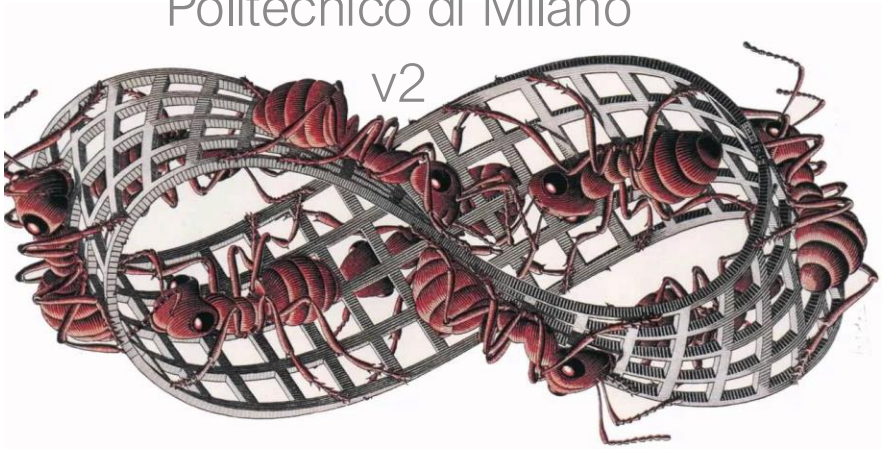
Advanced Computer Architectures
(High Performance Processors and Systems)

Dynamic Scheduling: Tomasulo vs Scoreboard

Politecnico di Milano

v2



Alessandro Verosimile < alessandro.verosimile@polimi.it>
Marco D. Santambrogio <marco.santambrogio@polimi.it>

EXE ON APRIL 23: ONLINE

EXE ON APRIL 23: ONLINE

NO EMAILS RECEIVED
THEREFORE CONFIRMED

Key Idea: dynamic scheduling

- Problem: data dependences that cannot be hidden with bypassing or forwarding cause hardware stalls of the pipeline

- Solution: allow instructions behind a stall to proceed
  - HW rearranges the instruction execution to reduce stalls

- Enables out-of-order execution and completion (commit)
  - Out-of order execution introduces possibility of WAR, WAW data hazards.

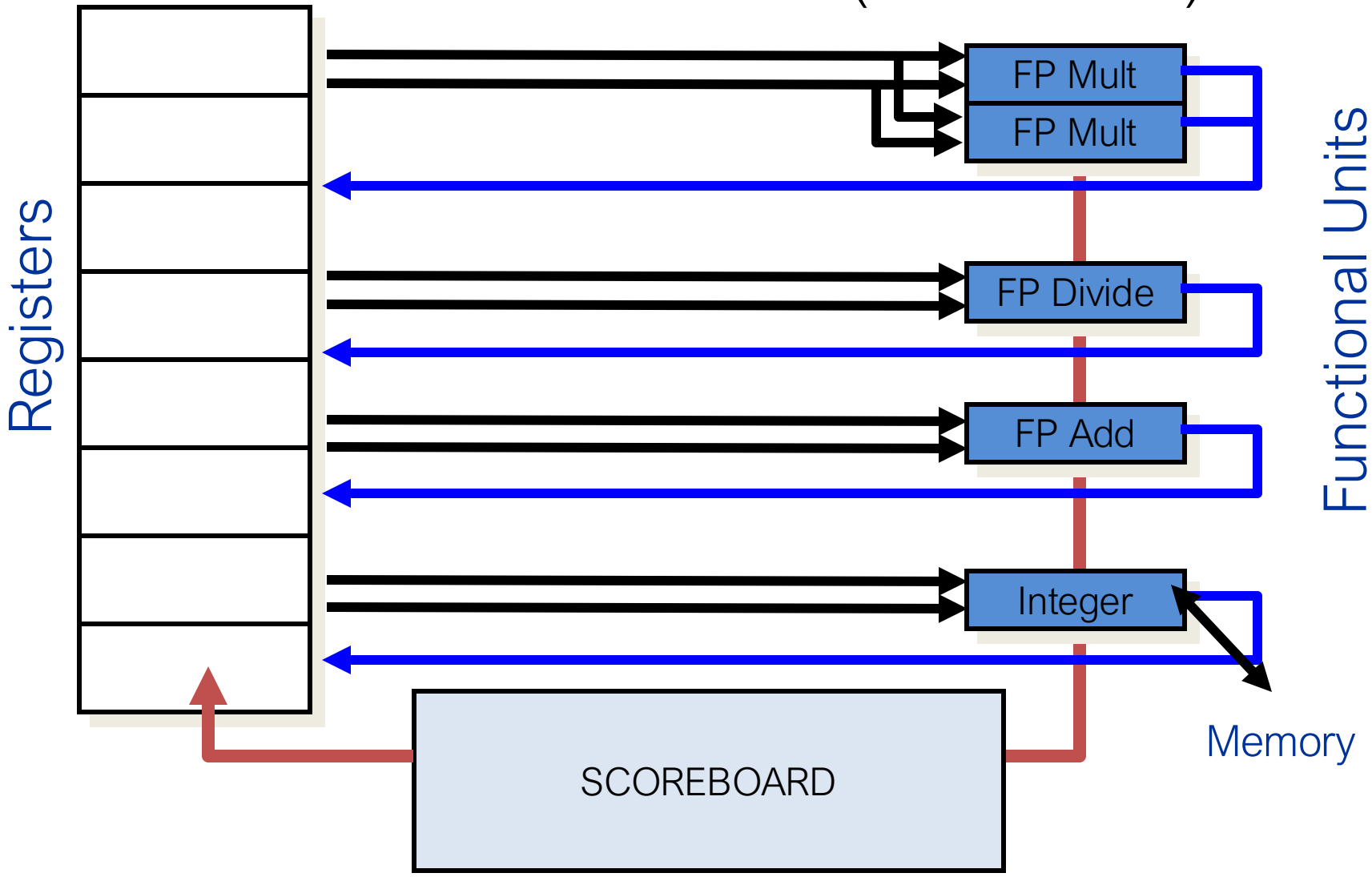
When is it Safe to Issue an Instruction?

Suppose a data structure keeps track of all the instructions in all the functional units

The following checks need to be made before the Issue stage can dispatch an instruction

- Is the required function unit available?
- Is the input data available? → RAW?
- Is it safe to write the destination? → WAR? WAW?
- Is there a structural conflict at the WB stage?

Scoreboard Architecture (CDC 6600)



Execution Process

- Issue
  - Functional unit is free (structural)
  - Active instructions do not have same Rd (WAW)
- Read Operands
  - Checks availability of source operands
  - Resolves RAW hazards dynamically (out-of-order execution)
- Execution
  - Functional unit begins execution when operands arrive
  - Notifies the scoreboard when it has completed execution
- Write result
  - Scoreboard checks WAR hazards
  - Stalls the completing instruction if necessary

Scoreboard structure: three parts

1. Instruction status
2. Functional Unit status

   Indicates the state of the functional unit (FU):

   Busy – Indicates whether the unit is busy or not
   Op - The operation to perform in the unit (+,-, etc.)
   Fi - Destination register
   Fj, Fk – Source register numbers
   Qj, Qk – Functional units producing source registers
   Rj, Rk – Flags indicating when Fj, Fk are ready

3. Register result status

   Indicates which functional unit will write each register.
   Blank if no pending instructions will write that register.

