Lecture - 6

Instruction-Level Parallelism
Dynamic Pipelines

Dr. Soner Onder

CS 4431

Michigan Technological University

Dynamic Pipelines

- A dynamic pipeline is a pipeline where instructions can “step-out” of the pipeline until some condition is satisfied:
  - Sleep
- They can “step-in” to the pipeline when it is appropriate:
  - Wake-up
  - Select

- This allows dynamic scheduling of instructions.

Advantages of Dynamic Scheduling

- Dynamic scheduling - hardware rearranges the instruction execution to reduce stalls while maintaining data flow and exception behavior

- It handles cases when dependences unknown at compile time
  - it allows the processor to tolerate unpredictable delays such as cache misses, by executing other code while waiting for the miss to resolve

- It allows code that compiled for one pipeline to run efficiently on a different pipeline

- It simplifies the compiler

- Hardware speculation, a technique with significant performance advantages, builds on dynamic scheduling (later).

HW Schemes: Instruction Parallelism

- Key idea: Allow instructions behind stall to proceed

```
DIVD   F0,F2,F4
ADDD   F10,F0,F8
SUBD   F12,F8,F14
```

- Enables out-of-order execution and allows out-of-order completion (e.g., SUBD)
  - In a dynamically scheduled pipeline, all instructions still pass through issue stage in order (in-order issue)

- Will distinguish when an instruction *begins execution* and when it *completes execution*; between 2 times, the instruction is *in execution (or, in-flight).*

- Note: Dynamic execution creates WAR and WAW hazards and makes exceptions harder

Dynamic Scheduling Step 1

- Simple pipeline had 1 stage to check both structural and data hazards: Instruction Decode (ID), also called Instruction Issue
- Split the ID pipe stage of simple 5-stage pipeline into 2 stages:
- *Issue*—Decode instructions, check for structural hazards
- *Read operands*—Wait until no data hazards, then read operands

