OUT-OF-ORDER EXECUTION

Mahdi Nazm Bojnordi

Assistant Professor

School of Computing

University of Utah

Overview

- Announcement
  - Homework 3 submission deadline: Oct. 2nd
- This lecture
  - Tomasulo algorithm
    - Three-step OoO scheduling
    - Hardware implementation
    - Four-step algorithm
    - Reorder buffer

Recall: Dynamic Scheduling

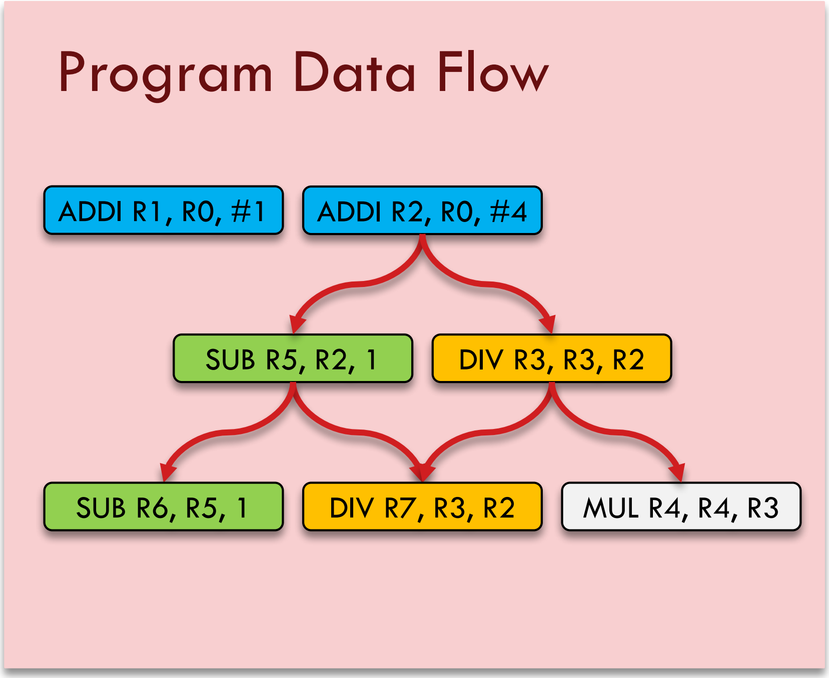
- The main idea is to issue dynamic instructions out of program order while maintaining data flow

Program Code

```
ADDI R1, R0, #1
ADDI R2, R0, #4
DIV R3, R3, R2
SUB R5, R2, 1
MUL R4, R4, R3
DIV R7, R3, R2
SUB R6, R5, 1
```

Recall: Dynamic Scheduling

- The main idea is to issue dynamic instructions out of program order while maintaining data flow



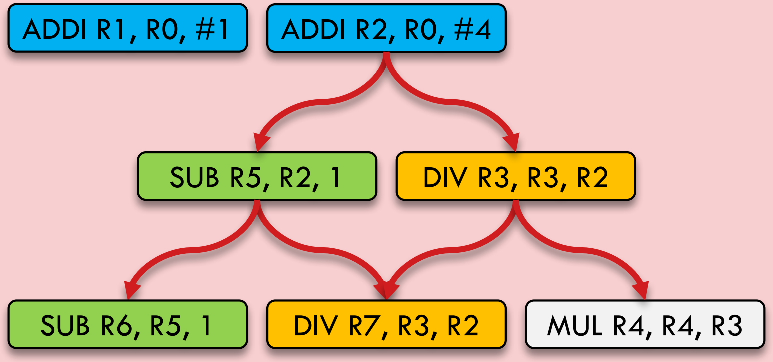
Program Code

```
ADDI R1, R0, #1
ADDI R2, R0, #4
DIV R3, R3, R2
SUB R5, R2, 1
MUL R4, R4, R3
DIV R7, R3, R2
SUB R6, R5, 1
```

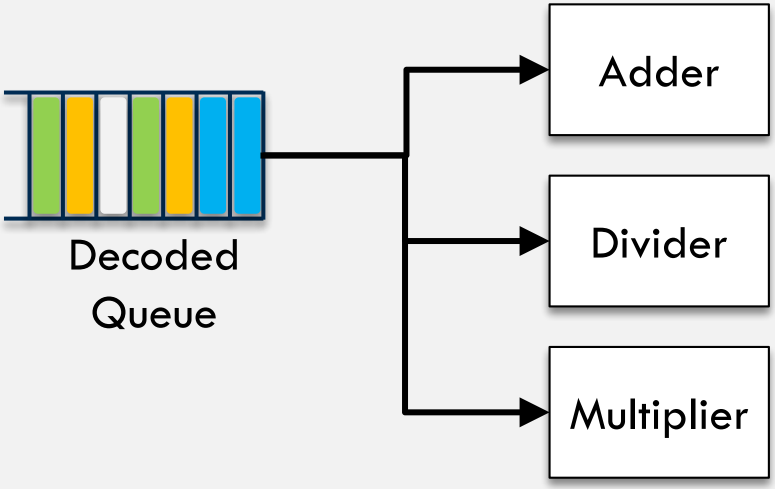
Recall: Dynamic Scheduling

- Instructions are stored in the decoded queue (or instruction queue) prior to execution.

Program Data Flow

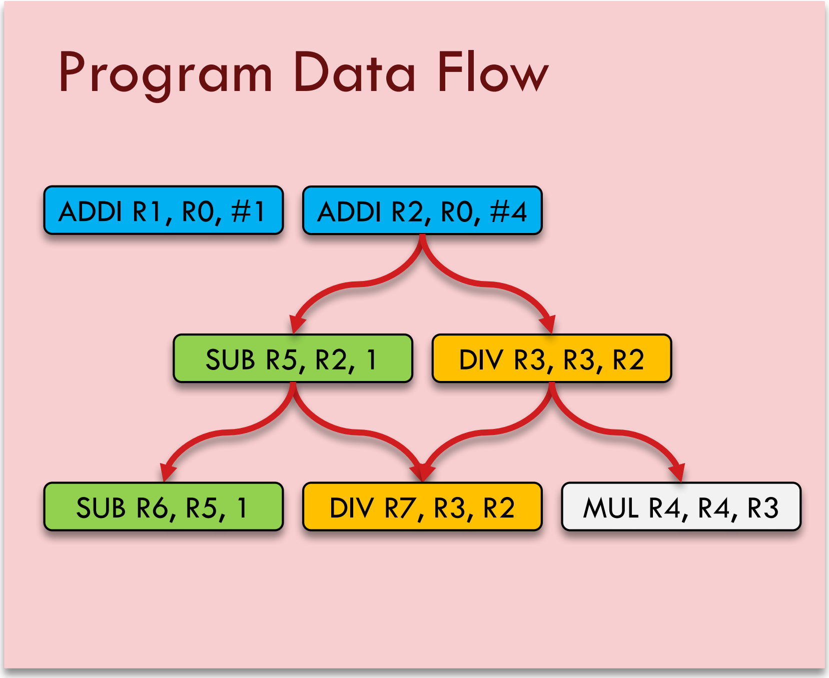


Functional Units



Dynamic Scheduling

- Dispatch decoded instructions to functional units
  - Wait until ready to go for execution



