C22

Register Results Status

F0	
F2	
F4	
F6	
F8	
F10	~~Divide~~

Functional Unit Status

Time	*Name*	*Busy*	*Op*	*Fi*	*Fj*	*Fk*	*Qj*	*Qk*	*Rj*	*Rk*
	Integer	No								
	Mult1	No								
	Mult2	No								
	Add	No								
0	Divide	No								

Dest. S1 S2 FU for j FU for k Fj ok? Fk ok?

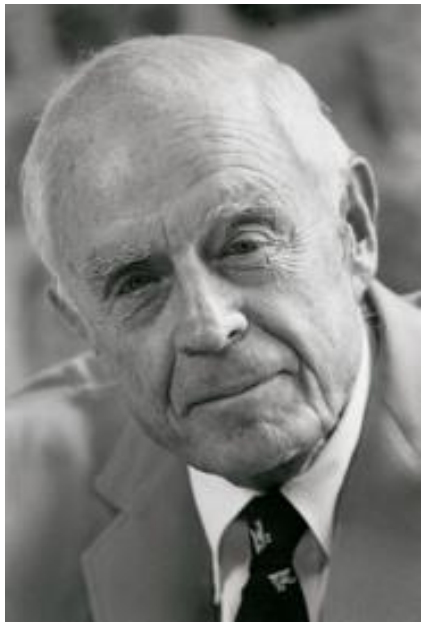
Outlines

- Pipeline Hazards
- Dynamic Scheduling: Scoreboarding
- Dynamic Scheduling: Tomasulo's Alg.

Early History

Then-CEO Thomas Watson Jr. wrote a memo to his employees:

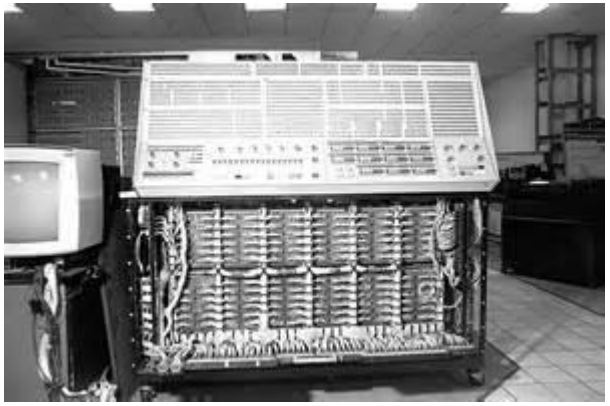
"Last week, Control Data ... announced the 6600 system. I understand that in the laboratory developing the system there are only 34 people including the janitor.



