

CS 631-02 RISC-V Assembly

Project 01 Observations

ssh beagle

tabs / spaces

atlang.h

```
#define SCAN_TOKEN_LEN 32 33
```

Redundancy

scanning →

parsing → parse-expression

TK → OP
error error

parse_operand()

TK_INTLIT, TK_BINLIT, TK_HEXLIT

Overflow

1) use `uint64_t`

2) `uint32_t`

`value_1 = value`

`value = (value * base) + sum` ←

`if (value < value_1) {`

`OF`

`}`

3)

`if (value_1 > (INTMAX >> 1)) {`

`OF`

`}`

`if (value_1 > (INTMAX / base)) {`

`OF`

`}`

if (value > (INTMAX - dig) / base) ?

$$\text{value} = (\text{value} * \text{base}) + \text{dig}$$

$$\frac{\text{value} - \text{dig}}{\text{base}}$$

$$(\text{val} * \text{base}) + \text{dig} > \frac{\text{MAX} - \text{dig}}{\text{base}}$$

$$\text{val} > (\text{MAX} - \text{dig}) / \text{base}$$

Check neg?

uint32_t value;

int32_t tmp;

tmp = (int32_t) value;

if (tmp < 0) ?

value = (uint32_t) (-tmp);

>

$$sh = 32 - width$$

$$value = ((int32_t) value) << sh >> sh;$$

ASR

$$r = (v1) >> v2$$

↑

$$r = \underline{(int32_t) v1} >> \underline{v2}$$

RISC-V Assembly

Vocabulary

instruction

operands
registers

32 register
64 bits wide



add a0, a1, a2

$a_0 = a_1 + a_2$

dest
operand

source
operands

x_0, x_1, \dots, x_{31}

ABI

a_0, a_1, \dots

t_0, t_1, \dots

s_0, s_1, \dots

SP

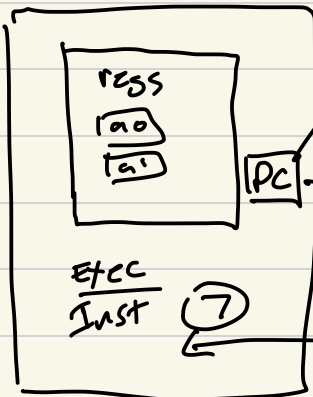
ra

immediate operands

addi a0, a1, 7

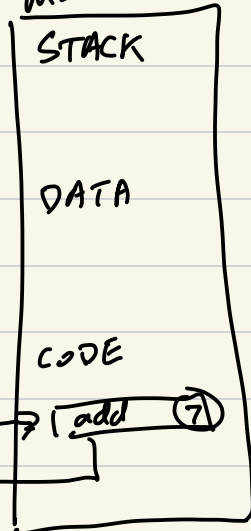
programming model

Processor



Program Counter

Memory



Instruction Type

Data Processing

add, sub, mul, div
and, or

Control

j, b jumps / branches

Memory

ld	/sd	64
lw	/sw	32
lb	/sb	8

Assembly

directives

labels

instructions

real inst

pseudo inst