

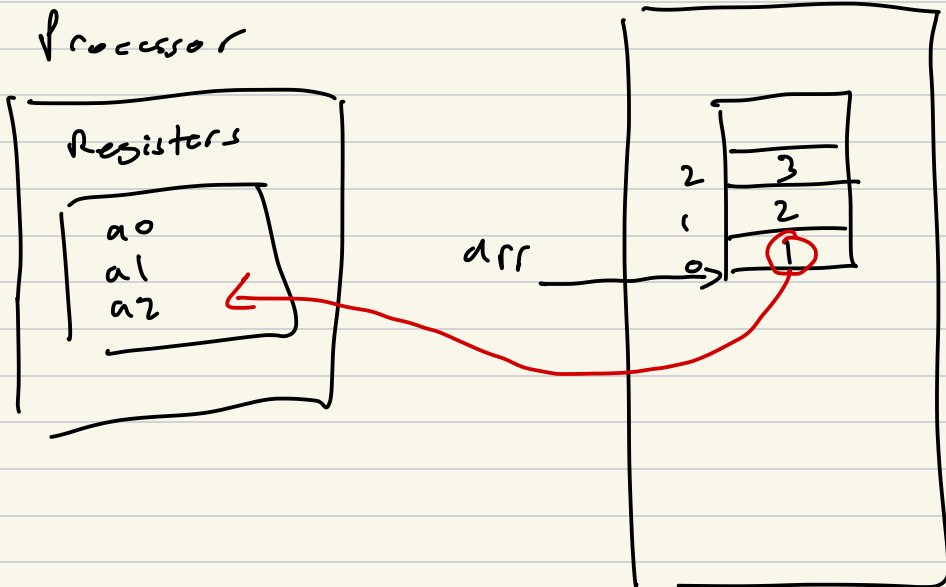
CS315-01 RISC-V Assembly 2

Functions, Arrays, Control

Types of instructions

- 1) data processing
- 2) control
- 3) memory

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



Memory instructions

load/store

lw to, (a0) # to = *a0;

↑ ↑ ↑
load dest addr
word
32 bit
value

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:

conditional branch
unconditional jump

main:

add t0, zero, zero

j next

add
add

next:

addi t0, zero, 1

Conditional branch

ble branch on less than
or equal

ble a0, a1, else

else:

```

if (x == 0) {
    y = 1;
}
}
done:
{
}
bne a0, zero, done
li to, 1

```

Loops / iteration

ASM

```

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

```