CPE 631 Lecture 10: Instruction Level Parallelism and Its Dynamic Exploitation

Aleksandar Milenković, milenka@ece.uah.edu
Electrical and Computer Engineering
University of Alabama in Huntsville

Techniques to exploit parallelism

<table>
<thead>
<tr>
<th>Technique (Section in the textbook)</th>
<th>Reduces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwarding and bypassing (Section A.2)</td>
<td>Data hazard (DH) stalls</td>
</tr>
<tr>
<td>Delayed branches (A.3)</td>
<td>Control hazard stalls</td>
</tr>
<tr>
<td>Basic dynamic scheduling (A.3)</td>
<td>DH stalls (RAW)</td>
</tr>
<tr>
<td>Dynamic scheduling with register renaming (3.2)</td>
<td>WAR and WAW stalls</td>
</tr>
<tr>
<td>Dynamic branch prediction (3.4)</td>
<td>CH stalls</td>
</tr>
<tr>
<td>Issuing multiple instruction per cycle (3.6)</td>
<td>Ideal CPI</td>
</tr>
<tr>
<td>Speculation (3.7)</td>
<td>Data and control stalls</td>
</tr>
<tr>
<td>Dynamic memory disambiguation (3.2, 3.7)</td>
<td>RAW stalls w. memory</td>
</tr>
<tr>
<td>Loop Unrolling (4.1)</td>
<td>CH stalls</td>
</tr>
<tr>
<td>Basic compiler pipeline scheduling (A.2, 4.1)</td>
<td>DH stalls</td>
</tr>
<tr>
<td>Compiler dependence analysis (4.4)</td>
<td>Ideal CPI, DH stalls</td>
</tr>
<tr>
<td>Software pipelining and trace scheduling (4.3)</td>
<td>Ideal CPI and DH stalls</td>
</tr>
<tr>
<td>Compiler speculation (4.4)</td>
<td>Ideal CPI, and D/CH stalls</td>
</tr>
</tbody>
</table>

Dynamically Scheduled Pipelines

Outline

- Tomasulo’s algorithm
Scoreboard Limitations

- Amount of parallelism among instructions
  - can we find independent instructions to execute
- Number of scoreboard entries
  - how far ahead the pipeline can look for independent instructions (we assume a window does not extend beyond a branch)
- Number and types of functional units
  - avoid structural hazards
- Presence of antidependences and output dependences
  - WAR and WAW stalls become more important

Tomasulo’s Algorithm

- Used in IBM 360/91 FPU (before caches)
- Goal: high FP performance without special compilers
- Conditions:
  - Small number of floating point registers (4 in 360) prevented interesting compiler scheduling of operations
  - Long memory accesses and long FP delays
  - This led Tomasulo to try to figure out how to get more effective registers — renaming in hardware!
- Why Study 1966 Computer?
  - The descendants of this have flourished!
  - Alpha 21264, HP 8000, MIPS 10000, Pentium III, PowerPC 604, …

Tomasulo’s Algorithm (cont’d)

- Control & buffers distributed with Function Units (FU)
  - FU buffers called “reservation stations” => buffer the operands of instructions waiting to issue;
- Registers in instructions replaced by values or pointers to reservation stations (RS) => register renaming
  - avoids WAR, WAW hazards
  - More reservation stations than registers, so can do optimizations compilers can’t
- Results to FU from RS, not through registers, over Common Data Bus that broadcasts results to all FUs
- Load and Stores treated as FUs with RSs as well
- Integer instructions can go past branches, allowing FP ops beyond basic block in FP queue

Tomasulo-based FPU for MIPS

- Control & buffers distributed with Function Units (FU)
- Registers in instructions replaced by values or pointers to reservation stations (RS) => register renaming
- Results to FU from RS, not through registers, over Common Data Bus that broadcasts results to all FUs
- Load and Stores treated as FUs with RSs as well
- Integer instructions can go past branches, allowing FP ops beyond basic block in FP queue
Reservation Station Components

- **Op**: Operation to perform in the unit (e.g., + or −)
- **Vj, Vk**: Value of Source operands
  - Store buffers have V field, result to be stored
- **Qj, Qk**: Reservation stations producing source registers (value to be written)
  - Note: Qj/Qk=0 => source operand is already available in Vj/Vk
- **Busy**: Indicates reservation station or FU is busy

Register result status—Indicates which functional unit will write each register, if one exists. Blank when no pending instructions that will write that register.