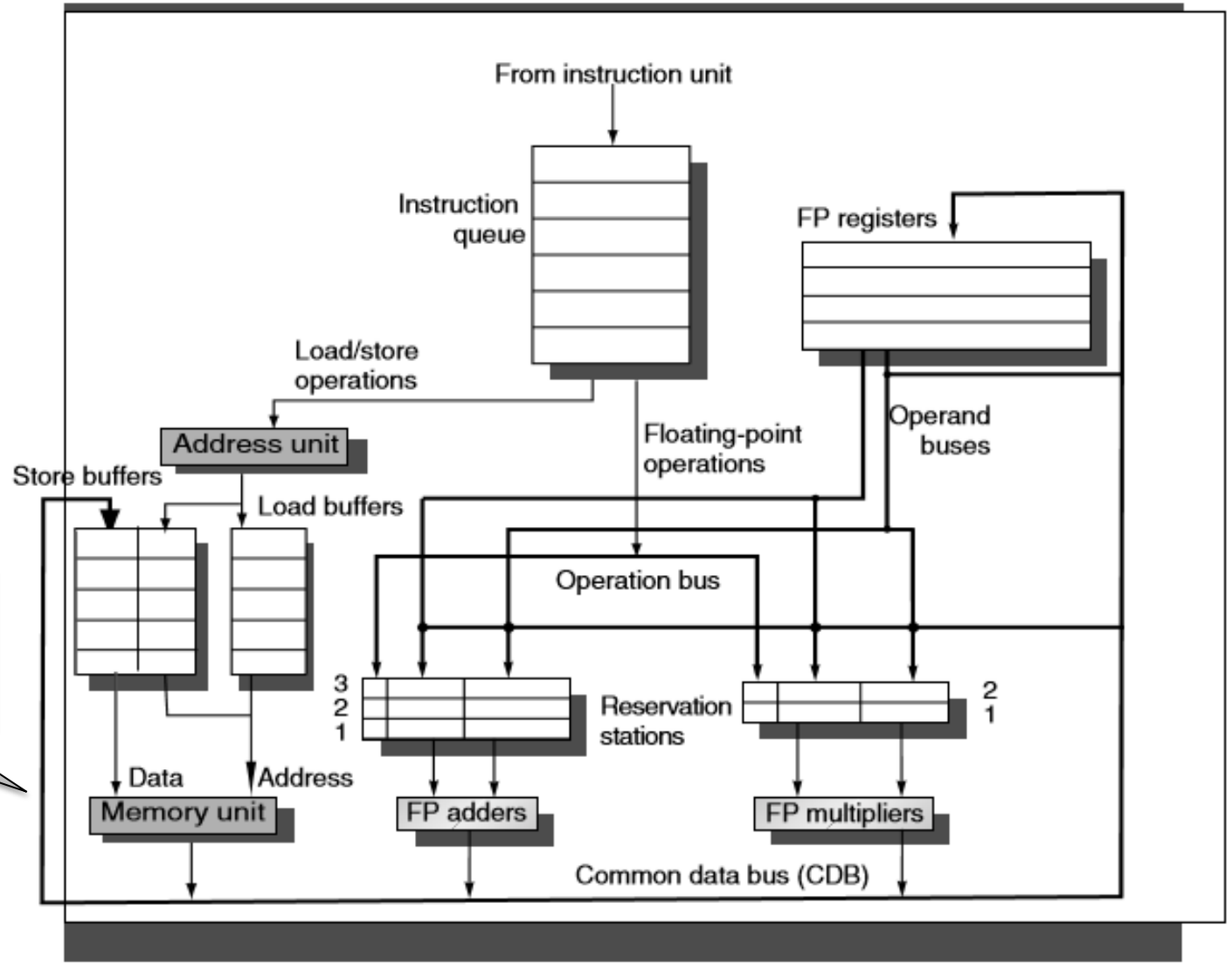
I fail to understand why we have lost our industry leadership position by letting someone else offer the world's most powerful computer.

IBM 360/91

- Announced in 1964 as a competitor to CDC 6600
- Dynamically scheduling FP unit (Tomasulo's algorithm)
  - It only had 2 FUs: 1 adder and 1 multiplier/divider
- Pipelined rather than multiple functional units
  - Adder support 3 instructions and Multiplier supports 2 instructions
  - In this class we discuss the alg. as if there were multiple FUs



Basic Structure Implementing Tomasulo's Alg.



CDB:
data + source tag
("come from" bus)

Reservation Stations

- Reservation stations (RS)
  - Fetches and buffers an operand as soon as it is available
  - When all operands are present, enable the associated FU
- Load/Store buffers
  - Load and Stores are treated as FUs with RSs as well
  - Behave almost exactly like reservation stations
- Both reservation stations and load/store buffers have tags
  - Essentially names for a set of virtual registers used in renaming
  - Shows which unit produces a result needed as a source operand

Register Renaming

- Registers in instructions are replaced by tags or pointers to reservation stations(RS) - called register renaming
- Register renaming eliminates name dependences
- Structural hazards:
  - A free reservation station of the right type must be available
  - Multiple reservation stations may compete for a shared FU
- More reservation stations than registers
  - Offer optimizations that compilers cannot

Tomasulo's Algorithm:

- Reservation Station Fields
  - *Op* : Operation to perform in the unit
  - *Vj, Vk* : Value of source operands
  - *Qj, Qk* : RS producing source registers
  - *Busy* : Indicates RS or FU is busy

- Instruction Status (Three stages)
  - Issue: get instruction from FP Op Queue (in-order)
  - Execution: operate on operand (may be out of order)
  - Write result: finish execution (may be out of order)

- Data flow approach: operations proceed as soon as their operands are available (instruction wake up)