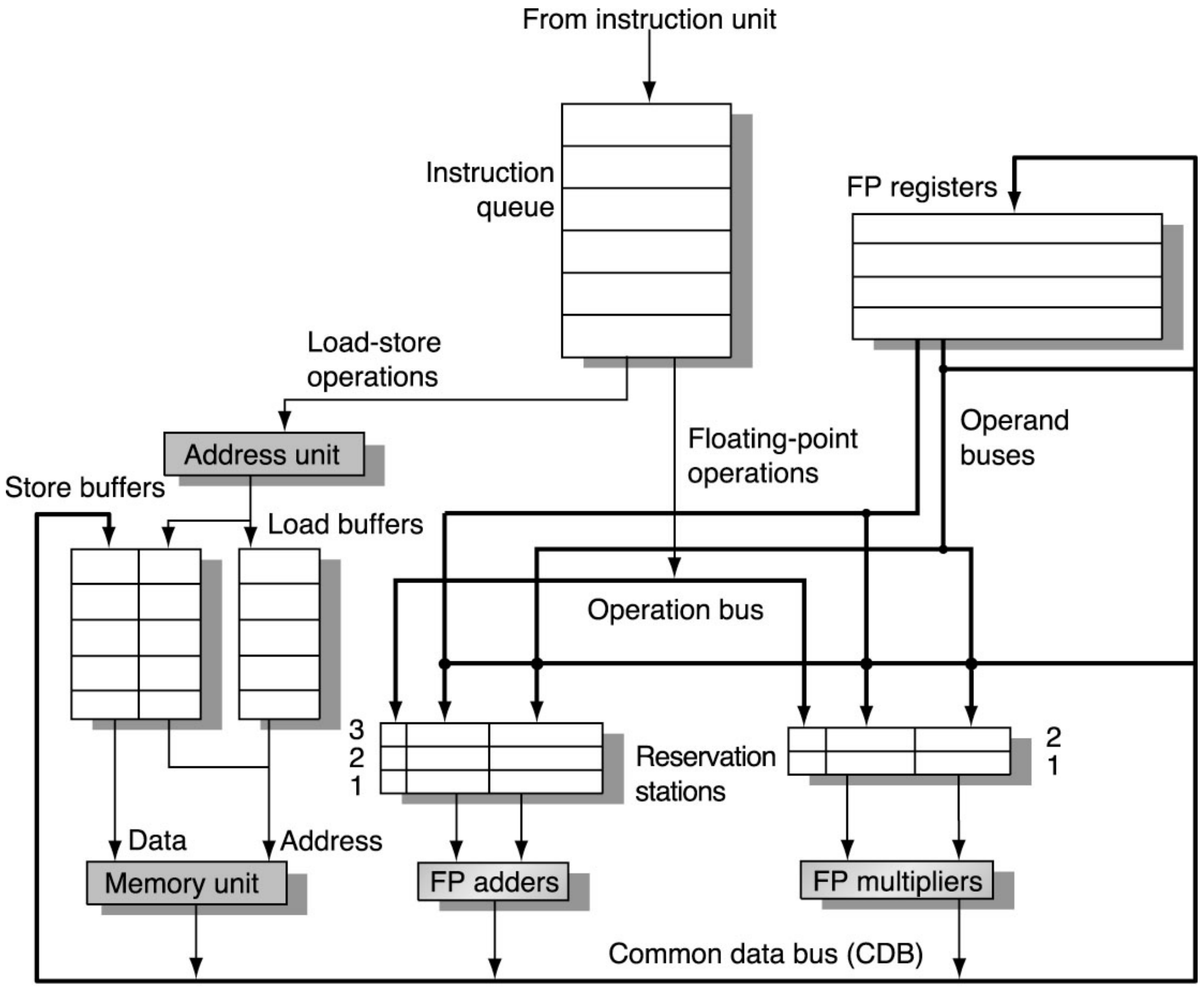
A Dynamic Algorithm: Tomasulo's

- For IBM 360/91 (before caches!)
  - ⇒ Long memory latency
- Goal: High Performance without special compilers
- Small number of floating point registers (4 in 360) prevented interesting compiler scheduling of operations
  - This led Tomasulo to try to figure out how to get more effective registers — renaming in hardware!
- Why Study 1966 Computer?
- The descendants of this have flourished!
  - Alpha 21264, Pentium 4, AMD Opteron, Power 5, …

Tomasulo 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, 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
- Load and Stores treated as FUs with RSs as well
- Integer instructions can go past branches (predict taken), allowing FP ops beyond basic block in FP queue

Tomasulo Organization



Why can 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 scheme offers 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**

Reservation Station Components

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

Vj, Vk: Value of Source operands

- Store buffers has V field, result to be stored

