

CS315-02 RISC-V

Project 01

Lab 02

RISC-V Assembly

basic structure of Assembly code

more instructions

argument passing

array access

if / else

loops

gdb

instructions

operands

add

t0, t1, t2

directive
↓

dest
operand

source
operands

· global add1_s

add1_s:

addi a0, a0, 1

ret

arguments: a0, a1, a2, ...

ret val in a0

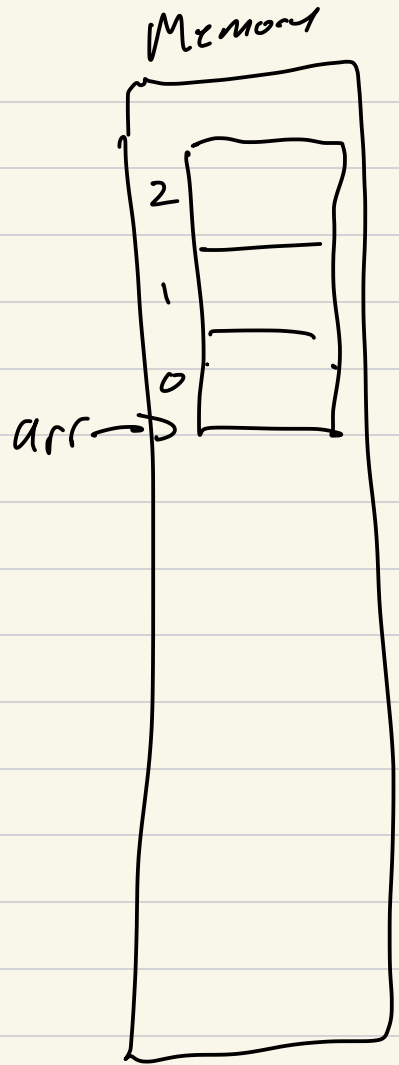
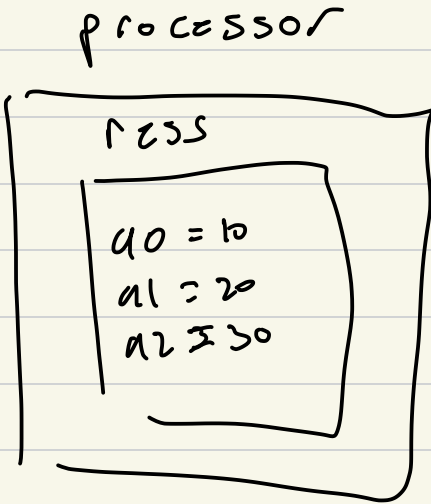
add t0, t1, t2

reg

addi t0, t1, 1

immediate

Arrays



a0 ptr (address)

Memo-
load / store

lw t0, (a0)

↑ dest ↑ source

t0 = *a0

Control

if/else

for loop

{

if (val > 0) {

 r = 1;

} else {

 r = 0;

}



ASM

l: do - interval

ble a0, zero, else

li t0, 1
done

else:

li t0, 0

done:

mv a0, t0

li t0, 1

addi t0, zero, 1

ble <=

breq ==

blt <

bge >=

bgt >

loops

loop (int n)

int i;

int sum = 0;

for (i = 0; i < n; i++) {

sum = sum + i;

}

a0 - int n
to - int i
t1 - int sum

loop_s:

i: to, 0

i: t1, 0

loop: