High-level view

- Out-of-order pipeline
  - Two decoupled pipelines: fetch/dispatch and issue/execute
  - Pipelines decoupled by buffers
    - Many names: reservation stations, issue queues/buffers, scheduling queues
      - "instruction window"

Early dynamically scheduled machines

- CDC 6600
  - Centralized control: "CDC scoreboard"
  - Many replicated functional units
  - All values pass through the register file
  - Stall on WAR/WAW hazards

- IBM 360/91 (Tomasulo's algorithm)
  - Distributed control: "reservation stations"
    - Several, fully-pipelined functional units (equivalent to replicating functional units)
    - Values broadcast to waiting instructions and register file in parallel (via the Common Data Bus)
    - Introduced register renaming: handles WAR/WAW hazards without stalling

CDC 6600 (MIPS version)

- Four stages after fetch
  - Dispatch
    - Check for structural and WAR hazards
      - Structural: Stall in dispatch stage if FU busy.
      - WAR: Stall in dispatch stage if an outstanding instruction in the scoreboard writes the same destination register.
    - Enter instruction into scoreboard and determine data dependences.
    - Route instruction to a free FU, where it waits until data operands are available
  - Issue
    - Wait for operands to become ready.
    - scoreboard signals when operands are ready.
      - Instruction reads registers from the register file.
      - then issues to FU for execution.
  - Execute
    - Write result
      - Check for WAR hazard. Stall if an outstanding prior instruction in the scoreboard needs the same register being written, and the read has not yet taken place
Warning – H&P naming

- H&P uses different names than what we will use
  - We: Dispatch  They: Issue
  - We: Issue  They: Read operands
  - We: Execute  They: Execute
  - We: Write result  They: Write result

CDC 6600 (MIPS version)

- Registers
  - FP MULT (1)
  - FP MULT (2)
  - FP DIV
  - FP ADD
- Integer Unit
  - (integer MIPS pipeline)
- Scoreboard
  - Control/status

Scoreboard

- Three data structures
  1. Instruction status
     - Which stage the instruction is in
  2. Functional unit status
     - Busy - FU is busy executing an instruction
     - Op - what instruction is the FU busy with
     - F_i - destination register
     - F_j, F_k - source registers
     - Q_j, Q_k - functional units producing src regs
     - R_j, R_k - flags indicating src regs are ready
  3. Register result status
     - Which FU is going to write each register
Running example

Example used for CDC and Tomasulo

- L.D F6, 34 (R2)
- L.D F2, 45 (R3)
- MUL.D F0, F2, F4
- SUB.D F8, F6, F2
- DIV.D F10, F0, F6
- ADD.D F6, F8, F2

CDC 6600 example

What happens when the first instruction is dispatched?

Instruction status

<table>
<thead>
<tr>
<th>L.D F6, 34 (R2)</th>
<th>L.D F2, 45 (R3)</th>
<th>MUL.D F0, F2, F4</th>
<th>SUB.D F8, F6, F2</th>
<th>DIV.D F10, F0, F6</th>
<th>ADD.D F6, F8, F2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Functional-unit status

- DIV: no
- ADD: no
- MULT2: no
- MULT1: no
- Integer: yes

Register-result status

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

CDC 6600 example (1)
### CDC 6600 example (2)

On the next cycle, the instr. is issued. What else happens?

#### Instruction status

<table>
<thead>
<tr>
<th>L</th>
<th>S</th>
<th>P</th>
<th>R</th>
<th>F</th>
<th>Q</th>
<th>Fk</th>
<th>Fj</th>
<th>Fi</th>
<th>op</th>
<th>busy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
</tbody>
</table>

#### Functional-unit status

<table>
<thead>
<tr>
<th>FU</th>
<th>Integer</th>
<th>F0</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Register-result status

<table>
<thead>
<tr>
<th>L.D</th>
<th>F6 , 34(R2)</th>
<th>L.D</th>
<th>F2 , 45(R3)</th>
<th>MUL.D</th>
<th>F0 , F2, F4</th>
<th>SUB.D</th>
<th>F8 , F6, F2</th>
<th>DIV.D</th>
<th>F10, F0, F6</th>
<th>ADD.D</th>
<th>F6 , F8, F2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CDC 6600 example (3)