Three Stages of Tomasulo Algorithm

1. **Issue**—get instruction from FP Op Queue
   - If reservation station free (no structural hazard), control issues instr & sends operands (renames registers)
2. **Execute**—operate on operands (EX)
   - When both operands ready then execute; if not ready, watch Common Data Bus for result
3. **Write result**—finish execution (WB)
   - Write it on Common Data Bus to all awaiting units; mark reservation station available

Normal data bus: data + destination ("go to" bus)
- 64 bits of data + 4 bits of Functional Unit source address
- Write if matches expected Functional Unit (produces result)
- Does the broadcast

Example speed: 2 clocks for Fl., pt., +, −; 10 for *; 40 ciks for /

Tomasulo Example

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Issue</th>
<th>Exec</th>
<th>Write</th>
<th>Busy</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F0 F6 3+ R2</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>34+R2</td>
<td></td>
</tr>
<tr>
<td>LD F2 F4 4+ R3</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTD F0 F2 F4</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBD F8 F6 F2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIVD F10 F0 F6</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

- Clock: F0 F2 F4 F6 F8 F10 ... F30
- Clock cycle counter

Tomasulo Example Cycle 1

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Issue</th>
<th>Exec</th>
<th>Write</th>
<th>Busy</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F0 F6 3+ R2</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>34+R2</td>
<td></td>
</tr>
<tr>
<td>LD F2 F4 4+ R3</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTD F0 F2 F4</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUBD F8 F6 F2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIVD F10 F0 F6</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

- Clock: F0 F2 F4 F6 F8 F10 ... F30
- Clock cycle counter
Tomasulo Example Cycle 2

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F6 34+ R2</td>
<td>1</td>
<td>Yes</td>
<td>34+R2</td>
</tr>
<tr>
<td>MULT F0 F2 F4</td>
<td>2</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>ADD F8 F6 F2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reservation Station: S1 S2 RS RS

Time Name Busy Op Vj Vk Qj Qk
Add1 No
Add2 No
Add3 No
Mult1 No
Mult2 No

Register result status:

Clock

F0 F2 F4 F6 F8 F10 F12 ...

Note: Can have multiple loads outstanding

14/02/2005 UAH-CPE631

Tomasulo Example Cycle 3

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F6 34+ R2</td>
<td>1</td>
<td>Yes</td>
<td>34+R2</td>
</tr>
<tr>
<td>MULT F0 F2 F4</td>
<td>2</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD F8 F6 F2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reservation Station: S1 S2 RS RS

Time Name Busy Op Vj Vk Qj Qk
Add1 Yes SUBD M(A1) Load2
Add2 No
Add3 No
Mult1 Yes MULTD R(F4) Load2
Mult2 No

Register result status:

Clock

F0 F2 F4 F6 F8 F10 F12 ...

6 FU

Mult1 M(A2) M(A1) Add1 Mult2

Note: registers names are removed (“renamed”) in Reservation Stations; MULT issued

Load1 completing: what is waiting for Load1?

14/02/2005 UAH-CPE631

Tomasulo Example Cycle 4

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F6 34+ R2</td>
<td>1</td>
<td>Yes</td>
<td>34+R2</td>
</tr>
<tr>
<td>MULT F0 F2 F4</td>
<td>2</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>ADD F8 F6 F2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reservation Station: S1 S2 RS RS

Time Name Busy Op Vj Vk Qj Qk
Add1 Yes MULTD M(A1) Load2
Add2 No
Add3 No
Mult1 Yes MULT M(A2) R(F4) Load2
Mult2 No

Register result status:

Clock

F0 F2 F4 F6 F8 F10 F12 ...

Load2 completing; what is waiting for Load2?

14/02/2005 UAH-CPE631

Tomasulo Example Cycle 5

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F6 34+ R2</td>
<td>1</td>
<td>Yes</td>
<td>34+R2</td>
</tr>
<tr>
<td>MULT F0 F2 F4</td>
<td>2</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>ADD F8 F6 F2</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Reservation Station: S1 S2 RS RS

Time Name Busy Op Vj Vk Qj Qk
Add1 Yes MULT D M(A1) Mult1
Add2 No
Add3 No
Mult1 Yes MULTD M(A2) R(F4)
Mult2 Yes ADD Mult1
Mult3 No

Register result status:

Clock

F0 F2 F4 F6 F8 F10 F12 ...

Mult1 M(A2) M(A1) Add1 Mult2

Timer starts down for Add1, Mult1

14/02/2005 UAH-CPE631
Tomasulo Example Cycle 6

Reservation Stations: Exec Write

<table>
<thead>
<tr>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F6 34+ R2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>Load1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD F2 45+ R3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>Load2</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTD F0 F2 F4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F10 F0 F6</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

Clock

Issue ADDD here despite none dependency on F6?

