

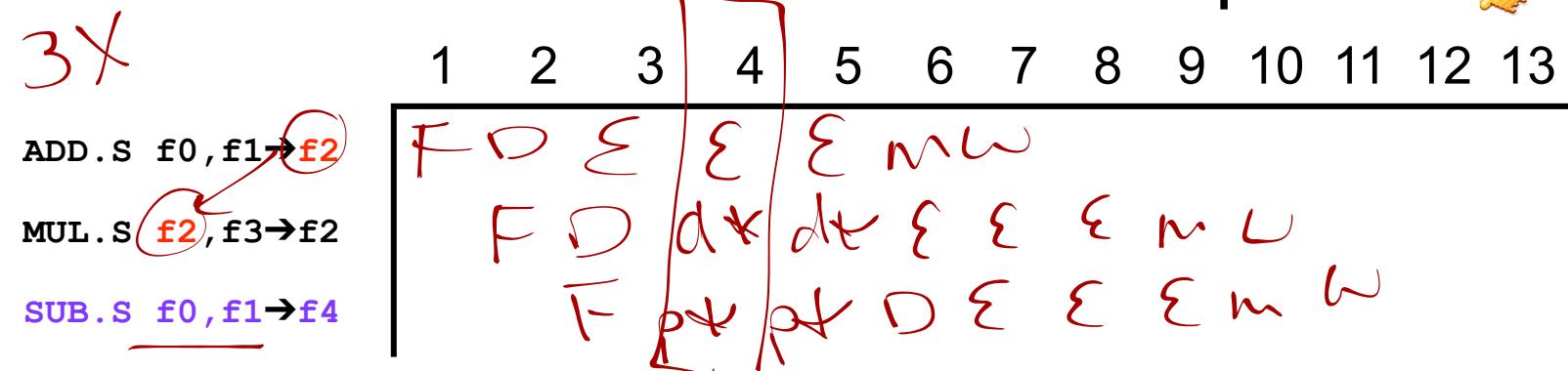
ECE 552: Dynamic Scheduling

Natalie Enright Jerger

Lecture notes based on slides created by Amir Roth of University of Pennsylvania with sources that included University of Wisconsin slides by Mark Hill, Guri Sohi, Jim Smith, and David Wood.

Lecture notes enhanced by Milo Martin, Mark Hill, and David Wood with sources that included Profs. Asanovic, Falsafi, Hoe, Lipasti, Shen, Smith, Sohi, Vijaykumar, and Wood

The Problem With In-Order Pipelines



- What's happening in cycle 4?

MUL.S stalls due to RAW hazard

- ok - this is fundamental problem

SUB.S stalls due to pipeline hazard

why? SUB.S can't complete decode b/c

MUL.S is in its way

That is the only reason & it's not a fundamental one

- Why can't SUB.S go into D in cycle 4 and E+ in cycle 5?

Static Instruction Scheduling

- **Issue:** time @ which insns begin execute
 - **Schedule:** order in which insns execute
 - **Scheduling:** re-arranging insns to enable rapid issue
- Static: by Compiler
- Requires knowledge of pipeline & program dependencies
- Requires large scheduling scope full of independent insns
- Ex: Loop unrolling - increases scope for loops

Motivation: Dynamic Scheduling

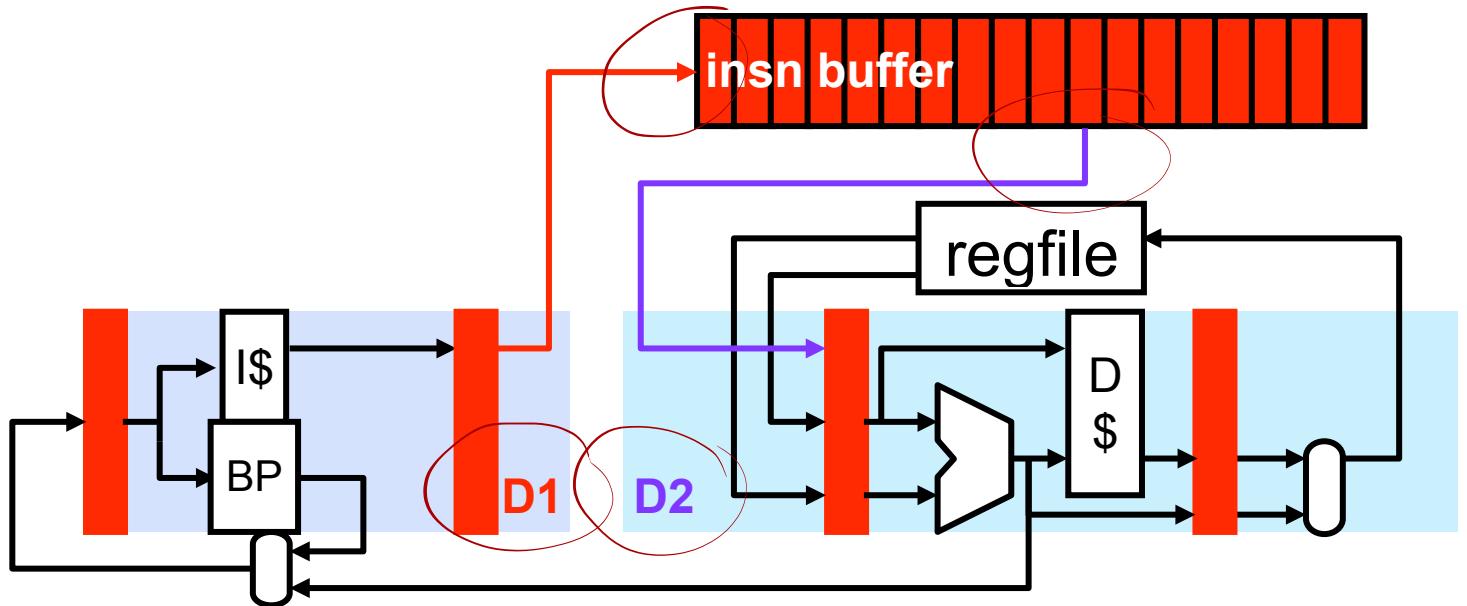
- **Dynamic scheduling (out-of-order execution)**
 - Execute insns in non-sequential (non-VonNeumann) order...
 - + Reduce RAW stalls
 - + Increase pipeline and functional unit (FU) utilization
 - Original motivation was to increase FP unit utilization

- + Expose more opportunities for parallel issue (ILP)
 - Not in-order → can be in parallel
- ...but make it appear like sequential execution
 - Important
 - But difficult

Before We Continue

- If we can do this in software...
- ...why build complex (slow-clock, high-power) hardware?
 - + Performance portability
 - Don't want to recompile for new machines
 - + more info available
 - memory addresses, branch directions
cache misses
 - + more registers available
 - compiler might not have enough
to fix WAR/WAW hazards
 - + easier to speculate & recover from
mis-speculation
 - Flush instead of recovery code
 - But compiler has a much larger scope
 - compiler does as much as it can
 - hardware does rest

Instruction Buffer

- Trick: **insn buffer** (many names for this buffer)
 - Basically a bunch of latches for holding insns
 - Gives us our scheduling scope
- Split D into two pieces
 - D1: Accumulate decoded insn in buffer
In-order
 - D2: Buffer sends insns down rest of pipeline out of order!