Exercise

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Scoreboard – Clk=1

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist			Issue	Read Op	Exec Co.	Write R.
Mult1													addd	1			
mult2													multd				
Add1	YES	ADD	F0	F2	F4			YES	YES		S1		multd				
Add2													addd				
F0	F2	F4	F6	F8	F10	F12	F14										
ADD1																	

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=2

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist			Issue	Read Op	Exec Co.	Write R.
Mult1	YES	MULT	F2	F6	F8			YES	YES		S2		**addd**	1	2		
mult2	NO												**multd**	2			
Add1	YES	ADD	F0	F2	F4			YES	YES	2	S1		**multd**				
Add2	NO												**addd**				
F0	**F2**	**F4**	**F6**	**F8**	**F10**	**F12**	**F14**										
ADD1	MULT1																

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=3

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	YES	MULT	F2	F6	F8			YES	YES	4	S2
mult2	YES	MULT	F10	F0	F2	ADD1	MULT1	NO	NO		S3
Add1	YES	ADD	F0	F2	F4			NO	NO	1	S1
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
ADD1	MULT1				MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2		
multd	2	3		
multd	3			
addd				

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=4

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	YES	MULT	F2	F6	F8			NO	NO	3	S2
mult2	YES	MULT	F10	F0	F2	ADD1	MULT1	NO	NO		S3
Add1	YES	ADD	F0	F2	F4			NO	NO	0	S1
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
ADD1	MULT1				MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	
multd	2	3		
multd	3			
addd				

WAW

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=5

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	YES	MULT	F2	F6	F8			NO	NO	2	S2
mult2	YES	MULT	F10	F0	F2	✓	MULT1	YES	NO		S3
Add1	NO										
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
	MULT1				MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3		
multd	3			
addd				

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=6

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist			Issue	Read Op	Exec Co.	Write R.
Mult1	YES	MULT	F2	F6	F8			NO	NO	1	S2		**addd**	1	2	4	5
mult2	YES	MULT	F10	F0	F2		MULT1	YES	NO		S3		**multd**	2	3		
Add1	YES	ADD2	F0	F12	F14			YES	YES		S4		**multd**	3			
Add2	NO												**addd**	6			
F0	**F2**	**F4**	**F6**	**F8**	**F10**	**F12**	**F14**										
ADD1	MULT1				MULT2												

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=7

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist			Issue	Read Op	Exec Co.	Write R.
Mult1	YES	MULT	F2	F6	F8			NO	NO	0	S2		**addd**	1	2	4	5
mult2	YES	MULT	F10	F0	F2		MULT1	YES	NO		S3		**multd**	2	3	7	
Add1	YES	ADD2	F0	F12	F14			YES	YES	2	S4		**multd**	3			
Add2	NO												**addd**	6	7		
F0	**F2**	**F4**	**F6**	**F8**	**F10**	**F12**	**F14**										
ADD1	MULT1				MULT2												

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=8

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist			Issue	Read Op	Exec Co.	Write R.
Mult1	NO												addd	1	2	4	5
mult2	YES	MULT	F10	F0	F2		✓	YES	YES		S3		multd	2	3	7	8
Add1	YES	ADD2	F0	F12	F14			NO	NO	1	S4		multd	3			
Add2	NO												addd	6	7		
F0	F2	F4	F6	F8	F10	F12	F14										
ADD1					MULT2												

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=9

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

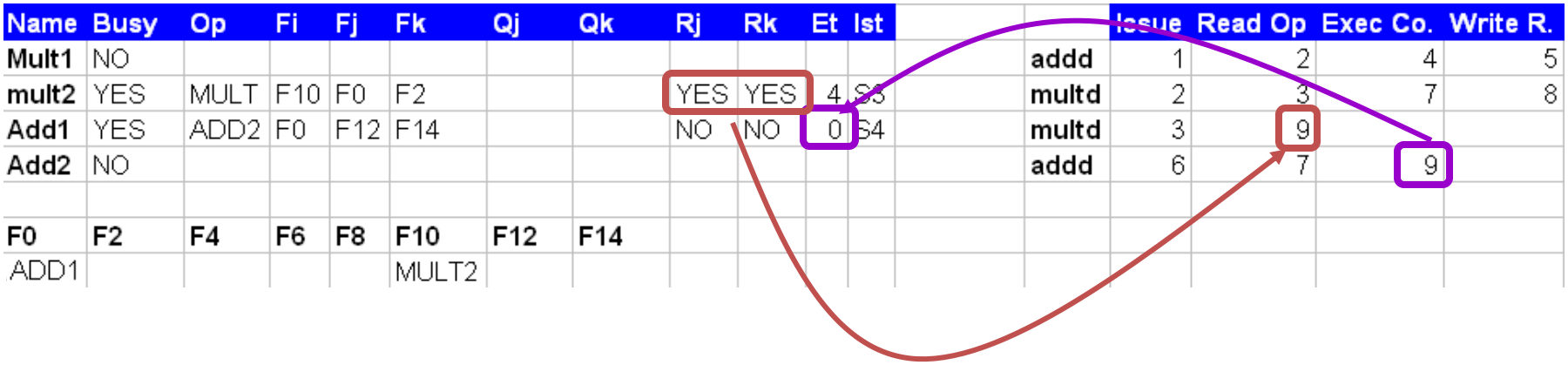
S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2			YES	YES	4	S3
Add1	YES	ADD2	F0	F12	F14			NO	NO	0	S4
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
ADD1					MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9		
addd	6	7	9	



ADD: 2cc

MULT: 4cc

Scoreboard – Clk=10

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2			NO	NO	3	S3
Add1	NO										
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
					MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9		
addd	6	7	9	10

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=11

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist			Issue	Read Op	Exec Co.	Write R.
Mult1	NO												**addd**	1	2	4	5
mult2	YES	MULT	F10	F0	F2			NO	NO	2	S3		**multd**	2	3	7	8
Add1	NO												**multd**	3	9		
Add2	NO												**addd**	6	7	9	10
F0	**F2**	**F4**	**F6**	**F8**	**F10**	**F12**	**F14**										
					MULT2												

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=12

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2			NO	NO	1	S3
Add1	NO										
Add2	NO										
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9		
addd	6	7	9	10

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=13

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2			NO	NO	0	S3
Add1	NO										
Add2	NO										
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9	13	
addd	6	7	9	10

ADD: 2cc

MULT: 4cc

Scoreboard – Clk=14

S1: ADDD F0, F2, F4

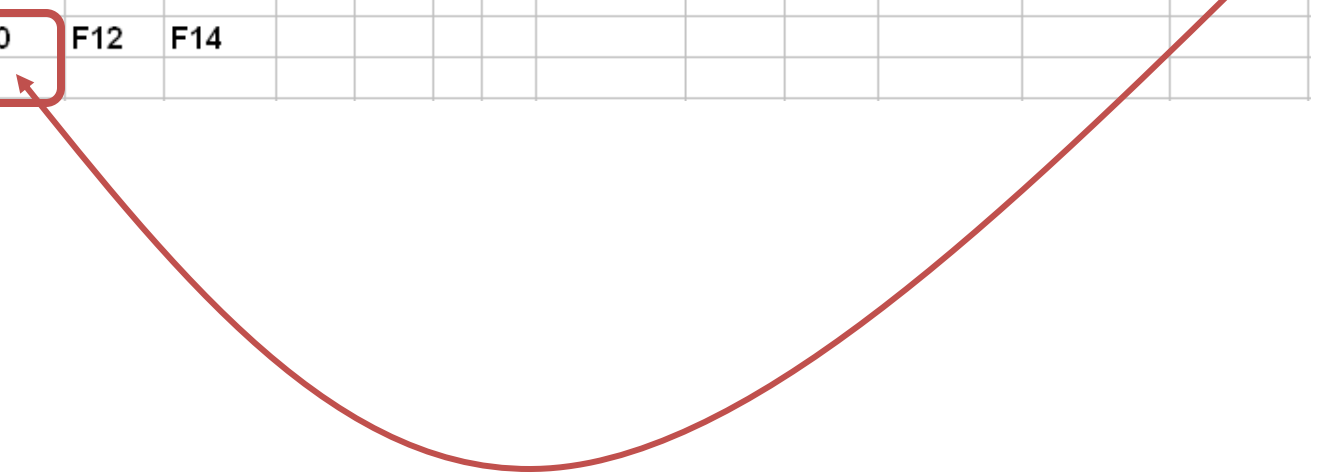
S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	NO										
Add1	NO										
Add2	NO										
F0	F2	F4	F6	F8	F10	F12	F14				

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9	13	14
addd	6	7	9	10



ADD: 2cc

MULT: 4cc

Scoreboard – Clk=14

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	NO										
Add1	NO										
Add2	NO										
F0	F2	F4	F6	F8	F10	F12	F14				

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9	13	14
addd	6	7	9	10

ADD: 2cc

MULT: 4cc

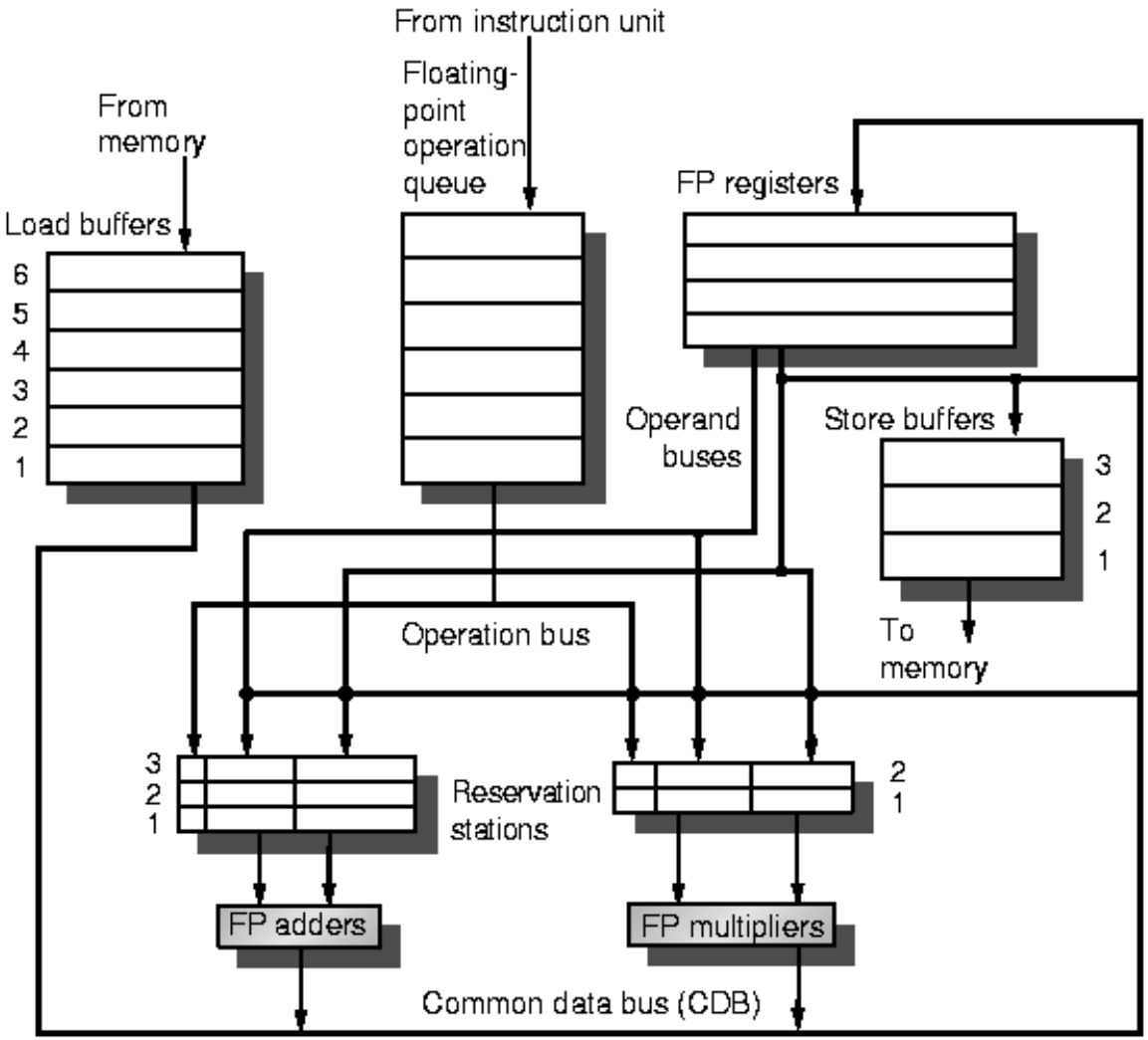
Tomasulo Approach

- Another approach to eliminate stalls
  - Combines scoreboard with
  - Register renaming (to avoid WAR and WAW)

- Designed for the IBM 360/91
  - High FP performance for the whole 360 family
  - Four double precision FP registers
  - Long memory access and long FP delays

- Can support overlapped execution of multiple iterations of a loop

Tomasulo Algorithm for an FPU



Execution Process

- Issue
  - Empty reservation station or buffer
  - Send operands to the reservation station
  - Use name of reservation station for operands
- Execute
  - Execute operation if operands are available
  - Monitor CDB for availability of operands
- Write result
  - When result is available, write it to the CDB

Reservation Station Components

- Tag identifying the RS
- OP = the operation to perform on the component
- Vj, Vk = Value of the source operands
- Qj,Qk = Pointers to RS that produce Vj,Vk
  Zero value = Source op. is already available in Vj or Vk
- Busy = Indicates RS Busy
- Note: Only one of V-field or Q-field is valid for each operand

Tomasulo – Clk=1

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime	
mult1							
mult2							
add1	ADD	R(F2)	R(F4)				
add2							
F0	F2	F4	F6	F8	F10	F12	F14
ADD1							

	Issue	Exec Co.	Write R.
addd	1		
multd			
multd			
addd			

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=2

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1	MULT	R(F6)	R(F8)			
mult2						
add1	ADD	R(F2)	R(F4)			1
add2						

	Issue	Exec Co.	Write R.
addd	1		
multd	2		
multd			
addd			

F0	F2	F4	F6	F8	F10	F12	F14
ADD1	MULT1						

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=3

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1	MULT	R(F6)	R(F8)			3		addd	1	3	
mult2	MULT			ADD1	MULT1			multd	2		
add1	ADD	R(F2)	R(F4)			0		multd	3		
add2								addd			
F0	F2	F4	F6	F8	F10	F12	F14				
ADD1	MULT1				MULT2						

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=4

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1	MULT	R(F6)	R(F8)			2		addd	1	3	4
mult2	MULT	M(A1)		✓	MULT1			multd	2		
add1								multd	3		
add2	ADD	R(F12)	R(F14)					addd	4		
F0	F2	F4	F6	F8	F10	F12	F14				
ADD2	MULT1				MULT2						

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=5

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1	MULT	R(F6)	R(F8)			1		**addd**	1	3	4
mult2	MULT	M(A1)			MULT1			**multd**	2		
add1								**multd**	3		
add2	ADD	R(F12)	R(F14)			1		**addd**	4		
F0	**F2**	**F4**	**F6**	**F8**	**F10**	**F12**	**F14**				
ADD2	MULT1				MULT2						

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=6

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1	MULT	R(F6)	R(F8)			0
mult2	MULT	M(A1)			MU	
add1						
add2	ADD	R(F12)	R(F14)			0

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	
multd	3		
addd	4	6	

F0	F2	F4	F6	F8	F10	F12	F14
ADD2	MULT1				MULT2		

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=7

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime	
mult1	MULT	R(F6)	R(F8)			0	
mult2	MULT	M(A1)			MULT1		
add1							
add2							
F0	F2	F4	F6	F8	F10	F12	F14
	MULT1				MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	
multd	3		
addd	4		7

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=8

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1								addd	1	3	4
mult2	MULT	M(A1)	M(M1)					multd	2	6	8
add1								multd	3		
add2								addd	4	6	7
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=9

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1						
mult2						3
add1						
add2						

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	8
multd	3		
addd	4	6	7

F0	F2	F4	F6	F8	F10	F12	F14
					MULT2		

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=10

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1								addd	1	3	4
mult2						2		multd	2	6	8
add1								multd	3		
add2								addd	4	6	7
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=11

