Tomasulo’s algorithm

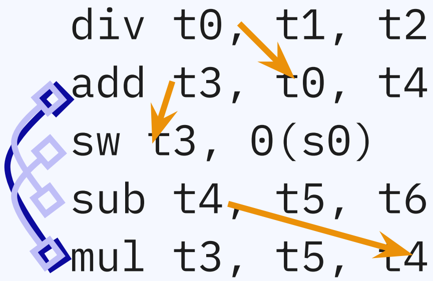
Tomasulo’s algorithm intuition

Developed by Robert Tomasulo for IBM 360/91

Minimize RAW hazards by tracking data dependences and reordering

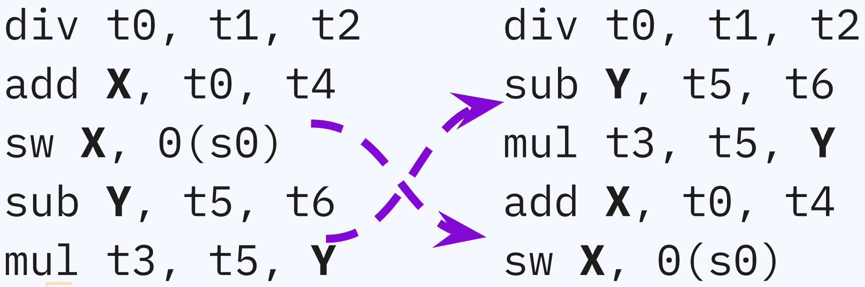
Minimize WAR and WAW hazards by *register renaming*

Limited to code within basic blocks (we’ll come back to branching soon)



```
div t0, t1, t2
add t3, t0, t4
sw  t3, 0(s0)
sub t4, t5, t6
mul t3, t5, t4
```

What if the hardware had two temporary registers, X and Y?



```
div t0, t1, t2
add X, t0, t4
sw  X, 0(s0)
sub Y, t5, t6
mul t3, t5, Y
```

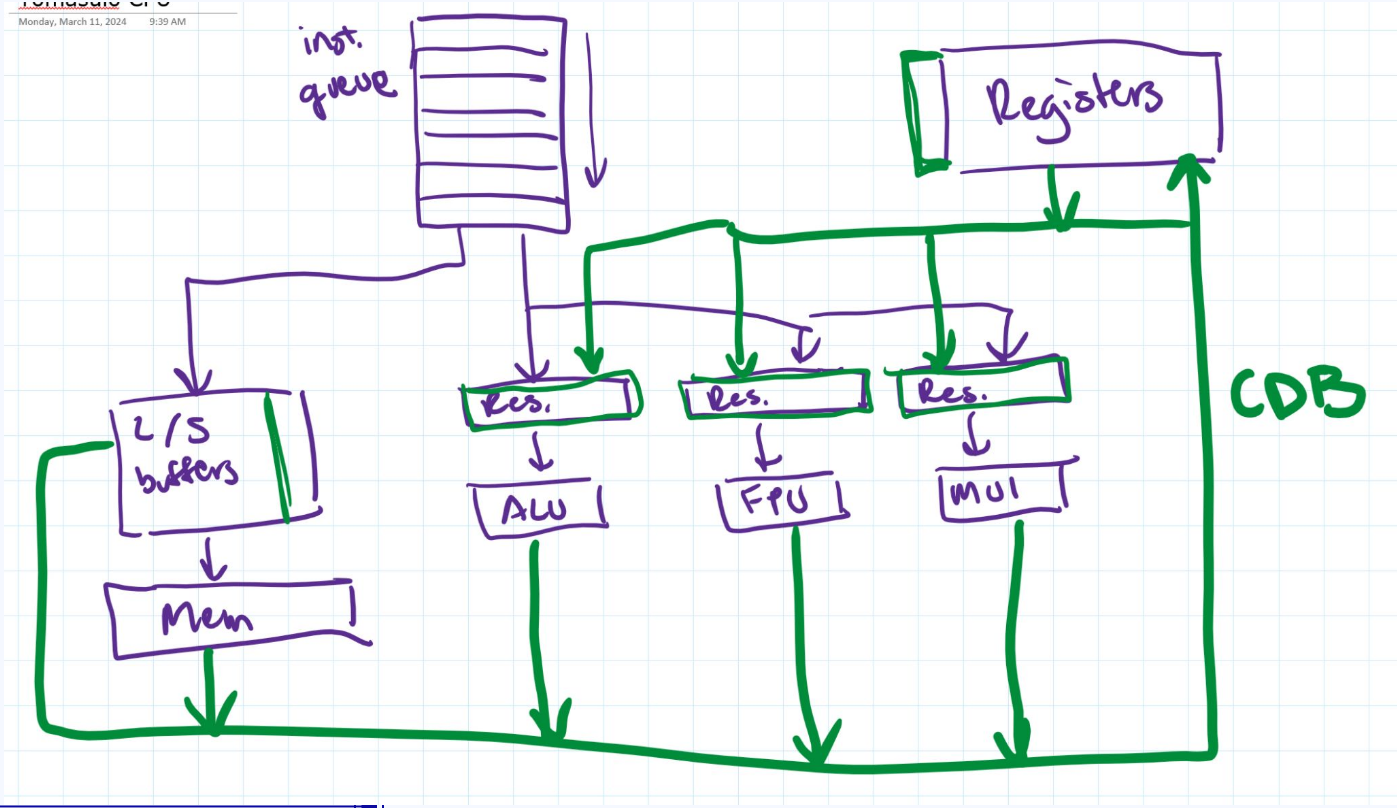
```
div t0, t1, t2
sub Y, t5, t6
mul t3, t5, Y
add X, t0, t4
sw  X, 0(s0)
```

now the div might have enough time to write to t0 before add needs it!