Scheduling Algorithm I: Tomasulo

- **Tomasulo's algorithm**
 - **Register renaming**: removes WAR/WAW hazards
 - **Reservation stations (RS)**: instruction buffer
 - **Common data bus (CDB)**: broadcasts results to RS
- First implementation: IBM 360/91 [1967]
 - Dynamic scheduling for FP units only

- Bypassing
- Our example: “Simple Tomasulo”
 - Dynamic scheduling for everything, including load/store
 - 2 examples (1 in class, 1 podcast)
 - Some inconsistencies terminology in podcast (diff textbook)

Recall dependences

- Dependences are a property of programs
 - whether dependence results in a hazard/stall is a property of pipeline organization
- insn j follows insn i in program order
- True/data dependence – 
 - insn i produces a result that may be used by insn j
 - insn j is data dependent on insn k and insn k is data dependent on insn i
 - Data dependent insn cannot execute simultaneously or be completely overlapped

Name dependences

- Occur when 2 insns use same register/mem location but there is no data flow b/t the insns

- Anti dependence - WAR
 - insn j writes a register that insn i reads
 - original ordering must be preserved to ensure i gets correct value
- Output dependence - WAW
 - insn i and insn j write the same reg/mem location
 - ordering must be preserved to ensure that value finally written corresponds to insn j

Register Renaming

- **Register renaming (in hardware)**
 - Change register names to eliminate WAR/WAW hazards
 - An elegant idea (like caching & pipelining)
 - Key: think of registers ($r1, f0\dots$) as **names**, not **storage locations**
 - Can have more locations than names
 - Can have multiple active versions of same name
- How does it work?
 - **Map-table**: maps names to most recent locations
 - SRAM indexed by name
 - on a write: allocate new location, note in map table
 - on a read: find location of most recent write via map table lookup

Small detail: must deallocate locations at some point

- Example

- Names: r_1, r_2, r_3, r_4
- Locations: $p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8$
- Original mapping: $r_1 \rightarrow p_1, r_2 \rightarrow p_2, r_3 \rightarrow p_3, r_4 \rightarrow p_4$
 p_5-p_8 are “free”

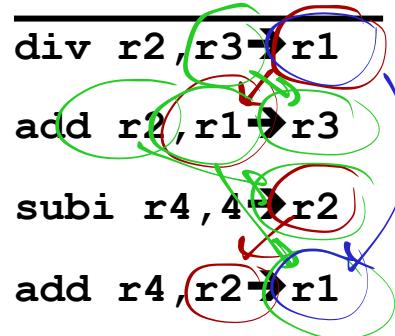
MapTable

| r_1 | r_2 | r_3 | r_4 |
|-------|-------|-------|-------|
| p_1 | p_2 | p_3 | p_4 |
| p_5 | p_2 | p_3 | p_4 |
| p_5 | p_2 | p_6 | p_4 |
| p_5 | p_7 | p_6 | p_4 |
| p_8 | p_7 | p_6 | p_4 |

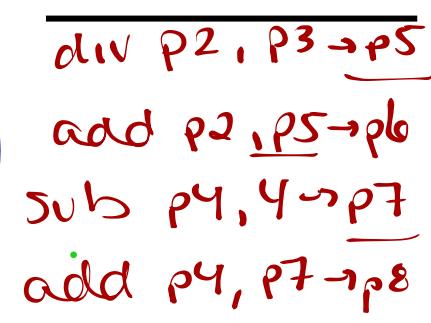
FreeList

| |
|----------------------|
| p_5, p_6, p_7, p_8 |
| p_6, p_7, p_8 |
| p_7, p_8 |
| p_8 |
| — |

Orig insns



Renamed insns

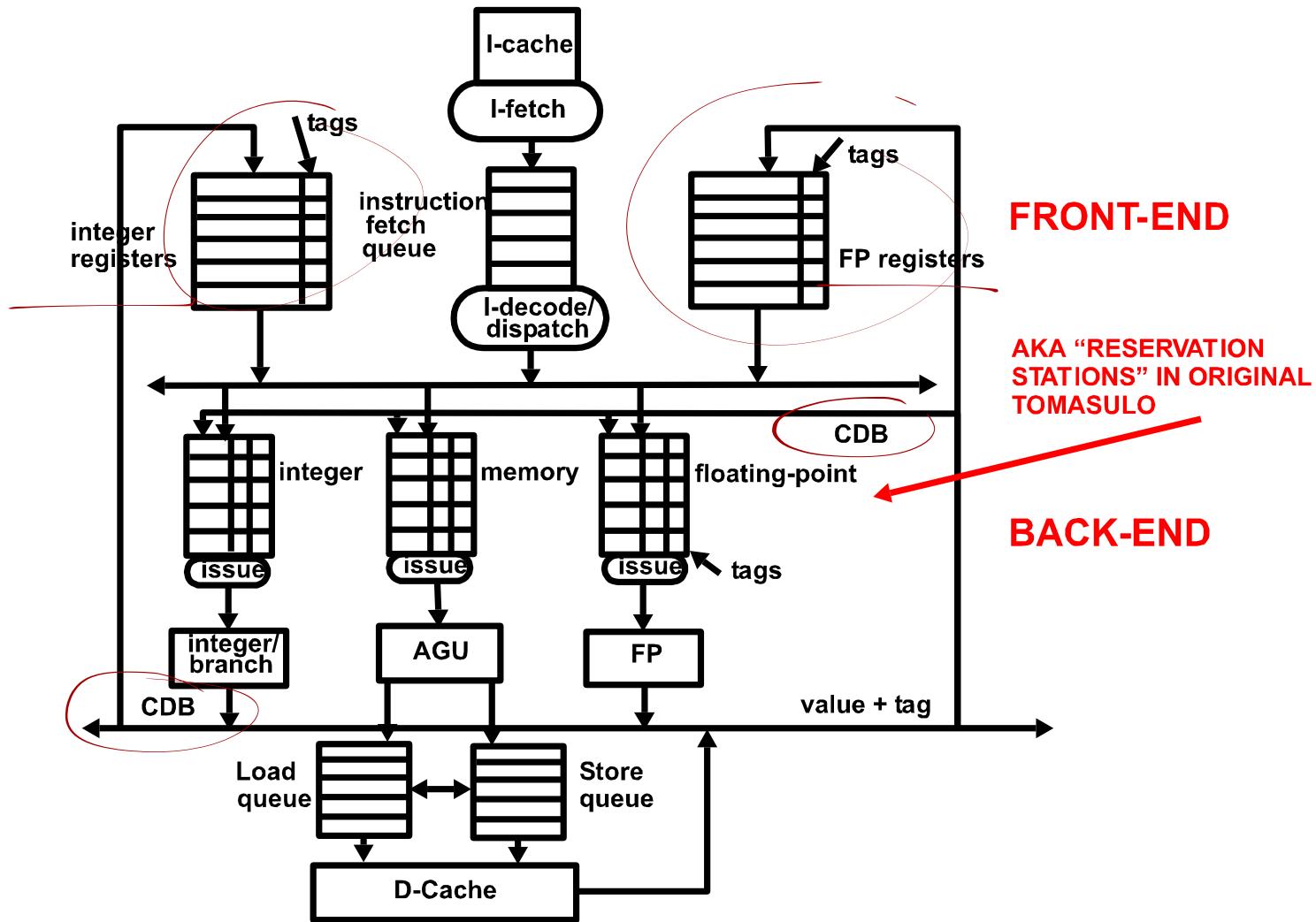


- Renaming

- + Removes **WAW** and **WAR** dependences
- + Leaves **RAW** intact!



Tomasulo Organization



Tomasulo Data Structures

- Reservation Stations (RS#)
 - FU, busy, op
 - **Q_j, Q_k**: source register tags (RS# of the RS that will produce this value)
 - **V_j, V_k**: source register value
- Map Table
 - **T**: tag (RS#) that will write this register
- Common Data Bus (CDB)

- Broadcasts $\langle RS\#, \text{value} \rangle$ of completed insns
- Tags interpreted as ready-bits++
 - $T == 0 \rightarrow \text{value is ready}$
 - $T != 0 \rightarrow \text{value is not ready, wait until CDB broadcasts T}$

New Pipeline

- New pipeline structure: F, **D**, **S**, X, **W**
 - F (fetch)
 - Same as before*
 - D (**Dispatch**)
 - Stall for structural hazard (RS)
 - Input reg ready? read value into RS
 - else read tag into RS
 - Set reg status (map table) - rename output reg
 - S (**Issue**)
 - wait for RAW hazards
 - else read reg values from RS & send to execute
 - X (execute) / memory (m)
 - Same - execute operation (includes mem)*
 - W (**writeback**)
 - Wait for structural hazard (CDB)
 - output reg status tag still matches?
 - clear, write result to reg