Tomasulo Example

Instruction Status

		j	*k*
1. LD	F6	34	R2
2. LD	F2	45	R3
3. MUL.D	F0	F2	F4
4. SUB.D	F8	F6	F2
5. DIV. D	F10	F0	F6
6. ADD.D	F6	F8	F2

IS	EX	WR

Execution start

Register Results

F0	
F2	
F4	
F6	
F8	
F10	

Reservation Stations

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

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Assumptions: Load (2 cycles), Add (2 cycles), Mult(10 cycles), Divi (40 cycles)

Tomasulo Example: Cycle 1

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1		
2. LD	F2	45	R3			
3. MUL.D	F0	F2	F4			
4. SUB.D	F8	F6	F2			
5. DIV. D	F10	F0	F6			
6. ADD.D	F6	F8	F2			

Register Results Status

F0	
F2	
F4	
F6	Load1
F8	
F10	

Reservation Stations

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

	Busy	*Addr*
Load1	Yes	34+R2
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 2

Instruction Status

		j	k	IS	EX	WR
1. LD	F6	34	R2	C1	C2	
2. LD	F2	45	R3	C2		
3. MUL.D	F0	F2	F4			
4. SUB.D	F8	F6	F2			
5. DIV. D	F10	F0	F6			
6. ADD.D	F6	F8	F2			

Register Results Status

F0	
F2	Load2
F4	
F6	Load1
F8	
F10	

Reservation Stations

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

	Busy	Addr.
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

Load Buffer

Tomasulo Example: Cycle 3

Instruction Status

		j	k	IS	EX	WR
1. LD	F6	34	R2	C1	C2	
2. LD	F2	45	R3	C2	C3	
3. MUL.D	F0	F2	F4	C3		
4. SUB.D	F8	F6	F2			
5. DIV. D	F10	F0	F6			
6. ADD.D	F6	F8	F2			

Register Results Status

F0	Mult1
F2	Load2
F4	
F6	Load1
F8	
F10	

Reservation Stations

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	ML		R[F4]	Load2	
	Mult2	No					

	Busy	Addr.
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

Load Buffer

Tomasulo Example: Cycle 4

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	
3. MUL.D	F0	F2	F4	C3		
4. SUB.D	F8	F6	F2	C4		
5. DIV. D	F10	F0	F6			
6. ADD.D	F6	F8	F2			

Register Results Status

F0	Mult1
F2	Load2
F4	
F6	M[A1]
F8	Add1
F10	

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	Yes	SU	M[A1]			Load2
	Add2	No					
	Add3	No					
	Mult1	Yes	ML		R[F4]	Load2	
	Mult2	No					

	Busy	*Addr.*
Load1	No	
Load2	Yes	45+R3
Load3	No	

Load Buffer

Tomasulo Example: Cycle 5

Instruction Status

		j	k	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3		
4. SUB.D	F8	F6	F2	C4		
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2			

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	M[A1]
F8	Add1
F10	Mult2

Reservation Stations

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	Add1	Yes	SU	M[A1]	M[A2]		
	Add2	No					
	Add3	No					
10	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	Addr.
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 6

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	
4. SUB.D	F8	F6	F2	C4	C6	
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2	C6		

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	Add2
F8	Add1
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
1	Add1	Yes	SU	M[A1]	M[A2]		
	Add2	Yes	AD		M[A2]	Add1	
	Add3	No					
9	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Q: Issue ADD.D here despite name dependency on F6?

Tomasulo Example: Cycle 7

Instruction Status

		j	k	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	
4. SUB.D	F8	F6	F2	C4	C6	
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2	C6		

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	Add2
F8	Add1
F10	Mult2

Reservation Stations

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
0	Add1	Yes	SU	M[A1]	M[A2]		
	Add2	Yes	AD		M[A2]	Add1	
	Add3	No					
8	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	Addr.
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 8

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	
4. SUB.D	F8	F6	F2	C4	C6	C8
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2	C6		

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	Add2
F8	M-M
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
2	Add2	Yes	AD	M-M	M[A2]		
	Add3	No					
7	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 9

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	
4. SUB.D	F8	F6	F2	C4	C6	C8
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2	C6	C9	