Program Data Flow

ADDI R1, R0, #1

ADDI R2, R0, #4

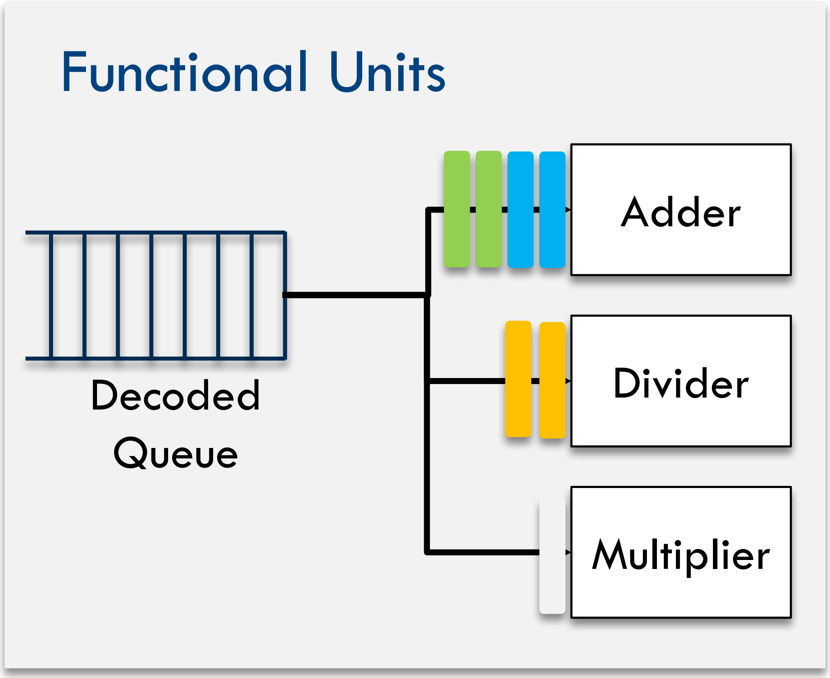
SUB R5, R2, 1

DIV R3, R3, R2

SUB R6, R5, 1

DIV R7, R3, R2

MUL R4, R4, R3



Functional Units

Decoded Queue

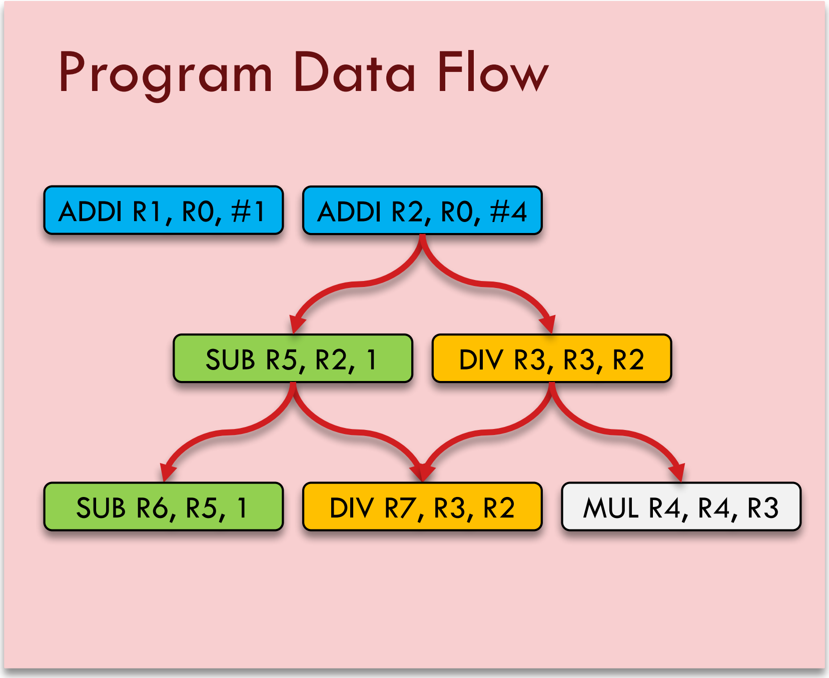
Adder

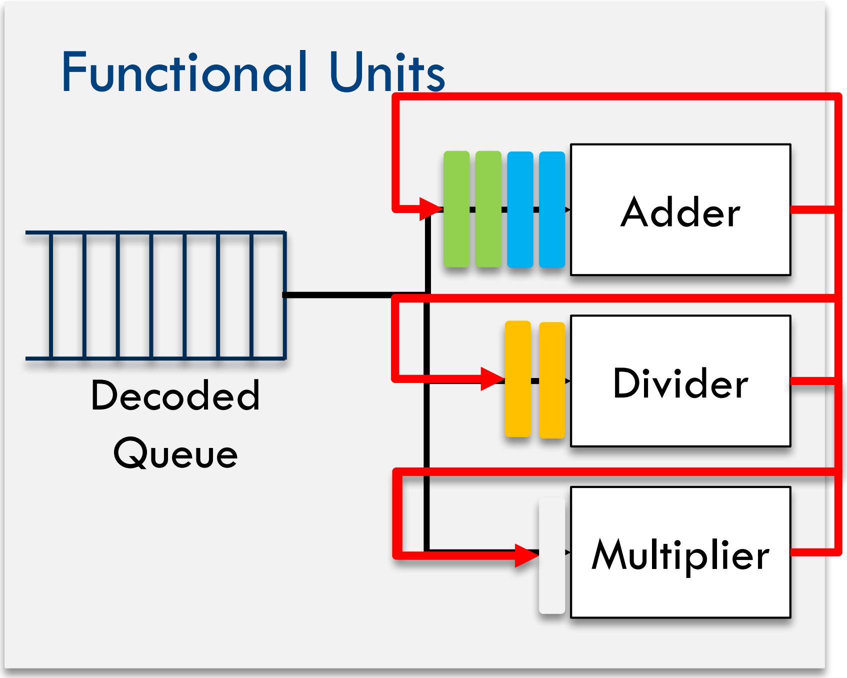
Divider

Multiplier

Dynamic Scheduling

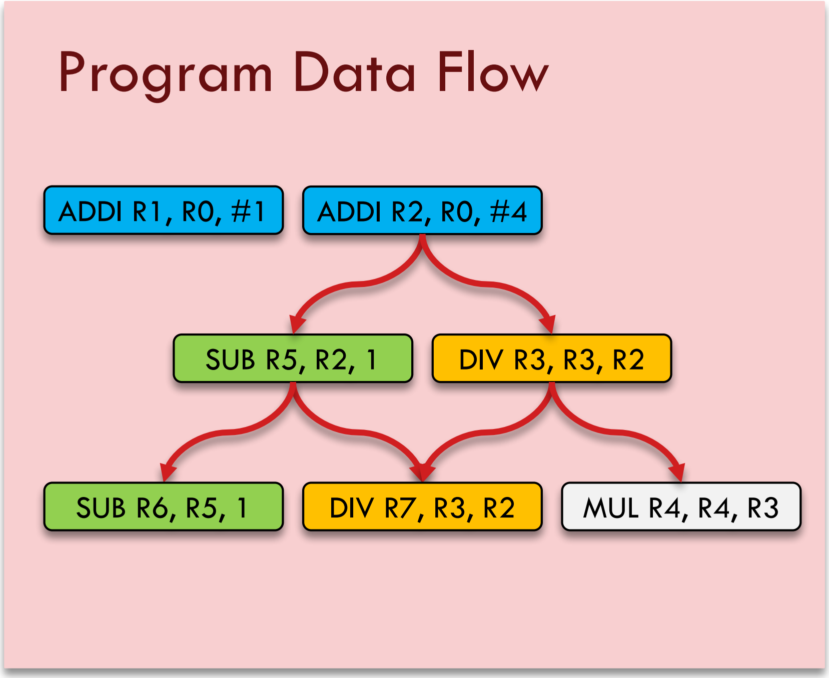
- When is the right time to go for execution?

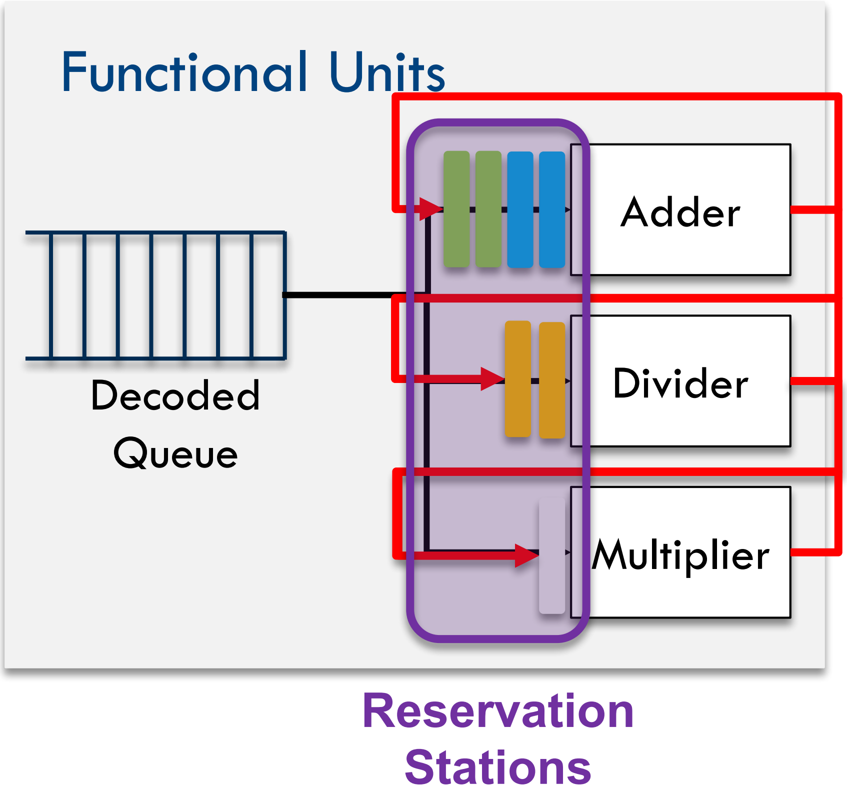




Dynamic Scheduling

- Reservation stations are used to keep the functional units busy





Tomasulo Algorithm

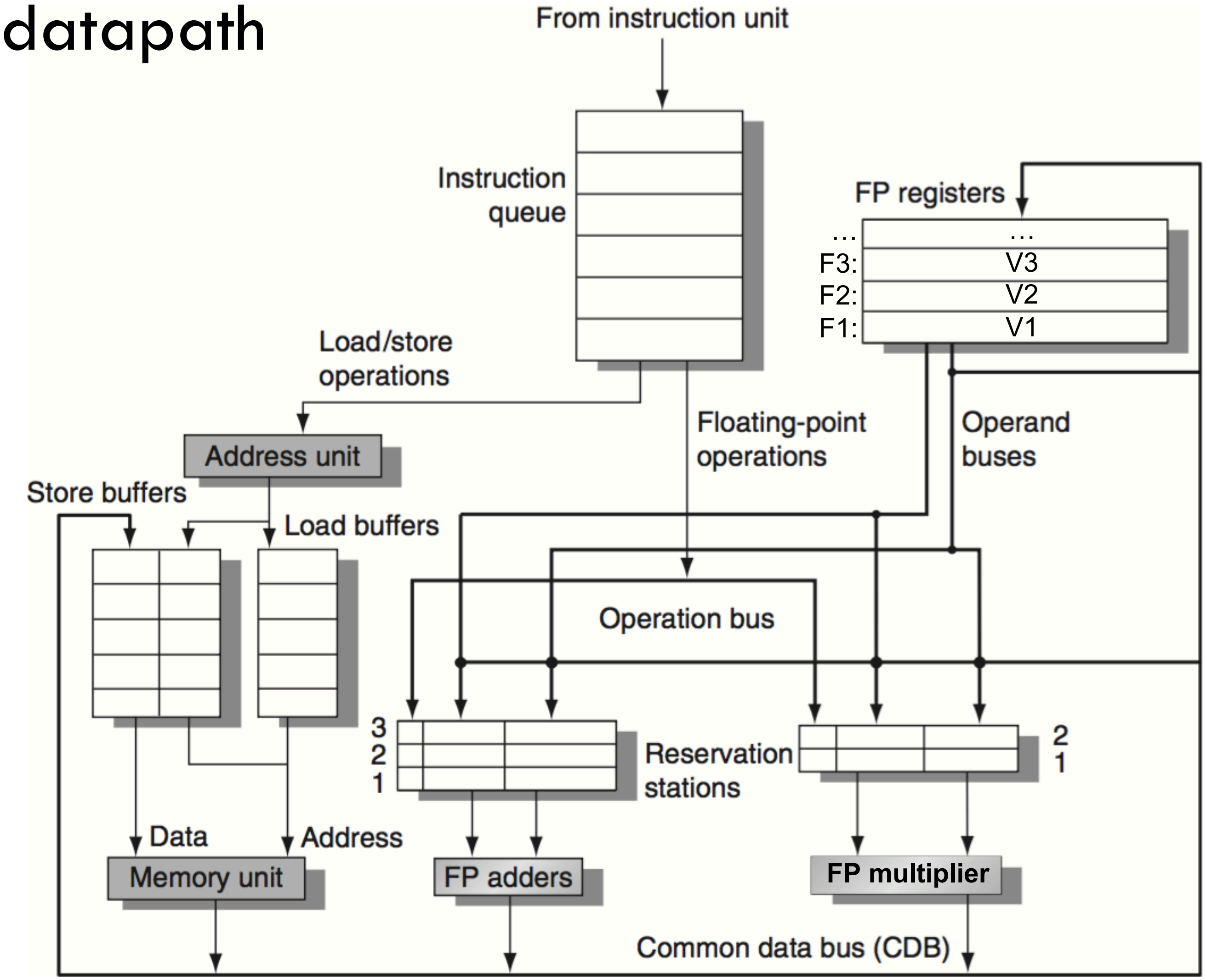
- Dispatch instructions to functional units
  - Use reservation stations (RS)
- Execute an instruction as soon as all of its operands are ready
  - Watch the common data bus (CDB)
- Remove false (anti- and output-) data dependence
  - Rename destination register to RS name

Three-Step Tomasulo Algorithm

- Issue: take an instruction from the instruction queue
  - If there are free reservation stations without structural hazards, rename and read/send operands or RS names
- Execute: operate on operand(s) when ready
  - If all of the operands are ready, execute; if not watch the common data bus
- Write result: update the register values
  - Write the result through CDB to all waiting reservation stations and the register file; release the RS entry