Worksheets for today

Link

The OOO CPU(sketch of H&P fig 3.6)



Tomasulo CPU

Monday, March 11, 2024 9:39 AM

Three stages of execution

Issue

- If reservation station available, issue to station; tracking source/dest data
- Source can be register (in which case, value is available) or reservation station (in which case, update res. station tags to wait for that result)
- If dest is register, update register file tags to say data will come from this res. station

Execute

- If operands available, execute instr; otherwise wait

Write

- Once result is ready, send on CDB (update registers, reservation stations)

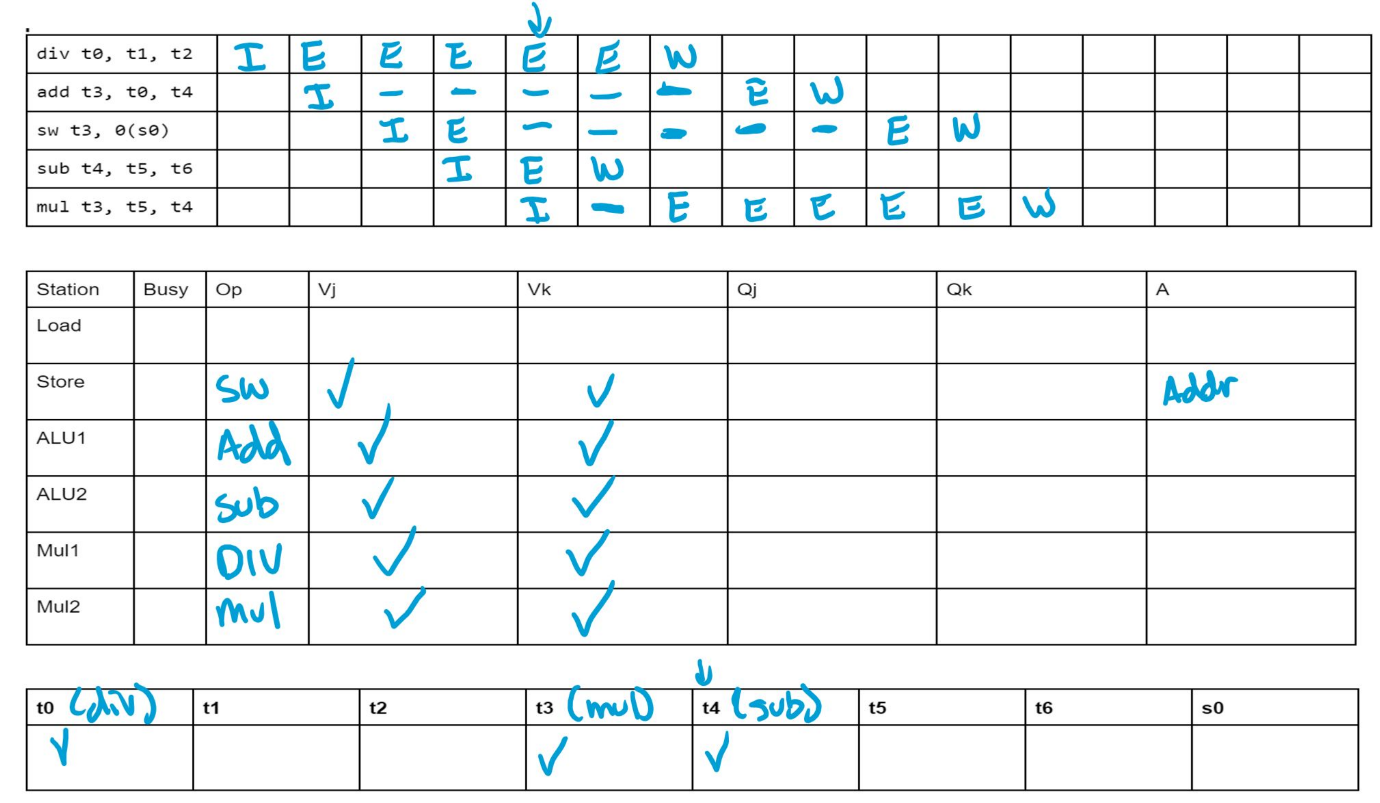
Execute latencies for our examples

Mul/div: 5 cycles

Load/store: 2 cycles (1 for addr, 1 for load/store)