14/02/2005 UAH-CPE631

Tomasulo Example Cycle 7

Reservation Stations: Exec Write

<table>
<thead>
<tr>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F6 34+ R2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>Load1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD F2 45+ R3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>Load2</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTD F0 F2 F4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F10 F0 F6</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F0 F6 F10</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

Clock

• Add1 (SUBD) completing: what is waiting for it?

14/02/2005 UAH-CPE631

Tomasulo Example Cycle 8

Reservation Stations: Exec Write

<table>
<thead>
<tr>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F6 34+ R2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>Load1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD F2 45+ R3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>Load2</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTD F0 F2 F4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIVD F10 F0 F6</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F3</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F3</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

Clock

14/02/2005 UAH-CPE631

Tomasulo Example Cycle 9

Reservation Stations: Exec Write

<table>
<thead>
<tr>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD F6 34+ R2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>Load1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LD F2 45+ R3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>Load2</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTD F0 F2 F4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIVD F10 F0 F6</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F0 F6 F10</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDD F6 F8 F3</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

Clock

14/02/2005 UAH-CPE631
Tomasulo Example Cycle 10

Reservation Stations: SI S2 RS RS

Clock

Time Name Busy Op Vj Vk Qj Qk

Add1 No

Add2 No

Add3 No

Add2 (ADDD) completing; what is waiting for it?

Write result of ADDD here?

All quick instructions complete in this cycle!

14/02/2005 UAH-CPE631

21

Tomasulo Example Cycle 11

Reservation Stations: SI S2 RS RS

Clock

Time Name Busy Op Vj Vk Qj Qk

Add1 No

Add2 No

Add3 No

Add2 (ADDD) completing; what is waiting for it?

Write result of ADDD here?

All quick instructions complete in this cycle!

14/02/2005 UAH-CPE631

22

Tomasulo Example Cycle 12

Reservation Stations: SI S2 RS RS

Clock

Time Name Busy Op Vj Vk Qj Qk

Add1 No

Add2 No

Add3 No

Add2 (ADDD) completing; what is waiting for it?

Write result of ADDD here?

All quick instructions complete in this cycle!

14/02/2005 UAH-CPE631

23

Tomasulo Example Cycle 13

Reservation Stations: SI S2 RS RS

Clock

Time Name Busy Op Vj Vk Qj Qk

Add1 No

Add2 No

Add3 No

Add2 (ADDD) completing; what is waiting for it?

Write result of ADDD here?

All quick instructions complete in this cycle!

14/02/2005 UAH-CPE631

24
Tomasulo Example Cycle 14

Reservation Stations:

<table>
<thead>
<tr>
<th>Time</th>
<th>Name</th>
<th>Busy</th>
<th>Op</th>
<th>Vj</th>
<th>Vk</th>
<th>Qj</th>
<th>Qk</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add1</td>
<td>No</td>
<td>F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add2</td>
<td>No</td>
<td>F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add3</td>
<td>No</td>
<td>F4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mult1</td>
<td>Yes</td>
<td>MULTD</td>
<td>M(A2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mult2</td>
<td>Yes</td>
<td>DIVD</td>
<td>M(A1)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

Clock 1: FU M*F4 M(A2) (M-M+)

Tomasulo Example Cycle 15

Reservation Stations:

<table>
<thead>
<tr>
<th>Time</th>
<th>Name</th>
<th>Busy</th>
<th>Op</th>
<th>Vj</th>
<th>Vk</th>
<th>Qj</th>
<th>Qk</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add1</td>
<td>No</td>
<td>F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add2</td>
<td>No</td>
<td>F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add3</td>
<td>No</td>
<td>F4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mult1</td>
<td>No</td>
<td>MULTD</td>
<td>M(A2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mult2</td>
<td>Yes</td>
<td>DIVD</td>
<td>M(A1)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

Clock 1: FU MULTD | M(A1) | M(A2)

• Mult1 (MULD) completing; what is waiting for it?

Tomasulo Example Cycle 16

Reservation Stations:

<table>
<thead>
<tr>
<th>Time</th>
<th>Name</th>
<th>Busy</th>
<th>Op</th>
<th>Vj</th>
<th>Vk</th>
<th>Qj</th>
<th>Qk</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add1</td>
<td>No</td>
<td>F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add2</td>
<td>No</td>
<td>F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add3</td>
<td>No</td>
<td>F4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mult1</td>
<td>Yes</td>
<td>DIVD</td>
<td>M*F4</td>
<td>M(A1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

Clock 1: FU M*F4 M(A2) (M-M+)

• Just waiting for Mult2 (DIVD) to complete

Tomasulo Example Cycle 55

Reservation Stations:

<table>
<thead>
<tr>
<th>Time</th>
<th>Name</th>
<th>Busy</th>
<th>Op</th>
<th>Vj</th>
<th>Vk</th>
<th>Qj</th>
<th>Qk</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add1</td>
<td>No</td>
<td>F0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add2</td>
<td>No</td>
<td>F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add3</td>
<td>No</td>
<td>F4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mult1</td>
<td>No</td>
<td>MULTD</td>
<td>M(A2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mult2</td>
<td>Yes</td>
<td>DIVD</td>
<td>M(A1)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status:

Clock 1: FU M*F4 M(A2) (M-M+)

• Mult1 (MULD) completing; what is waiting for it?
Tomasulo Example Cycle 56

**Instruction status:**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD</td>
<td>F6</td>
<td>34+</td>
<td>R2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>Load1</td>
</tr>
<tr>
<td>LD</td>
<td>F2</td>
<td>45+</td>
<td>R3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>Load2</td>
</tr>
<tr>
<td>MULTD</td>
<td>F0</td>
<td>F2</td>
<td>F4</td>
<td>3</td>
<td>15</td>
<td>16</td>
<td>Load3</td>
</tr>
<tr>
<td>MULTD</td>
<td>F8</td>
<td>F6</td>
<td>F2</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>ADDD</td>
<td>F0</td>
<td>F6</td>
<td>F2</td>
<td>5</td>
<td>56</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>ADDD</td>
<td>F8</td>
<td>F6</td>
<td>F2</td>
<td>6</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

**Reservation Stations:**

- S1
- S2
- RS
- RS

**Register result status:**

- F0
- F2
- F4
- F6
- F8
- F10
- F12
- F30

- M*F4 (M(A1))

- Mult2 (DIVD) is completing; what is waiting for it?

---

Tomasulo Example Cycle 57

**Instruction status:**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD</td>
<td>F6</td>
<td>34+</td>
<td>R2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>Load1</td>
</tr>
<tr>
<td>LD</td>
<td>F2</td>
<td>45+</td>
<td>R3</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>Load2</td>
</tr>
<tr>
<td>MULTD</td>
<td>F0</td>
<td>F2</td>
<td>F4</td>
<td>3</td>
<td>15</td>
<td>16</td>
<td>Load3</td>
</tr>
<tr>
<td>MULTD</td>
<td>F8</td>
<td>F6</td>
<td>F2</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>ADDD</td>
<td>F0</td>
<td>F6</td>
<td>F2</td>
<td>5</td>
<td>56</td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>ADDD</td>
<td>F8</td>
<td>F6</td>
<td>F2</td>
<td>6</td>
<td>10</td>
<td>11</td>
<td></td>
</tr>
</tbody>
</table>

**Reservation Stations:**

- S1
- S2
- RS
- RS

**Register result status:**

- F0
- F2
- F4
- F6
- F8
- F10
- F12
- F30

- M*F4 (M(A1))

- Mult2 (DIVD) is completing; what is waiting for it?

---

Tomasulo Drawbacks

- Complexity
  - delays of 360/91. MIPS 10000, Alpha 21264, IBM PPC 620 in CA:AQA 2/e, but not in silicon!
- Many associative stores (CDB) at high speed
- Performance limited by Common Data Bus
  - Each CDB must go to multiple functional units ➔ high capacitance, high wiring density
  - Number of functional units that can complete per cycle limited to one!
  - Multiple CDBs ➔ more FU logic for parallel assoc stores
- Non-precise interrupts!
  - We will address this later

---

Tomasulo Loop Example

**Loop:**

```
Loop: LD  F0  0(R1)
MULT  F4  F0  F2
SD    F4  0  R1
SUBI  R1  R1  #8
BNEZ  R1  Loop
```

- This time assume Multiply takes 4 clocks
- Assume 1st load takes 8 clocks
  - (L1 cache miss), 2nd load takes 1 clock (hit)
- To be clear, will show clocks for SUBI, BNEZ
  - Reality: integer instructions ahead of Fl, Pt. Instructions
- Show 2 iterations
### Loop Example

#### Instruction status:
<table>
<thead>
<tr>
<th>Iteration</th>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD F0 0 R1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULT D F4 F0 F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LD F0 0 R1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MULT D F4 F0 F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SD F4 0 R1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MULT D F4 F0 F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Reservation Stations:
- **S1**: S2 RS
- **F1**: F0 F2 F4 F6 F8 F10 F12 ... F30

#### Register result status
- **Value of Register used for address, iteration control**: R1

#### Added Store Buffers
- **Add1**: No
- **Add2**: No
- **Add3**: No
- **Mul1**: No
- **Mul2**: No

### Loop Example Cycle 1

#### Instruction status:
<table>
<thead>
<tr>
<th>Iteration</th>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD F0 0 R1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULT D F4 F0 F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SD F4 0 R1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MULT D F4 F0 F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Reservation Stations:
- **S1**: S2 RS
- **F1**: F0 F2 F4 F6 F8 F10 F12 ... F30

#### Register result status
- **Value of Register used for address, iteration control**: R1

### Loop Example Cycle 2

#### Instruction status:
<table>
<thead>
<tr>
<th>Iteration</th>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD F0 0 R1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULT D F4 F0 F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SD F4 0 R1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MULT D F4 F0 F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Reservation Stations:
- **S1**: S2 RS
- **F1**: F0 F2 F4 F6 F8 F10 F12 ... F30

#### Register result status
- **Value of Register used for address, iteration control**: R1

### Loop Example Cycle 3

#### Instruction status:
<table>
<thead>
<tr>
<th>Iteration</th>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD F0 0 R1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULT D F4 F0 F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SD F4 0 R1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MULT D F4 F0 F2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Reservation Stations:
- **S1**: S2 RS
- **F1**: F0 F2 F4 F6 F8 F10 F12 ... F30

#### Register result status
- **Value of Register used for address, iteration control**: R1

---

14/02/2005

© Aleksandar Milenkovich
**Loop Example Cycle 4**

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Busy</th>
<th>Addr</th>
<th>Fu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>Yes</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULT</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td>Yes</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SD</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td>Yes</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>

**Reservation Stations:**

- S1
- S2
- RS

**Time Name Busy Op Vj Vk Qj Qk Code:**

- Add1: No LD F0 0 R1
- Add2: No MULT D F0 F2
- Add3: No SD F4 0 R1
- Mult1: Yes Mult R(F2) Load1
- Mult2: No BNEZ R1 Loop

**Register result status**

- Check R1 F0 F2 F4 F6 F8 F10 F12 ... F30

**Loop Example Cycle 5**

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Busy</th>
<th>Addr</th>
<th>Fu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>Yes</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULT</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td>Yes</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SD</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td>Yes</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>

**Reservation Stations:**

- S1
- S2
- RS

**Time Name Busy Op Vj Vk Qj Qk Code:**

- Add1: No LD F0 0 R1
- Add2: No MULT D F0 F2
- Add3: No SD F4 0 R1
- Mult1: Yes Mult R(F2) Load1
- Mult2: No BNEZ R1 Loop

**Register result status**

- Check R1 F0 F2 F4 F6 F8 F10 F12 ... F30

**Loop Example Cycle 6**

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Busy</th>
<th>Addr</th>
<th>Fu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>Yes</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULT</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td>Yes</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SD</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td>Yes</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>

**Reservation Stations:**

- S1
- S2
- RS

**Time Name Busy Op Vj Vk Qj Qk Code:**

- Add1: No LD F0 0 R1
- Add2: No MULT D F0 F2
- Add3: No SD F4 0 R1
- Mult1: Yes Mult R(F2) Load1
- Mult2: No BNEZ R1 Loop

**Register result status**

- Check R1 F0 F2 F4 F6 F8 F10 F12 ... F30

**Loop Example Cycle 7**

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Instruction</th>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Busy</th>
<th>Addr</th>
<th>Fu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>Yes</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULT</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td>Yes</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SD</td>
<td>0</td>
<td>3</td>
<td></td>
<td></td>
<td>Yes</td>
<td>80</td>
<td></td>
</tr>
</tbody>
</table>

**Reservation Stations:**

- S1
- S2
- RS

**Time Name Busy Op Vj Vk Qj Qk Code:**

- Add1: No LD F0 0 R1
- Add2: No MULT D F0 F2
- Add3: No SD F4 0 R1
- Mult1: Yes Mult R(F2) Load1
- Mult2: Yes Mult R(F2) Load2

**Register result status**

- Check R1 F0 F2 F4 F6 F8 F10 F12 ... F30
### Loop Example Cycle 8

**Reservation Stations:**
- S1
- S2
- RS

**Register result status**
- Check: R1
- F0: Load1
- F2: Mult1
- F4: Load2
- F6: Load3
- F8: Store1
- F10: Store2
- F12: Store3

1. **Instruction status:** Exec Write
2. **ITER Instruction**
   - j: 1
   - k: 1
   - Issue CompResult: Busy Addr Fu
   - Load1: Yes 80
   - Load2: Yes 72
   - Load3: No

3. **Reservation Stations:**
   - Add1: No
   - Add2: No
   - Add3: No
   - Mult1: Yes Mult1 (R(F2) Load1)
   - Mult2: Yes Mult2 (R(F2) Load2)

4. **Register result status**
   - Clock: 1

5. **Clock:**
   - F0: Load1
   - F2: Mult1
   - F4: Load2
   - F6: Load3
   - F8: Store1
   - F10: Store2
   - F12: Store3