Page 10

CDB broadcasts to RS: tag match?
clear tag, copy value
Free RS entry

Value/Copy-Based Register Renaming

- Tomasulo-style register renaming
 - Called “**value-based**” or “**copy-based**”
 - **Names:** architectural registers
 - **Storage locations:** register file & reservation stations
 - values can & do exist in both
 - Register file holds master (most recent) values
 - + RS copies eliminate WAR hazards
 - Storage locations referred to internally by RS# tags
 - Register table translates names to tags
 - Tag == 0 → value is in reg file
 - Tag != 0 → value is not ready & is being computed by RS#
 - CDB broadcasts values with tags attached
 - so insns know what value they are looking for

Value-Based Renaming Example

I.S [r1+0] → f1 (allocated RS#2)
 $\text{mapTable}[r1] == 0 \rightarrow RS[2].VK = \text{regfile}[r1]$
 $\text{mapTable}[f1] = RS\#2$

MUL.S f0, f1 → f2 (allocated RS#4)
 $\text{maptable}[f_0] == 0 \rightarrow RS[4].V_j = RF[f_0]$
 $\text{maptable}[f_1] == RS\#2 \rightarrow RS[4].Q_k = RS\#2$

ADD.S f7, f8 → f0

Can write regfile [f0] before the MUL.S executes

L.S [r1+4] → f1

| Map Table | |
|-----------|-------|
| Reg | Value |
| f0 | |
| f1 | RS#2 |
| f2 | RS#4 |
| r1 | |

Can write regfile [f1] before MUL.S executes

Can write regfile [f1] before 1st L.S.

Reservation Stations

| T | FU | busy | op | Qj | Qk | Vj | Vk |
|---|-----|------|-------|----|------|------|------|
| 2 | LD | yes | L.S | - | - | - | [r1] |
| 4 | FPJ | yes | MUL.S | - | RS#2 | [f0] | - |

New Pipeline Diagram

- Alternative pipeline diagram

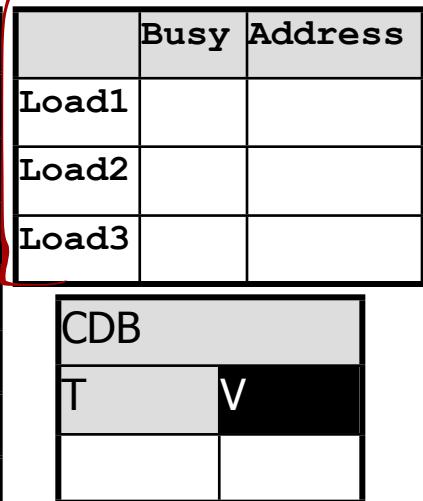
- Down: insns
- Across: pipeline stages
- In boxes: cycles
- Basically: stages ↔ cycles
- Convenient for out-of-order

| Insn Status | | | | |
|--------------------|---|---|---|---|
| Insn | D | S | X | W |
| L.S 34 (r2) → f6 | | | | |
| L.S 45 (r3) → f2 | | | | |
| MUL.S f2, f4 → f0 | | | | |
| SUB.S f6, f2 → f8 | | | | |
| DIV.S f0, f6 → f10 | | | | |
| ADD.S f8, f2 → f6 | | | | |

Tomasulo Data Structures

Insn Status

| Insn | D | S | X | W |
|--------------------|---|---|---|---|
| LD.S 34(r2) → f6 | | | | |
| LD.S 45(r3) → f2 | | | | |
| MUL.S f2, f4 → f0 | | | | |
| SUB.S f6, f2 → f8 | | | | |
| DIV.S f0, f6 → f10 | | | | |
| ADD.S f8, f2 → f6 | | | | |



Reservation Stations

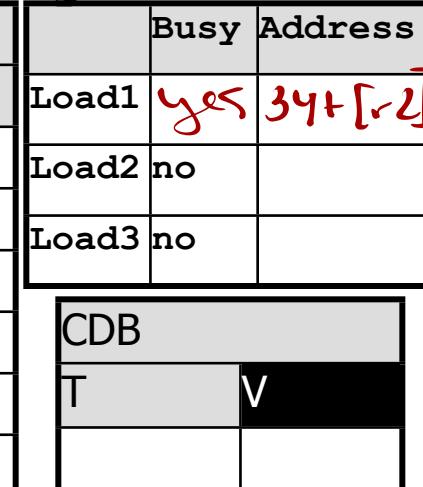
| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|----|----|----|----|----|
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | no | | | | | |
| Mult2 | no | | | | | |

Dispatch: Allocate RS, input ready → copy into RS
Set reg status for output

Tomasulo: Cycle 1

Insn Status

| Insn | D | S | X | W |
|--------------------|----|---|---|---|
| LD.S 34(r2) → f6 | C1 | | | |
| LD.S 45(r3) → f2 | | | | |
| MUL.S f2, f4 → f0 | | | | |
| SUB.S f6, f2 → f8 | | | | |
| DIV.S f0, f6 → f10 | | | | |
| ADD.S f8, f2 → f6 | | | | |



Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|----|----|----|----|----|
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | no | | | | | |
| Mult2 | no | | | | | |

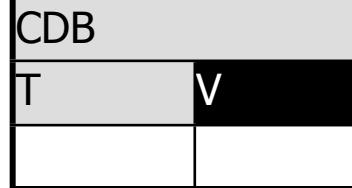
| Reg | Table |
|-----|-------|
| f0 | |
| f2 | |
| f4 | |
| f6 | |
| f8 | |
| f10 | |
| ... | |
| r2 | |
| r3 | |

| Op | Cycles |
|-----|--------|
| ADD | 2 |
| L.S | 2 |
| MUL | 10 |
| DIV | 40 |

Tomasulo: Cycle 2

| Insn Status | | | | |
|--------------------|----|----|---|---|
| Insn | D | S | X | W |
| LD.S 34 (r2) → f6 | c1 | C2 | | |
| LD.S 45 (r3) → f2 | C2 | | | |
| MUL.S f2, f4 → f0 | | | | |
| SUB.S f6, f2 → f8 | | | | |
| DIV.S f0, f6 → f10 | | | | |
| ADD.S f8, f2 → f6 | | | | |

| | Busy | Address | Map Table |
|-------|------|-----------|-----------|
| Load1 | Yes | 34 + [R2] | Reg T |
| Load2 | Yes | 45 + [r3] | f0 |
| Load3 | No | | f2 Load2 |
| | | | f4 |
| | | | f6 Load1 |
| | | | f8 |
| | | | f10 |
| | | | ... |
| r2 | | | |
| r3 | | | |



Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|----|----|----|----|----|
| Add1 | No | | | | | |
| Add2 | No | | | | | |
| Add3 | No | | | | | |
| Mult1 | No | | | | | |
| Mult2 | No | | | | | |

| Ex latencies | |
|--------------|--------|
| Op | Cycles |
| ADD | 2 |
| L.S | 2 |
| MUL | 10 |
| DIV | 40 |

can have multiple loads outstanding

Issue(s): Wait for RAW hazards

- If no RAW → read values from RS
to send to functional unit

Tomasulo: Cycle 3

| Insn Status | | | | | Busy | Address | Map Table | |
|--------------------|----|----|---|---|-------|---------|-----------|----------|
| Insn | D | S | X | W | | | Reg | T |
| LD.S 34 (r2) → f6 | c1 | c2 | | | Load1 | Yes | 34+ [R2] | f0 MULT1 |
| LD.S 45 (r3) → f2 | c2 | C3 | | | Load2 | Yes | 45+ [R3] | f2 Load2 |
| MUL.S f2, f4 → f0 | C3 | | | | Load3 | No | | f4 |
| SUB.S f6, f2 → f8 | | | | | CDB | | | f6 Load1 |
| DIV.S f0, f6 → f10 | | | | | T | V | | f8 |
| ADD.S f8, f2 → f6 | | | | | | | | f10 |
| | | | | | | | ... | |
| | | | | | | | r2 | |
| | | | | | | | r3 | |

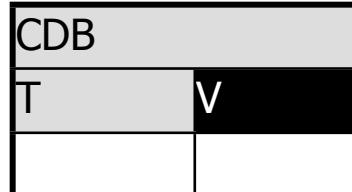
| Reservation Stations | | | | | | |
|----------------------|------|-------------|----|----|----|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | yes | MUL.S Load1 | | - | - | [F4] |
| Mult2 | no | | | | | |

