

CS315-01 RISC-V Assembly 2

Functions, Arrays, Control

Type of instructions

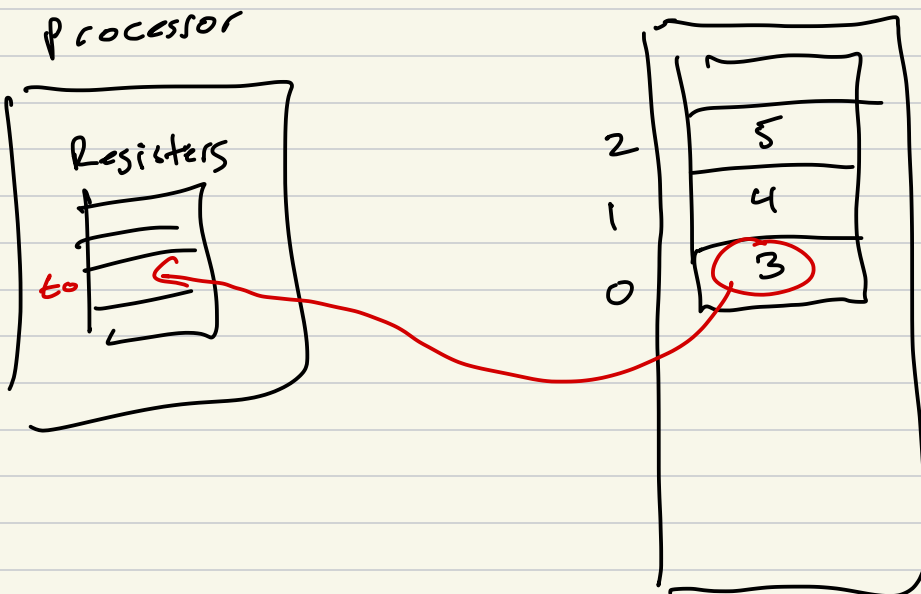
1) data processing

2) Control

3) Memory

Arrays

`int arr[3] = {3, 4, 5};`



Memory Instructions

Load / Store

lw to, (ao)

↑ ↑ ↑

load dest address
word (pointer)

32 bits

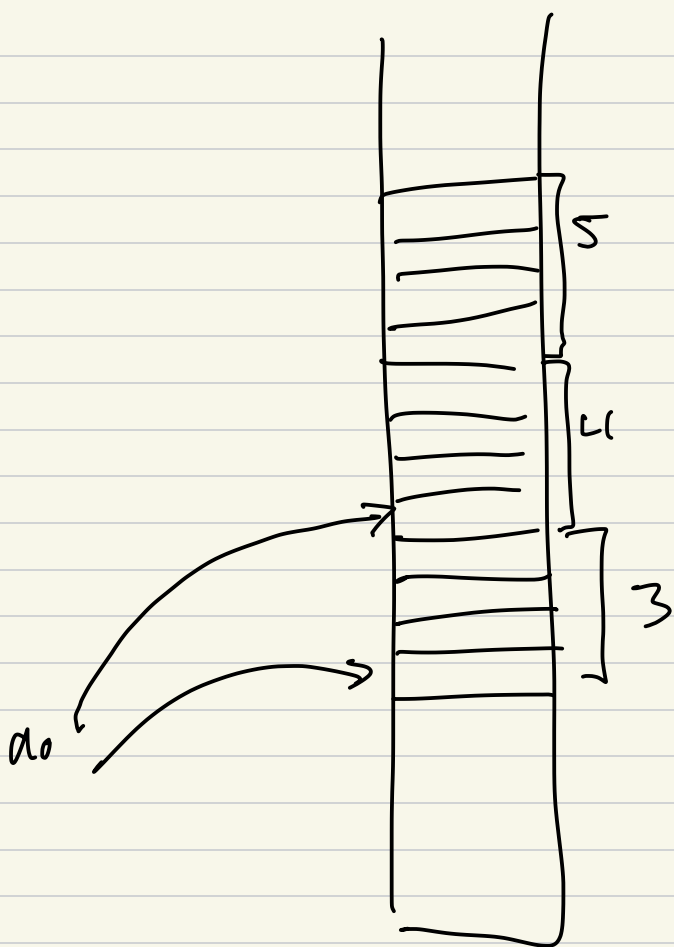
to = *ao.

int add3arr (int arr[]);

↑
ao
(addr)

Memory
byte
addressable





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
```

```
addi t1, zero, 1
```

```
j done
```

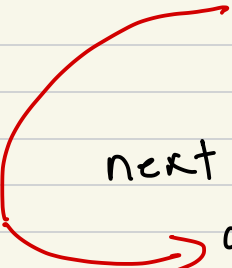
```
else:
```

```
addi t1, zero, 0
```

```
done:
```

1) unconditional jump
conditional branch

1) main:
 add to, zero, zero
 j next
 add
 add
next:
 addi to, zero, 1



Conditional Branch

ble branch on less than
or equal

ble to, t1, else

$t_0 \leq t_1$?
.

Loops / Iteration

ASM

```
int loopsum(int n) {  
    int i;  
    int sum = 0;  
    for (i = 0; i < n; i++) {  
        sum = sum + 1;  
    }  
    return sum;  
}
```

```
# to = int i;  
# t1 = int sum
```

```
loopsum:
```

```
li to, 0  
li t1, 0
```

```
loop:
```

```
bge to, a0, done
```

```
add t1, t1, to
```

```
addi to, to, 1
```

```
j loop
```

```
done:
```

```
mv a0, t1
```

```
ret
```