

CS 631-01 RISC-V Assembly

Project 01 Observations

ssh

tabs/spaces

ntlang.h

33

#define SCAN_TOKEN_LEN ~~32~~

Redundancy

scanning

parsing - parse_expression()

map TK → OP
enum enum

]

parse_operand()

TK_INTLIT, TK_BINLIT, TK_HEXLIT

]

OVERFLOW

1) use uint64_t

2) uint32_t

value_1 = value

value = (value * base) + tmp

if (value < value_1) {
 overflow

}

3) uint32_t

if ((value > (UINT_MAX - tmp) / base) {
 OVERFLOW

}

$$\begin{aligned} \text{value}_1 &= (\text{value} * \text{base}) + \text{tmp} \\ \text{UINT_MAX} \\ \text{value}_1 - \text{tmp} &= \text{value} \\ \hline \text{base} \end{aligned}$$

Check neg?

uint32_t tmp;

int32_t tmp2

tmp2 = (int32_t) tmp;

if (tmp2 < 0) {

tmp = (uint32_t) (-tmp2);

}

ASR

(v2) >> v2

tmp2 = (int32_t) tmp

(uint32_t)(tmp2 >> v2)

RISC-V Assembly

Vocabulary
instruction

operands
registers

→ 32 regs
64 bits each

`add`
mnemonic

`a0, a1, a2`

`a0 = a1 + a2`

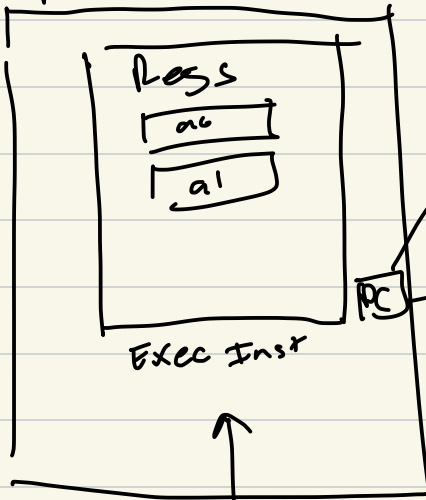
`x0, x1, ..., x31`

dest
reg

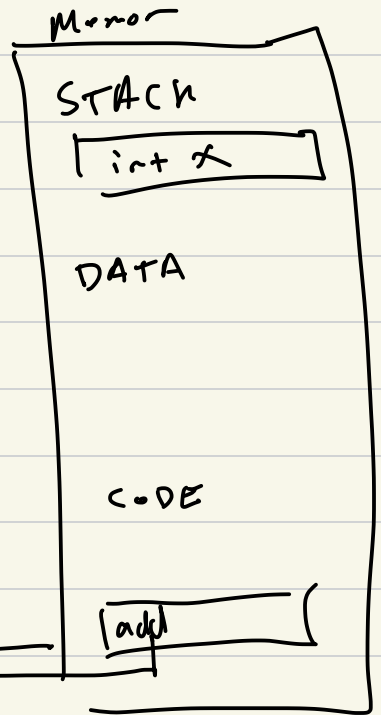
source
operands

`a0, a1`
`t0, t1`
`s0, s0`

Programming Model



Program Counter



instruction word

Instruction Types

Data Processing
add, sub

Control
j, b
jumps branches

Memory
ld / sd
lw / sw

Assembly

directives functions
labels control
instructions

real instruction

pseudo instruction