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

- Note: Qj,Qk=0 => ready
- Store buffers only have Qi for RS producing result

Busy: Indicates reservation station or FU is busy

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

Tomasulo Drawbacks

- Complexity
  - delays of 360/91, MIPS 10000, Alpha 21264, IBM PPC 620 in CA:AQA 2/e, but not in silicon!
- Many associative stores (CDB) at high speed
- Performance limited by Common Data Bus
  - Each CDB must go to multiple functional units ⇒high capacitance, high wiring density
  - Number of functional units that can complete per cycle limited to one!
    - Multiple CDBs ⇒ more FU logic for parallel assoc stores
- Non-precise interrupts!
  - We will address this later

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.

Three Stages of Tomasulo Algorithm

1. Issue—get instruction from FP Op Queue

   If reservation station free, the instruction is issued & operands are sent (renames registers).

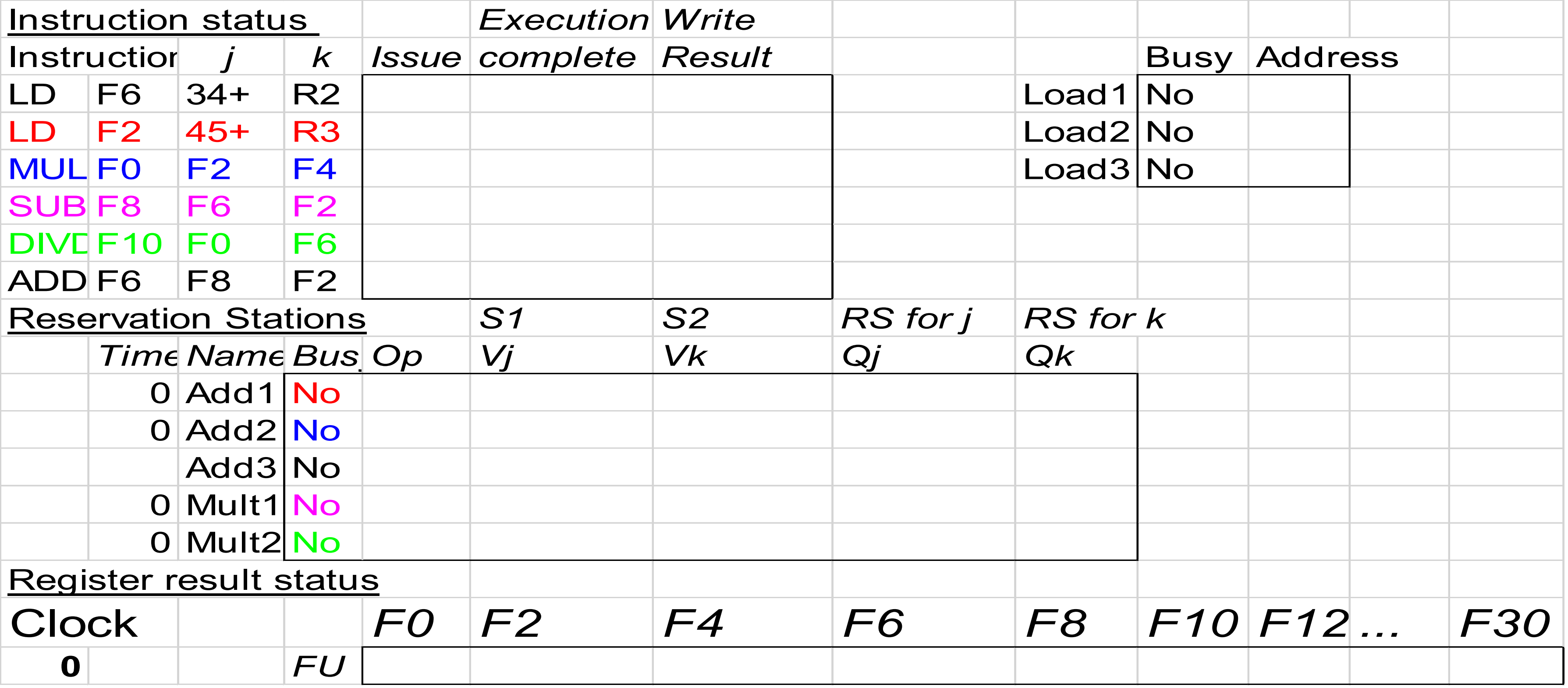
2. Execution—operate on operands (EX)

   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.

