Lecture 6: ILP HW Case Study— CDC 6600 Scoreboard & Tomasulo's Algorithm

Professor Alvin R. Lebeck
Computer Science 220
Fall 2001

Admin

- **HW #2**
- **Project Selection by October 2**
  - **Your own ideas?**
- **Short proposal due October 2**
  - **Content: problem definition, goal of project, metric for success**
  - **3 - 5 page document**
  - **5 - 10 minute presentation**
- **Status report due November 1.**
  - **document only**
- **Final report due December 6**
  - **8-10 page document**
  - **15-20 minute presentation**

Review: ILP

- Instruction Level Parallelism in SW or HW
- Loop level parallelism is easiest to see

Today

- SW parallelism dependencies defined for program, hazards if HW cannot resolve dependencies
- SW dependencies/Compiler sophistication determine if compiler can unroll loops
  - Memory dependencies hardest to determine

Review: FP Loop Showing Stalls

```
1 Loop: LD    F0,0(R1)  ;F0=vector element
2       stall
3       ADDD  F4,F0,F2  ;add scalar in F2
4       stall
5       stall
6       SD    0(R1),F4  ;store result
7       SUBI  R1,R1,8   ;decrement pointer 8B (DW)
8       BNEZ  R1,Loop   ;branch R1!=zero
9       stall           ;delayed branch slot
```

Instruction producing result	*Instruction using result*	*Latency in clock cycles*
FP ALU op	Another FP ALU op	3
FP ALU op	Store double	2
Load double	FP ALU op	1

- Rewrite code to minimize stalls?

Review: Unrolled Loop That Minimizes Stalls

```
1 Loop: LD     F0,0(R1)
2       LD     F6,-8(R1)
3       LD     F10,-16(R1)
4       LD     F14,-24(R1)
5       ADDD   F4,F0,F2
6       ADDD   F8,F6,F2
7       ADDD   F12,F10,F2
8       ADDD   F16,F14,F2
9       SD     0(R1),F4
10      SD     -8(R1),F8
11      SD     -16(R1),F12
12      SUBI   R1,R1,#32
13      BNEZ   R1,LOOP
14      SD     8(R1),F16    ; 8-32 = -24
```

- **What assumptions made when moved code?**
  - OK to move store past SUBI even though changes register
  - OK to move loads before stores: get right data?
  - When is it safe for compiler to do such changes?

14 clock cycles, or 3.5 per iteration

Review: Hazard Detection

- Assume all hazard detection in ID stage

1. Check for structural hazards.
2. Check for RAW data hazard.
3. Check for WAW data hazard.

- If any occur stall at ID stage
- This is called an in-order issue/execute machine, if any instruction stalls all later instructions stall.
  - Note that instructions may complete execution out of order.

Can we do better?

- **Problem: Stall in ID stage if any data hazard.**
- **Your task: Teams of two, propose a design to eliminate these stalls.**

```
MULD  F2, F3, F4
ADDD  F1, F2, F3
ADDD  F3, F4, F5
ADDD  F1, F4, F5
```

Long latency…

HW Schemes: Instruction Parallelism

- **Why in HW at run time?**
  - **Works when can't know dependencies**
  - **Simpler Compiler**
  - **Code for one machine runs well on another machine**
- **Key Idea: Allow instructions behind stall to proceed**

```
DIVD    F0, F2, F4
ADD     F10, F0, F8
SUBD    F8, F8, F14
```

  - **Enables out-of-order execution => out-of-order completion**
  - **ID stage check for both structural & data dependencies**

HW Schemes: Instruction Parallelism

- **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 allow instruction to execute whenever 1 & 2 hold, not waiting for prior instructions**

Scoreboard Implications

- **Out-of-order completion => WAR, WAW hazards?**
- **Solutions for WAR**
  - **Queue both the operation and copies of its operands**
  - **Read registers only during Read Operands stage**
- **For WAW, must detect hazard: stall until other completes**
- **Need to have multiple instructions in execution phase => multiple execution units or pipelined execution units**
- **Scoreboard keeps track of dependencies, state or operations**
- **Scoreboard replaces ID, EX, WB with 4 stages**

Four Stages of Scoreboard Control

1. Issue: decode instructions & check for structural hazards (ID1)

If a functional unit for the instruction is free and no other active instruction has the same destination register (WAW), the scoreboard issues the instruction to the functional unit and updates its internal data structure. If a structural or WAW hazard exists, then the instruction issue stalls, and no further instructions will issue until these hazards are cleared.

2. Read operands: wait until no data hazards, then read operands (ID2)

A source operand is available if no earlier issued active instruction is going to write it, or if the register containing the operand is being written by a currently active functional unit. When the source operands are available, the scoreboard tells the functional unit to proceed to read the operands from the registers and begin execution. The scoreboard resolves RAW hazards dynamically in this step, and instructions may be sent into execution out of order.

Four Stages of Scoreboard Control

3. Execution: operate on operands

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

4. Write Result: finish execution (WB)

Once the scoreboard is aware that the functional unit has completed execution, the scoreboard checks for WAR hazards. If none, it writes results. If WAR, then it stalls the instruction.

Example:

```
DIVD    F0,F2,F4
ADDD    F10,F0,F8
SUBD    F8,F8,F14
```

CDC 6600 scoreboard would stall SUBD until ADDD reads operands

Three Parts of the Scoreboard

1. **Instruction status**: which of 4 steps the instruction is in
2. **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
3. **Register result status**: Indicates which functional unit will write each register, if one exists. Blank when no pending instructions will write that register

INT ADD/SUB: 1 clock cycle

Load/store effective address computation ("execution" stage) combined with Memory access: 1 clock cycle

FP ADD/SUB: 2 clock cycles

FP MUL: 10 clock cycles

FP DIV: 40 clock cycles

Scoreboard Example Cycle 1

Instruction Status

Instruction	j	k	Issue	Read Operand	Execution Complete	Write Result
LD F6	34+	R2	1			
LD F2	45+	R3				
MULT F0	F2	F4				
SUBD F8	F6	F2				
DIVD F10	F0	F6				
ADDD F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F6		R2				Yes
Mult1	No								
Mult2	No								
Add	No								
Divide	No								

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
1	FU				Int					

Scoreboard Example Cycle 2

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2		
LD	F2	45+	R3				
MULT	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F6		R2				Yes
Mult1	No								
Mult2	No								
Add	No								
Divide	No								

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
2	FU				Int					

Scoreboard Example Cycle 3

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	
LD	F2	45+	R3				
MULT	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F6		R2				Yes
Mult1	No								
Mult2	No								
Add	No								
Divide	No								

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
3	FU				Int					

Scoreboard Example Cycle 4

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3				
MULT	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F6		R2				Yes
Mult1	No								
Mult2	No								
Add	No								
Divide	No								

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
4	FU				Int					

Scoreboard Example Cycle 5

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5			
MULT	F0	F2	F4				
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F2		R3				Yes
Mult1	No								
Mult2	No								
Add	No								
Divide	No								

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
5	FU		Int							

Scoreboard Example Cycle 6

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6		
MULT	F0	F2	F4	6			
SUBD	F8	F6	F2				
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F2		R3				Yes
Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
Mult2	No								
Add	No								
Divide	No								

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
6	FU	Mul1	Int							

Scoreboard Example Cycle 7

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	
MULT	F0	F2	F4	6			
SUBD	F8	F6	F2	7			
DIVD	F10	F0	F6				
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F2		R3				Yes
Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
Mult2	No								
Add	Yes	Sub	F8	F6	F2		Int	Yes	No
Divide	No								

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
7	FU	Mul1	Int			Add				

Scoreboard Example Cycle 8a

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	
MULT	F0	F2	F4	6			
SUBD	F8	F6	F2	7			
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	Yes	Load	F2		R3				Yes
Mult1	Yes	Mult	F0	F2	F4	Integer		No	Yes
Mult2	No								
Add	Yes	Sub	F8	F6	F2		Int	Yes	No
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
8	FU	Mul1	Int			Add	Div			

Scoreboard Example Cycle 8b

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	F0	F2	F4	6			
SUBD	F8	F6	F2	7			
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
Mult2	No								
Add	Yes	Sub	F8	F6	F2		Int	Yes	Yes
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
8	FU	Mul1				Add	Div			

Scoreboard Example Cycle 9

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9		
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
Mult2	No								
Add	Yes	Sub	F8	F6	F2		Int	Yes	Yes
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	...	F31
9	FU	Mul1				Add	Div			

Scoreboard Example Cycle 11

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	F0	F2	F4	6	9		
SUBD	F8	F6	F2	7	9	11	
DIVD	F10	F0	F6	8			
ADDD	F6	F8	F2				

Functional Unit Status

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
Mult2	No								
Add	Yes	Sub	F8	F6	F2		Int	Yes	Yes
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	...	F31
11	FU	Mul1				Add	Div			

Scoreboard Example Cycle 12

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
Mult2	No								
Add	No								
Divide	Yes	Div	F10	F0	F6	Mult1		No	Yes

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
12	FU	Mul1					Div			

Scoreboard Example Cycle 13

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
Mult2	No								
Add	Yes	Ad	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	…	F31
13	FU	Mul1			Add		Div			

Scoreboard Example Cycle 14

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
Mult2	No								
Add	Yes	Ad	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	…	F31
14	FU	Mul1			Add		Div			

Scoreboard Example Cycle 15

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
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	…	F31
15	FU	Mul1			Add		Div			

Scoreboard Example Cycle 16

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
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	…	F31
16	FU	Mul1			Add		Div			

Scoreboard Example Cycle 17

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
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	…	F31
17	FU	Mul1			Add		Div			

Scoreboard Example Cycle 18

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
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	…	F31
18	FU	Mul1			Add		Div			

Scoreboard Example Cycle 19

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	Yes	Mult	F0	F2	F4			Yes	Yes
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	…	F31
19	FU	Mul1			Add		Div			

Scoreboard Example Cycle 20

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	No								
Mult2	No								
Add	Yes	Add	F6	F8	F2			Yes	Yes
Divide	Yes	Div	F10	F0	F6	Mult1		Yes	Yes

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
20	FU				Add		Div			

Scoreboard Example Cycle 21

Instruction Status

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk
Integer	No								
Mult1	No								
Mult2	No								
Add	Yes	Add	F6	F8	F2			Yes	Yes
Divide	Yes	Div	F10	F0	F6	Mult1		Yes	Yes

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
21	FU				Add		Div			

Scoreboard Example Cycle 22

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

40 cycle Divide

Functional Unit Status

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

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
22	FU						Div			

Scoreboard Example Cycle 61

Instruction		j	k	Issue	Read Operand	Execution Complete	Write Result
LD	F6	34+	R2	1	2	3	4
LD	F2	45+	R3	5	6	7	8
MULT	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