ALU: 1 cycle

Example 1

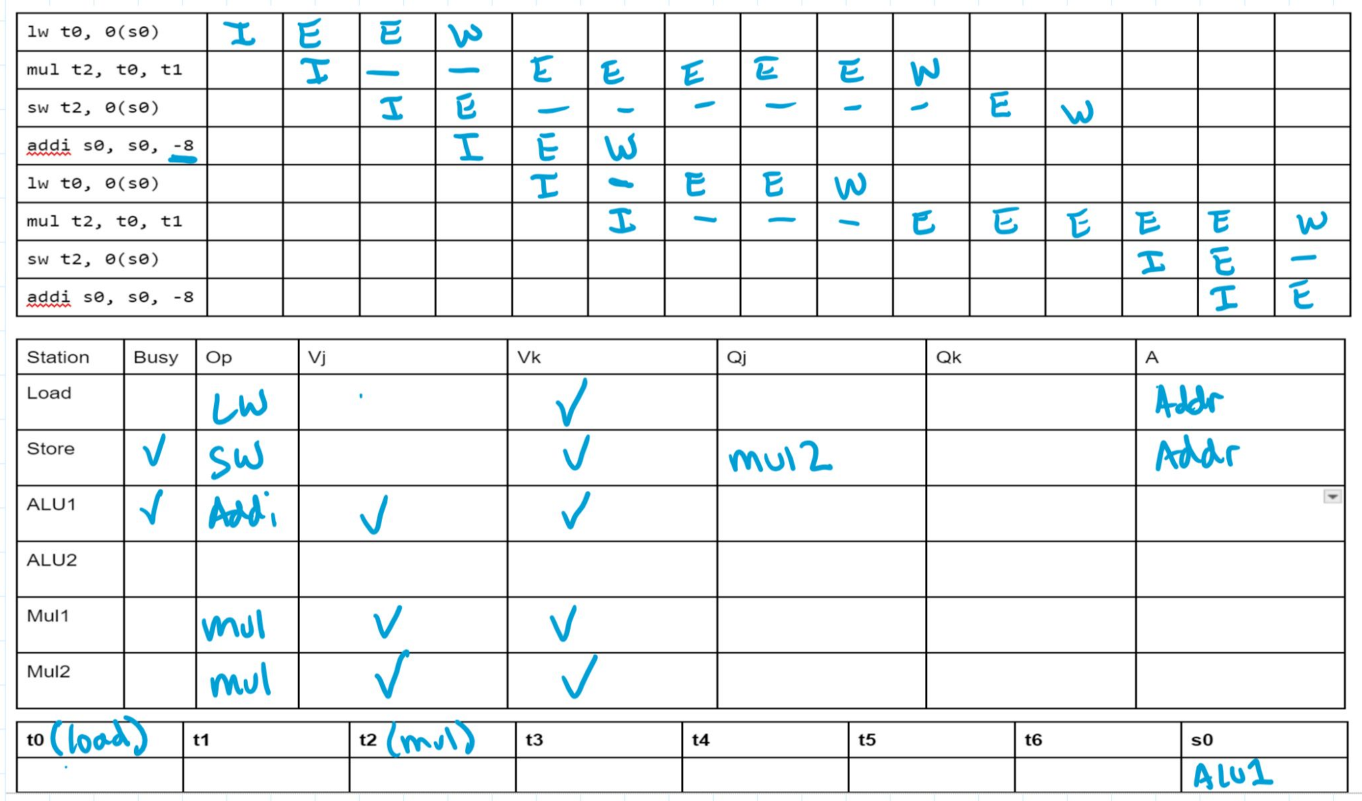


					↓											
div t0, t1, t2	I	E	E	E	E	E	W									
add t3, t0, t4		I	–	–	–	–	–	E	W							
sw t3, 0(s0)			I	E	–	–	–	–	–	E	W					
sub t4, t5, t6				I	E	W										
mul t3, t5, t4					I	–	E	E	E	E	E	W				

Station	Busy	Op	Vj	Vk	Qj	Qk	A
Load							
Store		sw	✓	✓			Addr
ALU1		Add	✓	✓			
ALU2		sub	✓	✓			
Mul1		DIV	✓	✓			
Mul2		mul	✓	✓			

t0 (div)	t1	t2	t3 (mul)	↓ t4 (sub)	t5	t6	s0
✓			✓	✓			

Example 2: unrolled loop



lw t0, 0(s0)	I	E	E	W											
mul t2, t0, t1		I	—	—	E	E	E	E	E	W					
sw t2, 0(s0)			I	E	—	—	—	—	—	—	E	W			
addi s0, s0, -8				I	E	W									
lw t0, 0(s0)					I	—	E	E	W						
mul t2, t0, t1						I	—	—	—	E	E	E	E	E	W
sw t2, 0(s0)													I	E	—
addi s0, s0, -8														I	E

Station	Busy	Op	Vj	Vk	Qj	Qk	A
Load		LW		✓			Addr
Store	✓	SW		✓	mul2		Addr
ALU1	✓	Addi	✓	✓			
ALU2							
Mul1		mul	✓	✓			
Mul2		mul	✓	✓			

t0 (load)	t1	t2 (mul)	t3	t4	t5	t6	s0
							ALU1