Tomasulo Example Cycle 0

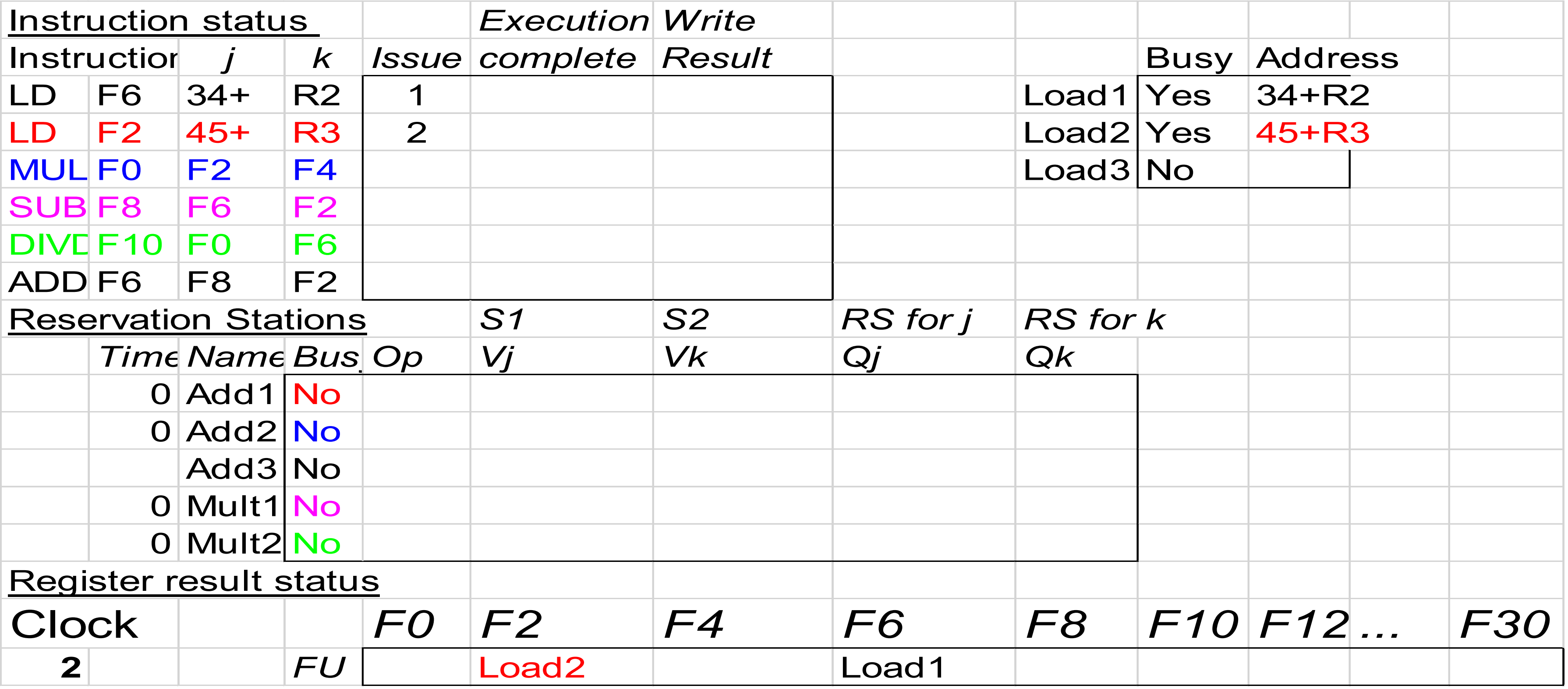


Instruction status					Execution	Write						
Instructior		j	k	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2					Load1	No			
LD	F2	45+	R3					Load2	No			
MUL	F0	F2	F4					Load3	No			
SUB	F8	F6	F2									
DIVD	F10	F0	F6									
ADD	F6	F8	F2									
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Bus	Op	Vj	Vk	Qj	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									

Tomasulo Example Cycle 1

Instruction status					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1				Load1	Yes	34+R2		
LD	F2	45+	R3					Load2	No			
MUL	F0	F2	F4					Load3	No			
SUB	F8	F6	F2									
DIVD	F10	F0	F6									
ADD	F6	F8	F2									
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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					Execution	Write						
Instructior		j	k	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2	1				Load1	Yes	34+R2		
LD	F2	45+	R3	2				Load2	Yes	45+R3		
MUL	F0	F2	F4					Load3	No			
SUB	F8	F6	F2									
DIVD	F10	F0	F6									
ADD	F6	F8	F2									
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Bus	Op	Vj	Vk	Qj	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					