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

Register Result Status

CLOCK		F0	F2	F4	F6	F8	F10	F12	…	F31
61	FU						Div			

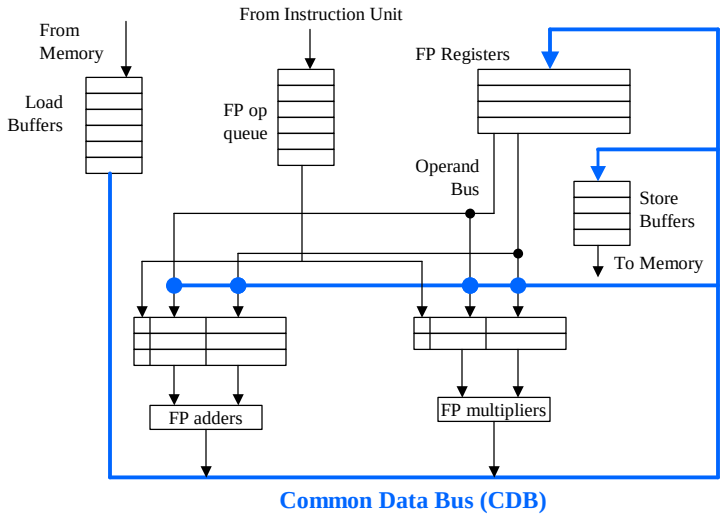
Scoreboard Summary

- Speedup 1.7 from compiler; 2.5 by hand
  BUT slow memory (no cache)
- Limitations of 6600 scoreboard
  - No forwarding
  - Limited to instructions in basic block (small *window*)
  - Number of functional units (structural hazards)
  - Wait for WAR hazards
  - Prevent WAW hazards
- How to design a datapath that eliminates these problems?

Tomasulo's Algorithm: Another Dynamic Scheme

- For IBM 360/91 about 3 years after CDC 6600
- 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
- Differences between Tomasulo Algorithm & Scoreboard
  - Control & buffers distributed with Function Units vs. centralized in scoreboard; called "reservation stations"
  - Register specifiers in instructions replaced by pointers to reservation station buffer (Everything can be solved with level of indirection!)
  - HW renaming of registers to avoid WAR, WAW hazards
  - Common Data Bus broadcasts results to all FUs
  - Load and Stores treated as FUs as well

Tomasulo Organization



Reservation Station Components

Op—Operation to perform in the unit (e.g., + or –)

Qj, Qk—Reservation stations producing source registers

Vj, Vk—Value of Source operands

Rj, Rk—Flags indicating when Vj, Vk are ready

Busy—Indicates reservation station and 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.

INT ADD/SUB: 1 clock cycle

Load/store effective address computation: 1 clock cycle

Load/store memory access: 1 clock cycle

FP ADD/SUB: 2 clock cycles

FP MUL: 10 clock cycles

FP DIV: 40 clock cycles

Three Stages of Tomasulo Algorithm

1. **Issue**—get instruction from FP Op Queue
   If reservation station free, the scoreboard issues instr & sends operands (renames registers).
2. **Execution**—operate on operands (EX)
   When both operands ready then execute; if not ready, watch CDB for result
3. **Write result**—finish execution (WB)
   Write on Common Data Bus to all awaiting units; mark reservation station available.

INT ADD/SUB: 1 clock cycle

Load/store effective address computation: 1 clock cycle

Load/store memory access: 1 clock cycle

FP ADD/SUB: 2 clock cycles

FP MUL: 10 clock cycles

FP DIV: 40 clock cycles

# of FP ADD/SUB Units: 3
# of FP MUL/DIV Units: 2
# of LD Units: 1
# of SD Units: 1

Tomasulo Example Cycle 0 (initial state)

Note: the "Names" are reservation stations associated with the corresponding Functional Unit
Load1, Load2, Load3 are Load Buffers, which are attached to only 1 Load unit
Mult2 handles DIVD

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2			
LD	F2	45+	R3			
MULT	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

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

Register result status

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

INT ADD/SUB: 1 clock cycle

Load/store effective address computation: 1 clock cycle

Load/store memory access: 1 clock cycle

FP ADD/SUB: 2 clock cycles

FP MUL: 10 clock cycles

FP DIV: 40 clock cycles

# of FP ADD/SUB Units: 3
# of FP MUL/DIV Units: 2
# of LD Units: 1
# of SD Units: 1

Tomasulo Example Cycle 1

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1		
LD	F2	45+	R3			
MULT	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 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
0	Mult2	No					

Register result status

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

Tomasulo Example Cycle 2

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1		
LD	F2	45+	R3	2		
MULT	F0	F2	F4			
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

Reservation Stations

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

Register result status

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

LD F2, 45(R3) cannot start execution at clock cycle #3 due to structural hazard!

