Condition Codes

- **Processor State Register (psr)**
- **Integer Condition Codes**
  - \( N \): Set if the last ALU result was negative
  - \( Z \): Set if the last ALU result was zero
  - \( V \): Set if the last ALU result overflowed
  - \( C \): Set if the last ALU instruction that modified the integer condition codes (icc) caused a carry out of, or a borrow into, bit 31

**cc** versions of the integer arithmetic instructions set all the codes
- **cc** versions of the logical instructions set only \( N \) and \( Z \)

Tests on the condition codes implement conditionals and loops

Carry and Overflow

- If the carry bit (\( C \)) is set, the last addition resulted in a carry or the last subtraction resulted in a borrow.

Carry is needed to implement arithmetic using numbers represented in several words, e.g. multiple-precision addition.

Overflow (\( V \)) indicates that the result of signed addition or subtraction doesn't fit.

Carry and Overflow

Branches

Branch instructions transfer control based on ACC.

**Branches**

Compare and Test

See page 26 in the Alpha Architecture Manual §4.4 in Paul

Tests on the condition codes are used to implement multiprecision arithmetic.

Integer condition codes — the cc field — hold 4 bits

<table>
<thead>
<tr>
<th>Address</th>
<th>Label</th>
<th>Cond</th>
<th>Target</th>
<th>Disp22</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>jcc</td>
<td>00 a</td>
<td>cond</td>
<td>010</td>
</tr>
</tbody>
</table>

Processors store register (psr)

Branches and Formal Instructions Set Condition Codes

Compare and Test

Compare two values

To test a single value

Branch instructions transfer control based on ACC.

Unconditional branches

Address of the branch instruction, where is the target address?

Branches are formal instructions set condition codes

Using \( d90 \) as a destination decides the result

Compare two values

Compare two values

To test a single value
Branches, cont'd

- comparisions
  - condition-code branches
    - raw condition-code branches
    - synthetic branch condition synonym
      - bnz ! Z
      - bz ! Z
      - bpos ! N
      - bneg ! N
      - bcc ! C
      - bgeu ! N
      - bcs ! C
      - blu
      - bvc ! V
      - bvs ! V

- comparisons
  - synthetic branches
    - signed
      - be ! Z
      - bz ! Z
      - bne ! Z
      - bnz ! Z
    - unsigned
      - bge ! N
      - bgeu ! N
      - bl ! N
      - bvc ! V
      - bvs ! V

Control Transfer

- normally, instructions are fetched and executed from sequential memory locations
- program counter, PC, is address of the current instruction, and the branches, control-transfer instructions change PC to something else
- PC-relative addressing is like register displacement addressing that uses PC as the base register

Control Transfer, cont'd

- normally, instructions are fetched and executed from sequential memory locations
- program counter, PC, is address of the current instruction, and the branches, control-transfer instructions change PC to something else
- PC-relative addressing is like register displacement addressing that uses PC as the base register

Branching Examples

- if-then-else
  - if (a > b)
  - c = a;
  - else
  - c = b;
  - becomes
  - #define a %l0
  - #define b %l1
  - #define c %l3
  - cmp a,b
  - ble L1;
  - nop
  - mov a,c
  - ba L2;
  - nop
  - L1: mov b,c
  - L2: ...

- if (x > y)
  - becomes
  - if (x > y)
  - if x <= y
  - else
  - becomes
  - if x <= y
  - if x > y

Control Transfer, cont'd

- branches
- jumping to an arbitrary location may require two branches, but branches are used to build conditionals and loops in small code blocks
- is multiplied by 4 because all instructions are word aligned
- condition execution does not depend on where it is loaded, i.e., all instructions use PC-relative addressing

- calls
- position-independent code is code whose correct execution does not depend on where it is loaded, i.e., all instructions use PC-relative addressing

- if-then-else
- program counter, PC, is address of the current instruction, and the branches, control-transfer instructions change PC to something else
- PC-relative addressing is like register displacement addressing that uses PC as the base register