Tomasulo Example Cycle 3

Instruction status					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3			Load1	Yes	34+R2		
LD	F2	45+	R3	2				Load2	Yes	45+R3		
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2									
DIVD	F10	F0	F6									
ADD	F6	F8	F2									
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Bus*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	0	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					

Tomasulo Example Cycle 4

Instruction status					*Execution*	*Write*						
Instruction		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2				Load2	Yes	45+R3		
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4								
DIVD	F10	F0	F6									
ADD	F6	F8	F2									
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	0	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				

Tomasulo Example Cycle 5

Instruction status					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5			Load2	Yes	45+R3		
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4								
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2									
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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			

Tomasulo Example Cycle 6

Instruction status					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4								
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6								
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Bus*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	2	Add1	Yes	SUBD	M(34+R2)	M(45+R3)						
	0	Add2	Yes	ADDD		M(45+R3)	Add1					
		Add3	No									
	10	Mult1	Yes	MULT	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					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4								
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6								
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Bus*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	1	Add1	Yes	SUBD	M(34+R2)	M(45+R3)						
	0	Add2	Yes	ADDD		M(45+R3)	Add1					
		Add3	No									
	9	Mult1	Yes	MULT	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					Execution	Write						
Instruction		j	k	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4	8							
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6								
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	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					*Execution*	*Write*
Instructior		*j*	*k*	*Issue*	*complete*	*Result*
LD	F6	34+	R2	1	3	4
LD	F2	45+	R3	2	5	6
MUL	F0	F2	F4	3		
SUB	F8	F6	F2	4	8	9
DIVD	F10	F0	F6	5		
ADD	F6	F8	F2	6		

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations				*S1*	*S2*	*RS for j*	*RS for k*
Time	*Name*	*Bus*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
0	Add1	No					
0	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
7	Mult1	Yes	MULT	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					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6								
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Bus*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	0	Add1	No									
	2	Add2	Yes	ADDD	M()–M()	M(45+R3)						
		Add3	No									
	6	Mult1	Yes	MULT	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					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6								
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Bus*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	0	Add1	No									
	1	Add2	Yes	ADDD	M()–M()	M(45+R3)						
		Add3	No									
	5	Mult1	Yes	MULT	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					Execution	Write			
Instructior		j	k	Issue	complete	Result		Busy	Address
LD	F6	34+	R2	1	3	4	Load1	No	
LD	F2	45+	R3	2	5	6	Load2	No	
MUL	F0	F2	F4	3			Load3	No	
SUB	F8	F6	F2	4	8	9			
DIVD	F10	F0	F6	5					
ADD	F6	F8	F2	6	12				

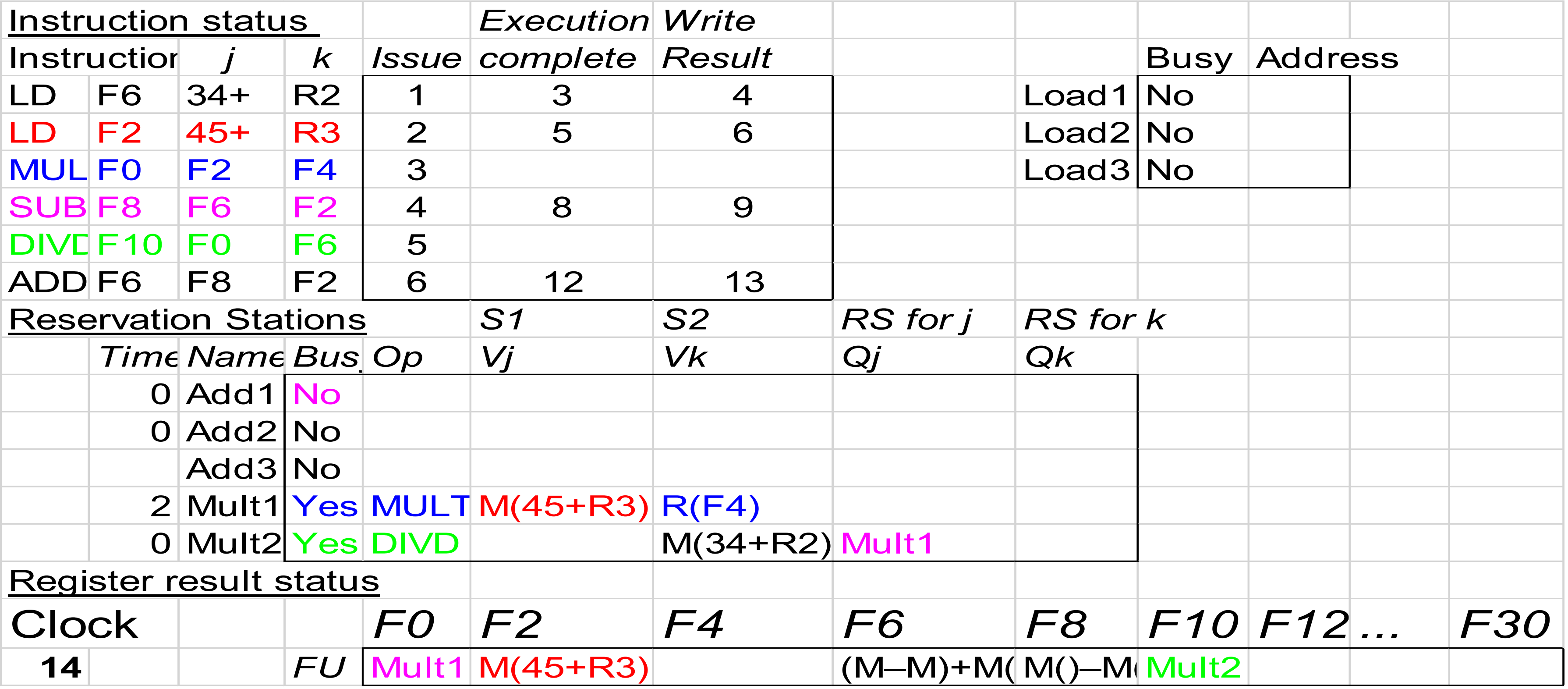
Reservation Stations				S1	S2	RS for j	RS for k
Time	Name	Bus	Op	Vj	Vk	Qj	Qk
0	Add1	No					
0	Add2	Yes	ADDD	M()–M()	M(45+R3)		
	Add3	No					
4	Mult1	Yes	MULT	M(45+R3)	R(F4)		
0	Mult2	Yes	DIVD		M(34+R2)	Mult1	