Tomasulo Example Cycle 3

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	
LD	F2	45+	R3	2		
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2			
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	Yes	34+R2
Load2	Yes	45+R3
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD		R(F4)	Load2	
0	Mult2	No					

Register result status

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

LD F2, 45(R3) starts execution at clock cycle #4
MULT is stuck because of F2!
F6 has been updated in the Register File, its value is the same as the value at M[34+R2]

Tomasulo Example Cycle 4

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2		
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	F0	F6			
ADDD	F6	F8	F2			

	Busy	Address
Load1	No	
Load2	Yes	45+R3
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
	Add1	Yes	SUBD	M(34+R2)			Load2
0	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD		R(F4)	Load2	
0	Mult2	No					

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
4	FU	Mult1	Load2		M(34+R2)	Add1				

MULT is still waiting!
SUBD has got F6 ready because of CDB-based broadcast, but waiting for F2!

Tomasulo Example Cycle 5

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2			

	Busy	Address
Load1	No	
Load2	Yes	45+R3
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	Yes	SUBD	M(34+R2)			Load2
0	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD		R(F4)	Load2	
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
5	FU	Mult1	Load2		M(34+R2)	Add1	Mult2			

MULT starts execution
SUBD starts execution

Tomasulo Example Cycle 6

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
2	Add1	Yes	SUBD	M(34+R2)	M(45+R3)		
0	Add2	Yes	ADDD		M(45+R3)	Add1	
	Add3	No					
10	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
6	FU	Mult1	M(45+R3)		Add2	Add1	Mult2			

Tomasulo Example Cycle 7

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4		
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
1	Add1	Yes	SUBD	M(34+R2)	M(45+R3)		
0	Add2	Yes	ADDD		M(45+R3)	Add1	
	Add3	No					
9	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
7	FU	Mult1	M(45+R3)		Add2	Add1	Mult2			

Tomasulo Example Cycle 8

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4	8	
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	Yes	SUBD	M(34+R2)	M(45+R3)		
0	Add2	Yes	ADDD		M(45+R3)	Add1	
	Add3	No					
8	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
8	FU	Mult1	M(45+R3)		Add2	Add1	Mult2			

Tomasulo Example Cycle 9

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
7	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
9	FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2			

Tomasulo Example Cycle 10

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
2	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
6	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
10	FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2			

Tomasulo Example Cycle 11

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
1	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
5	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
11	FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2			

Tomasulo Example Cycle 12

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	12	

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
4	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
12	FU	Mult1	M(45+R3)		Add2	M()–M()	Mult2			

Tomasulo Example Cycle 13

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
3	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
13	FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 14

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
2	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
14	FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 15

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3		
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
1	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
15	FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 16

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3	16	
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
16	FU	Mult1	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 17

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3	16	17
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
0	Mult2	Yes	DIVD	M*F4	M(34+R2)		

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
17	FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2		

Tomasulo Example Cycle 18

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3	16	17
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
40	Mult2	Yes	DIVD	M*F4	M(34+R2)		

Register result status

Clock	F0	F2	F4	F6	F8	F10	F12	...	F30
18	FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2		

Tomasulo Example Cycle 57

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3	16	17
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
1	Mult2	Yes	DIVD	M*F4	M(34+R2)		

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
57	FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 58

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3	16	17
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5	58	
ADDD	F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
	Add1	No					
0	Add2	No					
	Add3	No					
0	Mult1	No					
0	Mult2	Yes	DIVD	M*F4	M(34+R2)		

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
58	FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	Mult2			

Tomasulo Example Cycle 59

Instruction status

Instruction		j	k	Issue	Execution complete	Write Result
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MULT	F0	F2	F4	3	16	17
SUBD	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5	58	59
ADDD	F6	F8	F2	6	12	13

