Tomasulo's Algorithm: Execution Steps

issue & read (assume the instruction has been fetched)

- structural hazard detection for entry in reservation station **& reorder buffer**
  - issue if no hazard
  - stall if hazard
- read registers for source operands
  - **can come from either registers or reorder buffer**
  - **RAT indicates which**
- **allocate entry in reorder buffer**

execute

- RAW hazard detection for source operands
- snoop on common data bus for missing operands
  - **reservation station tag is entry number in reorder buffer**
- dispatch to functional unit when obtain both operand values

Tomasulo's Algorithm: Execution Steps

write result

- broadcast result & **reorder buffer entry** (tag) on the common data bus to reservation stations & **reorder buffer**

commit

- **retire the instruction at the head of the reorder buffer**
- **update register with result in reorder buffer or do a store**
- **remove instruction from reorder buffer**
- **if a mispredicted branch, restart with right-path instructions**

Example, 1

First load has written its result

~ Reorder Buffer

Instruction	Issue	Execute	Write Result	Commit
`ld F6, 34(R2)`	yes	yes	yes	
`ld F2, 45(R3)`	yes	yes		
`multd F0, F2, F4`	yes			
`subd F8, F6, F2`	yes			
`divd F10, F0, F6`	yes			
`addd F6, F8, F2`	yes			

Reservations Stations

Name	Busy	Op	V_j	V_k	Q_j	Q_k
Add1	yes	subd	(ROB 1)			ROB 2
Add2	yes	addd			ROB 4	ROB 2
Add3	no					
Mult1	yes	multd		(F4)	ROB 2	
Mult2	yes	div		(ROB 1)	ROB 3	

Register Status (Qj)

F0	F2	F4	F6	F8	F10	F12…
ROB 3	ROB 2		ROB 6	ROB 4	ROB 5	

Example, 2

First load has committed, second load has written

~ Reorder Buffer

Instruction	Issue	Execute	Write Result	Commit
~~`ld F6, 34(R2)`~~	~~yes~~	~~yes~~	~~yes~~	~~yes~~
`ld F2, 45(R3)`	yes	yes	yes	
`multd F0, F2, F4`	yes	yes		
`subd F8, F6, F2`	yes	yes		
`divd F10, F0, F6`	yes			
`addd F6, F8, F2`	yes			

Reservations Stations

Name	Busy	Op	V_j	V_k	Q_j	Q_k
Add1	yes	subd	(ROB 1)	(ROB 2)		
Add2	yes	addd		(ROB 2)	ROB 4	
Add3	no					
Mult1	yes	multd	(ROB 2)	(F4)		
Mult2	yes	div		(ROB 1)	ROB 3	

Register Status (Qj)

F0	F2	F4	F6	F8	F10	F12…
ROB 3	ROB 2		ROB 6	ROB 4	ROB 5	

Example, 3

second load has committed, subtd has written

~ Reorder Buffer

Instruction	Issue	Execute	Write Result	Commit
~~ld F6, 34(R2)~~	yes	yes	yes	yes
~~ld F2, 45(R3)~~	yes	yes	yes	yes
multd F0, F2, F4	yes	yes		
subd F8, F6, F2	yes	yes	yes	
divd F10, F0, F6	yes			
addd F6, F8, F2	yes	yes		

Reservations Stations

Name	Busy	Op	V_j	V_k	Q_j	Q_k
Add1	no					
Add2	yes	addd	(ROB 4)	(ROB 2)		
Add3	no					
Mult1	yes	multd	(ROB 2)	(F4)		
Mult2	yes	div		(ROB 1)	ROB 3	

Register Status (Qj)

F0	F2	F4	F6	F8	F10	F12…
ROB 3	(ROB 2)		ROB 6	ROB 4	ROB 5	

Example, 4

addd has written

~ Reorder Buffer

Instruction	Issue	Execute	Write Result	Commit
~~ld F6, 34(R2)~~	yes	yes	yes	yes
~~ld F2, 45(R3)~~	yes	yes	yes	yes
multd F0, F2, F4	yes	yes		
subd F8, F6, F2	yes	yes	yes	
divd F10, F0, F6	yes			
addd F6, F8, F2	yes	yes	yes	

Reservations Stations

Name	Busy	Op	V_j	V_k	Q_j	Q_k
Add1	no					
Add2	no					
Add3	no					
Mult1	yes	multd	(ROB 2)	(F4)		
Mult2	yes	div		(ROB 1)	ROB 3	

Register Status (Qj)

F0	F2	F4	F6	F8	F10	F12…
ROB 3	(ROB 2)		ROB 6	ROB 4	ROB 5	

Example, 5

multd has written

~ Reorder Buffer

Instruction	Issue	Execute	Write Result	Commit
~~ld F6, 34(R2)~~	yes	yes	yes	yes
~~ld F2, 45(R3)~~	yes	yes	yes	yes
multd F0, F2, F4	yes	yes	yes	
subd F8, F6, F2	yes	yes	yes	
divd F10, F0, F6	yes	yes		
addd F6, F8, F2	yes	yes	yes	

Reservations Stations

Name	Busy	Op	V_j	V_k	Q_j	Q_k
Add1	no					
Add2	no					
Add3	no					
Mult1	no					
Mult2	yes	div	(ROB 3)	(ROB 1)		

Register Status (Qj)

F0	F2	F4	F6	F8	F10	F12…
	(ROB 2)		ROB 6	ROB 4	ROB 5	

Example, 6

multd has committed

~ Reorder Buffer

Instruction	Issue	Execute	Write Result	Commit
~~ld F6, 34(R2)~~	yes	yes	yes	yes
~~ld F2, 45(R3)~~	yes	yes	yes	yes
~~multd F0, F2, F4~~	yes	yes	yes	yes
subd F8, F6, F2	yes	yes	yes	
divd F10, F0, F6	yes	yes		
addd F6, F8, F2	yes	yes	yes	

Reservations Stations

Name	Busy	Op	V_j	V_k	Q_j	Q_k
Add1	no					
Add2	no					
Add3	no					
Mult1	no					
Mult2	yes	div	(ROB 3)	(ROB 1)		

Register Status (Qj)

F0	F2	F4	F6	F8	F10	F12…
(ROB 3)	(ROB 2)		ROB 6	ROB 4	ROB 5	

Example, 7

subtd has committed

~ Reorder Buffer

Instruction	Issue	Execute	Write Result	Commit
~~ld F6, 34(R2)~~	yes	yes	yes	yes
~~ld F2, 45(R3)~~	yes	yes	yes	yes
~~multd F0, F2, F4~~	yes	yes	yes	yes
~~subd F8, F6, F2~~	yes	yes	yes	yes
divd F10, F0, F6	yes	yes		
addd F6, F8, F2	yes	yes	yes	

Reservations Stations

Name	Busy	Op	V_j	V_k	Q_j	Q_k
Add1	no					
Add2	no					
Add3	no					
Mult1	no					
Mult2	yes	div	(ROB 3)	(ROB 1)		

Register Status (Qj)

F0	F2	F4	F6	F8	F10	F12…
(ROB 3)	(ROB 2)		ROB 6	(ROB 4)	ROB 5	

Comparisons

- degree of pipelining
  - **Alpha, UltraSPARC, Pentium Pro:** most superpipelined
  - **MIPS R10K:** old style
- number of pipelines
  - **MIPS R10K:** integer, memory access, floating point all decoupled
  - **UltraSPARC 1:** integrated integer & floating point
  - **Pentium Pro:** integrated L1 & L2 data cache access