---

### Loop Example Cycle 9

**Reservation Stations:**
- S1
- S2
- RS

**Register result status**
- Check: R1
- F0: Load1
- F2: Mult2
- F4: Load2
- F6: Load3
- F8: Store1
- F10: Store2
- F12: Store3

1. **Instruction status:** Exec Write
2. **ITER Instruction**
   - j: 1
   - k: 1
   - Issue CompResult: Busy Addr Fu
   - Load1: Yes 80
   - Load2: Yes 72
   - Load3: No

3. **Reservation Stations:**
   - Add1: No
   - Add2: No
   - Add3: No
   - Mult1: Yes Mult2 (R(F2) Load1)
   - Mult2: Yes Mult2 (R(F2) Load2)

4. **Register result status**
   - Clock: 1

5. **Clock:**
   - F0: Load1
   - F2: Mult2
   - F4: Load2
   - F6: Load3
   - F8: Store1
   - F10: Store2
   - F12: Store3

---

### Loop Example Cycle 10

**Reservation Stations:**
- S1
- S2
- RS

**Register result status**
- Check: R1
- F0: Load1
- F2: Mult1
- F4: Load2
- F6: Load3
- F8: Store1
- F10: Store2
- F12: Store3

1. **Instruction status:** Exec Write
2. **ITER Instruction**
   - j: 1
   - k: 1
   - Issue CompResult: Busy Addr Fu
   - Load1: Yes 80
   - Load2: Yes 72
   - Load3: No

3. **Reservation Stations:**
   - Add1: No
   - Add2: No
   - Add3: No
   - Mult1: Yes Mult1 (R(F2) Load1)
   - Mult2: Yes Mult2 (R(F2) Load2)

4. **Register result status**
   - Clock: 1

5. **Clock:**
   - F0: Load1
   - F2: Mult1
   - F4: Load2
   - F6: Load3
   - F8: Store1
   - F10: Store2
   - F12: Store3

---

### Loop Example Cycle 11

**Reservation Stations:**
- S1
- S2
- RS

**Register result status**
- Check: R1
- F0: Load1
- F2: Mult2
- F4: Load2
- F6: Load3
- F8: Store1
- F10: Store2
- F12: Store3

1. **Instruction status:** Exec Write
2. **ITER Instruction**
   - j: 1
   - k: 1
   - Issue CompResult: Busy Addr Fu
   - Load1: Yes 80
   - Load2: Yes 72
   - Load3: No

3. **Reservation Stations:**
   - Add1: No
   - Add2: No
   - Add3: No
   - Mult1: Yes Mult1 (R(F2) Load1)
   - Mult2: Yes Mult2 (R(F2) Load2)

4. **Register result status**
   - Clock: 1

5. **Clock:**
   - F0: Load1
   - F2: Mult2
   - F4: Load2
   - F6: Load3
   - F8: Store1
   - F10: Store2
   - F12: Store3
### Loop Example Cycle 12

<table>
<thead>
<tr>
<th>Time</th>
<th>Name</th>
<th>Busy</th>
<th>Op</th>
<th>Vj</th>
<th>Vk</th>
<th>Qj</th>
<th>Qk</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add1</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add2</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Mul1</td>
<td>Yes</td>
<td>Mul</td>
<td>80</td>
<td>R2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Mul2</td>
<td>Yes</td>
<td>Mul</td>
<td>72</td>
<td>R2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status

<table>
<thead>
<tr>
<th>Clock</th>
<th>FU</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Load3</td>
</tr>
</tbody>
</table>

Loop Example Cycle 13

<table>
<thead>
<tr>
<th>Time</th>
<th>Name</th>
<th>Busy</th>
<th>Op</th>
<th>Vj</th>
<th>Vk</th>
<th>Qj</th>
<th>Qk</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add1</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Add2</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Mul1</td>
<td>Yes</td>
<td>Mul</td>
<td>80</td>
<td>R2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Mul2</td>
<td>Yes</td>
<td>Mul</td>
<td>72</td>
<td>R2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status

<table>
<thead>
<tr>
<th>Clock</th>
<th>FU</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Load3</td>
</tr>
</tbody>
</table>

Loop Example Cycle 14

<table>
<thead>
<tr>
<th>Time</th>
<th>Name</th>
<th>Busy</th>
<th>Op</th>
<th>Vj</th>
<th>Vk</th>
<th>Qj</th>
<th>Qk</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add1</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add2</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Add3</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Mul1</td>
<td>Yes</td>
<td>Mul</td>
<td>80</td>
<td>R2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Mul2</td>
<td>Yes</td>
<td>Mul</td>
<td>72</td>
<td>R2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status