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations

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

Register result status

Clock		F0	F2	F4	F6	F8	F10	F12	...	F30
59	FU	M*F4	M(45+R3)		(M–M)+M()	M()–M()	M*F4/M			

Tomasulo vs. Scoreboard

- **Is tomasulo better?**
- **Finish in 59 cycles vs. 61 for scoreboard, why?**
- **We do reach the divide 3 cycles earlier… Simultaneous read of operand for SUBD and MULT**

3 LD Units
3 SD Units

LD/SD effective address computation: 1 clock cycle

1st LD: 7 clock cycles to get value from memory (after effective address is known)
2nd LD: 3 clock cycles to get value from memory (after effective address is known)

SD: cycles to send value to memory (after effective address is known)

SUBI is executed in INT unit (not shown here!), takes 1 clock cycle

Tomasulo Loop Example

```
Loop: LD     F0   0    R1
      MULTD  F4   F0   F2
      SD     F4   0    R1
      SUBI   R1   R1   #8
      BNEZ   R1   Loop
```

- **Multiply takes 4 clocks**
- **Loads may have cache misses**

Loop Example Cycle 0

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1			
MULT F4	F0	F2	1			
SD F4	0	R1	1			
LD F0	0	R1	2			
MULT F4	F0	F2	2			
SD F4	0	R1	2			

	Busy	Address	Qi
Load1	No		
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations

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

Code:

```
LD    F0   0   R1
MULT  F4   F0  F2
SD    F4   0   R1
SUBI  R1   R1  #8
BNEZ  R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
0	80	Qi								

Loop Example Cycle 1

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1		
MULT	F4	F0	F2	1			
SD	F4	0	R1	1			
LD	F0	0	R1	2			
MULT	F4	F0	F2	2			
SD	F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations

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

Code:

```
LD   F0   0  R1
MULT F4   F0 F2
SD   F4   0  R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
1	80	Qi	Load1							

Loop Example Cycle 2

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1		
MULT	F4	F0	F2	1	2		
SD	F4	0	R1	1			
LD	F0	0	R1	2			
MULT	F4	F0	F2	2			
SD	F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load1	
0	Mult2	No					

Code:

```
LD   F0   0  R1
MULT F4   F0 F2
SD   F4   0  R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
2	80	Qi	Load1		Mult1					

Loop Example Cycle 3

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1		
MULT	F4	F0	F2	1	2		
SD	F4	0	R1	1	3		
LD	F0	0	R1	2			
MULT	F4	F0	F2	2			
SD	F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load1	
0	Mult2	No					

Code:

```
LD   F0   0  R1
MULT F4   F0 F2
SD   F4   0  R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
3	80	Qi	Load1		Mult1					

Loop Example Cycle 4

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1		
MULT	F4	F0	F2	1	2		
SD	F4	0	R1	1	3		
LD	F0	0	R1	2			
MULT	F4	F0	F2	2			
SD	F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load1	
0	Mult2	No					

Code:

```
LD   F0   0  R1
MULT F4   F0 F2
SD   F4   0  R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
4	72	Qi	Load1		Mult1					

Loop Example Cycle 5

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1		
MULT	F4	F0	F2	1	2		
SD	F4	0	R1	1	3		
LD	F0	0	R1	2			
MULT	F4	F0	F2	2			
SD	F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load1	
0	Mult2	No					

Code:

```
LD   F0  0  R1
MULT F4  F0 F2
SD   F4  0  R1
SUBI R1  R1 #8
BNEZ R1  Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
5	72	Qi	Load1		Mult1					

Loop Example Cycle 6

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1		
MULT	F4	F0	F2	1	2		
SD	F4	0	R1	1	3		
LD	F0	0	R1	2	6		
MULT	F4	F0	F2	2			
SD	F4	0	R1	2			

	Busy	Address	Qi
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load1	
0	Mult2	No					