Hardware Implementation

- Example FP datapath

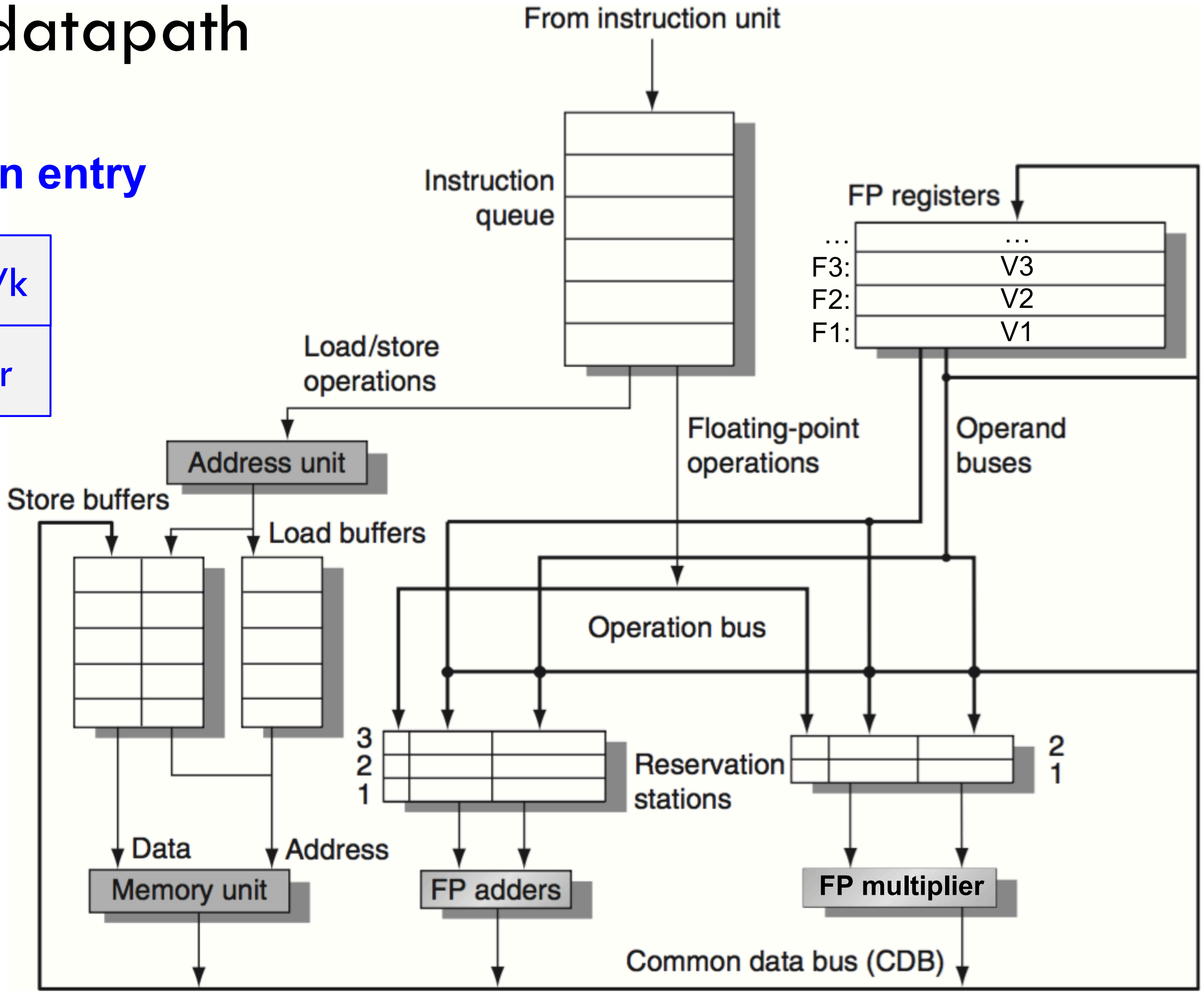


Hardware Implementation

- Example FP datapath

Reservation station entry

Busy	Op	Vj	Vk
Qj	Qk	Addr	



Hardware Implementation

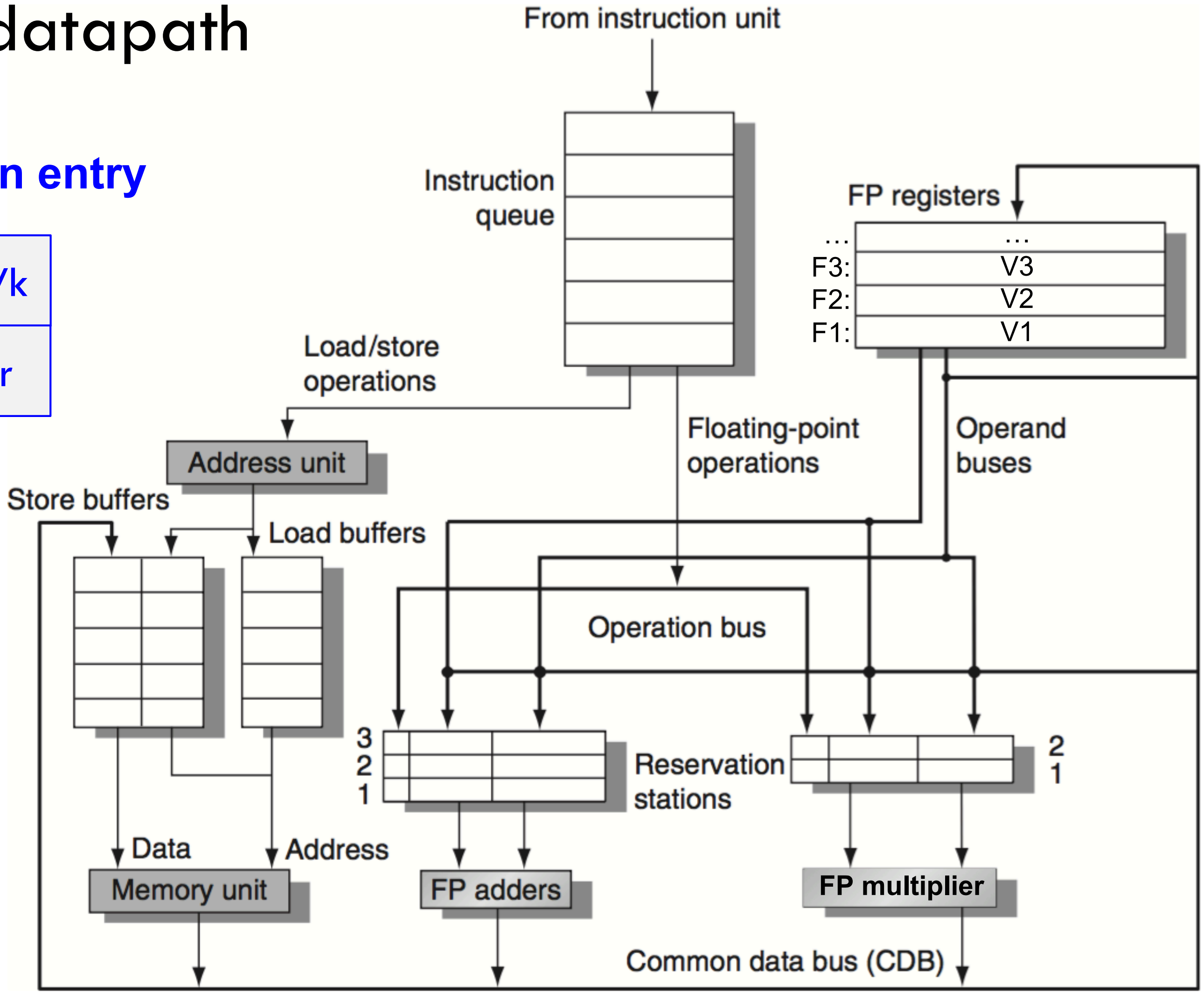
- Example FP datapath

Reservation station entry

Busy	Op	Vj	Vk
Qj	Qk	Addr	

Code:

```
ADD  F1, F2, F3
MUL  F6, F1, F3
```



Hardware Implementation

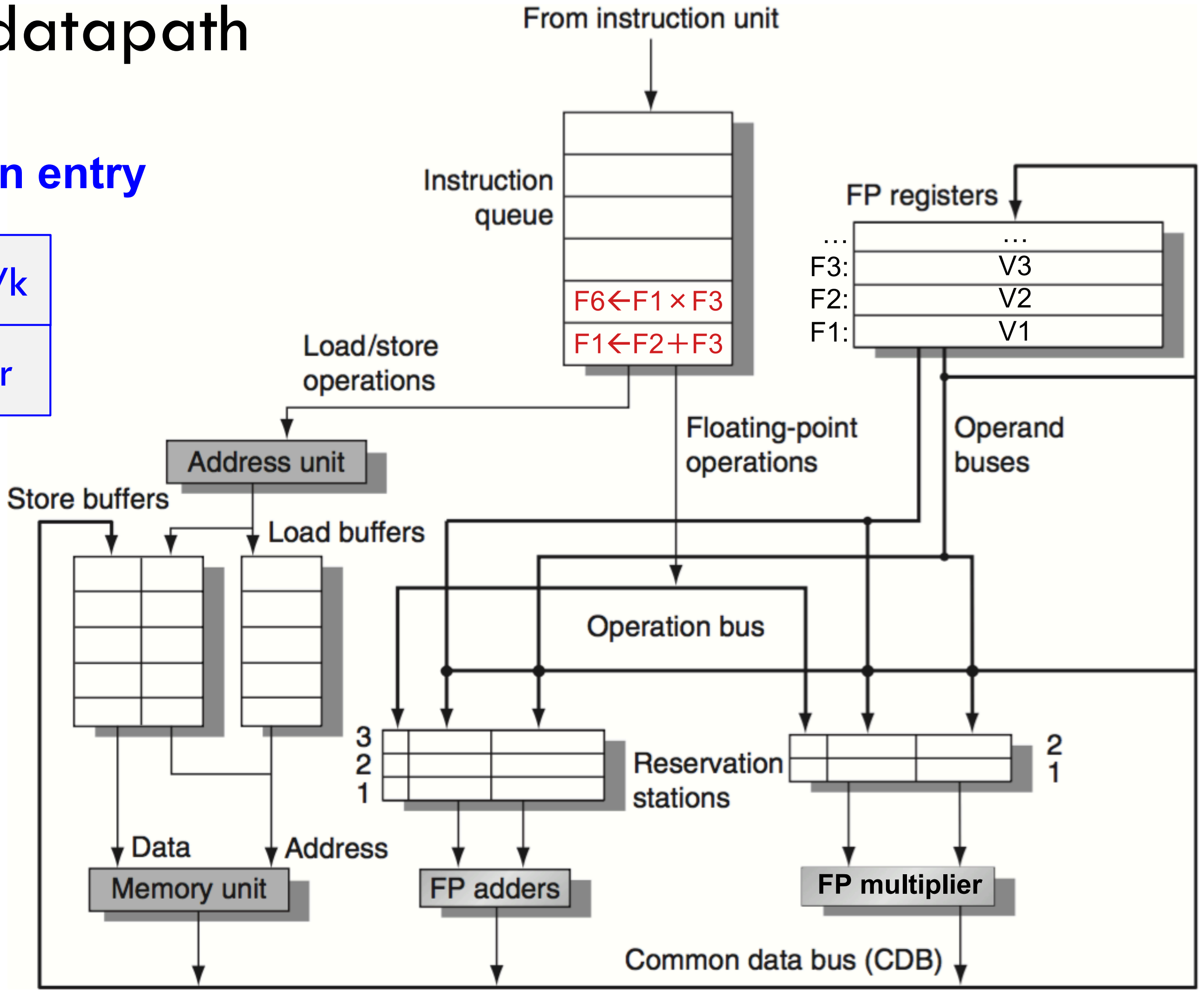
- □ Example FP datapath

Reservation station entry

Busy	Op	Vj	Vk
Qj	Qk	Addr	

Code:

```
ADD  F1, F2, F3
MUL  F6, F1, F3
```



Hardware Implementation

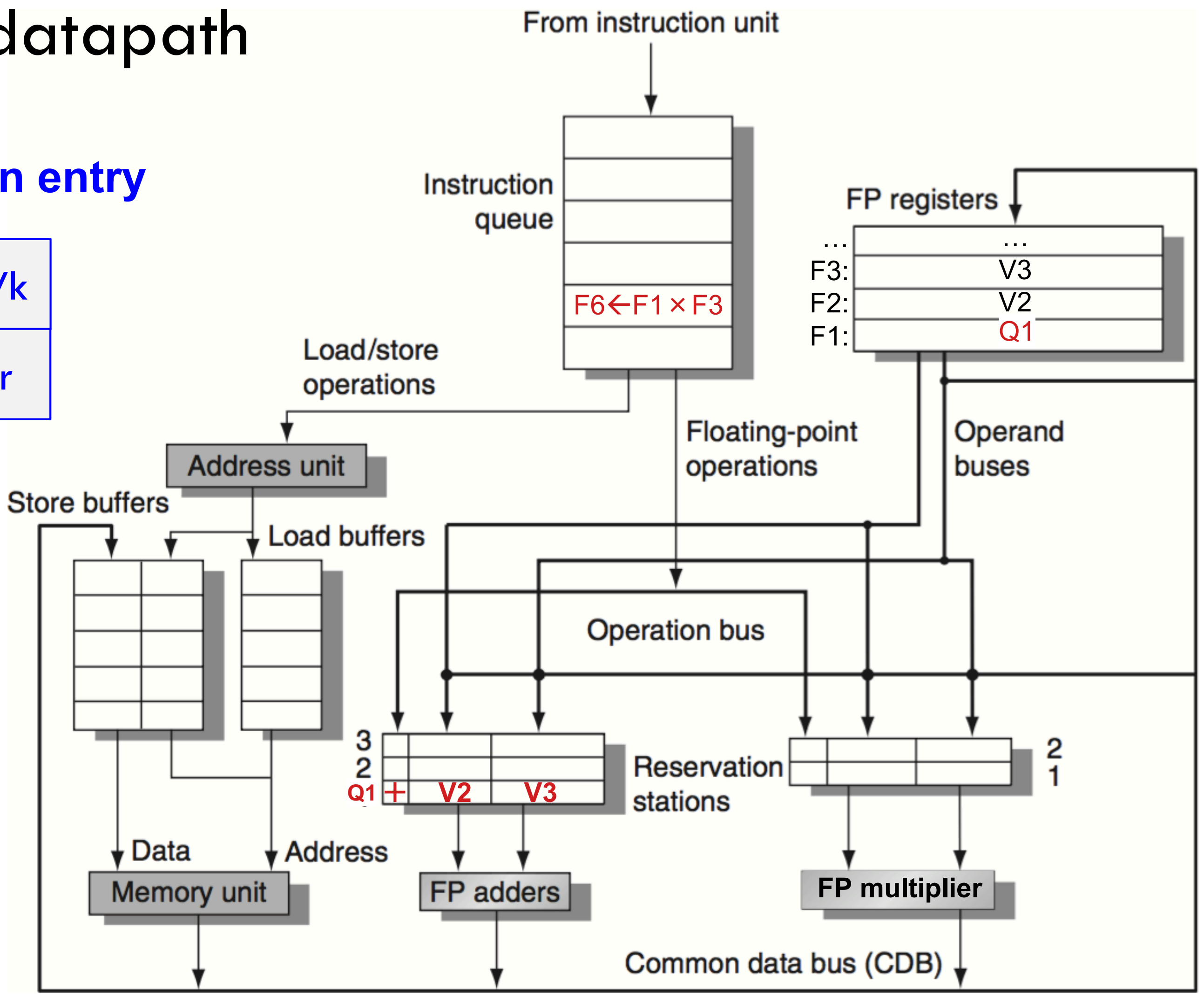
- Example FP datapath

Reservation station entry

Busy	Op	Vj	Vk
Qj	Qk	Addr	

Code:

```
ADD  F1, F2, F3
MUL  F6, F1, F3
```



Hardware Implementation

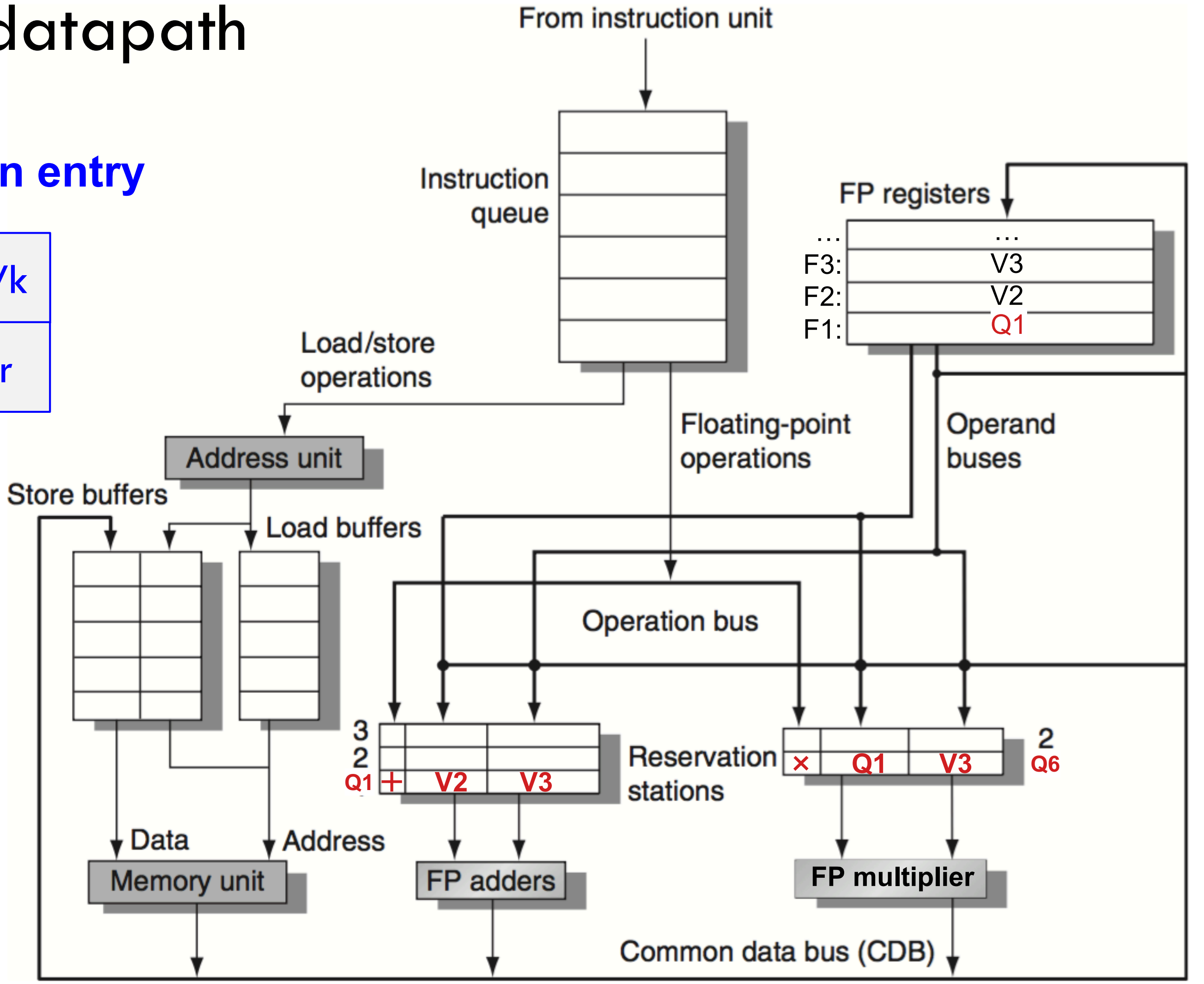
- Example FP datapath

Reservation station entry

Busy	Op	Vj	Vk
Qj	Qk	Addr	

Code:

```
ADD  F1, F2, F3
MUL  F6, F1, F3
```



Hardware Implementation

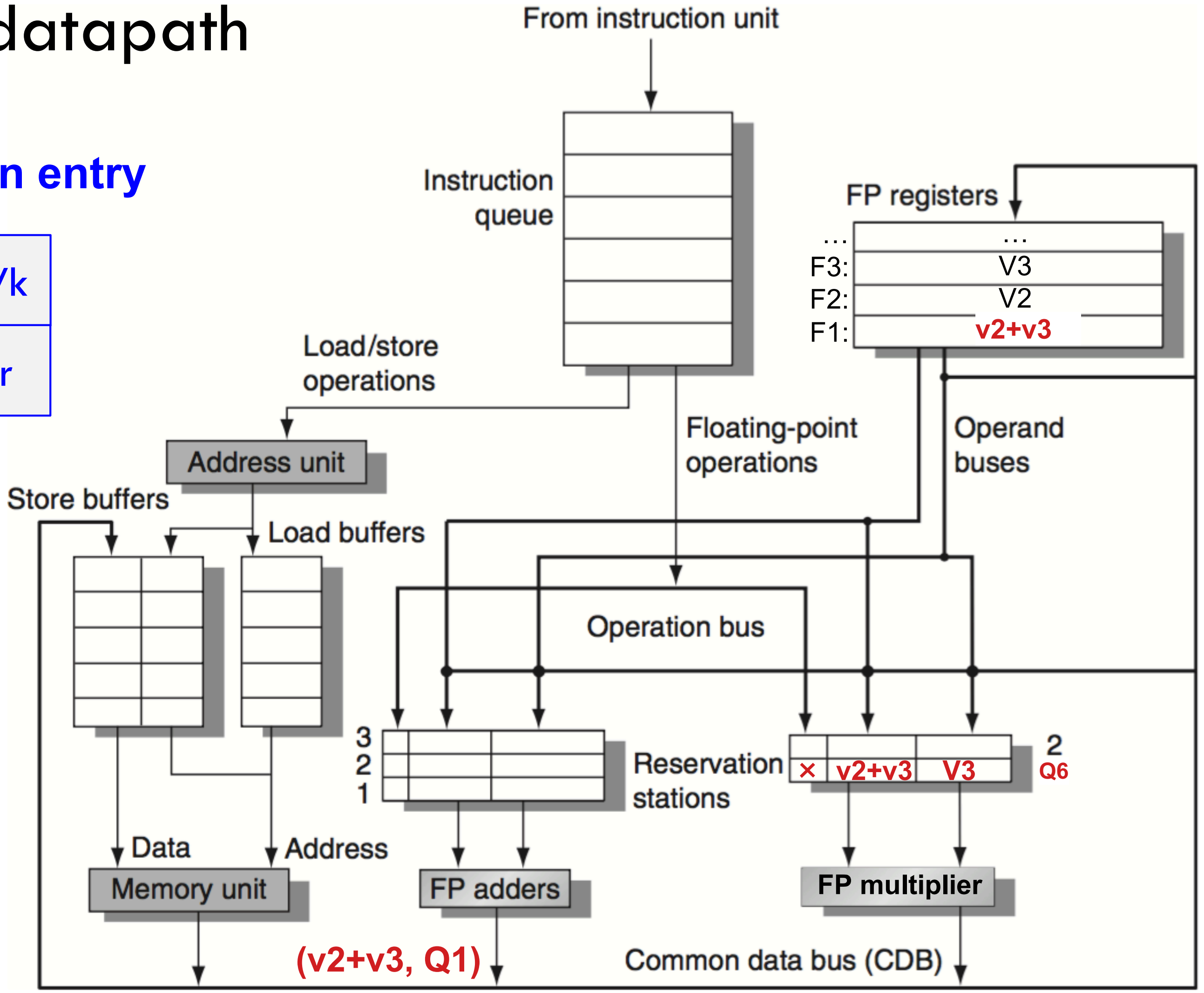
- Example FP datapath

Reservation station entry

Busy	Op	Vj	Vk
Qj	Qk	Addr	

Code:

```
ADD  F1, F2, F3
MUL  F6, F1, F3
```



Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD F6	43+	R2			
LD F2	45+	R3			
MUL F0	F2	F4			
SUB F8	F6	F2			
DIV F10	F0	F6			
ADD F6	F8	F2			

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	NO					
2	0	add3	NO					
10	0	mult1	NO					
40	0	divide	NO					

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	…	F30
Clock **0**	FU	value	value	value	value	value	value	value		

Example: Out-of-order Execution

Instruction Status

Instruction		j	k	issue	complete	write
LD	F6	43+	R2	1		
LD	F2	45+	R3			
MUL	F0	F2	F4			
SUB	F8	F6	F2			
DIV	F10	F0	F6			
ADD	F6	F8	F2			

		Busy	Address	Time
2	load1	YES	43+R2	2
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	NO					
2	0	add3	NO					
10	0	mult1	NO					
40	0	divide	NO					

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	… F30
Clock 1	FU	value	value	value	load1	value	value	value	

Example: Out-of-order Execution

Instruction Status

Instruction	j	k		issue	complete	write
LD	F6	43+	R2	1		
LD	F2	45+	R3	2		
MUL	F0	F2	F4			
SUB	F8	F6	F2			
DIV	F10	F0	F6			
ADD	F6	F8	F2			

		Busy	Address	Time
2	load1	YES	43+R2	1
2	load2	YES	45+R3	2
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	NO					
2	0	add3	NO					
10	0	mult1	NO					
40	0	divide	NO					