#### Instruction status

<table>
<thead>
<tr>
<th>L</th>
<th>S</th>
<th>P</th>
<th>R</th>
<th>F</th>
<th>Q</th>
<th>Fk</th>
<th>Fj</th>
<th>Fi</th>
<th>op</th>
<th>busy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
</tbody>
</table>

#### Functional-unit status

<table>
<thead>
<tr>
<th>FU</th>
<th>Integer</th>
<th>F0</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Register-result status

<table>
<thead>
<tr>
<th>L.D</th>
<th>F6 , 34(R2)</th>
<th>L.D</th>
<th>F2 , 45(R3)</th>
<th>MUL.D</th>
<th>F0 , F2, F4</th>
<th>SUB.D</th>
<th>F8 , F6, F2</th>
<th>DIV.D</th>
<th>F10, F0, F6</th>
<th>ADD.D</th>
<th>F6 , F8, F2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### CDC 6600 example (4)

#### Instruction status

<table>
<thead>
<tr>
<th>L</th>
<th>S</th>
<th>P</th>
<th>R</th>
<th>F</th>
<th>Q</th>
<th>Fk</th>
<th>Fj</th>
<th>Fi</th>
<th>op</th>
<th>busy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>F</td>
</tr>
</tbody>
</table>

#### Functional-unit status

<table>
<thead>
<tr>
<th>FU</th>
<th>Integer</th>
<th>F0</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Register-result status

<table>
<thead>
<tr>
<th>L.D</th>
<th>F6 , 34(R2)</th>
<th>L.D</th>
<th>F2 , 45(R3)</th>
<th>MUL.D</th>
<th>F0 , F2, F4</th>
<th>SUB.D</th>
<th>F8 , F6, F2</th>
<th>DIV.D</th>
<th>F10, F0, F6</th>
<th>ADD.D</th>
<th>F6 , F8, F2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---
### CDC 6600 example (5)

#### Instruction status

<table>
<thead>
<tr>
<th>Step</th>
<th>ADDR</th>
<th>OP</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

#### Functional-unit status

- ADD: Integer
- MULT1: Integer
- MUL.D: Integer
- SUB.D: Integer
- DIV.D: Integer

<table>
<thead>
<tr>
<th>FU</th>
<th>BUSY</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
<th>R10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULT1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUL.D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUB.D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIV.D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Register-result status

- L.D   F6 , 34(R2)  
- L.D   F2 , 45(R3)  
- MUL.D F0 , F2, F4  
- SUB.D F8 , F6, F2  
- DIV.D F10, F0, F6  
- ADD.D F6 , F8, F2  

### CDC 6600 example (6)

#### Instruction status

<table>
<thead>
<tr>
<th>Step</th>
<th>ADDR</th>
<th>OP</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

#### Functional-unit status

- ADD: Integer
- MULT1: Integer
- MUL.D: Integer
- SUB.D: Integer
- DIV.D: Integer

<table>
<thead>
<tr>
<th>FU</th>
<th>BUSY</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
<th>R10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULT1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUL.D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUB.D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIV.D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Register-result status

- L.D   F6 , 34(R2)  
- L.D   F2 , 45(R3)  
- MUL.D F0 , F2, F4  
- SUB.D F8 , F6, F2  
- DIV.D F10, F0, F6  
- ADD.D F6 , F8, F2  

### CDC 6600 example (7)

#### Instruction status

<table>
<thead>
<tr>
<th>Step</th>
<th>ADDR</th>
<th>OP</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

#### Functional-unit status

- ADD: Integer
- MULT1: Integer
- MUL.D: Integer
- SUB.D: Integer
- DIV.D: Integer

<table>
<thead>
<tr>
<th>FU</th>
<th>BUSY</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
<th>R10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULT1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUL.D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUB.D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIV.D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Register-result status