F2 renamed → mult will get value from Load2 not from f2
 F4 is ready so value is written to V_k in RS
 1st LD.S starts executing - 2 cycles

Tomasulo: Cycle 4

| Insn Status | | | | |
|--------------------|----|----|----|---|
| Insn | D | S | X | W |
| LD.S 34 (r2) → f6 | c1 | c2 | C4 | |
| LD.S 45 (r3) → f2 | c2 | c3 | | |
| MUL.S f2, f4 → f0 | c3 | | | |
| SUB.S f6, f2 → f8 | C4 | | | |
| DIV.S f0, f6 → f10 | | | | |
| ADD.S f8, f2 → f6 | | | | |

| | Busy | Address | Map Table |
|-------|------|---------|-----------|
| Load1 | Yes | 34+[R2] | Reg T |
| Load2 | Yes | 45+[R3] | f0 Mult1 |
| Load3 | No | | f2 Load2 |
| | | | f4 |
| | | | f6 Load1 |
| | | | f8 Add1 |
| | | | f10 |
| | | | ... |
| | | | r2 |
| | | | r3 |



Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|-------|-------|-------|----|------|
| Add1 | yes | SUB.S | Load1 | Load2 | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Load2 | | | [F4] |
| Mult2 | no | | | | | |

2nd LD.S starts executing

both f6 & f2 renamed → tags in Qj & Qk

| Ex latencies | |
|--------------|--------|
| Op | Cycles |
| ADD | 2 |
| L.S | 2 |
| MUL | 10 |
| DIV | 40 |

Tomasulo: Cycle 5

| Insn Status | | | | |
|--------------------|----|----|----|----|
| Insn | D | S | X | W |
| LD.S 34 (r2) → f6 | c1 | c2 | c4 | C5 |
| LD.S 45 (r3) → f2 | c2 | c3 | C5 | |
| MUL.S f2, f4 → f0 | c3 | | | |
| SUB.S f6, f2 → f8 | c4 | | | |
| DIV.S f0, f6 → f10 | C5 | | | |
| ADD.S f8, f2 → f6 | | | | |

| | Busy | Address |
|-------|------|-----------|
| Load1 | no | |
| Load2 | Yes | 45 + [R3] |
| Load3 | no | |

| CDB | |
|-------|--------|
| T | V |
| Load1 | mem(1) |

| Map Table | |
|-----------|-------|
| Reg | T |
| f0 | Mult1 |
| f2 | Load2 |
| f4 | |
| f6 | |
| f8 | Add1 |
| f10 | MULT2 |
| ... | |
| r2 | |
| r3 | |

Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|-------|-------|-------|--------|------|
| Add1 | Yes | SUB.S | | Load2 | mem(1) | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Load2 | | | [F4] |
| Mult2 | yes | DIV.S | MULT1 | | | [F6] |

Writeback - CDB available?
 output tag matches? update reg value,
 clear tag

CDB broadcasts to RS - anyone waiting on Load1
 (tag match)? → clear tag, write value into
 V field

Load1 - free RS entry

For DIV.S - no pending instruction writing file
 (Register written in 1st half of cycle)

Tomasulo: Cycle 6

| Insn Status | | | | |
|--------------------|----|----|----|----|
| Insn | D | S | X | W |
| LD.S 34(r2) → f6 | c1 | c2 | c4 | c5 |
| LD.S 45(r3) → f2 | c2 | c3 | c5 | c6 |
| MUL.S f2, f4 → f0 | c3 | c6 | | |
| SUB.S f6, f2 → f8 | c4 | c6 | | |
| DIV.S f0, f6 → f10 | c5 | | | |
| ADD.S f8, f2 → f6 | c6 | | | |

| | Busy | Address | Map Table |
|-------|------|---------|-----------|
| Load1 | no | | Reg T |
| Load2 | no | | f0 Mult1 |
| Load3 | no | | f2 |
| | | | f4 |
| | | | f6 ADD2 |
| | | | f8 Add1 |
| | | | f10 Mult2 |
| | | | ... |
| | | | r2 |
| | | | r3 |

| CDB | |
|-------|--------|
| T | V |
| LOAD2 | mem(2) |

Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|-------|-------|----|--------|--------|
| Add1 | Yes | SUB.S | | | Mem(1) | mem(2) |
| Add2 | yes | ADD.S | ADD1 | | | [F2] |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | | | mem(2) | [F4] |
| Mult2 | Yes | DIV.S | Mult1 | | | [F6] |

| Ex latencies | |
|--------------|--------|
| Op | Cycles |
| ADD | 2 |
| L.S | 2 |
| MUL | 10 |
| DIV | 40 |

Load2 writing result - anyone waiting?

- mult1 & add1 - values written from DB into Vj, Vk
- mult1 & add1 can issue - start executing next cycle

maptable[f2] cleared → value in f2, no pending insn writing f2

Tomasulo: Cycle 7

no change

DIV.S & ADD.S - waiting on values - can't issue

MUL.S & SUB.S - executing

Tomasulo: Cycle 8

| Insn Status | | | | |
|--------------------|----|----|----|----|
| Insn | D | S | X | W |
| LD.S 34(r2) → f6 | c1 | c2 | c4 | c5 |
| LD.S 45(r3) → f2 | c2 | c3 | c5 | c6 |
| MUL.S f2, f4 → f0 | c3 | c6 | | |
| SUB.S f6, f2 → f8 | c4 | c6 | C8 | |
| DIV.S f0, f6 → f10 | c5 | | | |
| ADD.S f8, f2 → f6 | c6 | | | |

| | Busy | Address |
|-------|------|---------|
| Load1 | | |
| Load2 | | |
| Load3 | | |
| CDB | | |
| T | V | |
| | | |

Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|-------|-------|----|--------|--------|
| Add1 | Yes | SUB.S | | | Mem(1) | Mem(2) |
| Add2 | Yes | ADD.S | Add1 | | | [F2] |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | | | Mem(2) | [F4] |
| Mult2 | Yes | DIV.S | Mult1 | | | [F6] |

| Map Table | |
|-----------|-------|
| Reg | T |
| f0 | Mult1 |
| f2 | |
| f4 | |
| f6 | Add2 |
| f8 | Add1 |
| f10 | Mult2 |
| ... | |
| r2 | |
| r3 | |

| Ex latencies | |
|--------------|--------|
| Op | Cycles |
| ADD | 2 |
| L.S | 2 |
| MUL | 10 |
| DIV | 40 |

Tomasulo: Cycle 9

| Insn Status | | | | |
|--------------------|----|------|----|------|
| Insn | D | S | X | W |
| LD.S 34 (r2) → f6 | c1 | c2 | c4 | c5 |
| LD.S 45 (r3) → f2 | c2 | c3 | c5 | c6 |
| MUL.S f2, f4 → f0 | c3 | c6 | | |
| SUB.S f6, f2 → f8 | c4 | c6 | c8 | (C9) |
| DIV.S f0, f6 → f10 | c5 | | | |
| ADD.S f8, f2 → f6 | c6 | (C9) | | |

| | Busy | Address |
|-------|------|---------|
| Load1 | | |
| Load2 | | |
| Load3 | | |

| CDB | |
|------|---------|
| T | V |
| Add1 | (m - m) |

| Map Table | |
|-----------|---------|
| Reg T | |
| f0 | Mult1 |
| f2 | |
| f4 | |
| f6 | Add2 |
| f8 | (m - m) |
| f10 | Mult2 |
| ... | |
| r2 | |
| r3 | |

Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|-------|-------|----|---------|------|
| Add1 | | | | | | |
| Add2 | Yes | ADD.S | | | (m - m) | [F2] |
| Add3 | No | | | | | |
| Mult1 | Yes | MUL.S | | | Mem (2) | [F4] |
| Mult2 | Yes | DIV.S | Mult1 | | | [F6] |

ADD.S - both operands available - can issue.

| Ex latencies | |
|--------------|--------|
| Op | Cycles |
| ADD | 2 |
| L.S | 2 |
| MUL | 10 |
| DIV | 40 |