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	…	F30
Clock 2	FU	value	load2	value	load1	value	value	value		

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD F6	43+	R2	1	3	
LD F2	45+	R3	2		
MUL F0	F2	F4	3		
SUB F8	F6	F2			
DIV F10	F0	F6			
ADD F6	F8	F2			

		Busy	Address	Time
2	load1	YES	43+R2	0
2	load2	YES	45+R3	1
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	NO					
2	0	add3	NO					
10	0	mult1	YES	MULT		value	load2	
40	0	divide	NO					

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	… F30
Clock 3	FU	mult1	load2	value	load1	value	value	value	

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD	F6	43+ R2	1	3	4
LD	F2	45+ R3	2	4	
MUL	F0	F2	F4	3		
SUB	F8	F6	F2	4		
DIV	F10	F0	F6			
ADD	F6	F8	F2			

		Busy	Address	Time
2	load1	NO		0
2	load2	YES	45+R3	0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	YES	SUB	value			load2
2	0	add2	NO					
2	0	add3	NO					
10	0	mult1	YES	MULT		value	load2	
40	0	divide	NO					

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	…	F30
Clock 4	FU	mult1	load2	value	value	add1	value	value		

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD F6	43+	R2	1	3	4
LD F2	45+	R3	2	4	5
MUL F0	F2	F4	3		
SUB F8	F6	F2	4		
DIV F10	F0	F6	5		
ADD F6	F8	F2			

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	2	add1	YES	SUB	value	value		
2	0	add2	NO					
2	0	add3	NO					
10	10	mult1	YES	MULT	value	value		
40	0	divide	YES	DIV		value	mult1	

Register Result Status

Clock 5		F0	F2	F4	F6	F8	F10	F12	…	F30
	FU	mult1	value	value	value	add1	divide	value		

Example: Out-of-order Execution

Instruction Status

Instruction	j	k		issue	complete	write
LD	F6	43+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MUL	F0	F2	F4	3		
SUB	F8	F6	F2	4		
DIV	F10	F0	F6	5		
ADD	F6	F8	F2	6		

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	1	add1	YES	SUB	value	value		
2	0	add2	YES	ADD		value	add1	
2	0	add3	NO					
10	9	mult1	YES	MULT	value	value		
40	0	divide	YES	DIV		value	mult1	

Register Result Status

	F0	F2	F4	F6	F8	F10	F12	…	F30
FU	mult1	value	value	add2	add1	divide	value		

Clock 6

Example: Out-of-order Execution

Instruction Status

Instruction	j	k		issue	complete	write
LD	F6	43+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MUL	F0	F2	F4	3		
SUB	F8	F6	F2	4	7	
DIV	F10	F0	F6	5		
ADD	F6	F8	F2	6		

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	YES	SUB	value	value		
2	0	add2	YES	ADD		value	add1	
2	0	add3	NO					
10	8	mult1	YES	MULT	value	value		
40	0	divide	YES	DIV		value	mult1	

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	… F30
Clock 7	FU	mult1	value	value	add2	add1	divide	value	

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD F6	43+	R2	1	3	4
LD F2	45+	R3	2	4	5
MUL F0	F2	F4	3		
SUB F8	F6	F2	4	7	8
DIV F10	F0	F6	5		
ADD F6	F8	F2	6		

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	2	add2	YES	ADD	value	value		
2	0	add3	NO					
10	7	mult1	YES	MULT	value	value		
40	0	divide	YES	DIV		value	mult1	

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	… F30
Clock **8**	FU	mult1	value	value	add2	value	divide	value	

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD F6	43+	R2	1	3	4
LD F2	45+	R3	2	4	5
MUL F0	F2	F4	3		
SUB F8	F6	F2	4	7	8
DIV F10	F0	F6	5		
ADD F6	F8	F2	6		

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	1	add2	YES	ADD	value	value		
2	0	add3	NO					
10	6	mult1	YES	MULT	value	value		
40	0	divide	YES	DIV		value	mult1	

Register Result Status

	F0	F2	F4	F6	F8	F10	F12	… F30
FU	mult1	value	value	add2	value	divide	value	

Clock 9

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD	F6	43+ R2	1	3	4
LD	F2	45+ R3	2	4	5
MUL	F0	F2	F4	3		
SUB	F8	F6	F2	4	7	8
DIV	F10	F0	F6	5		
ADD	F6	F8	F2	6	10	

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	YES	ADD	value	value		
2	0	add3	NO					
10	5	mult1	YES	MULT	value	value		
40	0	divide	YES	DIV		value	mult1	

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	… F30
Clock 10	FU	mult1	value	value	add2	value	divide	value	

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD	F6	43+ R2	1	3	4
LD	F2	45+ R3	2	4	5
MUL	F0	F2	F4	3		
SUB	F8	F6	F2	4	7	8
DIV	F10	F0	F6	5		
ADD	F6	F8	F2	6	10	11

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	NO					
2	0	add3	NO					
10	4	mult1	YES	MULT	value	value		
40	0	divide	YES	DIV		value	mult1	

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	… F30
Clock 11	FU	mult1	value	value	value	value	divide	value	

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD F6	43+	R2	1	3	4
LD F2	45+	R3	2	4	5
MUL F0	F2	F4	3	15	
SUB F8	F6	F2	4	7	8
DIV F10	F0	F6	5		
ADD F6	F8	F2	6	10	11

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	NO					
2	0	add3	NO					
10	0	mult1	YES	MULT	value	value		
40	0	divide	YES	DIV		value	mult1	

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	… F30
Clock **15**	FU	mult1	value	value	value	value	divide	value	

Example: Out-of-order Execution

Instruction Status

Instruction		j	k	issue	complete	write
LD	F6	43+	R2	1	3	4
LD	F2	45+	R3	2	4	5
MUL	F0	F2	F4	3	15	16
SUB	F8	F6	F2	4	7	8
DIV	F10	F0	F6	5		
ADD	F6	F8	F2	6	10	11

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	NO					
2	0	add3	NO					
10	0	mult1	NO					
40	40	divide	YES	DIV	value	value		

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	…	F30
Clock **16**	FU	value	value	value	value	value	divide	value		

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD F6	43+	R2	1	3	4
LD F2	45+	R3	2	4	5
MUL F0	F2	F4	3	15	16
SUB F8	F6	F2	4	7	8
DIV F10	F0	F6	5	56	
ADD F6	F8	F2	6	10	11

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	NO					
2	0	add3	NO					
10	0	mult1	NO					
40	0	divide	YES	DIV	value	value		

Register Result Status

		F0	F2	F4	F6	F8	F10	F12	…	F30
Clock **56**	FU	value	value	value	value	value	divide	value		

Example: Out-of-order Execution

Instruction Status

Instruction	j	k	issue	complete	write
LD F6	43+	R2	1	3	4
LD F2	45+	R3	2	4	5
MUL F0	F2	F4	3	15	16
SUB F8	F6	F2	4	7	8
DIV F10	F0	F6	5	56	57
ADD F6	F8	F2	6	10	11

		Busy	Address	Time
2	load1	NO		0
2	load2	NO		0
2	load3	NO		0

Reservation Stations

	Time	Name	Busy	Op	Vj	Vk	Qj	Qk
2	0	add1	NO					
2	0	add2	NO					
2	0	add3	NO					
10	0	mult1	NO					
40	0	divide	NO					

Register Result Status

Clock **57**

	F0	F2	F4	F6	F8	F10	F12	… F30
FU	value	value	value	value	value	value	value	

Summary of Tomasulo Algorithm

- Data hazards

- Structural hazards

- Precise exception handling

Summary of Tomasulo Algorithm

- Data hazards
  - RAW is handled by forwarding over CDB
  - WAR and WAW are removed by RS-based renaming
- Structural hazards
  - Multiple FUs may be accessing CDB simultaneously
- Precise exception handling
  - Not possible because of OoO writeback to register file

Summary of Tomasulo Algorithm

- Data hazards
  - RAW is handled by forwarding over CDB
  - WAR and WAW are removed by RS-based renaming
- Structural hazards
  - Multiple FUs may be accessing CDB simultaneously
    - Solution: delay conflicting instructions at issue and RS
- Precise exception handling
  - Not possible because of OoO writeback to register file
    - Solution: maintain the destination value in ROB (IW)

Four-Step Tomasulo Algorithm

- Issue (dispatch)
  - If RS and ROB slots are free; read/rename operands
- Execution
  - Execute operation as soon as the operand values are ready
- Write result
  - Send result to ROB and reservation stations via CDB
- Commit (retire)
  - Update register file for the head of ROB

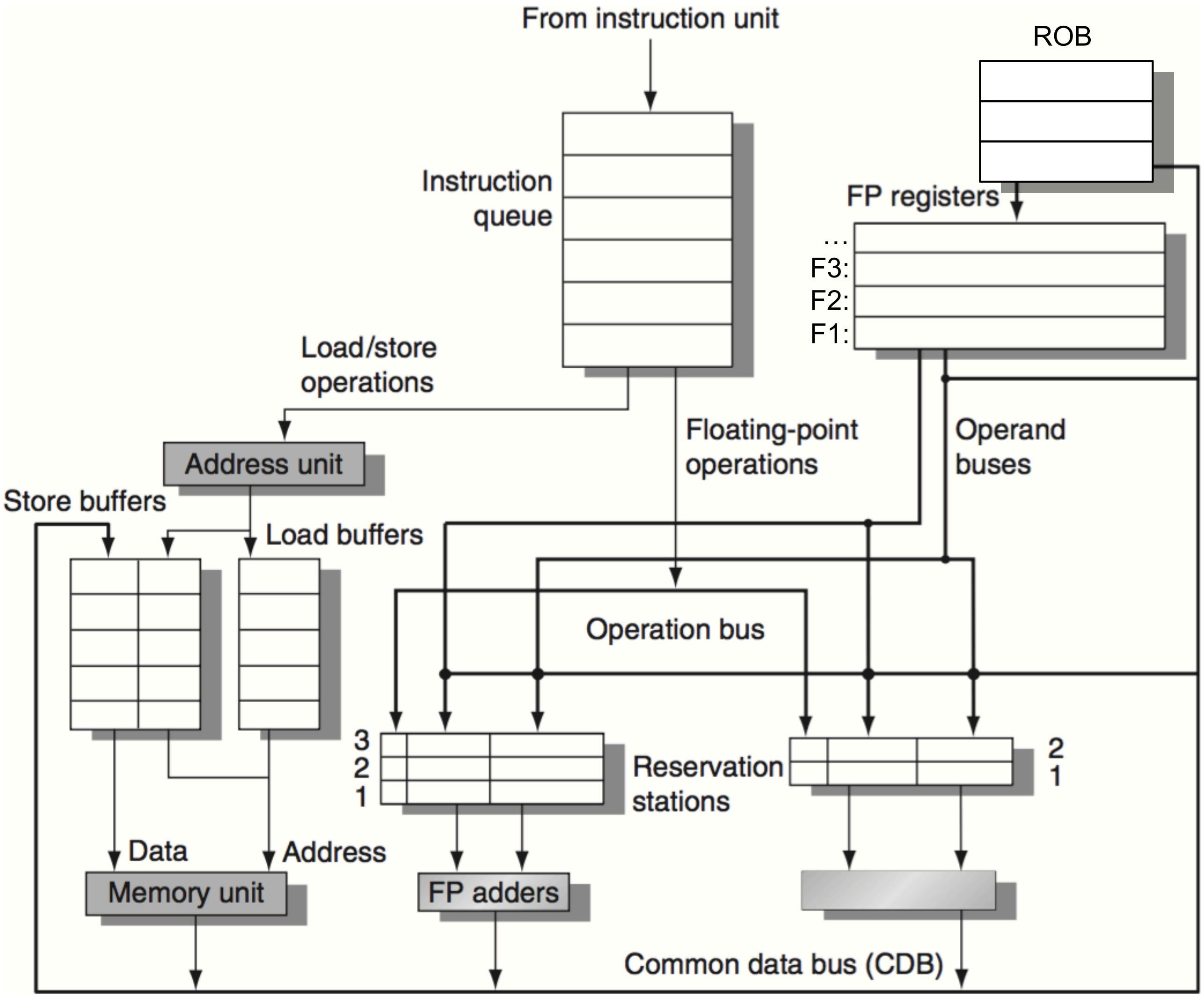
Four-Step Tomasulo Algorithm

ROB Entry

Valid	Result	Exception
Program Counter		

Code

```
MUL  F1, F2, F3
ADD  F1, F2, F3
```



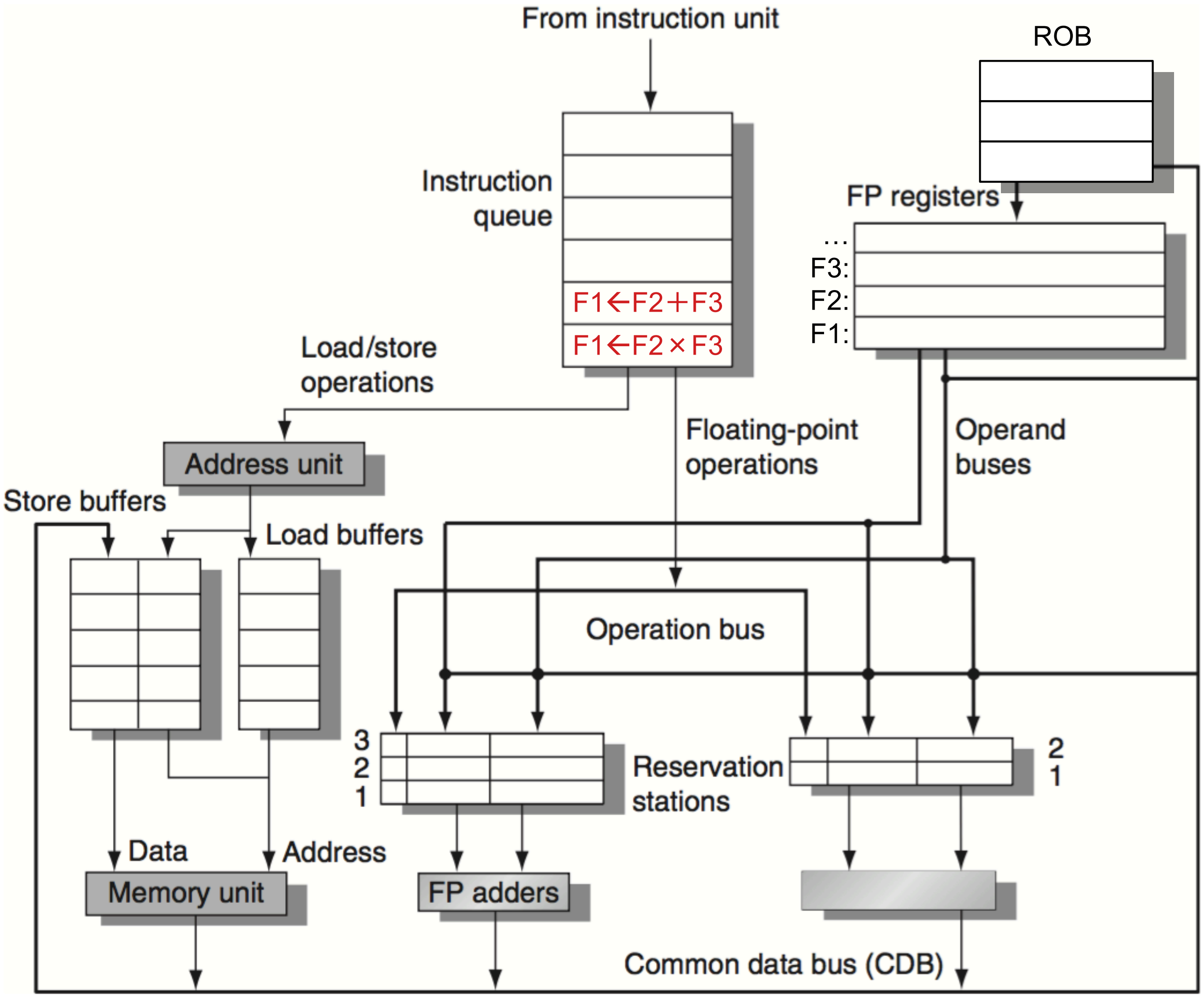
Four-Step Tomasulo Algorithm

ROB Entry

Valid	Result	Exception
Program Counter		

Code

```
MUL  F1, F2, F3
ADD  F1, F2, F3
```



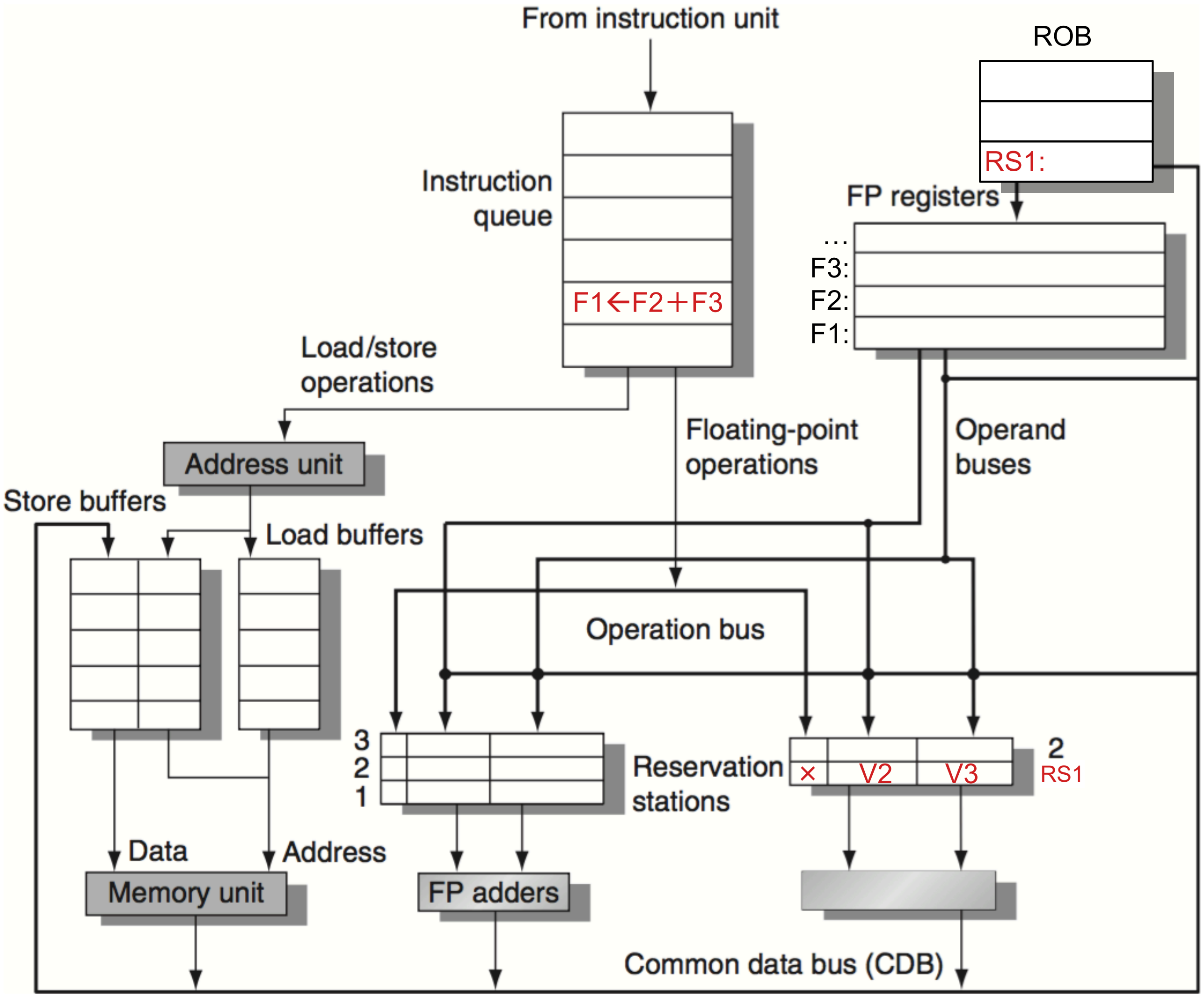
Four-Step Tomasulo Algorithm

ROB Entry

Valid	Result	Exception
Program Counter		

Code

```
MUL  F1, F2, F3
ADD  F1, F2, F3
```



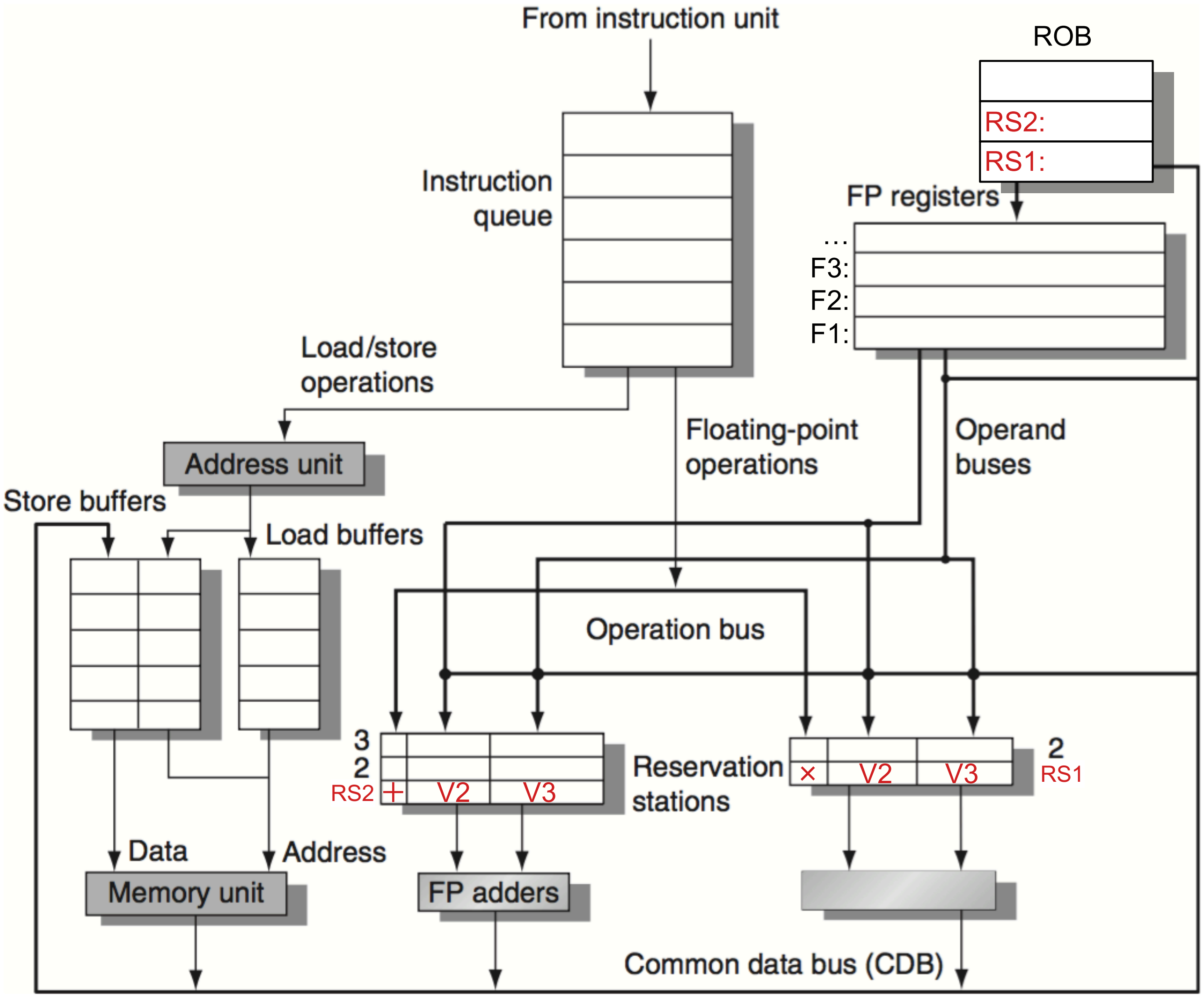
