

A note on division of signed integers

Overview

- $(\text{left} + \text{right}) / 2$
- left and right are signed integers
- We know that multiplication by a power of two can be accomplished by shifting left
- What about signed division?

MIPS compiler output

- $(\text{left} + \text{right}) / 2$: left is \$2, right is \$3
 - - `addu $3,$3,$2`
 - `srl $2,$3,31`
 - `addu $2,$2,$3`
 - `sra $2,$2,1`
- } Er, what?
- What is the point of the middle two instructions?

The code

- `addu$3,$3,$2` # add left + right
 - `srl $2,$3,31` # get sign bit (1 or 0)
 - `addu$2,$2,$3` # add 1 or 0 to sum
 - `sra $2,$2,1` # divide by 2 w/ shift
- MIPS code to say:
 - if \$left + \$right is negative, add one.

Examples

- $(-5 + 3)/2$
- $-5 + 3 = -2$ 1110
- Try with approach:
 - Sign bit: 1; add 1 1111
 - Shift right arithmetic: 1111 (-1)
- Try without:
 - Sign bit 1: don't add 1110
 - Shift right arithmetic: 1111 (-1)
- Same result! Hmm....

Examples

- $(-5 + 3)/2$
- $-5 + 2 = -3$ 1101
- Try with approach:
 - Sign bit: 1; add 1 1110
 - Shift right arithmetic: 1111 (-1)
- Try without:
 - Sign bit 1: don't add 1101
 - Shift right arithmetic: 1110 (-2)
- Different result! Hmm....

Summary

- If dividing a negative odd number by a power of two by using shifting, must add one first.
- Thankfully easy to do
 - srl \$2,\$3,\$31
 - addu \$2,\$2,\$3