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1								addd	1	3	4
mult2						1		multd	2	6	8
add1								multd	3		
add2								addd	4	6	7
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=12

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1								addd	1	3	4
mult2						0		multd	2	6	8
add1								multd	3	12	
add2								addd	4	6	7
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=13

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1								addd	1	3	4
mult2								multd	2	6	8
add1								multd	3	12	13
add2								addd	4	6	7
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

ADD: 2cc

MULT: 4cc

Scoreboard Vs Tomasulo

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9	13	14
addd	6	7	9	10

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	8
multd	3	12	13
addd	4	6	7

- Can we do better?
  - YES!!!... Lets go back to clk6 with Tomasulo...

https://flic.kr/p/8puKvo

Tomasulo – Clk=6

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1	MULT	R(F6)	R(F8)			0
mult2	MULT	M(A1)			MU	
add1						
add2	ADD	R(F12)	R(F14)			0

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	
multd	3		
addd	4	6	

F0	F2	F4	F6	F8	F10	F12	F14
ADD2	MULT1				MULT2		

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=7

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime		Issue	Exec Co.	Write R.
mult1								addd	1	3	4
mult2	MULT	M(A1)	M(M1)					multd	2	6	7
add1								multd	3		
add2	ADD	R(F12)	R(F14)			0		addd	4	6	

F0	F2	F4	F6	F8	F10	F12	F14
ADD2					MULT2		

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=8

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime	
mult1							
mult2	MULT	M(A1)	M(M1)			3	
add1							
add2							
F0	F2	F4	F6	F8	F10	F12	F14
					MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3		
addd	4	6	8

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=9

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime	
mult1							
mult2	MULT	M(A1)	M(M1)			2	
add1							
add2							
F0	F2	F4	F6	F8	F10	F12	F14
					MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3		
addd	4	6	8

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=10

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1								addd	1	3	4
mult2	MULT	M(A1)	M(M1)			1		multd	2	6	7
add1								multd	3		
add2								addd	4	6	8
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=11

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime	
mult1							
mult2	MULT	M(A1)	M(M1)			0	
add1							
add2							
F0	F2	F4	F6	F8	F10	F12	F14
					MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3	11	
addd	4	6	8

ADD: 2cc

MULT: 4cc

Tomasulo – Clk=12

S1: ADDD F0, F2, F4

S2: MULTD F2, F6, F8

S3: MULTD F10, F0, F2

S4: ADDD F0, F12, F14

Name	Op	Vj	Vk	Qj	Qk	Etime	
mult1							
mult2							
add1							
add2							
F0	F2	F4	F6	F8	F10	F12	F14
					MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3	11	12
addd	4	6	8

ADD: 2cc

MULT: 4cc

TOMASULO VS SCOREBOARD

Scoreboard Vs Tomasulo – Clk=1

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist			Issue	Read Op	Exec Co.	Write R.
Mult1													addd	1			
mult2													multd				
Add1	YES	ADD	F0	F2	F4			YES	YES		S1		multd				
Add2													addd				
F0	F2	F4	F6	F8	F10	F12	F14										
ADD1																	

Name	Op	Vj	Vk	Qj	Qk	Etime			Issue	Exec Co.	Write R.
mult1								addd	1		
mult2								multd			
add1	ADD	R(F2)	R(F4)					multd			
add2								addd			
F0	F2	F4	F6	F8	F10	F12	F14				
ADD1											

Scoreboard Vs Tomasulo – Clk=2

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	YES	MULT	F2	F6	F8			YES	YES		S2
mult2	NO										
Add1	YES	ADD	F0	F2	F4			YES	YES	2	S1
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
ADD1	MULT1						

	Issue	Read Op	Exec Co.	Write R.
addd	1	2		
multd	2			
multd				
addd				

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1	MULT	R(F6)	R(F8)			
mult2						
add1	ADD	R(F2)	R(F4)			1
add2						

F0	F2	F4	F6	F8	F10	F12	F14
ADD1	MULT1						

	Issue	Exec Co.	Write R.
addd	1		
multd	2		
multd			
addd			

Scoreboard Vs Tomasulo – Clk=3

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	YES	MULT	F2	F6	F8			YES	YES	4	S2
mult2	YES	MULT	F10	F0	F2	ADD1	MULT1	NO	NO		S3
Add1	YES	ADD	F0	F2	F4			NO	NO	1	S1
Add2	NO										

	Issue	Read Op	Exec Co.	Write R.
addd	1	2		
multd	2	3		
multd	3			
addd				

F0	F2	F4	F6	F8	F10	F12	F14
ADD1	MULT1				MULT2		

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1	MULT	R(F6)	R(F8)			3
mult2	MULT			ADD1	MULT1	
add1	ADD	R(F2)	R(F4)			0
add2						

	Issue	Exec Co.	Write R.
addd	1	3	
multd	2		
multd	3		
addd			

Scoreboard Vs Tomasulo – Clk=4

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	YES	MULT	F2	F6	F8			NO	NO	3	S2
mult2	YES	MULT	F10	F0	F2	ADD1	MULT1	NO	NO		S3
Add1	YES	ADD	F0	F2	F4			NO	NO	0	S1
Add2	NO										

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	
multd	2	3		
multd	3			
addd				

F0	F2	F4	F6	F8	F10	F12	F14
ADD1	MULT1				MULT2		

WAW

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1	MULT	R(F6)	R(F8)			2
mult2	MULT	M(A1)		✓	MULT1	
add1						
add2	ADD	R(F12)	R(F14)			

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2		
multd	3		
addd	4		

F0	F2	F4	F6	F8	F10	F12	F14
ADD2	MULT1				MULT2		

Scoreboard Vs Tomasulo – Clk=5

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	YES	MULT	F2	F6	F8			NO	NO	2	S2
mult2	YES	MULT	F10	F0	F2	✓	MULT1	YES	NO		S3
Add1	NO										
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
	MULT1				MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3		
multd	3			
addd				

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1	MULT	R(F6)	R(F8)			1
mult2	MULT	M(A1)			MULT1	
add1						
add2	ADD	R(F12)	R(F14)			1

F0	F2	F4	F6	F8	F10	F12	F14
ADD2	MULT1				MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2		
multd	3		
addd	4		

Scoreboard Vs Tomasulo – Clk=6

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	YES	MULT	F2	F6	F8			NO	NO	1	S2
mult2	YES	MULT	F10	F0	F2		MULT1	YES	NO		S3
Add1	YES	ADD2	F0	F12	F14			YES	YES		S4
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
ADD1	MULT1				MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3		
multd	3			
addd	6			

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1	MULT	R(F6)	R(F8)			0
mult2	MULT	M(A1)			MU	
add1						
add2	ADD	R(F12)	R(F14)			0

F0	F2	F4	F6	F8	F10	F12	F14
ADD2	MULT1				MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	
multd	3		
addd	4	6	

Scoreboard Vs Tomasulo – Clk=7

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Jst
Mult1	YES	MULT	F2	F6	F8			NO	NO	0	S2
mult2	YES	MULT	F10	F0	F2		MULT1	YES	NO		S3
Add1	YES	ADD2	F0	F12	F14			YES	YES	2	S4
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
ADD1	MULT1				MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	
multd	3			
addd	6	7		

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1						
mult2	MULT	M(A1)	M(M1)			
add1						
add2	ADD	R(F12)	R(F14)			0

F0	F2	F4	F6	F8	F10	F12	F14
ADD2					MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3		
addd	4	6	

Scoreboard Vs Tomasulo – Clk=8

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2		✓	YES	YES		S3
Add1	YES	ADD2	F0	F12	F14			NO	NO	1	S4
Add2	NO										

F0	F2	F4	F6	F8	F10	F12	F14
ADD2					MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3			
addd	6	7		

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1						
mult2	MULT	M(A1)	M(M1)			3
add1						
add2						

F0	F2	F4	F6	F8	F10	F12	F14
					MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3		
addd	4	6	8

Scoreboard Vs Tomasulo – Clk=9

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2			YES	YES	4	S3
Add1	YES	ADD2	F0	F12	F14			NO	NO	0	S4
Add2	NO										

	F0	F2	F4	F6	F8	F10	F12	F14
ADD2						MULT2		

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9		
addd	6	7	9	

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1						
mult2	MULT	M(A1)	M(M1)			2
add1						
add2						

	F0	F2	F4	F6	F8	F10	F12	F14
						MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3		
addd	4	6	8

Scoreboard Vs Tomasulo – Clk=10

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2			NO	NO	3	S3
Add1	NO										
Add2	NO										
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9		
addd	6	7	9	10

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1						
mult2	MULT	M(A1)	M(M1)			1
add1						
add2						
F0	F2	F4	F6	F8	F10	F12 F14
					MULT2	

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3		
addd	4	6	8

Scoreboard Vs Tomasulo – Clk=11

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2			NO	NO	2	S3
Add1	NO										
Add2	NO										
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9		
addd	6	7	9	10

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1						
mult2	MULT	M(A1)	M(M1)			0
add1						
add2						
F0	F2	F4	F6	F8	F10	F12
					MULT2	

F14

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3	11	
addd	4	6	8

Scoreboard Vs Tomasulo – Clk=12

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2			NO	NO	1	S3
Add1	NO										
Add2	NO										
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9		
addd	6	7	9	10

Name	Op	Vj	Vk	Qj	Qk	Etime
mult1						
mult2						
add1						
add2						
F0	F2	F4	F6	F8	F10	F12	F14
					MULT2		

	Issue	Exec Co.	Write R.
addd	1	3	4
multd	2	6	7
multd	3	11	12
addd	4	6	8

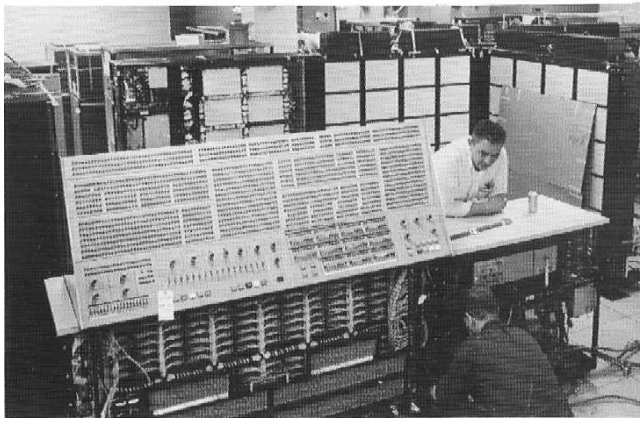
Scoreboard Vs Tomasulo – Clk=13

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	YES	MULT	F10	F0	F2			NO	NO	0	S3
Add1	NO										
Add2	NO										
F0	F2	F4	F6	F8	F10	F12	F14				
					MULT2						

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9	13	
addd	6	7	9	10

Tomasulo's waiting...



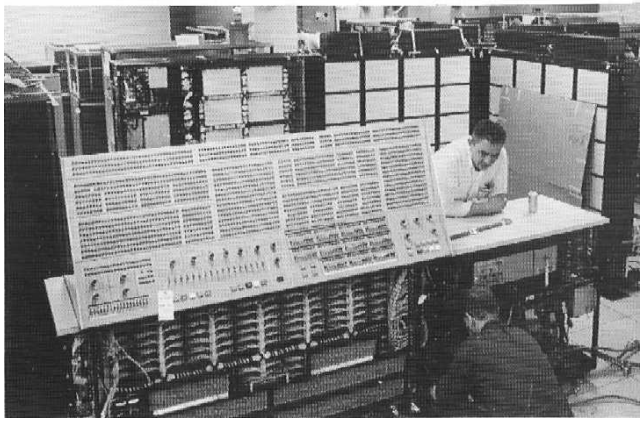
Scoreboard Vs Tomasulo – Clk=14

S1: ADDD F0, F2, F4
S2: MULTD F2, F6, F8
S3: MULTD F10, F0, F2
S4: ADDD F0, F12, F14

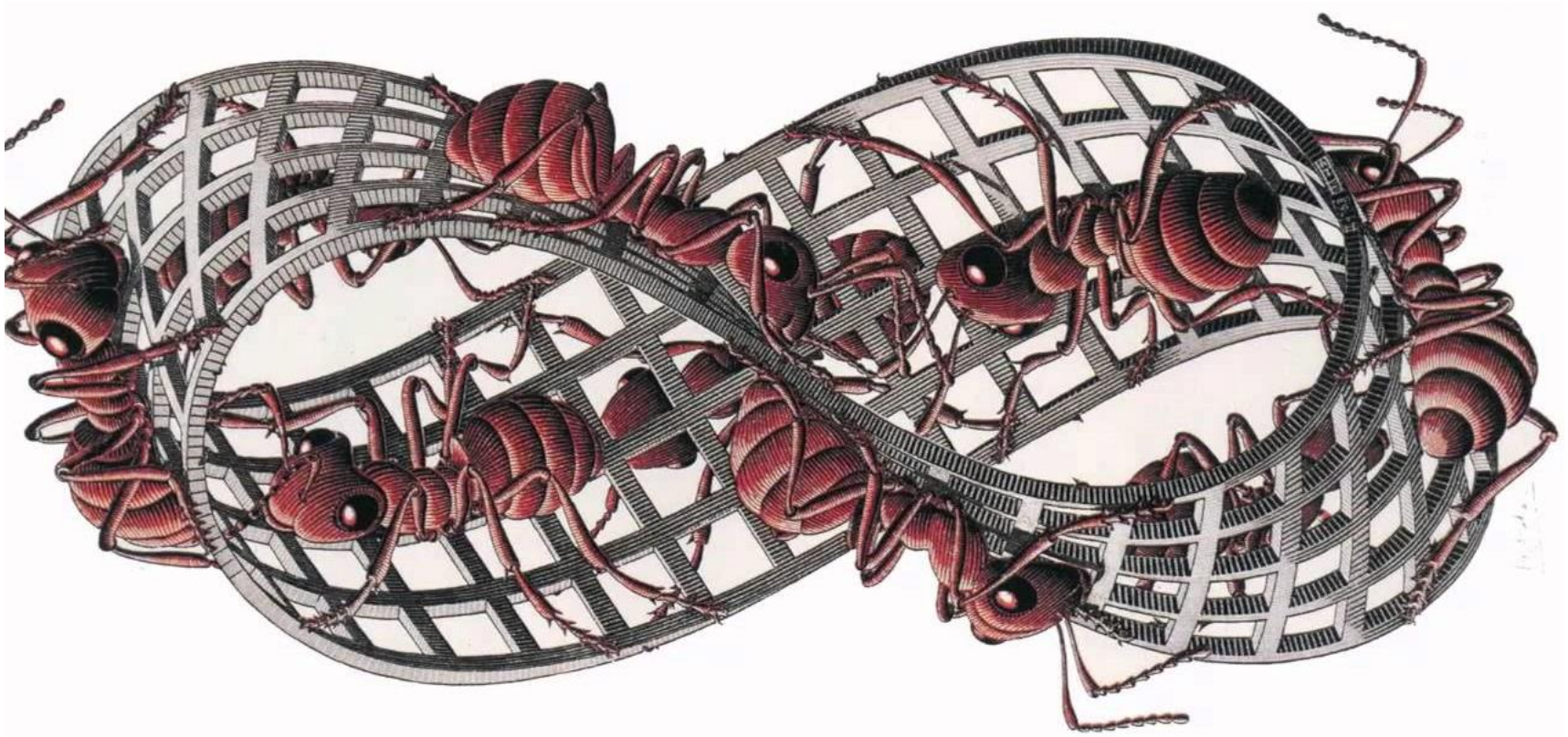
Name	Busy	Op	Fi	Fj	Fk	Qj	Qk	Rj	Rk	Et	Ist
Mult1	NO										
mult2	NO										
Add1	NO										
Add2	NO										
F0	F2	F4	F6	F8	F10	F12	F14				

	Issue	Read Op	Exec Co.	Write R.
addd	1	2	4	5
multd	2	3	7	8
multd	3	9	13	14
addd	6	7	9	10

Tomasulo's now sleeping...