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	Add2
F8	M-M
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
1	Add2	Yes	AD	M-M	M[A2]		
	Add3	No					
6	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 10

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	
4. SUB.D	F8	F6	F2	C4	C6	C8
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2	C6	C9	

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	Add2
F8	M-M
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
0	Add2	Yes	AD	M-M	M[A2]		
	Add3	No					
5	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 11

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	
4. SUB.D	F8	F6	F2	C4	C6	C8
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2	C6	C9	C11

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	M-M+M
F8	M-M
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
4	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 12

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	
4. SUB.D	F8	F6	F2	C4	C6	C8
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2	C6	C9	C11

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	M-M+M
F8	M-M
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
3	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 13

Instruction Status

		j	k	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	
4. SUB.D	F8	F6	F2	C4	C6	C8
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2	C6	C9	C11

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	M-M+M
F8	M-M
F10	Mult2

Reservation Stations

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
2	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	Addr.
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 14

Instruction Status

			j	*k*	IS	EX	WR
1.	LD	F6	34	R2	C1	C2	C4
2.	LD	F2	45	R3	C2	C3	C5
3.	MUL.D	F0	F2	F4	C3	C6	
4.	SUB.D	F8	F6	F2	C4	C6	C8
5.	DIV. D	F10	F0	F6	C5		
6.	ADD.D	F6	F8	F2	C6	C9	C11

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	M-M+M
F8	M-M
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
1	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 15

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	
4. SUB.D	F8	F6	F2	C4	C6	C8
5. DIV. D	F10	F0	F6	C5		
6. ADD.D	F6	F8	F2	C6	C9	C11

Register Results Status

F0	Mult1
F2	M[A2]
F4	
F6	M-M+M
F8	M-M
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
0	Mult1	Yes	ML	M[A2]	R[F4]		
	Mult2	Yes	DI		M[A1]	Mult1	

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 16

Instruction Status

			j	*k*	IS	EX	WR
1.	LD	F6	34	R2	C1	C2	C4
2.	LD	F2	45	R3	C2	C3	C5
3.	MUL.D	F0	F2	F4	C3	C6	C16
4.	SUB.D	F8	F6	F2	C4	C6	C8
5.	DIV. D	F10	F0	F6	C5		
6.	ADD.D	F6	F8	F2	C6	C9	C11

Register Results Status

F0	M*R
F2	M[A2]
F4	
F6	M-M+M
F8	M-M
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
40	Mult2	Yes	DI	M*R	M[A1]		

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 56

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	C16
4. SUB.D	F8	F6	F2	C4	C6	C8
5. DIV. D	F10	F0	F6	C5	C17	
6. ADD.D	F6	F8	F2	C6	C9	C11

Register Results Status

F0	M*R
F2	M[A2]
F4	
F6	M-M+M
F8	M-M
F10	Mult2

Reservation Stations

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
0	Mult2	Yes	DI	M*R	M[A1]		

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

Tomasulo Example: Cycle 57

Instruction Status

		j	*k*	IS	EX	WR
1. LD	F6	34	R2	C1	C2	C4
2. LD	F2	45	R3	C2	C3	C5
3. MUL.D	F0	F2	F4	C3	C6	C16
4. SUB.D	F8	F6	F2	C4	C6	C8
5. DIV. D	F10	F0	F6	C5	C17	C57
6. ADD.D	F6	F8	F2	C6	C9	C11

Register Results Status

F0	M*R
F2	M[A2]
F4	
F6	M-M+M
F8	M-M
F10	M*R/M

Reservation Stations

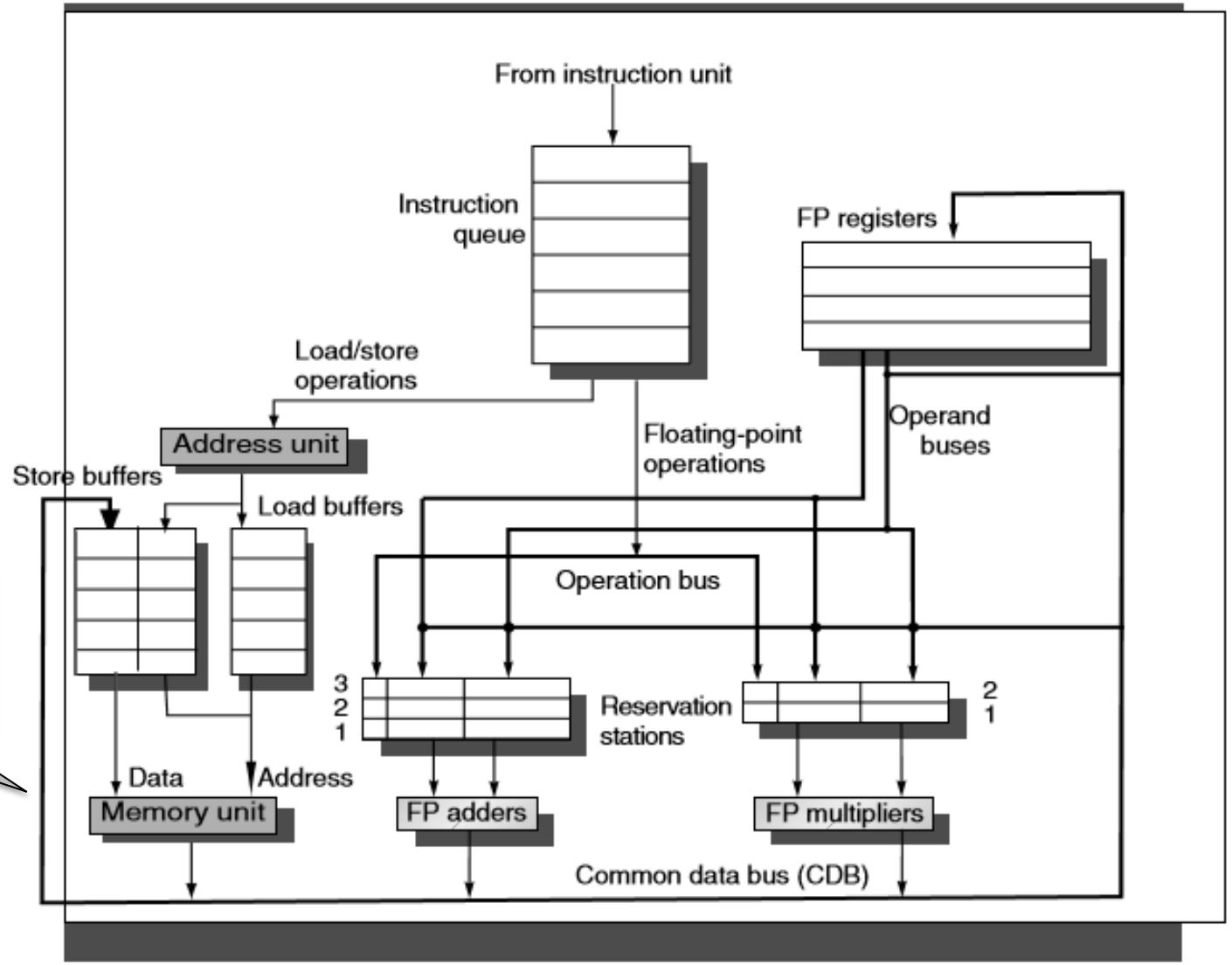
Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
	Mult2	No					

	Busy	*Addr.*
Load1	No	
Load2	No	
Load3	No	

Load Buffer

In-order issue, out-of-order execution, out-of-order completion

Basic Structure Review



CDB:
data + source tag
("come from" bus)

Tomasulo vs. Scoreboard

- **Key features of Tomasulo**
  - Reservation Stations (RS) for distributed control
  - Common Data Bus (CDB) broadcasts all results
  - Use tags to identify data values

- **Differences from Scoreboard**
  - Distributed hazard detection and control with RS
  - Results are bypassed to function unit
  - Common data bus for results

Q: Structure hazard for Tomasulo?

Summary

- Pipeline stall and bubble
- Dependency and hazards
- RAW, WAR, WAW
- Forwarding and pipeline interlock
- Functional units
- Dynamic scheduling and OoO
- Scoreboarding
- Tomasulo's Algorithm