Four-Step Tomasulo Algorithm

ROB Entry

Valid	Result	Exception
Program Counter		

Code

```
      MUL  F1, F2, F3
      ADD  F1, F2, F3
```



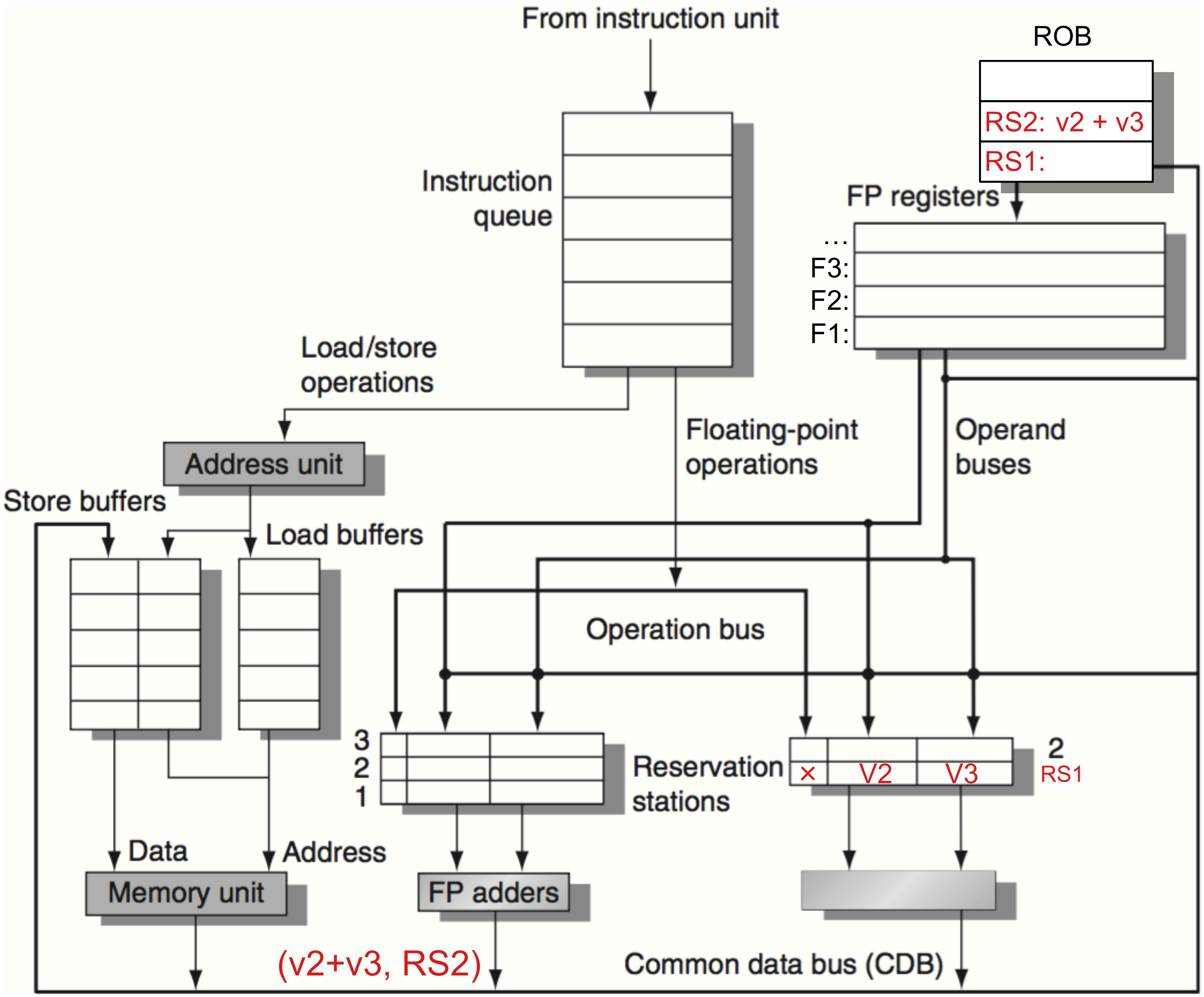
Four-Step Tomasulo Algorithm

ROB Entry

Valid	Result	Exception
Program Counter		

Code

```
MUL  F1, F2, F3
ADD  F1, F2, F3
```



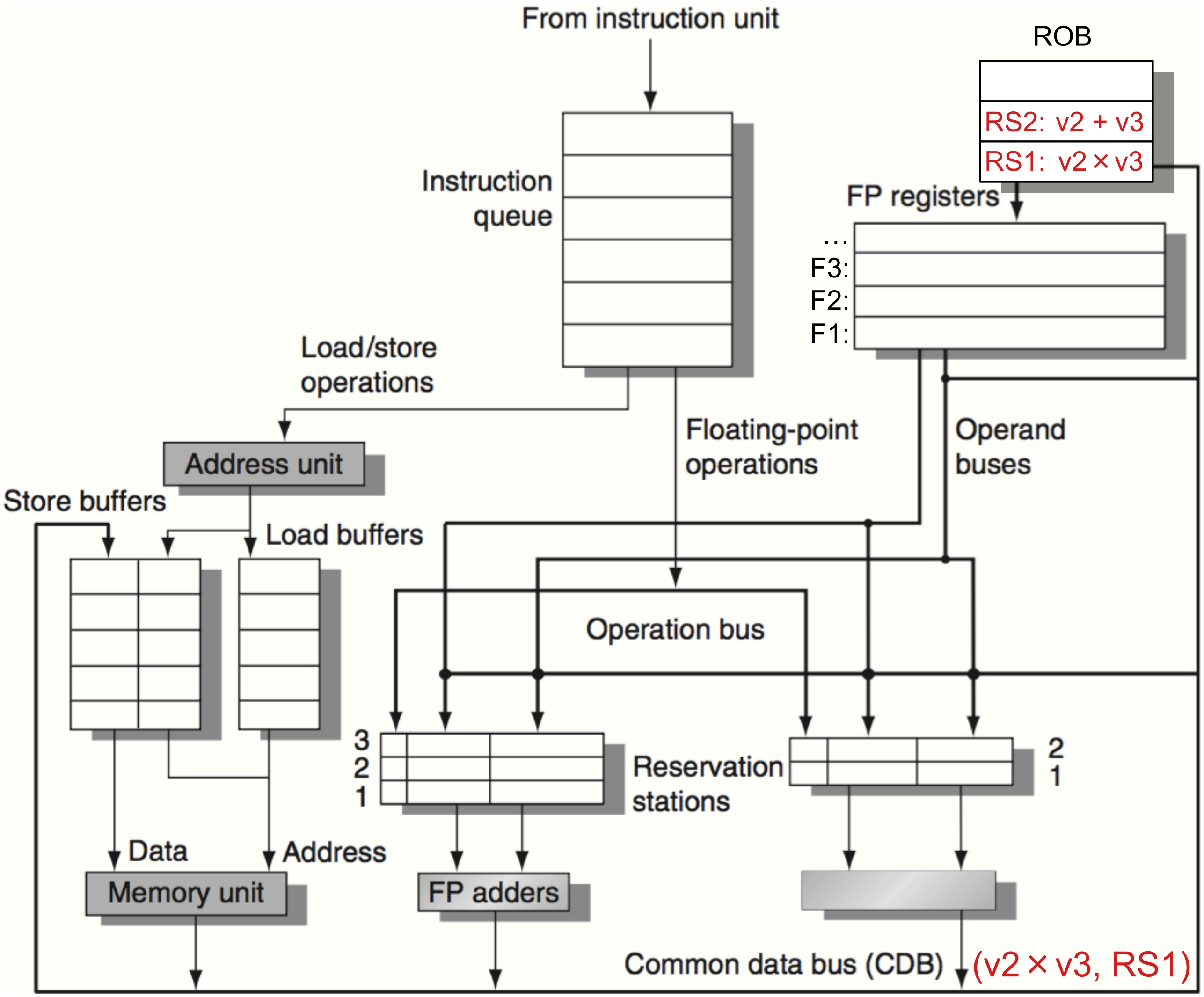
Four-Step Tomasulo Algorithm

ROB Entry

Valid	Result	Exception
Program Counter		

Code

```
MUL  F1, F2, F3
ADD  F1, F2, F3
```



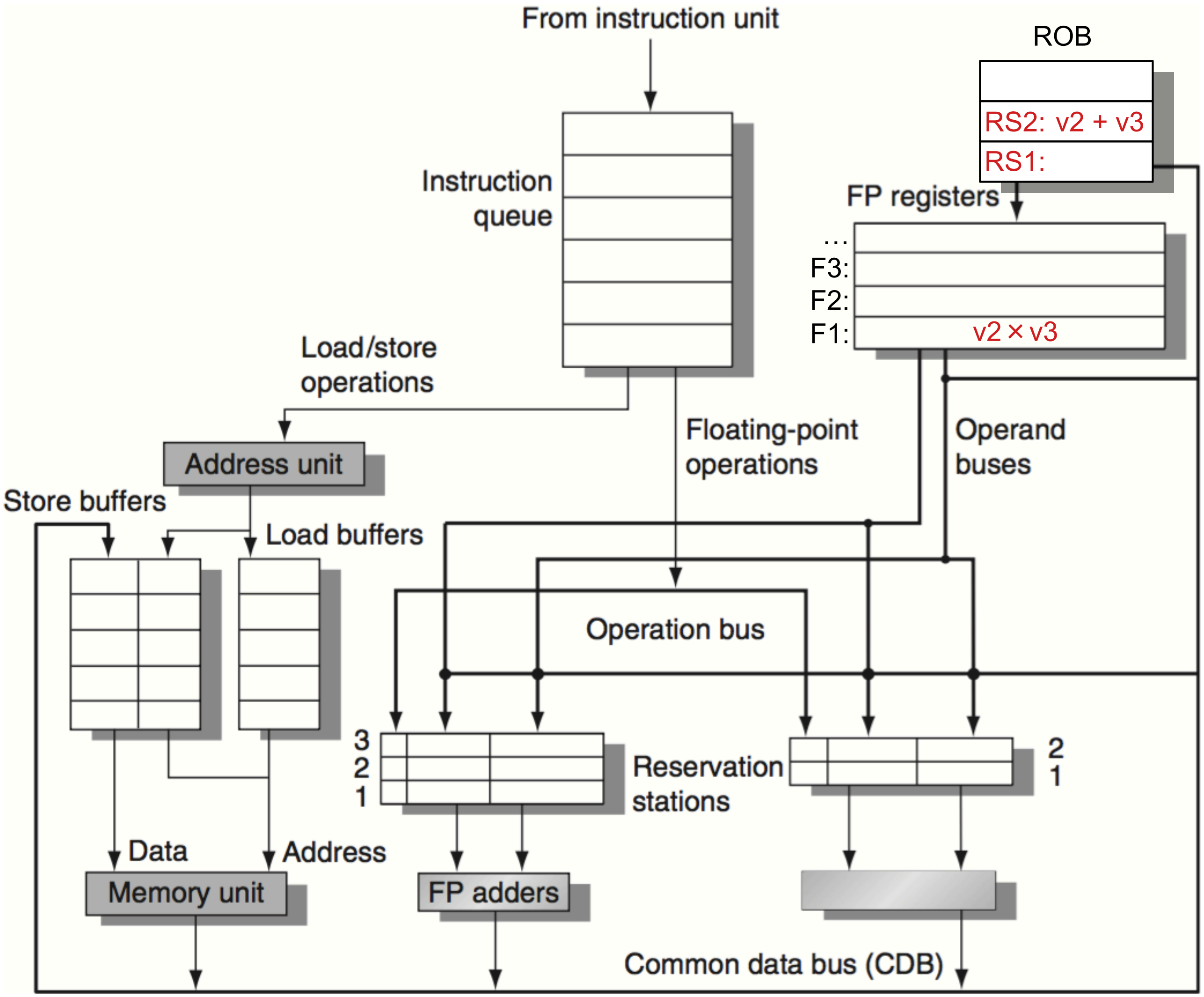
Four-Step Tomasulo Algorithm

ROB Entry

Valid	Result	Exception
Program Counter		

Code

MUL F1, F2, F3
ADD F1, F2, F3



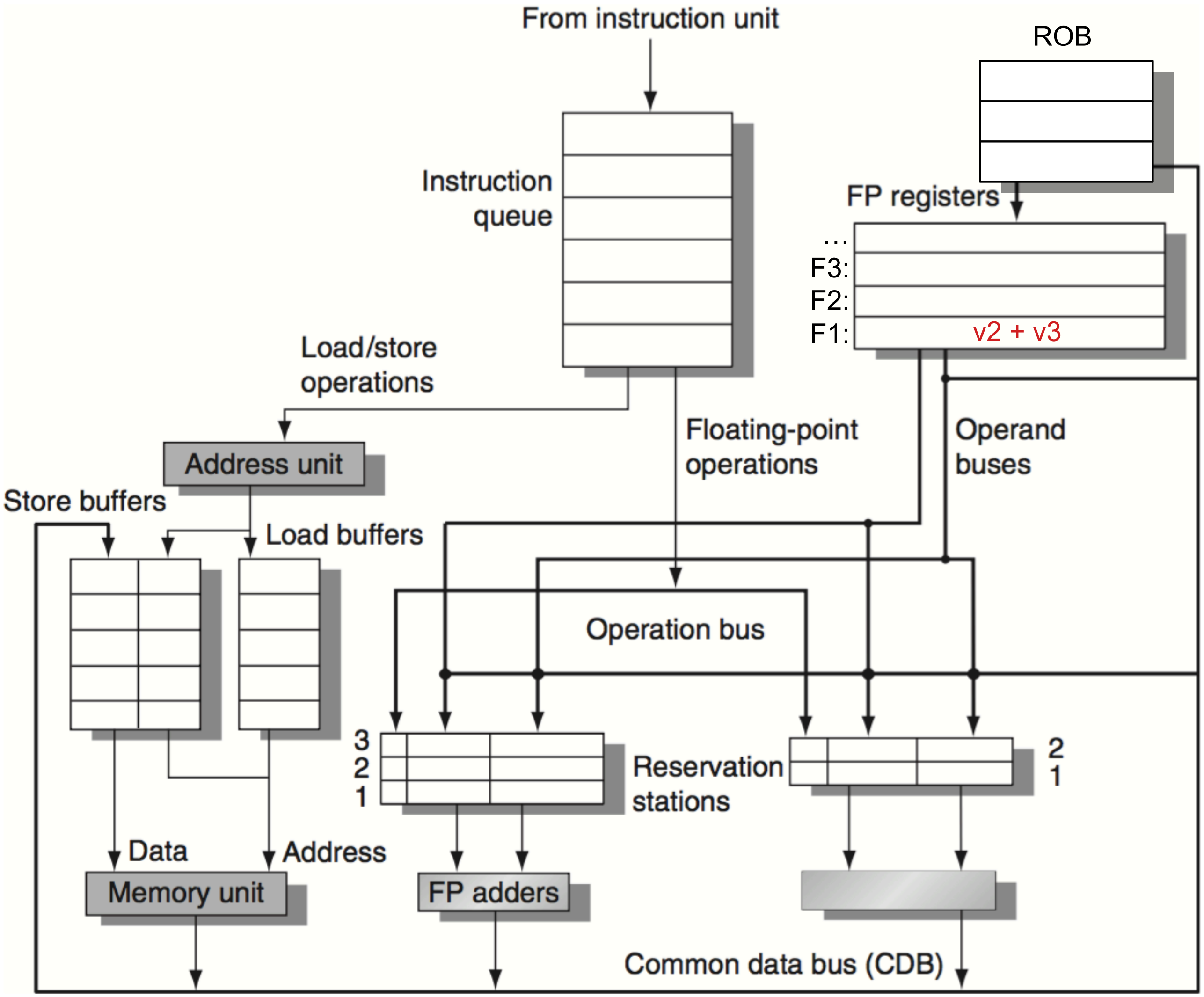
Four-Step Tomasulo Algorithm

ROB Entry

Valid	Result	Exception
Program Counter		

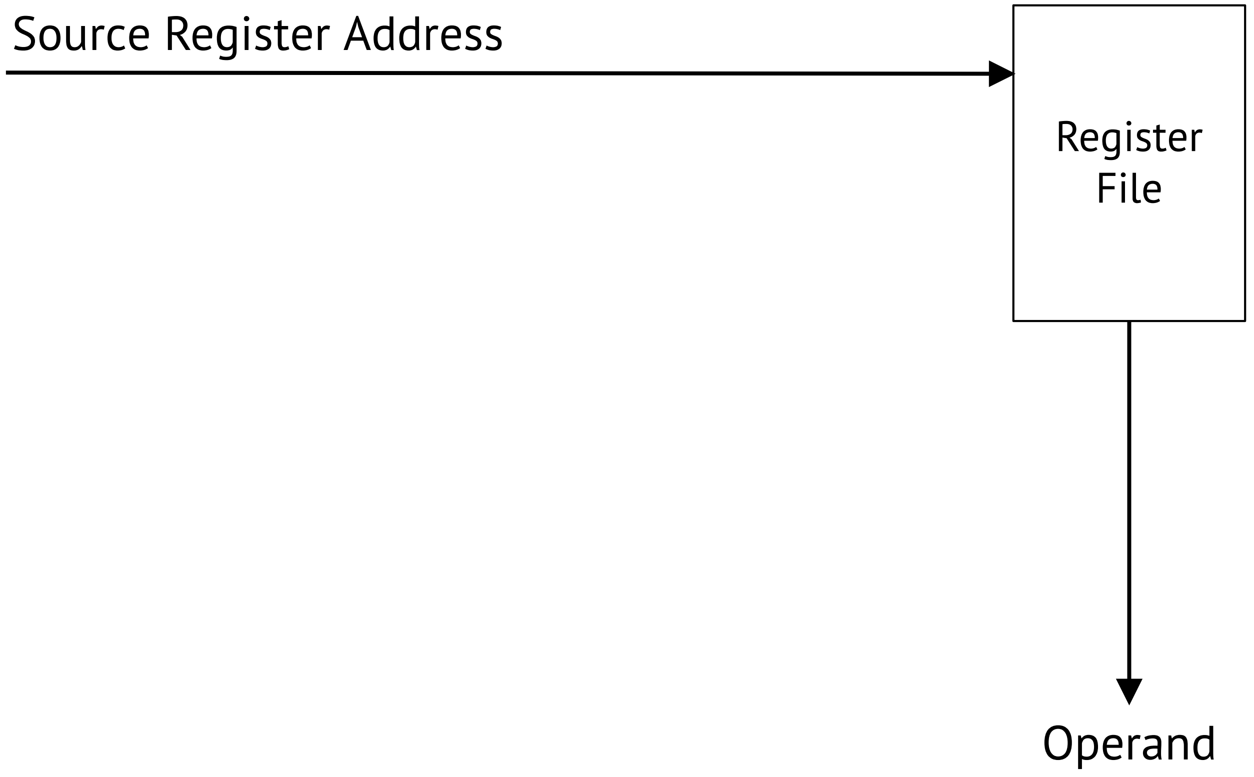
Code

```
MUL  F1, F2, F3
ADD  F1, F2, F3
```



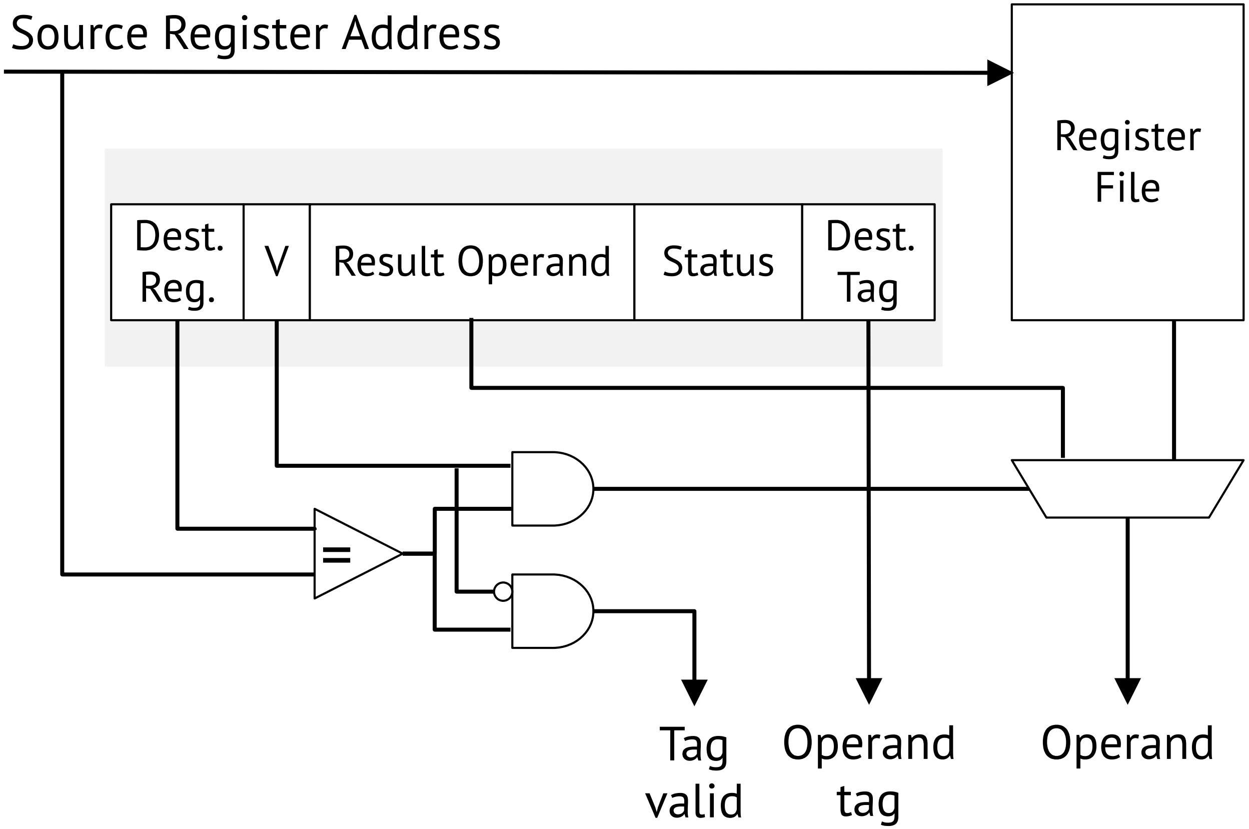
Operand Read

- Reading the register file



Operand Read/Search

- AMD K-5: ROB dependency check


Source Register Address
Register File
Dest. Reg.
V
Result Operand
Status
Dest. Tag
=
Tag valid
Operand tag
Operand