Tomasulo: Cycle 10

no change

ADD.S starts executing
DIV.S still waiting

Tomasulo: Cycle 11

| Insn Status | | | | |
|--------------------|----|----|-----|----|
| Insn | D | S | X | W |
| LD.S 34 (r2) → f6 | c1 | c2 | c4 | c5 |
| LD.S 45 (r3) → f2 | c2 | c3 | c5 | c6 |
| MUL.S f2, f4 → f0 | c3 | c6 | | |
| SUB.S f6, f2 → f8 | c4 | c6 | c8 | c9 |
| DIV.S f0, f6 → f10 | c5 | | | |
| ADD.S f8, f2 → f6 | c6 | c9 | C11 | |

| | Busy | Address | Map Table |
|-------|------|---------|-----------|
| Load1 | | | Reg T |
| Load2 | | | f0 Mult1 |
| Load3 | | | f2 |
| | | | f4 |
| | | | f6 Add2 |
| | | | f8 |
| | | | f10 Mult2 |
| | | | ... |
| r2 | | | |
| r3 | | | |

| Reservation Stations | | | | | | |
|----------------------|------|---------|-------|----|---------|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | Yes | ADD . S | | | (M-M) | [F2] |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL . S | | | Mem (2) | [F4] |
| Mult2 | Yes | DIV . S | Mult1 | | | [F6] |

Tomasulo: Cycle 12

| Insn Status | | | | |
|--------------------|----|----|-----|-------|
| Insn | D | S | X | W |
| LD.S 34 (r2) → f6 | c1 | c2 | c4 | c5 |
| LD.S 45 (r3) → f2 | c2 | c3 | c5 | c6 |
| MUL.S f2, f4 → f0 | c3 | c6 | | |
| SUB.S f6, f2 → f8 | c4 | c6 | c8 | c9 |
| DIV.S f0, f6 → f10 | c5 | | | |
| ADD.S f8, f2 → f6 | c6 | c9 | c11 | C 2 |

| | Busy | Address |
|-------|------|---------|
| Load1 | | |
| Load2 | | |
| Load3 | | |

| CDB | |
|-----|---------------|
| T | V |
| | Add12(m-m+F2) |

| Map Table | |
|-----------|-------|
| Reg | T |
| f0 | Mult1 |
| f2 | |
| f4 | |
| f6 | |
| f8 | |
| f10 | Mult2 |
| ... | |
| r2 | |
| r3 | |

| Ex latencies | |
|--------------|--------|
| Op | Cycles |
| ADD | 2 |
| L.S | 2 |
| MUL | 10 |
| DIV | 40 |

Tomasulo: Cycle 13-15

no change

MUL.S still executing - 10 cycle op

Tomasulo: Cycle 16

| Insn Status | | | | |
|--------------------|----|----|-----|-----|
| Insn | D | S | X | W |
| LD.S 34 (r2) → f6 | c1 | c2 | c4 | c5 |
| LD.S 45 (r3) → f2 | c2 | c3 | c5 | c6 |
| MUL.S f2, f4 → f0 | c3 | c6 | C16 | |
| SUB.S f6, f2 → f8 | c4 | c6 | c8 | c9 |
| DIV.S f0, f6 → f10 | c5 | | | |
| ADD.S f8, f2 → f6 | c6 | c9 | c11 | c12 |

| | Busy | Address | Map Table |
|-------|------|---------|-----------|
| Load1 | | | Reg T |
| Load2 | | | f0 Mult1 |
| Load3 | | | f2 |
| | | | f4 |
| | | | f6 |
| | | | f8 |
| | | | f10 Mult2 |
| | | | ... |
| r2 | | | |
| r3 | | | |

CDB

| | |
|---|---|
| T | V |
|---|---|

Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|-------|-------|----|---------|------|
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | | | Mem (2) | [F4] |
| Mult2 | Yes | DIV.S | Mult1 | | | [F6] |

Tomasulo: Cycle 17

| Insn Status | | | | | Busy | Address | Map Table |
|--------------------|----|-----|-----|-----|------|---------|-----------|
| Insn | D | S | X | W | | | Reg |
| LD.S 34 (r2) → f6 | c1 | c2 | c4 | c5 | | | T |
| LD.S 45 (r3) → f2 | c2 | c3 | c5 | c6 | | | f0 |
| MUL.S f2, f4 → f0 | c3 | c6 | c16 | C17 | | | f2 |
| SUB.S f6, f2 → f8 | c4 | c6 | c8 | c9 | | | f4 |
| DIV.S f0, f6 → f10 | c5 | C17 | | | | | f6 |
| ADD.S f8, f2 → f6 | c6 | c9 | c11 | c12 | | | f8 |

| Reservation Stations | | | | | | | |
|----------------------|------|-------|----|----|-------|-------|--|
| FU | busy | op | Qj | Qk | Vj | Vk | |
| Add1 | no | | | | | | |
| Add2 | no | | | | | | |
| Add3 | no | | | | | | |
| Mult1 | | | | | | | |
| Mult2 | Yes | DIV.S | | | MULT1 | MULT2 | |

| Ex latencies | |
|--------------|--------|
| Op | Cycles |
| ADD | 2 |
| L.S | 2 |
| MUL | 10 |
| DIV | 40 |

Fast Forward

- Nothing interesting happening...

DIV.S - 40 cycle op - starts executing in c18

Tomasulo: Cycle 57

| Insn Status | | | | | | Busy | Address | Map Table | |
|--------------------|----|-----|-----|-----|--|------|---------|-----------|---------|
| Insn | D | S | X | W | | | | Reg | Address |
| LD.S 34(r2) → f6 | c1 | c2 | c4 | c5 | | | | f0 | |
| LD.S 45(r3) → f2 | c2 | c3 | c5 | c6 | | | | f2 | |
| MUL.S f2, f4 → f0 | c3 | c6 | c16 | c17 | | | | f4 | |
| SUB.S f6, f2 → f8 | c4 | c6 | c8 | c9 | | | | f6 | |
| DIV.S f0, f6 → f10 | c5 | c17 | C57 | | | | | f8 | |
| ADD.S f8, f2 → f6 | c6 | c9 | c11 | c12 | | | | f10 | Mult2 |

| Reservation Stations | | | | | | |
|----------------------|------|-------|----|----|---------|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | no | | | | | |
| Mult2 | Yes | DIV.S | | | M* [F4] | [F6] |

Tomasulo: Cycle 58

| Insn Status | | | | | | Busy | Address | Map Table |
|--------------------|----|-----|-----|-----|--|------|---------|-----------|
| Insn | D | S | X | W | | | | Reg |
| LD.S 34 (r2) → f6 | c1 | c2 | c4 | c5 | | | | T |
| LD.S 45 (r3) → f2 | c2 | c3 | c5 | c6 | | | | f0 |
| MUL.S f2, f4 → f0 | c3 | c6 | c16 | c17 | | | | f2 |
| SUB.S f6, f2 → f8 | c4 | c6 | c8 | c9 | | | | f4 |
| DIV.S f0, f6 → f10 | c5 | c17 | c57 | CS8 | | | | f6 |
| ADD.S f8, f2 → f6 | c6 | c9 | c11 | c12 | | | | f8 |

| Reservation Stations | | | | | | |
|----------------------|------|----|----|----|----|----|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | no | | | | | |
| Mult2 | h6 | | | | | |

In-order dispatch, out of order execution & out of order completion

Tomasulo Summary

| Hazard | Tomasulo |
|-------------|-------------------------|
| Insn buffer | Stall in D - structural |
| FU | Wait in S - structural |
| RAW | Wait in S |
| WAR | none - renamed |
| WAW | none - renamed |

Dynamic Scheduling as Loop Unrolling

- Three steps of loop unrolling

- Step I: combine iterations

Increases scheduling scope - more flexibility

- Step II: pipeline schedule

Reduce impact of RAW hazards

- Step III: rename registers

Remove WAR/WAW violations that come from scheduling

Loop Example: SAX (SAXPY – PY)

- SAX** (Single-precision A X)

