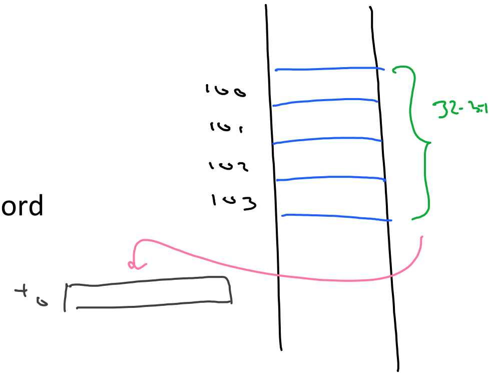
18 MIPS Extensions, FP

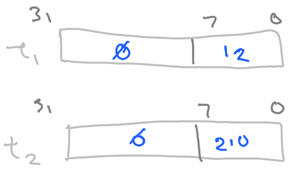
Thursday, October 15, 2015 10:01

LOAD/STORE

- Lw/sw
  - Load word, store word
  - A 32 bit quantity
- Lh/sh
  - Load half word, store half word
  - A 16 bit quantity
- Lb/sb
  - Load byte, store byte
  - A 8 bit quantity

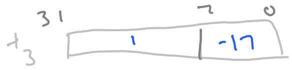


- What happens for lh/sh and lb/sb into a 32-bit register?



signed 8-bit
-128 127

Sign Extension



Big Indian, Little Indian -> figure it out for SPIPM

Put 4 bytes, see which 4 it loads

Apparently the same as the system, thus its little indian.

Lw $t0, 24($t7)
Addi $t0, $t1, -10

Zero extension

- Always fill with zeros, regardless of sign of the number
- OR, AND do zero extension

Lbu

- Load byte unsigned, is the lb instruction but only does zero extension

Nothing really unsigned about this.

Addu

- Add unsigned
- Confusing, not actually what you think it is

Addiu

- Immediate constant, sign extends the constants
- If you overflow in the addition, no error

Addi - exception on overflow
Addu, addiu - not soing unsigned arithmetic, but no exception on overflow

Floating Point Numbers

Not on homework

Fixed point

31
10
0
128.75

Floating point

$\underbrace{1.2345}_{\text{mantissa}} * 10^{\overset{\text{}}{12}}$

$1.2345 * 10^{-4}$ exponent

fraction

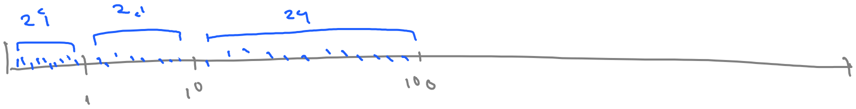
Formats

- 24 + 7 = 32 bits   there is another bit but nvm
- 53 + 10 but theres a bit missing somewhere
- Intel
  - 84 bits internally

Normalization

1.2345

Leading digit is not stored



$2^{4\text{-}bit}$ #'s   more dense, the lower the # is
the further out, you go, theres less coverage,

$a+1=a$ true for floating point #

the dist. btw 2 floating pt. #'s

Representation error

No exact floating point representation for X

Rounding error

Additional error introduced by arithmetic

Anything you do in FP, is prob wrong

$1.10 \times 2^{3} + 1.11 \times 2^{-1}$

$11000 \times 2^{-1} + 1.11 \times 2^{-1}$

Catorstrophic cancellation