- L.D   F6 , 34(R2)  
- L.D   F2 , 45(R3)  
- MUL.D F0 , F2, F4  
- SUB.D F8 , F6, F2  
- DIV.D F10, F0, F6  
- ADD.D F6 , F8, F2  

---

**ECE 463/521, Profs. Gehringer, Rotenberg, & Conte, Dept. of ECE, NC State University**
### CDC 6600 example (8)

<table>
<thead>
<tr>
<th>Instruction status</th>
<th>RETRIEVED</th>
<th>PENDING</th>
<th>EXECUTED</th>
<th>WRITE RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>L S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Functional-unit status

<table>
<thead>
<tr>
<th></th>
<th>F0</th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrived</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Pending</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Issue</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Dispatch</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Register-result status

<table>
<thead>
<tr>
<th></th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
</tr>
</thead>
<tbody>
<tr>
<td>L.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MUL.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ADD.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DIV.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

### CDC 6600 example (9)

<table>
<thead>
<tr>
<th>Instruction status</th>
<th>RETRIEVED</th>
<th>PENDING</th>
<th>EXECUTED</th>
<th>WRITE RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>L S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Functional-unit status

<table>
<thead>
<tr>
<th></th>
<th>F0</th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrived</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Pending</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Issue</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Dispatch</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Register-result status

<table>
<thead>
<tr>
<th></th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
</tr>
</thead>
<tbody>
<tr>
<td>L.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MUL.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ADD.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DIV.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

---

### CDC 6600 example (10)

<table>
<thead>
<tr>
<th>Instruction status</th>
<th>RETRIEVED</th>
<th>PENDING</th>
<th>EXECUTED</th>
<th>WRITE RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>L S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L S</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REG</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Functional-unit status

<table>
<thead>
<tr>
<th></th>
<th>F0</th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrived</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Pending</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Issue</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Dispatch</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Register-result status

<table>
<thead>
<tr>
<th></th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
<th>F10</th>
</tr>
</thead>
<tbody>
<tr>
<td>L.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MUL.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ADD.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>DIV.D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
CDC 6600 example (11)

MULTD about to write result...

<table>
<thead>
<tr>
<th>Instruction status</th>
<th>ISSUE</th>
<th>EXECE</th>
<th>WRITE RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>L.D F2, 34(R2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L.D F2, 43(R3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MUL D F2, F4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD F8, F6, F2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WAR (F6) F2, F6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD F8, F6, F2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FUNCTIONAL UNIT STATUS

<table>
<thead>
<tr>
<th>Slot</th>
<th>Busy</th>
<th>F0</th>
<th>F2</th>
<th>F4</th>
<th>F6</th>
<th>F8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REGISTER RESULT STATUS

<table>
<thead>
<tr>
<th>Slot</th>
<th>Rk</th>
<th>Rj</th>
<th>Qk</th>
<th>Qj</th>
<th>Fk</th>
<th>Fj</th>
<th>Fi</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

COMMANDS:

- WRITE RESULT
- EXECUTE
- ISSUE
- DISPATCH
- WAR (F6)
- RAW (F0)

CDC 6600 timing diagram

Instructions:
- S.D (3 cycles)
- L.D (1 cycle)
- MUL.D (10 cycles)
- SUB.D (2 cycles)
- DIV.D (40 cycles)
- ADD.D (2 cycles)

Related hazards:
- RAW hazards
- Structural hazards
- Anti- and output dependences

Remaining bottlenecks

- CDC 6600 does a good job of dynamic scheduling around RAW hazards
- Remaining performance limitations
  - Amount of Instruction-level parallelism (ILP) in the program
  - Increase size of window to look farther ahead
  - Number of scoreboard entries (window size)
  - Number and type of functional units, register ports, etc.
  - Structural hazards
  - Anti- and output dependences

- Dynamic scheduling exposes more WAW + WAR hazards because early (DOO) writes are possible
- WAR made worse in CDC due to late reads (read operands when finally issuing)
- WAW handled like a structural hazard in dispatch
Tomasulo's Algorithm

- Born of necessity
  - Used in IBM 360/91 floating-point unit
  - Many long-latency operations
    - Need dynamic scheduling: mitigate long stalls
  - ISA specified only 4 floating-point registers
    - Need register renaming: with only 4 registers, WAW/WAR hazards pop up quickly
    - Especially in floating-point code: loops by definition cause repeated writes to same registers
    - Renaming: recognize and give unique names to different dynamic instances of the same register specifier

Key aspects of Tomasulo’s Alg.

