CS 315-01 RISC-V Emulation Lab03 JAL Mem

Lab03 due tonight 11:59 pm

Project04 published

Lab03 exam problems - coming soon

Exam - 1 note sheet allowed

Extra OH 4:30 pm - 5:30 pm

Lab03

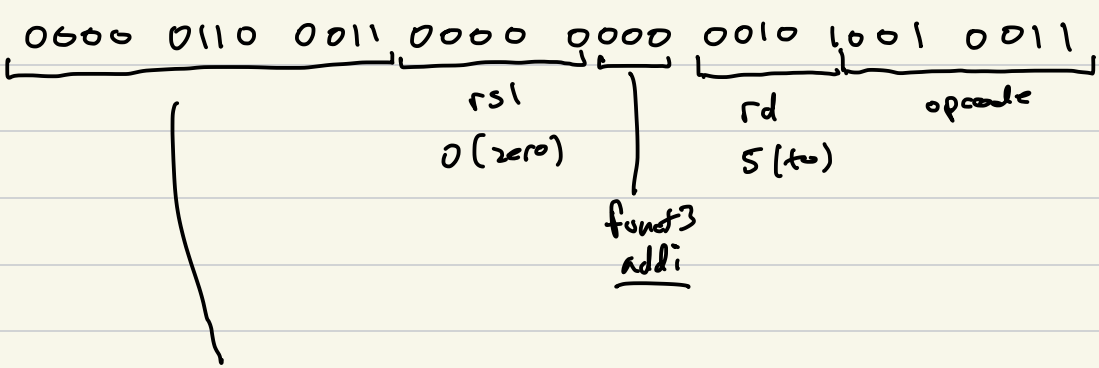
starter	rv_emu.c	131	loc
solution	rv_emu.c	225	loc

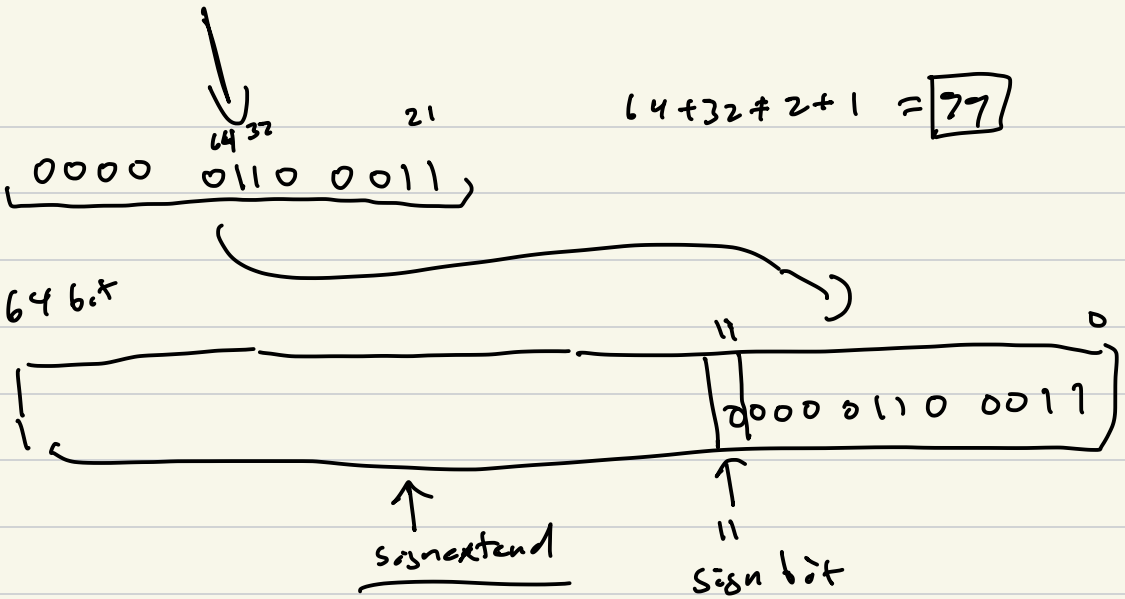
---

li t0, 99 → addi t0, zero, 99

↓

0x06300293





64+32+2+1 = 77

uint64_t

imm11_0 = get_bits(iw, 20, 12);

int64_t imm;

imm = sign_extend(imm11_0, 11)

sign bit

← shift left

→ shift right (arith)