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					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6	12	13						
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	3	Mult1	Yes	MULT	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					Execution	Write						
Instructior		j	k	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6	12	13						
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Bus	Op	Vj	Vk	Qj	Qk				
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	2	Mult1	Yes	MULT	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					Execution	Write						
Instructior		j	k	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3				Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6	12	13						
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Bus	Op	Vj	Vk	Qj	Qk				
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	1	Mult1	Yes	MULT	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			

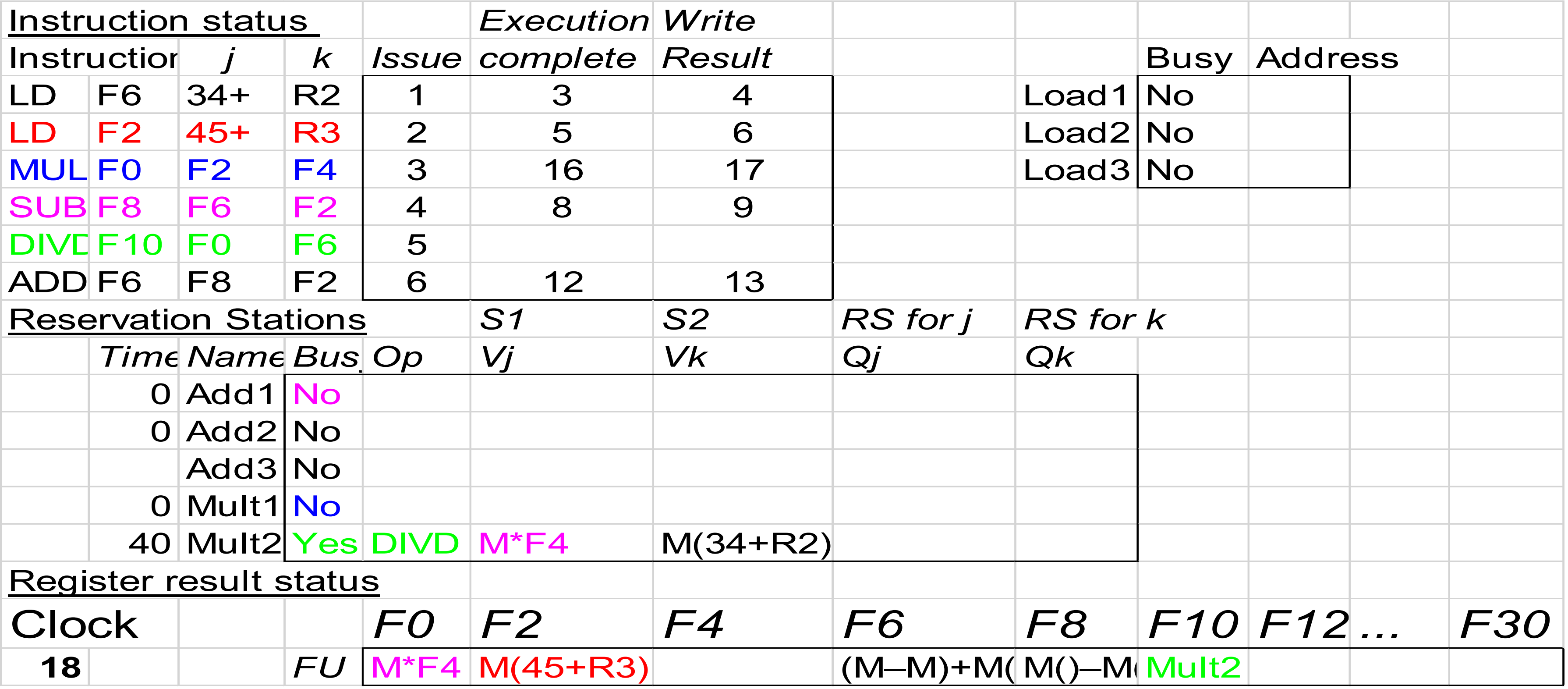
Tomasulo Example Cycle 16

Instruction status					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3	16			Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6	12	13						
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Bus*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	0	Add1	No									
	0	Add2	No									
		Add3	No									
	0	Mult1	Yes	MULT	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					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3	16	17		Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6	12	13						
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3	16	17		Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6	12	13						
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Bus*	*Op*	*Vj*	*Vk*	*Qj*	*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					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3	16	17		Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5								
ADD	F6	F8	F2	6	12	13						
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Bus*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*				
	0	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					Execution	Write						
Instructior		j	k	Issue	complete	Result			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3	16	17		Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5	58							
ADD	F6	F8	F2	6	12	13						
Reservation Stations					S1	S2	RS for j	RS for k				
	Time	Name	Busy	Op	Vj	Vk	Qj	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
58			FU	M*F4	M(45+R3)		(M–M)+M(	M()–M(	Mult2			

Tomasulo Example Cycle 59

Instruction status					*Execution*	*Write*						
Instructior		*j*	*k*	*Issue*	*complete*	*Result*			Busy	Address		
LD	F6	34+	R2	1	3	4		Load1	No			
LD	F2	45+	R3	2	5	6		Load2	No			
MUL	F0	F2	F4	3	16	17		Load3	No			
SUB	F8	F6	F2	4	8	9						
DIVD	F10	F0	F6	5	58	59						
ADD	F6	F8	F2	6	12	13						
Reservation Stations					*S1*	*S2*	*RS for j*	*RS for k*				
	Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*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*
59			*FU*	M*F4	M(45+R3)		(M–M)+M(	M()–M	M*F4/M			

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
- Load have cache misses

Loop Example Cycle 0

Instruction status						Executi	Write					
Instructio		j	k	iteration	Issue	comple	Result		Busy	Address		
LD	F0	0	R1	1				Load1	No			
MUL	F4	F0	F2	1				Load2	No			
SD	F4	0	R1	1				Load3	No		Qi	
LD	F0	0	R1	2				Store1	No			
MUL	F4	F0	F2	2				Store2	No			
SD	F4	0	R1	2				Store3	No			
Reservation Stations					S1	S2	RS fo	RS for k				
	Tim	Name	Bus	Op	Vj	Vk	Qj	Qk	Code:			
	0	Add1	No						LD	F0	0	R1
	0	Add2	No						MUL	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	No						SUB	R1	R1	#8
	0	Mult2	No						BNE	R1	Loop	
Register result status												
Clock		**R1**		F0	F2	F4	F6	F8	F10	F1	…	F30
0		**80**	Qi									

Loop Example Cycle 1

Instruction status						Executi	Write					
Instructio		j	k	iteration	Issue	comple	Result		Busy	Address		
LD	F0	0	R1	1	1			Load1	Yes	80		
MUL	F4	F0	F2	1				Load2	No			
SD	F4	0	R1	1				Load3	No		Qi	
LD	F0	0	R1	2				Store1	No			
MUL	F4	F0	F2	2				Store2	No			
SD	F4	0	R1	2				Store3	No			
Reservation Stations					S1	S2	RS fo	RS for	k			
	Tim	Name	Bus	Op	Vj	Vk	Qj	Qk	Code:			
	0	Add1	No						LD	F0	0	R1
	0	Add2	No						MUL	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	No						SUB	R1	R1	#8
	0	Mult2	No						BNE	R1	Loop	
Register result status												
Clock		R1		F0	F2	F4	F6	F8	F1(	F1	…	F3(
1		80	Qi	Load1								

Loop Example Cycle 2

Instruction status						Executi	Write					
Instructio		j	k	iteration	Issue	comple	Result		Busy	Address		
LD	F0	0	R1	1	1			Load1	Yes	80		
MUL	F4	F0	F2	1	2			Load2	No			
SD	F4	0	R1	1				Load3	No		Qi	
LD	F0	0	R1	2				Store1	No			
MUL	F4	F0	F2	2				Store2	No			
SD	F4	0	R1	2				Store3	No			
Reservation Stations					S1	S2	RS fo	RS for	k			
	Tim	Name	Bus	Op	Vj	Vk	Qj	Qk	Code:			
	0	Add1	No						LD	F0	0	R1
	0	Add2	No						MUL	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	Yes	MULTD		R(F2)	Load1		SUB	R1	R1	#8
	0	Mult2	No						BNE	R1	Loop	
Register result status												
Clock		R1		F0	F2	F4	F6	F8	F1(	F1	…	F3(
2		80	Qi	Load1		Mult1						