<table>
<thead>
<tr>
<th>Clock</th>
<th>FU</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Load3</td>
</tr>
</tbody>
</table>

Loop Example Cycle 15

<table>
<thead>
<tr>
<th>Time</th>
<th>Name</th>
<th>Busy</th>
<th>Op</th>
<th>Vj</th>
<th>Vk</th>
<th>Qj</th>
<th>Qk</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add1</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Add2</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Add3</td>
<td>No</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Mul1</td>
<td>Yes</td>
<td>Mul</td>
<td>80</td>
<td>R2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Mul2</td>
<td>Yes</td>
<td>Mul</td>
<td>72</td>
<td>R2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Register result status

<table>
<thead>
<tr>
<th>Clock</th>
<th>FU</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Load3</td>
</tr>
</tbody>
</table>
### Loop Example Cycle 16

**Reservation Stations:**

- **S1**
- **S2**
- **RS**

**Register result status**

<table>
<thead>
<tr>
<th>Check</th>
<th>R1</th>
<th>F0</th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
<th>F12</th>
<th>...</th>
<th>F30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ITER Instruction**

<table>
<thead>
<tr>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Addr</th>
<th>Fu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD</td>
<td>F0</td>
<td>0</td>
<td>R1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULTD</td>
<td>F4</td>
<td>F0</td>
<td>F2</td>
<td>Yes</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LD</td>
<td>F0</td>
<td>0</td>
<td>R1</td>
<td>Yes</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MULTD</td>
<td>F4</td>
<td>F0</td>
<td>F2</td>
<td>Yes</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SD</td>
<td>F4</td>
<td>0</td>
<td>R1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Register result status**

- **Clock**
  - **R1**
    - **F0**
    - **F2**
    - **F4**
    - **F6**
    - **F8**
    - **F10**
    - **F12**
    - **F30**

### Loop Example Cycle 17

**Reservation Stations:**

- **S1**
- **S2**
- **RS**

**Register result status**

<table>
<thead>
<tr>
<th>Check</th>
<th>R1</th>
<th>F0</th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
<th>F12</th>
<th>...</th>
<th>F30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>17</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ITER Instruction**

<table>
<thead>
<tr>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Addr</th>
<th>Fu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD</td>
<td>F0</td>
<td>0</td>
<td>R1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULTD</td>
<td>F4</td>
<td>F0</td>
<td>F2</td>
<td>Yes</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LD</td>
<td>F0</td>
<td>0</td>
<td>R1</td>
<td>Yes</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MULTD</td>
<td>F4</td>
<td>F0</td>
<td>F2</td>
<td>Yes</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SD</td>
<td>F4</td>
<td>0</td>
<td>R1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Register result status**

- **Clock**
  - **R1**
    - **F0**
    - **F2**
    - **F4**
    - **F6**
    - **F8**
    - **F10**
    - **F12**
    - **F30**

### Loop Example Cycle 18

**Reservation Stations:**

- **S1**
- **S2**
- **RS**

**Register result status**

<table>
<thead>
<tr>
<th>Check</th>
<th>R1</th>
<th>F0</th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
<th>F12</th>
<th>...</th>
<th>F30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>18</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ITER Instruction**

<table>
<thead>
<tr>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Addr</th>
<th>Fu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD</td>
<td>F0</td>
<td>0</td>
<td>R1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULTD</td>
<td>F4</td>
<td>F0</td>
<td>F2</td>
<td>Yes</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LD</td>
<td>F0</td>
<td>0</td>
<td>R1</td>
<td>Yes</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MULTD</td>
<td>F4</td>
<td>F0</td>
<td>F2</td>
<td>Yes</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SD</td>
<td>F4</td>
<td>0</td>
<td>R1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Register result status**

- **Clock**
  - **R1**
    - **F0**
    - **F2**
    - **F4**
    - **F6**
    - **F8**
    - **F10**
    - **F12**
    - **F30**

### Loop Example Cycle 19

**Reservation Stations:**

- **S1**
- **S2**
- **RS**

**Register result status**

<table>
<thead>
<tr>
<th>Check</th>
<th>R1</th>
<th>F0</th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
<th>F12</th>
<th>...</th>
<th>F30</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>19</td>
<td>64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**ITER Instruction**

<table>
<thead>
<tr>
<th>j</th>
<th>k</th>
<th>Issue</th>
<th>Comp</th>
<th>Result</th>
<th>Busy</th>
<th>Addr</th>
<th>Fu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LD</td>
<td>F0</td>
<td>0</td>
<td>R1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MULTD</td>
<td>F4</td>
<td>F0</td>
<td>F2</td>
<td>Yes</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>LD</td>
<td>F0</td>
<td>0</td>
<td>R1</td>
<td>Yes</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>MULTD</td>
<td>F4</td>
<td>F0</td>
<td>F2</td>
<td>Yes</td>
<td>75</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>SD</td>
<td>F4</td>
<td>0</td>
<td>R1</td>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Register result status**