Dynamic Scheduling: Tomasulo and Loops



Tomasulo Loop Example

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

- Assume Multiply takes 4 clocks
- Assume first load takes 8 clocks (cache miss), second load takes 1 clock (hit)
- To be clear, will show clocks for SUBI, BNEZ
- Reality: integer instructions ahead

Loop Example

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1			
1	MULTD	F4	F0	F2			
1	SD	F4	0	R1			
2	LD	F0	0	R1			
2	MULTD	F4	F0	F2			
2	SD	F4	0	R1			

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations:

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

Code:

LD	F0	0	R1
MULTD	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**	*Fu*									

Rename Table!

Loop Example Cycle 1

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1		
1	MULTD	F4	F0	F2			
1	SD	F4	0	R1			
2	LD	F0	0	R1			
2	MULTD	F4	F0	F2			
2	SD	F4	0	R1			

	Busy	*Addr*	*Fu*
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations:

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

Code:

```
LD     F0  0     R1
MULTD  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**	*Fu*	Load1								

Loop Example Cycle 2

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1		
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1			
2	LD	F0	0	R1			
2	MULTD	F4	F0	F2			
2	SD	F4	0	R1			

	Busy	*Addr*	*Fu*
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	No		
Store2	No		
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*S1 Vk*	*S2 Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd		R(F2)	Load1	
	Mult2	No					

Code:

LD	F0	0	R1
MULTD	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**	*Fu*	Load1		Mult1						

Loop Example Cycle 3

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1		
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1			
2	MULTD	F4	F0	F2			
2	SD	F4	0	R1			

	Busy	*Addr*	*Fu*
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*S1 Vk*	*S2 Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd		R(F2)	Load1	
	Mult2	No					

Code:

```
LD    F0  0  R1
MULTD 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**	*Fu*	Load1		Mult1						

Implicit renaming sets up “DataFlow” graph

Loop Example Cycle 3

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1		
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1			
2	MULTD	F4	F0	F2			
2	SD	F4	0	R1			

	Busy	*Addr*	*Fu*
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd		R(F2)	Load1	
	Mult2	No					

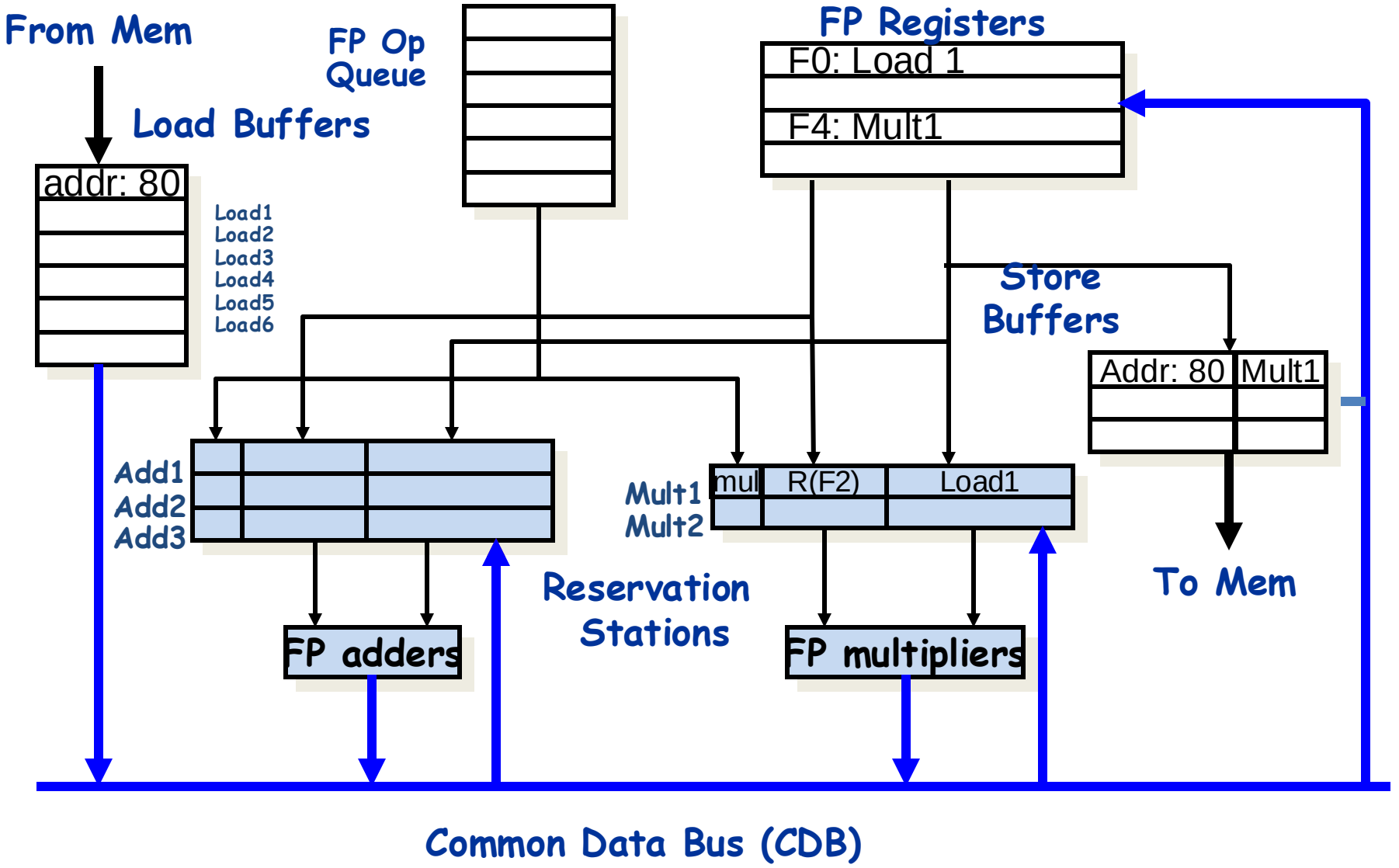
Code:

LD	F0	0	R1
MULTD	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**	*Fu*	Load1		Mult1						

Implicit renaming sets up “DataFlow” graph



What does this mean physically?

Loop Example Cycle 4

Instruction status:

ITER	Instruction		j	k	Issue	Exec Comp	Write Result
1	LD	F0	0	R1	1		
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1			
2	MULTD	F4	F0	F2			
2	SD	F4	0	R1			

	Busy	Addr	Fu
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations:

Time	Name	Busy	Op	Vj	Vk	Qj	Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd		R(F2)	Load1	
	Mult2	No					

S1 S2 RS

Code:

```
LD    F0  0  R1
MULTD 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	80	Fu	Load1		Mult1						

Dispatching SUBI Instruction…

Loop Example Cycle 5

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1		
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1			
2	MULTD	F4	F0	F2			
2	SD	F4	0	R1			

	Busy	*Addr*	*Fu*
Load1	Yes	80	
Load2	No		
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*S1 Vk*	*S2 Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd		R(F2)	Load1	
	Mult2	No					

Code:

```
LD    F0  0   R1
MULTD 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**	*Fu*	Load1		Mult1						

… and, BNEZ instruction

Loop Example Cycle 6

Instruction status:

ITER	Instruction		j	k	Issue	Exec Comp	Write Result
1	LD	F0	0	R1	1		
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6		
2	MULTD	F4	F0	F2			
2	SD	F4	0	R1			

	Busy	Addr	Fu
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 Qj	RS Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd	R(F2)	Load1		
	Mult2	No					