Code:

```
LD   F0  0  R1
MULT F4  F0 F2
SD   F4  0  R1
SUBI R1  R1 #8
BNEZ R1  Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
6	72	Qi	Load1		Mult1					

Loop Example Cycle 7

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD	F0	0	R1	1	1			Load1	Yes	80	
MULT	F4	F0	F2	1	2			Load2	Yes	72	
SD	F4	0	R1	1	3			Load3	No		Qi
LD	F0	0	R1	2	6			Store1	Yes	80	Mult1
MULT	F4	F0	F2	2	7			Store2	No		
SD	F4	0	R1	2				Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI	R1	R1	#8
0	Mult2	Yes	MULTD		R(F2)	Load2		BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
7	72	Qi	Load2		Mult2					

Loop Example Cycle 8

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD	F0	0	R1	1	1			Load1	Yes	80	
MULT	F4	F0	F2	1	2			Load2	Yes	72	
SD	F4	0	R1	1	3			Load3	No		Qi
LD	F0	0	R1	2	6			Store1	Yes	80	Mult1
MULT	F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD	F4	0	R1	2	8			Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load1		SUBI	R1	R1	#8
0	Mult2	Yes	MULTD		R(F2)	Load2		BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
8	72	Qi	Load2		Mult2					

Loop Example Cycle 9

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6		
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	Qi
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load1	
0	Mult2	Yes	MULTD		R(F2)	Load2	

Code:

```
LD   F0  0  R1
MULT F4  F0 F2
SD   F4  0  R1
SUBI R1  R1 #8
BNEZ R1  Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
9	64	Qi	Load2		Mult2					

Loop Example Cycle 10

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result
LD F0	0	R1	1	1	9	10
MULT F4	F0	F2	1	2		
SD F4	0	R1	1	3		
LD F0	0	R1	2	6	10	
MULT F4	F0	F2	2	7		
SD F4	0	R1	2	8		

	Busy	Address	Qi
Load1	No		
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
4	Mult1	Yes	MULTD	M(80)	R(F2)		
0	Mult2	Yes	MULTD		R(F2)	Load2	

Code:

```
LD   F0  0  R1
MULT F4  F0 F2
SD   F4  0  R1
SUBI R1  R1 #8
BNEZ R1  Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
10	64	Qi	Load2		Mult2					

Loop Example Cycle 11

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1	9	10
MULT	F4	F0	F2	1	2		
SD	F4	0	R1	1	3		
LD	F0	0	R1	2	6	10	11
MULT	F4	F0	F2	2	7		
SD	F4	0	R1	2	8		

	Busy	Address	Qi
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
3	Mult1	Yes	MULTD	M(80)	R(F2)		
4	Mult2	Yes	MULTD	M(72)	R(F2)		

Code:			
LD	F0	0	R1
MULT	F4	F0	F2
SD	F4	0	R1
SUBI	R1	R1	#8
BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
11	64	Qi			Mult2					

Loop Example Cycle 12

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1	9	10
MULT	F4	F0	F2	1	2		
SD	F4	0	R1	1	3		
LD	F0	0	R1	2	6	10	11
MULT	F4	F0	F2	2	7		
SD	F4	0	R1	2	8		

	Busy	Address	Qi
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
2	Mult1	Yes	MULTD	M(80)	R(F2)		
3	Mult2	Yes	MULTD	M(72)	R(F2)		

Code:			
LD	F0	0	R1
MULT	F4	F0	F2
SD	F4	0	R1
SUBI	R1	R1	#8
BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
12	64	Qi			Mult2					

Loop Example Cycle 13

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1	9	10
MULT	F4	F0	F2	1	2		
SD	F4	0	R1	1	3		
LD	F0	0	R1	2	6	10	11
MULT	F4	F0	F2	2	7		
SD	F4	0	R1	2	8		

	Busy	Address	
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
1	Mult1	Yes	MULTD	M(80)	R(F2)			SUBI	R1	R1	#8
2	Mult2	Yes	MULTD	M(72)	R(F2)			BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
13	64	Qi			Mult2					

Loop Example Cycle 14

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result
LD	F0	0	R1	1	1	9	10
MULT	F4	F0	F2	1	2	14	
SD	F4	0	R1	1	3		
LD	F0	0	R1	2	6	10	11
MULT	F4	F0	F2	2	7		
SD	F4	0	R1	2	8		

	Busy	Address	
Load1	No		
Load2	No		
Load3	Yes	64	Qi
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD	M(80)	R(F2)			SUBI	R1	R1	#8
1	Mult2	Yes	MULTD	M(72)	R(F2)			BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
14	64	Qi			Mult2					

Loop Example Cycle 15

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
MULT F4	F0	F2	1	2	14	15	Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(F2
MULT F4	F0	F2	2	7	15		Store2	Yes	72	Mult2
SD F4	0	R1	2	8			Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	No						SUBI	R1	R1	#8
0	Mult2	Yes	MULTD	M(72)	R(F2)			BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
15	64	Qi			Mult2					

Loop Example Cycle 16

Instruction status

Instruction	j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
MULT F4	F0	F2	1	2	14	15	Load2	No		
F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(F2
MULT F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(72
SD F4	0	R1	2	8			Store3	No		

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
0	Mult2	No						BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
16	64	Qi			Mult1					

Loop Example Cycle 17

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
MULT	F4	F0	F2	1	2	14	15	Load2	No		
SD	F4	0	R1	1	3			Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(F2
MULT	F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(72
SD	F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load3	
0	Mult2	No					

Code:

```
LD   F0   0  R1
MULT F4   F0 F2
SD   F4   0  R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
17	64	Qi			Mult1					

Loop Example Cycle 18

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
MULT	F4	F0	F2	1	2	14	15	Load2	No		
SD	F4	0	R1	1	3	18		Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	Yes	80	M(80)*R(F2
MULT	F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(72
SD	F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD		R(F2)	Load3	
0	Mult2	No					

Code:

```
LD   F0   0  R1
MULT F4   F0 F2
SD   F4   0  R1
SUBI R1   R1 #8
BNEZ R1   Loop
```

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
18	56	Qi			Mult1					

Loop Example Cycle 19

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
MULT	F4	F0	F2	1	2	14	15	Load2	No		
SD	F4	0	R1	1	3	18	19	Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	No		
MULT	F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(72
SD	F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
0	Mult2	No						BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
19	56	Qi			Mult1					

Loop Example Cycle 20

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
MULT	F4	F0	F2	1	2	14	15	Load2	No		
SD	F4	0	R1	1	3	18	19	Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	No		
MULT	F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(72
SD	F4	0	R1	2	8	20		Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
0	Mult2	No						BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
20	56	Qi			Mult1					

Loop Example Cycle 21

Instruction status

Instruction		j	k	iteration	Issue	Execution complete	Write Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
MULT	F4	F0	F2	1	2	14	15	Load2	No		
SD	F4	0	R1	1	3	~~18~~	~~19~~	Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	No		
MULT	F4	F0	F2	2	7	15	16	Store2	No		
SD	F4	0	R1	2	8	20	21	Store3	Yes	64	Mult1

Reservation Stations

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS for j Qj	RS for k Qk	Code:			
0	Add1	No						LD	F0	0	R1
0	Add2	No						MULT	F4	F0	F2
0	Add3	No						SD	F4	0	R1
0	Mult1	Yes	MULTD		R(F2)	Load3		SUBI	R1	R1	#8
0	Mult2	No						BNEZ	R1	Loop	

Register result status

Clock	R1		F0	F2	F4	F6	F8	F10	F12 ...	F30
21	56	Qi			Mult1					

Tomasulo Summary

- **Prevents Register as bottleneck**
- **Avoids WAR, WAW hazards of Scoreboard**
- **Allows loop unrolling in HW**
- **Not limited to basic blocks (provided branch prediction)**
- **Lasting Contributions**
  - **Dynamic scheduling**
  - **Register renaming**
  - **Load/store disambiguation**