CS 315-01 RISC-V Assembly 2

Functions, Arrays, Control

Project01 explaining code quality

Lab02 today Tue 10th

Exam problems Wed 11th

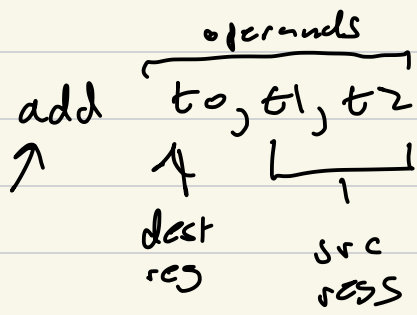
Project02 coming by Fri

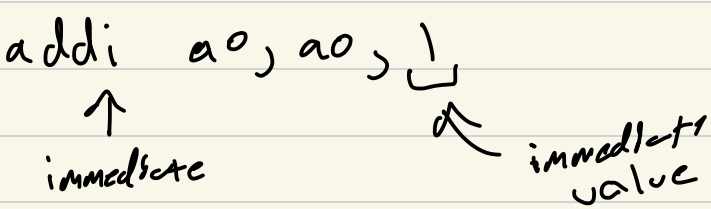
RISC-V Assembly

- More instructions
- array access
- if/then/else
- loops
- gdb

Instructions and Registers

32 64 bits wide





registers: x0, x1, ..., x31

arguments a0, a1, a2, a3, ...

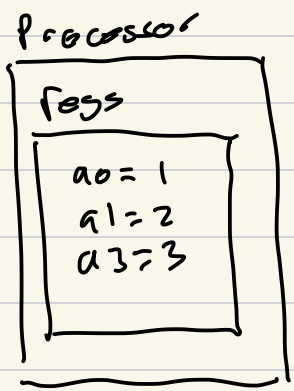
return value a0

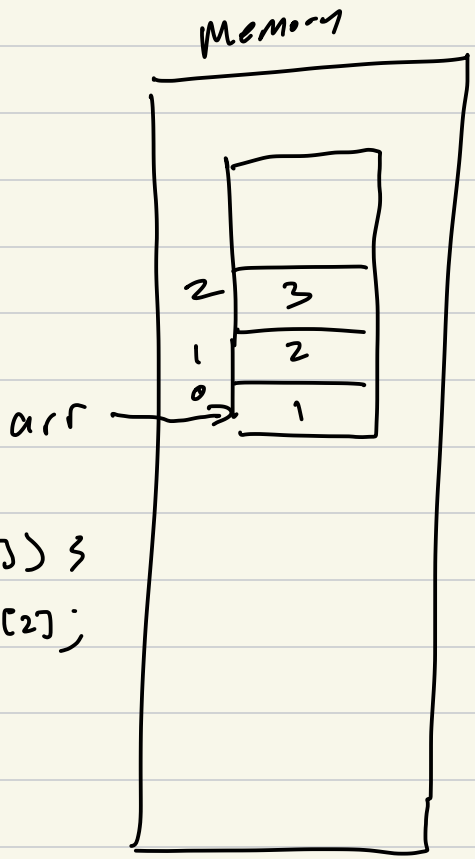
3 categories of instructions

- data processing
- control
- memory

Arrays

`int arr[3] = {1, 2, 3};`






```
sum3arr_c(int arr[]) {
  return arr[0] + arr[1] + arr[2];
}
```

load / store

```
lw t0, (a0)      #  t0 = *a0;
```

- `lw` ↑ load word (32 bit value)
- `t0` ↑ dst reg
- `(a0)` ↑ addr

4 bytes (int)

Control Statements

if / then / else

C

```
int val; int r;

if (val > 0) {
    r = 1;
} else {
    r = 0;
}
```

ASM

```
a0 - int val
t1 - int r;

    ble a0, zero, else
    li t0, 1
    j done
else:
    li t0, 0
done:
```

blc is a conditional branch

j is an unconditional jump

loops

C

```
loopsum(int n)

int i;
int sum=0;
for (i=0;i<n;i++){
    sum=sum+i;
}
```

ASM

```
# t0 - int i
# t1 - int sum

loopsum_s:
    li t0,0
    li t1,0

loop:
    bge t0,t1,done
    add t1,t1,t0
    addi t0,t0,1
    j loop

done:
    mv a0,t1
    ret
```