Code:

```
LD    F0  0   R1
MULTD 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	Fu	Load2		Mult1						

Notice that F0 never sees Load from location 80

Loop Example Cycle 7

Instruction status:

ITER	Instruction		j	k	Issue	Exec Comp	Write Result
1	LD	F0	0	R1	1		
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6		
2	MULTD	F4	F0	F2	7		
2	SD	F4	0	R1			

	Busy	Addr	Fu
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	No		
Store3	No		

Reservation Stations:

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

Code:

```
LD    F0  0   R1
MULTD 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	Fu	Load2		Mult2						

Register file completely detached from iteration 1

Loop Example Cycle 8

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1		
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6		
2	MULTD	F4	F0	F2	7		
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*S1 Vk*	*S2 Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd		R(F2)	Load1	
	Mult2	Yes	Multd		R(F2)	Load2	

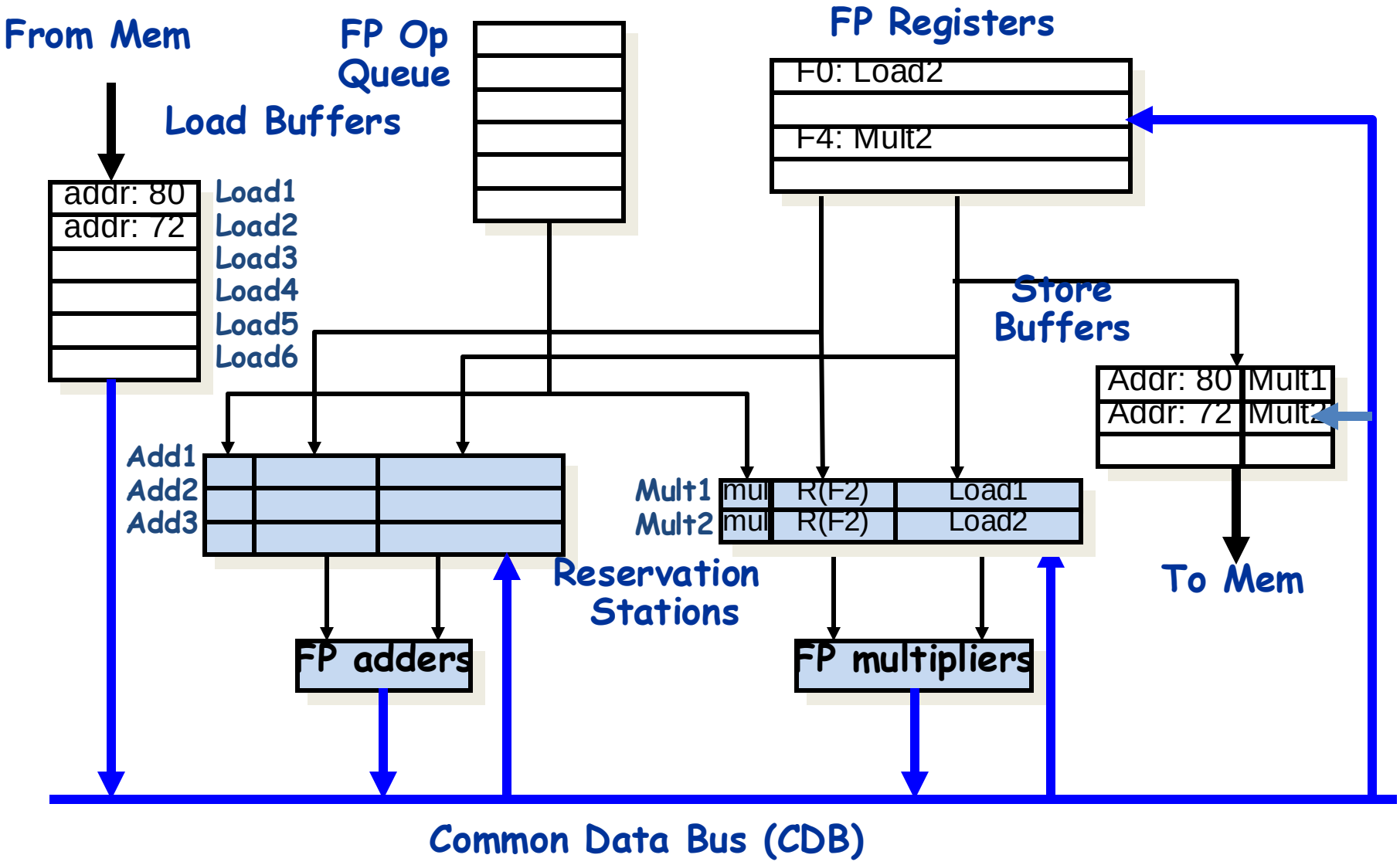
Code:

LD	F0	0	R1
MULTD	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**	*Fu*	Load2		Mult2						

First and Second iteration completely overlapped



What does this mean physically?

Loop Example Cycle 9

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6		
2	MULTD	F4	F0	F2	7		
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
Load1	Yes	80	
Load2	Yes	72	
Load3	No		
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*S1 Qj*	*S2 Qk*	*RS*
	Add1	No						
	Add2	No						
	Add3	No						
	Mult1	Yes	Multd		R(F2)	Load1		
	Mult2	Yes	Multd		R(F2)	Load2		

Code:

```
LD    F0 0  R1
MULTD 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	**72**	*Fu*	Load2		Mult2						

Loop Example Cycle 9

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6		
2	MULTD	F4	F0	F2	7		
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
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 Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd	R(F2)	Load1		
	Mult2	Yes	Multd	R(F2)	Load2		

Code:

```
LD    F0 0  R1
MULTD 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	**72**	*Fu*	Load2		Mult2						

Load1 completing: who is waiting? Note: Dispatching SUBI

Loop Example Cycle 10

Instruction status:

ITER	Instruction		j	k	Issue	Exec Comp	Write Result
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6	10	
2	MULTD	F4	F0	F2	7		
2	SD	F4	0	R1	8		

	Busy	Addr	Fu
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	S1 Vk	S2 Qj	S2 Qk	RS
	Add1	No						
	Add2	No						
	Add3	No						
4	Mult1	Yes	Multd	M[80]	R(F2)			
	Mult2	Yes	Multd		R(F2)	Load2		

Code:

LD	F0	0	R1
MULTD	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	Fu	Load2		Mult2						

Loop Example Cycle 11

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7		
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
3	Mult1	Yes	Multd	M[80]	R(F2)		
4	Mult2	Yes	Multd	M[72]	R(F2)		

