

CS 315-01 RISC-V

Project 01

Lab 02

RISC-V

basic structure of Assembly code

More instruction

array access

if / then

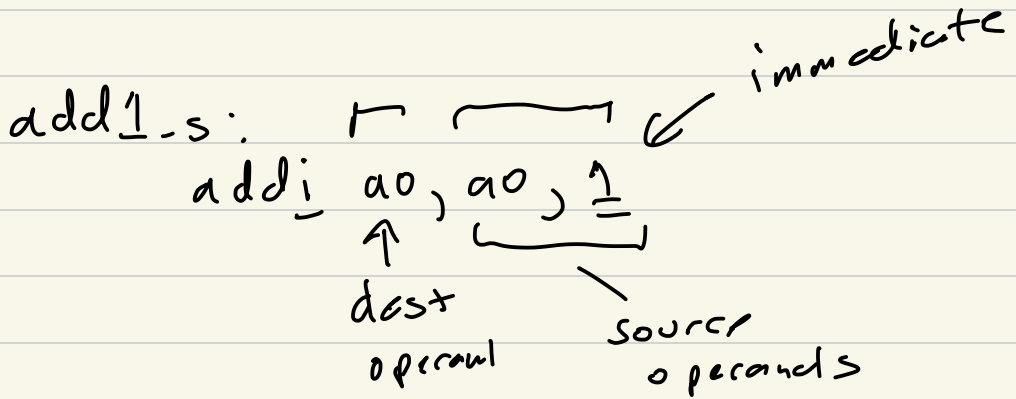
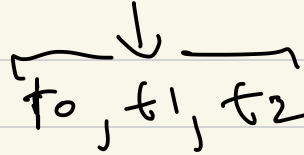
loops

gdb

instruction

operands

add



arguments

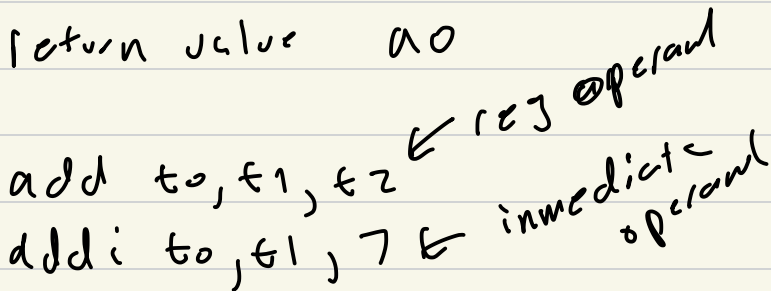
a0, a1, a2, a3, a4, ...

return value

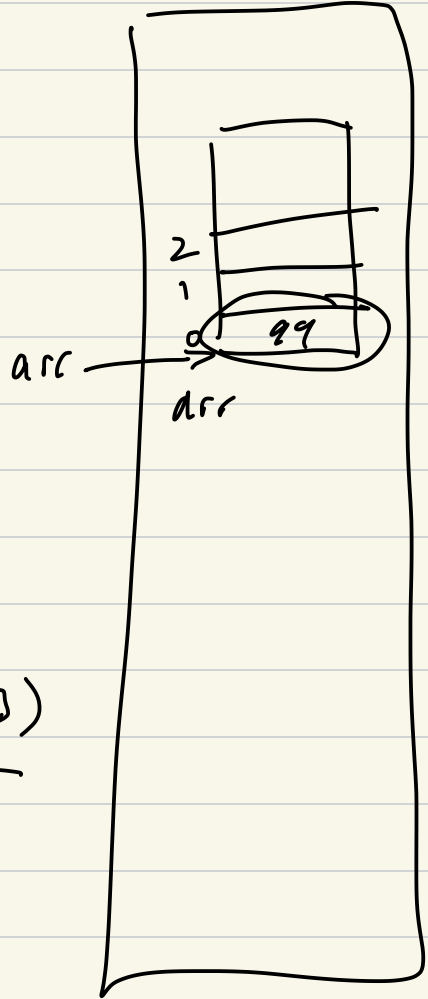
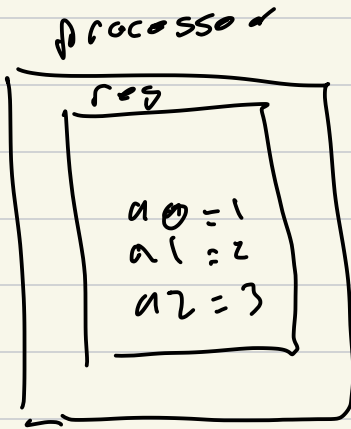
a0

add r0, r1, r2

addi r0, r1, 7



# Arrays



a0 ptr / address

a0 = arr (&arr[0])

load / store

lw to, (a0)  
    ↑      ↑  
    dst   addr  
to = \*a0 ;

Control

if/then

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

ASM

```
ble val a0, zero, else
```

```
li t0, 1
j done
```

else:

```
li t0, 0
```

done:

loopSum(int n)

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

```
# t0 - int i;
# t1 - int sum
loopSum-s:
```

```
li t0, 0
```

```
li t1, 0
```

loop:

```
bgr t0, a0, done
```

```
add t1, t1, t0
```

```
- addi t0, t0, 1
```

```
j loop
```

done:

```
mv a0, t1
```

```
ret
```