- Only because there won't be room in the diagrams for SAXPY

```
for (i=N; i>0; i++)
    Z[i] = A*X[i];
```

```

0: L.S X(r1), f1          // loop
1: MUL.S f0, f1, f2        // A in f0
2: S.S f4, Z(r1)
3: SUBI r1, 8, r1          // i in r1
4: BNEZ r1, Loop           //
```

- Consider two iterations

L.S, MUL.S, S.S, SUBI, L.S, MUL.S, S.S

Assumptions

- Multiply takes 4 clock

- Assume 1st load takes 8 clocks (cache miss), 2nd load takes 1 clock (hit)
- Assume R1 initially has value of 80

Tomasulo Loop Example: Cycle 1

| Insn Status | | D | S | X | W |
|-------------------|---|---|---|---|---|
| Insn | | | | | |
| L.S #0(r1) → f0 | ✓ | | | | |
| MUL.S f0, f2 → f4 | ✓ | | | | |
| S.S f4 → #0(r1) | | | | | |
| SUBI r1, 8 → r1 | | | | | |
| BNEZ r1, Loop | | | | | |
| L.S #0(r1) → f0 | ✓ | | | | |
| MUL.S f0, f2 → f4 | ✓ | | | | |
| S.S f4 → #0(r1) | | | | | |

| | Busy | Addr | T |
|-----|------|------|---|
| Ld1 | yes | 80 | |
| Ld2 | no | | |
| Ld3 | no | | |
| St1 | no | | |
| St2 | no | | |
| St3 | no | | |

| CDB |
|-----|
| T V |

| Reg Status |
|------------|
| Reg T |
| f0 Ld1 |
| f2 |
| f4 |
| f6 |
| f8 |
| f10 |
| ... |
| r1 |

| Reservation Stations | | | | | | |
|----------------------|------|----|----|----|----|----|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | no | | | | | |
| Mult2 | no | | | | | |

Tomasulo Loop Example: Cycle 2

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|----|---|---|-----|------|------|---|-----------|
| Insn | D | S | X | W | Ld1 | Yes | 80 | | Reg T |
| L.S #0(r1) → f0 | c1 | C2 | | | Ld2 | no | | | f0 Ld1 |
| MUL.S f0, f2 → f4 | C2 | | | | Ld3 | no | | | f2 |
| S.S f4 → #0(r1) | | | | | St1 | no | | | f4 MULT1 |
| SUBI r1, 8 → r1 | | | | | St2 | no | | | f6 |
| BNEZ r1, Loop | | | | | St3 | no | | | f8 |
| L.S #0(r1) → f0 | | | | | CDB | | | | f10 |
| MUL.S f0, f2 → f4 | | | | | T | V | | | ... |
| S.S f4 → #0(r1) | | | | | | | | | r1 |

| Reservation Stations | | | | | | |
|----------------------|------|------|-----|----|----|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | yes | MULS | Ld1 | | | [F2] |
| Mult2 | no | | | | | |

Tomasulo Loop Example: Cycle 3

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|----|---|---|-----|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | Yes | 80 | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | | | Ld2 | no | | | f0 Ld1 |
| MUL.S f0, f2 → f4 | c2 | | | | Ld3 | no | | | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | yes | 80 | Mult1 | f4 Mult1 |
| SUBI r1, 8 → r1 | | | | | St2 | no | | | f6 |
| BNEZ r1, Loop | | | | | St3 | no | | | f8 |
| L.S #0(r1) → f0 | | | | | CDB | | | | |
| MUL.S f0, f2 → f4 | | | | | T V | | | | |
| S.S f4 → #0(r1) | | | | | | | | | |

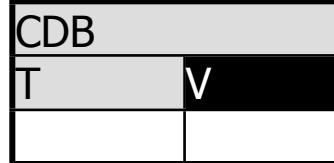
| Reservation Stations | | | | | | |
|----------------------|------|-------|-----|----|----|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Ld1 | | | [F2] |
| Mult2 | no | | | | | |

Rename sets up dataflow graph

Tomasulo Loop Example: Cycle 4

| Insn Status | | | | |
|-------------------|----|----|---|---|
| Insn | D | S | X | W |
| L.S #0(r1) → f0 | c1 | c2 | | |
| MUL.S f0, f2 → f4 | c2 | | | |
| S.S f4 → #0(r1) | c3 | | | |
| SUBI r1, 8 → r1 | C4 | | | |
| BNEZ r1, Loop | | | | |
| L.S #0(r1) → f0 | | | | |
| MUL.S f0, f2 → f4 | | | | |
| S.S f4 → #0(r1) | | | | |

| | Busy | Addr | T | Map Table |
|-----|------|------|-------|-----------|
| Ld1 | Yes | 80 | | |
| Ld2 | No | | | Reg T |
| Ld3 | No | | | f0 Ld1 |
| St1 | Yes | 80 | Mult1 | f2 |
| St2 | No | | | f4 Mult1 |
| St3 | No | | | |



| f6 | | | |
|-----|--------|--|--|
| f8 | | | |
| f10 | | | |
| ... | | | |
| r1 | AelolJ | | |

Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|-------|-----|----|------|------|
| Add1 | yes | Subi | | | [r1] | 8 |
| Add2 | No | | | | | |
| Add3 | No | | | | | |
| Mult1 | Yes | MUL.S | Ld1 | | | [F2] |
| Mult2 | No | | | | | |

Tomasulo Loop Example: Cycle 5

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|----|---|---|-----|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | Yes | 80 | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | | | Ld2 | no | | | f0 Ld1 |
| MUL.S f0, f2 → f4 | c2 | | | | Ld3 | no | | | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 Mult1 |
| SUBI r1, 8 → r1 | c4 | C5 | | | St2 | no | | | f6 |
| BNEZ r1, Loop | C5 | | | | St3 | no | | | f8 |
| L.S #0(r1) → f0 | | | | | | CDB | | | f10 |
| MUL.S f0, f2 → f4 | | | | | | T | V | | ... |
| S.S f4 → #0(r1) | | | | | | | | | r1 Add1 |

| Reservation Stations | | | | | | |
|----------------------|------|-------|-----|----|------|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | Yes | SUBI | | | [r1] | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Ld1 | | | [F2] |
| Mult2 | no | | | | | |

Tomasulo Loop Example: Cycle 6

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|----|----|---|-----|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | Yes | 80 | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | | | Ld2 | Yes | Add1 | → | f0 Ld2 |
| MUL.S f0, f2 → f4 | c2 | | | | Ld3 | no | | | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 Mult1 |
| SUBI r1, 8 → r1 | c4 | c5 | C6 | | St2 | no | | | f6 |
| BNEZ r1, Loop | c5 | | | | St3 | no | | | f8 |
| L.S #0(r1) → f0 | C6 | | | | CDB | | | | f10 |
| MUL.S f0, f2 → f4 | | | | | T | V | | | ... |
| S.S f4 → #0(r1) | | | | | | | | | r1 Add1 |

| Reservation Stations | | | | | | |
|----------------------|------|-------|-----|----|------|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | Yes | SUBI | | | [r1] | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Ld1 | | | [F2] |
| Mult2 | no | | | | | |

FO will never get result of Ld1 →
if b/c dependent insn (MUL.S) will
get value directly from Ld1 (CDB)

Branches: 2 options - ① No speculation &
branch must be resolved before younger
insn can execute
② Speculation - must have way to undo
- come back to this later

Lab 3: Assume perfect branch prediction

Tomasulo Loop Example: Cycle 7

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|----|----|----|------|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | Yes | 80 | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | | | Ld2 | Yes | 72 | | f0 Ld2 |
| MUL.S f0, f2 → f4 | c2 | | | | Ld3 | no | | | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 MULT2 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | C7 | St2 | no | | | f6 |
| BNEZ r1, Loop | c5 | C7 | | | St3 | no | | | f8 |
| L.S #0(r1) → f0 | c6 | | | | CDB | | | | f10 |
| MUL.S f0, f2 → f4 | C7 | | | | T | V | | | ... |
| S.S f4 → #0(r1) | | | | | Add1 | 72 | | | r1 |

| Reservation Stations | | | | | | |
|----------------------|------|-------|-----|----|----|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Ld1 | | | [F2] |
| Mult2 | Yes | MUL.S | Ld2 | | | [F2] |