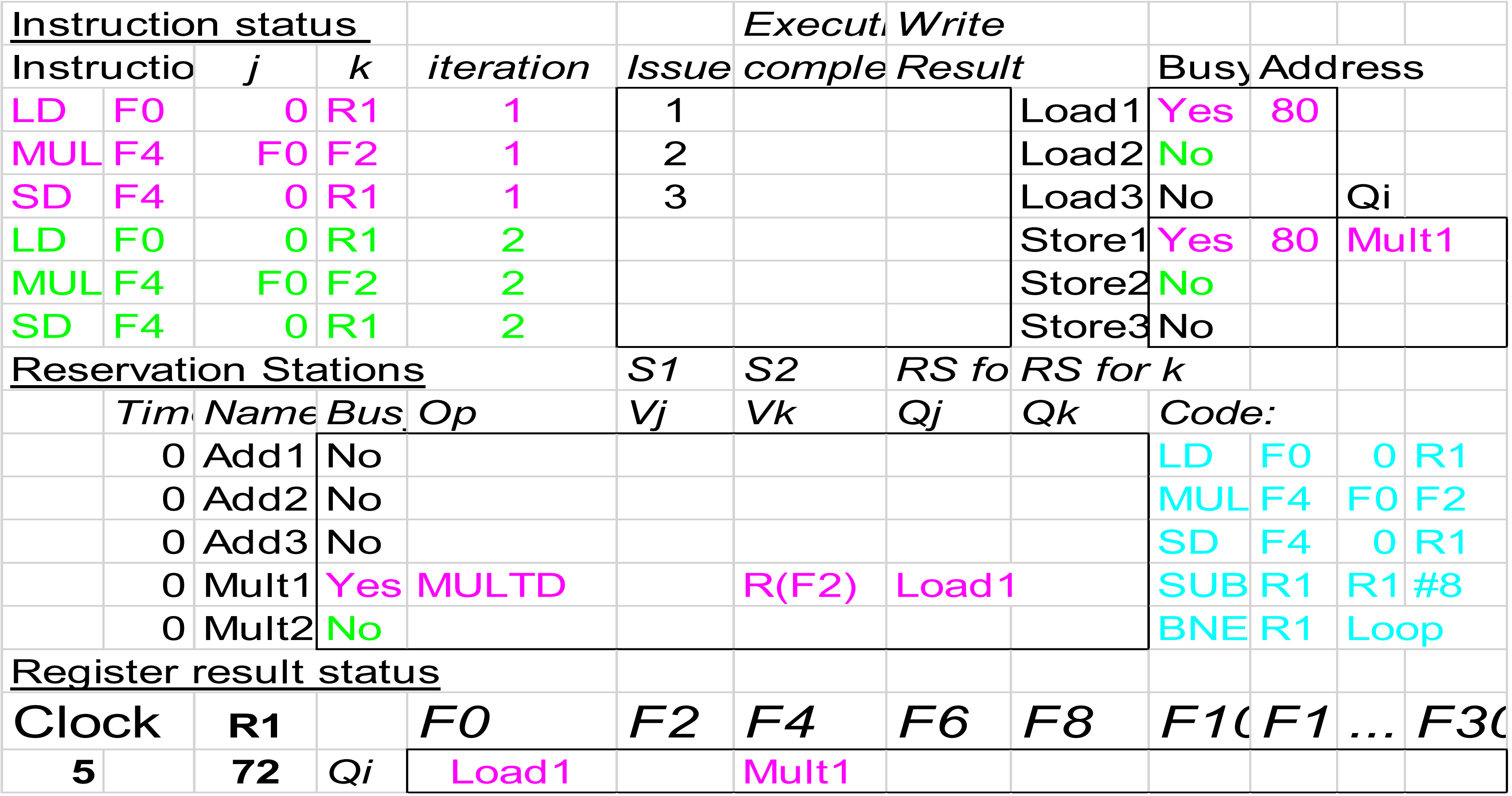
Loop Example Cycle 3

Instruction status						Executi	Write					
Instructio		j	k	iteration	Issue	comple	Result		Busy	Address		
LD	F0	0	R1	1	1			Load1	Yes	80		
MUL	F4	F0	F2	1	2			Load2	No			
SD	F4	0	R1	1	3			Load3	No		Qi	
LD	F0	0	R1	2				Store1	Yes	80	Mult1	
MUL	F4	F0	F2	2				Store2	No			
SD	F4	0	R1	2				Store3	No			
Reservation Stations					S1	S2	RS fo	RS for k				
	Tim	Name	Bus	Op	Vj	Vk	Qj	Qk	Code:			
	0	Add1	No						LD	F0	0	R1
	0	Add2	No						MUL	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	Yes	MULTD		R(F2)	Load1		SUB	R1	R1	#8
	0	Mult2	No						BNE	R1	Loop	
Register result status												
Clock		R1		F0	F2	F4	F6	F8	F1(	F1	…	F3(
3		80	Qi	Load1		Mult1						

Loop Example Cycle 4

Instruction status						Executi	Write					
Instructio		j	k	iteration	Issue	comple	Result		Busy	Address		
LD	F0	0	R1	1	1			Load1	Yes	80		
MUL	F4	F0	F2	1	2			Load2	No			
SD	F4	0	R1	1	3			Load3	No		Qi	
LD	F0	0	R1	2				Store1	Yes	80	Mult1	
MUL	F4	F0	F2	2				Store2	No			
SD	F4	0	R1	2				Store3	No			
Reservation Stations					S1	S2	RS fo	RS for k				
	Tim	Name	Busy	Op	Vj	Vk	Qj	Qk	Code:			
	0	Add1	No						LD	F0	0	R1
	0	Add2	No						MUL	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	Yes	MULTD		R(F2)	Load1		SUB	R1	R1	#8
	0	Mult2	No						BNE	R1	Loop	
Register result status												
Clock		R1		F0	F2	F4	F6	F8	F1(	F1	...	F3(
4		72	Qi	Load1		Mult1						