Code:

```
LD    F0  0  R1
MULTD 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**	*Fu*	Load3		Mult2						

Next load in sequence

Loop Example Cycle 12

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7		
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*S1 Vk*	*S2 Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
2	Mult1	Yes	Multd	M[80]	R(F2)		
3	Mult2	Yes	Multd	M[72]	R(F2)		

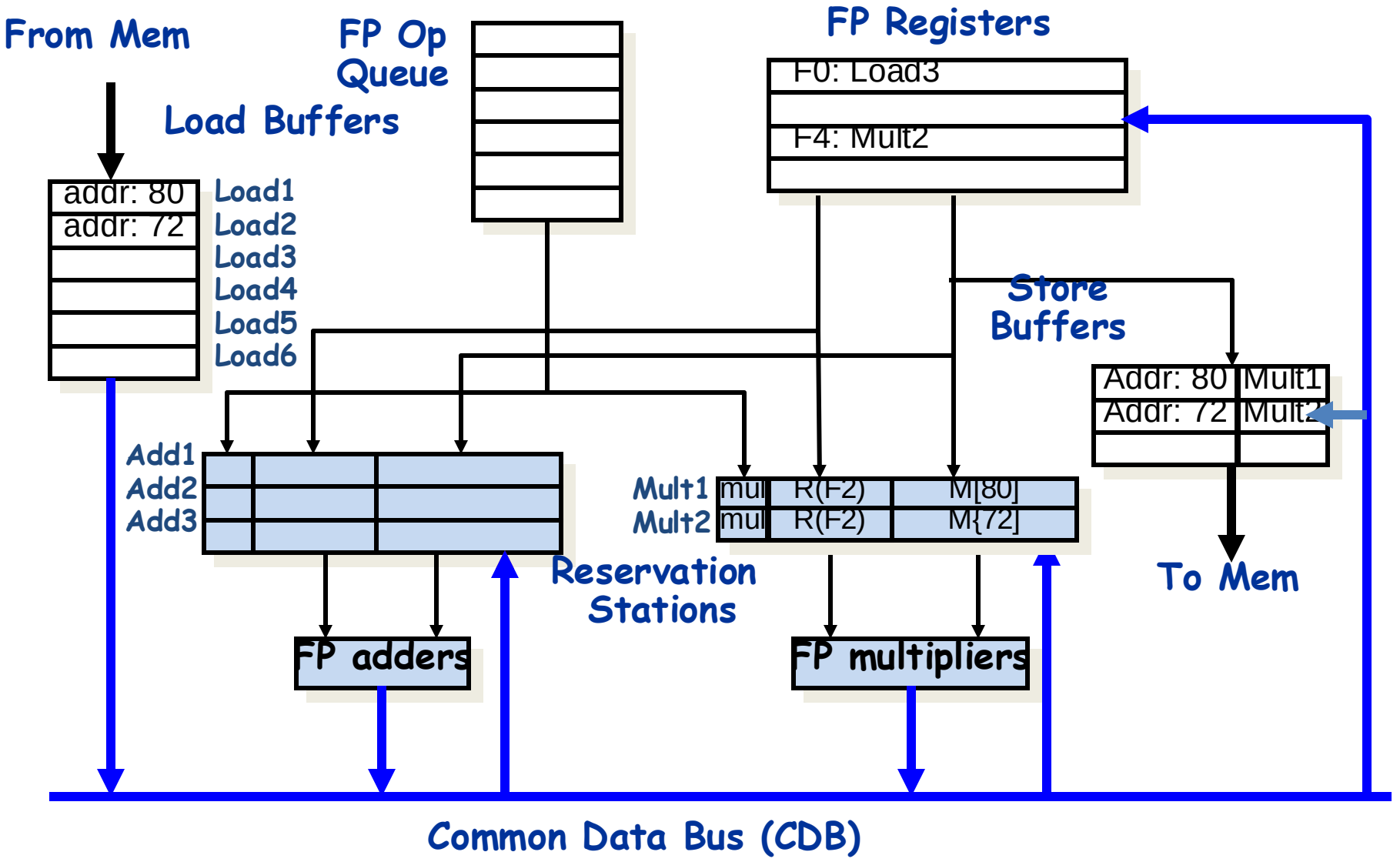
Code:

```
LD    F0  0  R1
MULTD 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**	*Fu*	Load3		Mult2						

Why not issue third multiply?



What does this mean physically?

Loop Example Cycle 13

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2		
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7		
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
1	Mult1	Yes	Multd	M[80]	R(F2)		
2	Mult2	Yes	Multd	M[72]	R(F2)		

Code:

```
LD    F0  0   R1
MULTD 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**	*Fu*	Load3		Mult2						

Loop Example Cycle 14

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2	14	
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7		
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	Yes	80	Mult1
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*S1 Vk*	*S2 Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
0	Mult1	Yes	Multd	M[80]	R(F2)		
1	Mult2	Yes	Multd	M[72]	R(F2)		

Code:

LD	F0	0	R1
MULTD	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**	*Fu*	Load3		Mult2						

Mult1 completing. Who is waiting?

Loop Example Cycle 15

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2	14	15
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7	15	
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	Yes	80	[80]*R2
Store2	Yes	72	Mult2
Store3	No		

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*S1 Vk*	*S2 Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	No					
0	Mult2	Yes	Multd	M[72]	R(F2)		

Code:

LD	F0	0	R1
MULTD	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**	*Fu*	Load3		Mult2						

Mult2 completing. Who is waiting?

Loop Example Cycle 16

Instruction status:

ITER	Instruction		j	k	Issue	Exec Comp	Write Result
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2	14	15
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7	15	16
2	SD	F4	0	R1	8		

	Busy	Addr	Fu
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	Yes	80	[80]*R2
Store2	Yes	72	[72]*R2
Store3	No		

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd	R(F2)	Load3		
	Mult2	No					

Code:

```
LD    F0  0  R1
MULTD 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	Fu	Load3		Mult1						

Loop Example Cycle 17

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2	14	15
1	SD	F4	0	R1	3		
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7	15	16
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	Yes	80	[80]*R2
Store2	Yes	72	[72]*R2
Store3	Yes	64	Mult1

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*S1 Vk*	*S2 Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd		R(F2)	Load3	
	Mult2	No					

Code:

```
LD    F0  0  R1
MULTD 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**	*Fu*	Load3		Mult1						

Loop Example Cycle 18

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2	14	15
1	SD	F4	0	R1	3	18	
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7	15	16
2	SD	F4	0	R1	8		

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	Yes	80	[80]*R2
Store2	Yes	72	[72]*R2
Store3	Yes	64	Mult1

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*Vk*	*S1 Qj*	*S2 Qk*	*RS*
	Add1	No						
	Add2	No						
	Add3	No						
	Mult1	Yes	Multd		R(F2)	Load3		
	Mult2	No						

Code:

LD	F0	0	R1
MULTD	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	**64**	*Fu*	Load3		Mult1						

Loop Example Cycle 19

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2	14	15
1	SD	F4	0	R1	3	18	19
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7	15	16
2	SD	F4	0	R1	8	19	

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	No		
Store2	Yes	72	[72]*R2
Store3	Yes	64	Mult1

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*S1 Vj*	*S2 Vk*	*RS Qj*	*Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd		R(F2)	Load3	
	Mult2	No					

Code:

```
LD    F0  0   R1
MULTD 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	**64**	*Fu*	Load3		Mult1						

Loop Example Cycle 20

Instruction status:

ITER	Instruction		*j*	*k*	*Issue*	*Exec Comp*	*Write Result*
1	LD	F0	0	R1	1	9	10
1	MULTD	F4	F0	F2	2	14	15
1	SD	F4	0	R1	3	18	19
2	LD	F0	0	R1	6	10	11
2	MULTD	F4	F0	F2	7	15	16
2	SD	F4	0	R1	8	19	20

	Busy	*Addr*	*Fu*
Load1	No		
Load2	No		
Load3	Yes	64	
Store1	No		
Store2	No		
Store3	Yes	64	Mult1

Reservation Stations:

Time	*Name*	*Busy*	*Op*	*Vj*	*S1 Vk*	*S2 Qj*	*RS Qk*
	Add1	No					
	Add2	No					
	Add3	No					
	Mult1	Yes	Multd		R(F2)	Load3	
	Mult2	No					

Code:

```
LD    F0  0   R1
MULTD 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	**64**	*Fu*	Load3		Mult1						

Why can Tomasulo overlap iterations of loops?

- Register renaming
  - Multiple iterations use different physical destinations for registers (dynamic loop unrolling).
  - Replace static register names from code with dynamic register "pointers"
  - Effectively increases size of register file
  - Permit instruction issue to advance past integer control flow operations.
- Crucial: integer unit must "get ahead" of floating point unit so that we can issue multiple iterations

  ⇒ Branch Prediction

Other idea: Tomasulo building "DataFlow" graph.

Tomasulo Drawbacks

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

Summary

- HW exploiting ILP
  - Works when can't know dependence at compile time.
  - Code for one machine runs well on another
- Reservations stations: renaming to larger set of registers + buffering source operands
  - Prevents registers as bottleneck
  - Avoids WAR, WAW hazards of Scoreboard
  - Allows loop unrolling in HW
- Not limited to basic blocks (integer units gets ahead, beyond branches)
- Lasting Contributions
  - Dynamic scheduling
  - Register renaming
  - Load/store disambiguation

Advanced Computer Architectures

(High Performance Processors and Systems)

Dynamic Scheduling: Tomasulo vs Scoreboard

Politecnico di Milano

v1