RegFile decoupled from computation
1st & 2nd iterations
removed WAW hazards

Tomasulo Loop Example: Cycle 8

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|----|----|----|-----|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | Yes | 80 | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | | | Ld2 | Yes | 72 | | f0 Ld2 |
| MUL.S f0, f2 → f4 | c2 | | | | Ld3 | no | | | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 Mult2 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | Yes | 72 | MULT2 | f6 |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | no | | | f8 |
| L.S #0(r1) → f0 | c6 | c8 | | | CDB | | | | f10 |
| MUL.S f0, f2 → f4 | c7 | | | | T | V | | | ... |
| S.S f4 → #0(r1) | c8 | | | | | | | | r1 |

| Reservation Stations | | | | | | |
|----------------------|------|-------|-----|----|----|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Ld1 | | | [F2] |
| Mult2 | Yes | MUL.S | Ld2 | | | [F2] |

L.S #2 - issue now b/c branch resolved
 So not speculative
 We'll add speculation later

Tomasulo Loop Example: Cycle 9

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|----|----|----|-----|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | Yes | 80 | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | | | Ld2 | Yes | 72 | | f0 Ld2 |
| MUL.S f0, f2 → f4 | c2 | | | | Ld3 | no | | | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 Mult2 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | Yes | 72 | Mult2 | |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | no | | | f6 |
| L.S #0(r1) → f0 | c6 | c8 | | | CDB | | | | f8 |
| MUL.S f0, f2 → f4 | c7 | | | | T V | | | | f10 |
| S.S f4 → #0(r1) | c8 | | | | | | | | ... |
| | | | | | | | | | r1 Add1 |

Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|-------|-----|----|-------|------|
| Add1 | yes | Subi | | | [-] | 8 |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Ld1 | | | [F2] |
| Mult2 | Yes | MUL.S | Ld2 | | | [F2] |

2nd Subi dispatched - not shown



Tomasulo Loop Example: Cycle 10

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|----|-----|----|-----|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | Yes | 80 | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | C10 | | Ld1 | Yes | 80 | | f0 Ld2 |
| MUL.S f0, f2 → f4 | c2 | | | | Ld2 | Yes | 72 | | f2 |
| S.S f4 → #0(r1) | c3 | | | | Ld3 | no | | | f4 Mult2 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St1 | Yes | 80 | Mult1 | f6 |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St2 | Yes | 72 | Mult2 | f8 |
| L.S #0(r1) → f0 | c6 | c8 | | | St3 | no | | | f10 |
| MUL.S f0, f2 → f4 | c7 | | | | CDB | | | | |
| S.S f4 → #0(r1) | c8 | | | | T V | | | | |

| Reservation Stations | | | | | | | |
|----------------------|------|-------|-----|----|------|------|---------|
| FU | busy | op | Qj | Qk | Vj | Vk | |
| Add1 | Yes | SUBI | | | [r1] | | r1 Add1 |
| Add2 | no | | | | | | |
| Add3 | no | | | | | | |
| Mult1 | Yes | MUL.S | Ld1 | | | [F2] | |
| Mult2 | Yes | MUL.S | Ld2 | | | [F2] | |

Ld1 completing execution - who is waiting?
 BNEZ #2 dispatched - not shown

Tomasulo Loop Example: Cycle 11

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|-----|-----|-----|----------|------|------|-------|-------------|
| Insn | D | S | X | W | Ld1 | h6 | | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | c10 | c11 | Ld2 | Yes | 72 | | f0 Ld2 Add1 |
| MUL.S f0, f2 → f4 | c2 | c11 | | | Ld3 | yes | | Add1 | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 Mult2 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | Yes | 72 | Mult2 | f6 |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | no | | | f8 |
| L.S #0(r1) → f0 | c6 | c8 | c11 | | CDB | | | | f10 |
| MUL.S f0, f2 → f4 | c7 | | | | T V | | | | ... |
| S.S f4 → #0(r1) | c8 | | | | Ld1 [80] | | | | r1 Add1 |

Reservation Stations

| FU | busy | op | Qj | Qk | Vj | Vk |
|-------|------|-------|-----|----|-------|------|
| Add1 | Yes | SUBI | | | [r1] | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | | | m[80] | [F2] |
| Mult2 | Yes | MUL.S | Ld2 | | | [F2] |

Ld1 puts result on CDB → value written into Vj for Mult1

Ld2: completing execution - who is waiting
L.S #3 would dispatch (not shown)

Tomasulo Loop Example: Cycle 12

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|-----|-----|-----|-----|-------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | no | | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | c10 | c11 | Ld2 | | | | f0 Ld3 |
| MUL.S f0, f2 → f4 | c2 | c11 | | | Ld3 | Yes | | Add1 | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 Mult2 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | Yes | 72 | Mult2 | f6 |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | no | | | f8 |
| L.S #0(r1) → f0 | c6 | c8 | c11 | C12 | CDB | | | | f10 |
| MUL.S f0, f2 → f4 | c7 | C12 | | | T | V | | | ... |
| S.S f4 → #0(r1) | c8 | | | | Ld2 | M[72] | | | r1 Add1 |

| Reservation Stations | | | | | | |
|----------------------|------|-------|----|----|-------|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | Yes | SUBI | | | [r1] | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | | | M[80] | [F2] |
| Mult2 | Yes | MUL.S | ✓ | | M[72] | [F2] |

Why not dispatch 3rd MUL.S? — structural hazard - no reservation station - must STALL

Tomasulo Loop Example: Cycle 13

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|-----|-----|-----|------|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | no | | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | c10 | c11 | Ld2 | no | | | f0 Ld3 |
| MUL.S f0, f2 → f4 | c2 | c11 | | | Ld3 | yes | le4 | | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 Mult2 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | Yes | 72 | Mult2 | f6 |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | no | | | f8 |
| L.S #0(r1) → f0 | c6 | c8 | c11 | c12 | CDB | | | | |
| MUL.S f0, f2 → f4 | c7 | c12 | | | T | V | | | f10 |
| S.S f4 → #0(r1) | c8 | | | | Add1 | le4 | | | ... |

| Reservation Stations | | | | | | |
|----------------------|------|-------|----|----|-------|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | | | M[80] | [F2] |
| Mult2 | Yes | MUL.S | | | M[72] | [F2] |

Subi not shown

Tomasulo Loop Example: Cycle 14

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|----------------------|------|-------|-----|-----|-------|----------------|------|-------|-----------|
| Insn | D | S | X | W | | Ld1 | no | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | c10 | c11 | Ld2 | no | | | f0 Ld3 |
| MUL.S f0, f2 → f4 | c2 | c11 | | | Ld3 | Yes | 64 | | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 Mult2 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | Yes | 72 | Mult2 | |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | no | | | |
| L.S #0(r1) → f0 | c6 | c8 | c11 | c12 | CDB | | | | f6 |
| MUL.S f0, f2 → f4 | c7 | c12 | | | T V | | | | f8 |
| S.S f4 → #0(r1) | c8 | | | | | | | | f10 |
| Reservation Stations | | | | | | | | | |
| FU | busy | op | Qj | Qk | Vj | V _k | | | |
| Add1 | no | | | | | | | | |
| Add2 | no | | | | | | | | |
| Add3 | no | | | | | | | | |
| Mult1 | Yes | MUL.S | | | M[80] | [F2] | | | |
| Mult2 | Yes | MUL.S | | | M[72] | [F2] | | | |

no change · waiting for
mults

Tomasulo Loop Example: Cycle 15

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|-----|-----|-----|-----|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | no | | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | c10 | c11 | Ld2 | no | | | f0 Ld3 |
| MUL.S f0, f2 → f4 | c2 | c11 | C15 | | Ld3 | Yes | 64 | | f2 |
| S.S f4 → #0(r1) | c3 | | | | St1 | Yes | 80 | Mult1 | f4 Mult2 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | Yes | 72 | Mult2 | |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | no | | | |
| L.S #0(r1) → f0 | c6 | c8 | c11 | c12 | | | | | |
| MUL.S f0, f2 → f4 | c7 | c12 | | | | | | | |
| S.S f4 → #0(r1) | c8 | | | | | | | | |

| Reservation Stations | | | | | | |
|----------------------|------|-------|----|----|-------|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | | | M[80] | [F2] |
| Mult2 | Yes | MUL.S | | | M[72] | [F2] |

mult1 15 completing - what's 15
wait for?