Loop Example Cycle 5



Instruction status						Executi	Write					
Instructio		j	k	iteration	Issue	comple	Result		Busy	Address		
LD	F0	0	R1	1	1			Load1	Yes	80		
MUL	F4	F0	F2	1	2			Load2	No			
SD	F4	0	R1	1	3			Load3	No		Qi	
LD	F0	0	R1	2				Store1	Yes	80	Mult1	
MUL	F4	F0	F2	2				Store2	No			
SD	F4	0	R1	2				Store3	No			
Reservation Stations					S1	S2	RS fo	RS for k				
	Tim	Name	Bus	Op	Vj	Vk	Qj	Qk	Code:			
	0	Add1	No						LD	F0	0	R1
	0	Add2	No						MUL	F4	F0	F2
	0	Add3	No						SD	F4	0	R1
	0	Mult1	Yes	MULTD		R(F2)	Load1		SUB	R1	R1	#8
	0	Mult2	No						BNE	R1	Loop	
Register result status												
Clock		**R1**		F0	F2	F4	F6	F8	F1(	F1	…	F3(
5		**72**	Qi	Load1		Mult1						

Loop Example Cycle 6

Instruction status					Execution	Write				
Instruction	j	k	iteration	Issue	complete	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				Store2	No		
SD F4	0	R1	2				Store3	No		

Reservation Stations				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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
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	7		
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

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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
7	72	Qi	Load2		Mult2					

Loop Example Cycle 8

Instruction status

					Execution	Write				
Instruction	j	k	iteration	Issue	complete	Result		Busy	Address	Qi
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		
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

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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
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

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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

					Execution	Write
Instruction	j	k	iteration	Issue	complete	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

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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

					Execution	Write				
Instruction	j	k	iteration	Issue	complete	Result		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
MULT F4	F0	F2	1	2			Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1
MULT F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD F4	0	R1	2	8			Store3	No		

Reservation Stations

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

						Execution	Write				
Instruction		j	k	iteration	Issue	complete	Result		Busy	Address	
LD	F0	0	R1	1	1	9	10	Load1	No		
MULT	F4	F0	F2	1	2			Load2	No		
SD	F4	0	R1	1	3			Load3	Yes	64	Qi
LD	F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1
MULT	F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD	F4	0	R1	2	8			Store3	No		

Reservation Stations

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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					Execution	Write				
Instruction	*j*	*k*	*iteration*	*Issue*	*complete*	*Result*		Busy	Address	
LD F0	0	R1	1	1	9	10	Load1	No		
MULT F4	F0	F2	1	2			Load2	No		
SD F4	0	R1	1	3			Load3	Yes	64	Qi
LD F0	0	R1	2	6	10	11	Store1	Yes	80	Mult1
MULT F4	F0	F2	2	7			Store2	Yes	72	Mult2
SD F4	0	R1	2	8			Store3	No		

Reservation Stations				*S1*	*S2*	*RS for j*	*RS for k*
Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
0	Add1	No					
0	Add2	No					
0	Add3	No					
1	Mult1	Yes	MULTD	M(80)	R(F2)		
2	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*
13	**64**	*Qi*			Mult2						

Loop Example Cycle 14

Instruction status					Execution	Write
Instruction	j	k	iteration	Issue	complete	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				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	Yes	MULTD	M(80)	R(F2)		
1	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
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(F
MULT F4	F0	F2	2	7	15		Store2	Yes	72	Mult2
SD F4	0	R1	2	8			Store3	No		

Reservation Stations

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	Qk
0	Add1	No					
0	Add2	No					
0	Add3	No					
0	Mult1	No					
0	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
15	64	Qi			Mult2					

Loop Example Cycle 16

Instruction status						Execution	Write				
Instruction		j	k	iteration	Issue	complete	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(F
MULT	F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(7
SD	F4	0	R1	2	8			Store3	No		

Reservation Stations				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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
16	64	Qi			Mult1						

Loop Example Cycle 17

Instruction status						Execution	Write				
Instruction		j	k	iteration	Issue	complete	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(F
MULT	F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(7
SD	F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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(F
MULT	F4	F0	F2	2	7	15	16	Store2	Yes	72	M(72)*R(7
SD	F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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(7
SD F4	0	R1	2	8			Store3	Yes	64	Mult1

Reservation Stations

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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
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(7
SD F4	0	R1	2	8	20		Store3	Yes	64	Mult1

Reservation Stations

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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
20	**56**	Qi			Mult1					

Loop Example Cycle 21

Instruction status

						Execution	Write				
Instruction		j	k	iteration	Issue	complete	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

				S1	S2	RS for j	RS for k
Time	Name	Busy	Op	Vj	Vk	Qj	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
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