1. Read register operands at dispatch stage
   - If operands are available, the data is buffered along with the instruction in “reservation stations”
   - CDC: only buffers the instruction, all operands are read from register file when all operands are ready (operands read at issue stage)
   - CDC – late reads / Tomasulo – early reads: early reads help WAR condition

2. Unavailable registers are renamed at dispatch stage
   - Waiting instructions replace register specifiers with a “tag” indicating the producer instruction
   - Register specifiers are used only once, at dispatch!
   - Renaming eliminates WAW/WAR hazards

3. Successive writes to a register
   - Only last one is actually used to update register: helps WAW condition
   - CDC: stall in dispatch until WAW hazard goes away

Other differences with CDC

- Distributed control
  - Reservation stations (versus scoreboard)
- Results broadcast to both register file and functional units
  - Result bus called the “Common Data Bus” (CDB)
  - Don’t have to wait for value to go through register file (i.e., use bypasses).
  - Functional units don’t contend for register file ports.
Tomasulo's Alg. (MIPS version)

Four stages after fetch
- Dispatch
  - Check for structural hazard
    - Stall in dispatch stage if no free reservation station
  - Read register file
    - Read data operands if available
    - Read "tag" if data operand unavailable; a tag is the reservation station number of the producer instruction
  - Route instruction plus data or tags to reservation station, where it waits until all operands are available
- Issue
  - Wait until operands become available on the CDB (match CDB tag against operand tags)
  - Grab operands from CDB and issue to FU (if free)
- Execute
- Write result
  - Broadcast result + tag to all reservation stations, store buffers, and register file via the CDB
  - Only write register file if the CDB tag matches the tag in the register file

Four stages (cont...)

Notice:
- Same four stages as CDC, but read register file at dispatch stage and replace register specifiers with tags
- No WAW / WAR checks (using tags eliminates these)
- Again, H&P uses different names than what we will use
- Worse, H&P specifies only three stages for Tomasulo
- But any dynamic scheduling inherently has dispatch, issue, execute, write result...
Register renaming

- Consider simple example with register reuse
  
  ```
  L.D    F0, 34(R2)
  ADD.D  F4, F0, F2
  L.D    F0, 45(R3)
  ADD.D  F8, F0, F6
  ```

- Dataflow graph with both true and false dependences
  
  - True dependence (RAW / F0)
  - Anti-dependence (WAR / F0)
  - Output dependence (WAW / F0)

- All instructions execute serially
  
  - Due to reuse of F0 by the loads
  - But those are 2 distinct instances of F0
  - Use different names for 2 instances of F0

Same program segment with F0 renamed

- Tomasulo Alg: use reservation station number (tag) of producer instruction (e.g. load buffer 1 = load1)
  
  ```
  L.D    load1, 34(R2)
  ADD.D  F4, load1, F2
  L.D    load2, 45(R3)
  ADD.D  F8, load2, F6
  ```

- Dataflow graph with only true dependences
  
  - Renaming removes output and anti-dependences
  - Parallelism is exposed

How (Tomasulo) renaming works (1)

- Use reservation station number (tag) of producer instruction (e.g. load buffer 1 = load1)
  
  ```
  L.D    F0 <-
  ADD.D  <- F0
  L.D    F0 <-
  ADD/D  <- F0
  ```
To memory

From memory

OPERAND

BUSES

OPERATION BUS

RESERVATION STATIONS

FP adders
FP multipliers

LOAD BUFFERS
STORE BUFFERS

FLT. PT. OPERATION QUEUE

From IF unit

4 FP REGISTERS

1 2 3 4

1 12 23

COMMON DATA BUS (CDB)

(5)

(6) Second load completes 1st

(7)