Tomasulo Loop Example: Cycle 16

| Insn Status | | | | | | Busy | Addr | T | <i>Value</i> | Map Table |
|----------------------|------|-------|-----|-----|-----|----------------|---------|------|--------------|-----------|
| Insn | | D | S | X | W | Ld1 | no | | | Reg T |
| L.S #0(r1) → f0 | | c1 | c2 | c10 | c11 | Ld2 | no | | | f0 Ld3 |
| MUL.S f0, f2 → f4 | | c2 | c11 | c15 | c16 | Ld3 | Yes | 64 | | f2 |
| S.S f4 → #0(r1) | | c3 | c16 | | | St1 | Yes | 80 | [80]xF2 | f4 Mult2 |
| SUBI r1, 8 → r1 | | c4 | c5 | c6 | c7 | St2 | Yes | 72 | Mult2 | |
| BNEZ r1, Loop | | c5 | c6 | c7 | c8 | St3 | no | | | |
| L.S #0(r1) → f0 | | c6 | c8 | c11 | c12 | CDB | | | | |
| MUL.S f0, f2 → f4 | | c7 | c12 | c16 | | T | V | | | f6 |
| S.S f4 → #0(r1) | | c8 | | | | MULT | [80]xJ2 | | | f8 |
| Reservation Stations | | | | | | | | | | |
| FU | busy | op | Qj | Qk | Vj | V _k | | | | |
| Add1 | no | | | | | | | | | |
| Add2 | no | | | | | | | | | |
| Add3 | no | | | | | | | | | |
| Mult1 | no | | | | | | | | | |
| Mult2 | Yes | MUL.S | | | | M[72] | | [F2] | | |

Tomasulo Loop Example: Cycle 17

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|-----|-----|-----|-------|---------|------|---------|-----------|
| Insn | D | S | X | W | Ld1 | no | | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | c10 | c11 | Ld1 | no | | | |
| MUL.S f0, f2 → f4 | c2 | c11 | c15 | c16 | Ld2 | no | | | f0 Ld3 |
| S.S f4 → #0(r1) | c3 | c16 | c17 | | Ld3 | Yes | 64 | | |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St1 | Yes | 80 | [80]*F2 | |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St2 | Yes | 72 | [72]*F2 | |
| L.S #0(r1) → f0 | c6 | c8 | c11 | c12 | St3 | no | | | |
| MUL.S f0, f2 → f4 | c7 | c12 | c16 | c17 | CDB | | | | |
| S.S f4 → #0(r1) | c8 | c17 | | | T | V | | | |
| | | | | | MULT2 | [72]*F2 | | | |

| Reservation Stations | | | | | | |
|----------------------|------|-------|-----|----|----|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | yes | MUL.S | Ld3 | | | [F2] |
| Mult2 | no | | | | | |

MUL.S #3 → Structural hazard resolved

Tomasulo Loop Example: Cycle 18

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|-----|-----|-----|-----|------|------|-----------|-----------|
| Insn | D | S | X | W | Ld1 | no | | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | c10 | c11 | Ld2 | no | | | f0 Ld3 |
| MUL.S f0, f2 → f4 | c2 | c11 | c15 | c16 | Ld3 | Yes | 64 | | f2 |
| S.S f4 → #0(r1) | c3 | c16 | c17 | c18 | St1 | ne | | | f4 Mult1 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | Yes | 72 | [72] * F2 | |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | yes | 64 | MULT1 | f6 |
| L.S #0(r1) → f0 | c6 | c8 | c11 | c12 | CDB | | | | f8 |
| MUL.S f0, f2 → f4 | c7 | c12 | c16 | c17 | | T | V | | f10 |
| S.S f4 → #0(r1) | c8 | c17 | c18 | | | | | | ... |
| | | | | | | | | | r1 |

| Reservation Stations | | | | | | |
|----------------------|------|-------|-----|----|----|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Ld3 | | | [F2] |
| Mult2 | no | | | | | |

S.S doesn't use CDB

Dispatch 3rd S.S. (stalled previously due to structural hazard on MUL.S)

Tomasulo Loop Example: Cycle 19

| Insn Status | | | | | | Busy | Addr | T | Map Table |
|-------------------|----|-----|-----|-----|-----|------|------|-------|-----------|
| Insn | D | S | X | W | Ld1 | no | | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | c10 | c11 | Ld2 | no | | | f0 Ld3 |
| MUL.S f0, f2 → f4 | c2 | c11 | c15 | c16 | Ld3 | Yes | 64 | | f2 |
| S.S f4 → #0(r1) | c3 | c16 | c17 | c18 | St1 | no | | | f4 Mult1 |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | | | | f6 |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | Yes | 64 | Mult1 | f8 |
| L.S #0(r1) → f0 | c6 | c8 | c11 | c12 | CDB | | | | |
| MUL.S f0, f2 → f4 | c7 | c12 | c16 | c17 | T | V | | | f10 |
| S.S f4 → #0(r1) | c8 | c17 | c18 | C19 | | | | | ... |
| | | | | | | | | | r1 |

| Reservation Stations | | | | | | |
|----------------------|------|-------|-----|----|----|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Ld3 | | | [F2] |
| Mult2 | no | | | | | |

Tomasulo can overlap loop iterations!

Why can Tomasulo overlap loop iterations?

- Register renaming

multiple iterations use different physical destinations for registers → dynamic loop unrolling

- Reservation stations

Buffer old values of registers - avoids WAR
stalling

Permit 000 issue - independent insns

- Other perspective:

Tomasulo builds data flow dependency
graph on the fly

Tomasulo's scheme offers 2 major advantages

- Distribution of hazard detection logic

Distributed reservation stations & CDB
if multiple insns waiting on a single
result & each has other operand
- insns can be released simultaneously
by broadcast on CDB

- Elimination of stalls for WAW and WAR hazards

- But...

1st store has page fault?

| Insn Status | | | | | | Busy | Addr | T | Reg Status |
|-------------------|----|-----|-----|-----|-----|------|------|---------|------------|
| Insn | D | S | X | W | Ld1 | no | | | Reg T |
| L.S #0(r1) → f0 | c1 | c2 | c10 | c11 | Ld2 | no | | | f0 Ld3 |
| MUL.S f0, f2 → f4 | c2 | c11 | c15 | c16 | Ld3 | Yes | 64 | | |
| S.S f4 → #100(r1) | c3 | c16 | c17 | c18 | St1 | no | | | |
| SUBI r1, 8 → r1 | c4 | c5 | c6 | c7 | St2 | Yes | 72 | [72]*F2 | |
| BNEZ r1, Loop | c5 | c6 | c7 | c8 | St3 | Yes | 64 | Mult1 | |
| L.S #0(r1) → f0 | c6 | c8 | c11 | c12 | CDB | | | | |
| MUL.S f0, f2 → f4 | c7 | c12 | c16 | c17 | T | V | | | |
| S.S f4 → #0(r1) | c8 | c17 | c18 | c19 | | | | | |

| Reservation Stations | | | | | | |
|----------------------|------|-------|-----|----|----|------|
| FU | busy | op | Qj | Qk | Vj | Vk |
| Add1 | no | | | | | |
| Add2 | no | | | | | |
| Add3 | no | | | | | |
| Mult1 | Yes | MUL.S | Ld3 | | | [F2] |
| Mult2 | no | | | | | |

4 younger insns have already updated values!

Dynamic Scheduling Summary

- Dynamic scheduling: out-of-order execution
 - Higher pipeline/FU utilization, improved performance
 - Easier and more effective in hardware than software
 - + More storage locations than architectural registers
 - + Dynamic handling of cache misses
- Instruction buffer: multiple F/D latches

- Implements large scheduling scope + “passing” functionality
- Split decode into in-order dispatch and out-of-order issue
 - Stall vs. wait
- Dynamic scheduling algorithms
 - Tomasulo: copy-based register renaming, full out-of-order
 - Next: Precise State and Speculation