Branche

Extract fields

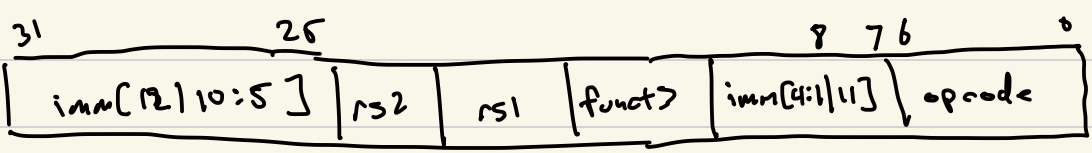
funct3

rs1

rs2

imm

1) get parts
2) combine parts
3) sign extend



imm[12] imm[11] imm[10:5] imm[4:1] 0

1) get parts

```
uint32_t imm12   = get_bits(iw, 31, 1);
uint32_t imm10_5 = get_bits(iw, 25, 6);
uint32_t imm4_1  = get_bits(iw, 8, 4);
uint32_t imm11   = get_bits(iw, 7, 1);
```

2) combine parts

```
uint64_t uimm;

uimm = (imm12 << 12) | (imm_11 << 11)
     | (imm10_5 << 5) | (imm4_1 << 1) | 0;
```

3) sign_extend

```
int64_t imm = sign_extend(uimm, 12);
```

If you take the branch

```
pc = pc + offset (imm)
```

Else

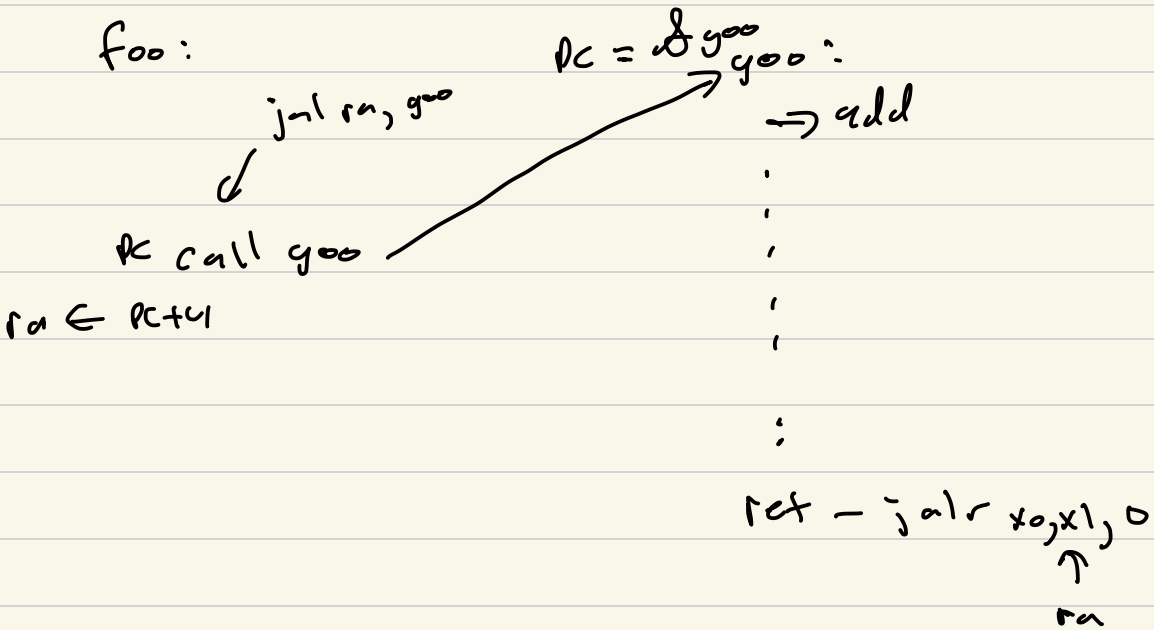
```
pc = pc + 4
```

JAL Jump and Link

imm

call jal x1, offset
ra

j jump → jal x0, offset
zero



Mem instructions - Loads & Stores

i-type

lw t0, offset(a0) → lw t0, 8(a0)

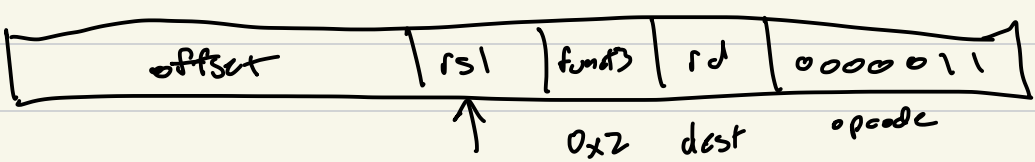
t0 = *(a0 + offset)

t0 = *((uint32_t *)(a0 + offset))

Target address

TA = a0 + offset (Imm)

rd = *((uint32_t *) TA);



lb uint8_t *

lw uint32_t *

ld uint64_t *

Stores → S-type

sw t0, offset(a0)

TA = a0 + offset (offset → imm)

*(a0 + offset) = t0