

EE2361 - lecture 7

9/21/2016

Control Instructions

alter the sequence of
instructions that are executed

jumps (goto)
branches
function calls

unconditional
⇒ this always happens
conditional
⇒ what happens
depends on a "condition"

Branch instructions

SR = Status Register

- Branches conditioned on flag bits in the SR
DC, N, Z, OV, C

4-bits

1000

+8

~~unsign~~ ^{unsign}

- Branches conditioned on operations on unsigned numbers

1000

-8

signed

2C

- Branches conditioned on results of signed operations

Difference between branch & goto?

goto uses an absolute 23-bit address for the target

branches use a 16-bit encoding of the target relative to instruction

→
$$PC = (PC + 2) + 2 \cdot \text{Slt} \ll 16$$

↑ ↑

new value old value

of PC

16-bit number
2C

16-bit literal
value 2C
Slt = signed left
16-bits

Since branches are relative addresses
 $\sim \pm 32k$ instructions away
from branch instruction is
this a problem

Note: This is not a problem

(1) We only have a small
region of program memory
with flash.

(2) Branches are typically "local"
because of the way software is coded

Branch Based on Condition Code

brc C, label ; branch if carry

Branch Based on Unsigned Arith Result

brc LTU, label ; branch if less than
(unsigned)

(same as brc NC, label)

$$A < B ? \equiv \frac{A - B}{\text{sub}} < 0$$

Example

signed less than

$A < B$?

Two possibilities

$A - B$ is negative, no overflow
 $N = 1$ $OV = 0$

$A - B$ is positive, overflow

for bra LT, loop

condition is $(N \& \& \underline{!OV}) \underline{||} (\underline{!N} \& \underline{OV})$ ✓

$\&\&$ is and, $||$ is or, $!$ is negate

For these comparisons

$$A < B$$

we did a subtraction

2 sub instruction

3 operand instruction, need a 3rd register

2 operand instruction, overwrite one register with result

⇒ to make thing more efficient

there are compare instructions

⇒ there ~~is~~ set SR bits

Example