Chapter “2”

“Instruction Level Parallelism”
(Multiple Issue and VLIW Processors)
If only 1 instruction is issued every clock cycle, then CPI ≥ 1.

Multiple-issue processors try to achieve CPI < 1 by issuing several instructions simultaneously.

Multiple-issue processors come in 3 flavors

1. Statically-scheduled superscalar processors
2. Dynamically-scheduled superscalar processors
3. VLIW (very long instruction word) processors
The 2 types of superscalar processors issue varying numbers of instructions per clock

- use in-order execution if they are statically scheduled, or
- out-of-order execution if they are dynamically scheduled

VLIW processors, in contrast, issue a fixed number of instructions formatted either as one large instruction or as a fixed instruction packet with the parallelism among instruction explicitly indicated by the instruction.
MULTIPLE ISSUE PROCESSORS (VLIW)

DEFINITION

- Each “instruction” has explicit coding for multiple operations
- Tradeoff between instruction space and simple decoding
  - The long instruction word has room for many operations
  - By definition, all the operations the compiler puts in the long instruction word are independent => execute in parallel
MULTIPLE ISSUE PROCESSORS (VLIW)

DEFINITION

- e.g. VLIW: 2 integer operations, 2 FP ops, 2 Memory refs, 1 branch
  » 16 to 24 bits per field => 7*16 or 112 bits to 7*24 or 168 bits wide
- Need compiling technique that schedules across several branches.
Unrolled Loop that Minimizes Stalls for Superscalar Processor

1 Loop: L.D F0,0 (R1)
2    L.D F6,-8 (R1)
3    L.D F10,-16 (R1)
4    L.D F14,-24 (R1)
5    ADD.D F4,F0,F2
6    ADD.D F8,F6,F2
7    ADD.D F12,F10,F2
8    ADD.D F16,F14,F2
9    S.D 0 (R1),F4
10   S.D -8 (R1),F8
11   S.D -16 (R1),F12
12   DSUBUI R1,R1,#32
13   BNEZ R1,LOOP
14   S.D 8 (R1),F16 ; 8-32 = -24

for (i=1000; i>0; i=i-1)
    x[i] = x[i] + s;

14 clock cycles, or 14/4=3.5 cycles per iteration; 9 registers needed
## MULTIPLE ISSUE PROCESSORS (VLIW)

### LOOP UNROLLING IN VLIW

<table>
<thead>
<tr>
<th>Memory reference 1</th>
<th>Memory reference 2</th>
<th>FP operation 1</th>
<th>FP op. 2</th>
<th>Int. op/branch</th>
<th>Clock</th>
</tr>
</thead>
<tbody>
<tr>
<td>L.D F0,0(R1)</td>
<td>L.D F6,-8(R1)</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>L.D F10,-16(R1)</td>
<td>L.D F14,-24(R1)</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>L.D F18,-32(R1)</td>
<td>L.D F22,-40(R1)</td>
<td>ADD.D F4,F0,F2</td>
<td>ADD.D F8,F6,F2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>L.D F26,-48(R1)</td>
<td></td>
<td>ADD.D F12,F10,F2</td>
<td>ADD.D F16,F14,F2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S.D 0(R1),F4</td>
<td>ADD.D F28,F26,F2</td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>S.D -16(R1),F12</td>
<td>S.D -24(R1),F16</td>
<td></td>
<td></td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>S.D -32(R1),F20</td>
<td>S.D -40(R1),F24</td>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>S.D -0(R1),F28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>BNEZ R1,LOOP</td>
<td>9</td>
</tr>
</tbody>
</table>

Unrolled 7 times to avoid delays

7 results in 9 clocks, or 9/7 = 1.3 cycles per iteration

Average: 23/9 = 2.5 ops per cycle, 23/45 = 51% efficiency

Note: Need more registers in VLIW (15 vs. 9 in SS)