- **Clock**
  - **R1**
    - **F0**
    - **F2**
    - **F4**
    - **F6**
    - **F8**
    - **F10**
    - **F12**
    - **F30**
### Loop Example Cycle 20

<table>
<thead>
<tr>
<th>Time</th>
<th>Name</th>
<th>Busy</th>
<th>Op</th>
<th>Vj</th>
<th>Fk</th>
<th>Qj</th>
<th>Qk</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Add1</td>
<td>No</td>
<td>LD</td>
<td>F0</td>
<td>0</td>
<td>R1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Add2</td>
<td>No</td>
<td>MULTD</td>
<td>F4</td>
<td>F0</td>
<td>F2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Add3</td>
<td>No</td>
<td>SD</td>
<td>F4</td>
<td>0</td>
<td>R1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Reservation Stations: S1 S2 RS

- S1: Add1, Add2, Add3, Mult1, Mult2
- S2: Load1, Mult1
- RS: Load1, Mult1

### Register result status

- Add1: No
- Add2: No
- Add3: No
- Mult1: Yes
- Mult2: No

### Tomasulo's scheme offers 2 major advantages

1. **the distribution of the hazard detection logic**
   - Distributed reservation stations and the CDB
   - If multiple instructions waiting on single result, & each instruction has other operand, then instructions can be released simultaneously by broadcast on CDB
   - If a centralized register file were used, the units would have to read their results from the registers when register buses are available.

2. **the elimination of stalls for WAW and WAR hazards**

### Why can Tomasulo overlap iterations of loops?

- **Register renaming**
  - Multiple iterations use different physical destinations for registers (dynamic loop unrolling)

- **Reservation stations**
  - Permit instruction issue to advance past integer control flow operations
  - Also buffer old values of registers - totally avoiding the WAR stall that we saw in the scoreboard

- **Other perspective:** Tomasulo building data flow dependency graph on the fly

### What about Precise Interrupts?

- Tomasulo had:
  - In-order issue, out-of-order execution, and out-of-order completion
  - Need to "fix" the out-of-order completion aspect so that we can find precise breakpoint in instruction stream.
Relationship between precise interrupts and speculation

- Speculation is a form of guessing
- Important for branch prediction:
  - Need to “take our best shot” at predicting branch direction
- If we speculate and are wrong, need to back up and restart execution to point at which we predicted incorrectly:
  - This is exactly same as precise exceptions!
- Technique for both precise interrupts/exceptions and speculation: in-order completion or commit

HW support for precise interrupts

- Need HW buffer for results of uncommitted instructions: reorder buffer
  - 3 fields: instr, destination, value
  - Use reorder buffer number instead of reservation station
  - Supplies operands between execution complete & commit
  - (Reorder buffer can be operand => more registers like RS)
  - Instructions commit
  - Once instruction commits, result is put into register
  - As a result, easy to undo speculated instructions on mispredicted branches or exceptions

Four Steps of Speculative Tomasulo Algorithm

1. Issue—get instruction from FP Op Queue
   - If reservation station and reorder buffer slot free, issue instr & send operands & reorder buffer no. for destination (this stage sometimes called “dispatch”)
2. Execution—operate on operands (EX)
   - When both operands ready then execute; if not ready, watch CDB for result; when both in reservation station, execute; checks RAW (sometimes called “issue”)
3. Write result—finish execution (WB)
   - Write on Common Data Bus to all awaiting FUs & reorder buffer; mark reservation station available.
4. Commit—update register with reorder result
   - When instr. at head of reorder buffer & result present, update register with result (or store to memory) and remove instr from reorder buffer. Mispredicted branch flushes reorder buffer (sometimes called “graduation”)

What are the hardware complexities with reorder buffer (ROB)?

- How do you find the latest version of a register?
  - (As specified by Smith paper) need associative comparison network
  - Could use future file or just use the register result status buffer to track which specific reorder buffer has received the value
- Need as many ports on ROB as register file
### Summary

- Reservations stations: implicit register 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)
- Today, helps cache misses as well
  - Don’t stall for L1 Data cache miss (insufficient ILP for L2 miss?)
- Lasting Contributions
  - Dynamic scheduling
  - Register renaming
  - Load/store disambiguation
- 360/91 descendants are Pentium III; PowerPC 604; MIPS R10000; HP-PA 8000; Alpha 21264