# CS 315.02 RISC-V Emulation DEV

Projects out today,
Project 03 exam problems

rv_emu.c given 97
rv_emu.c sol 203

addi a0, zero, 99

li a0, 99

mv a0, to

addi a0, to, 0

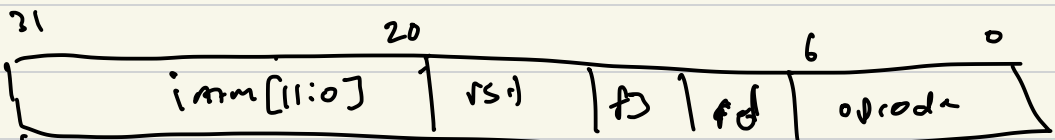
0x 0630 0513

0000 0110 0011 0000 0000 0101 0001 0011

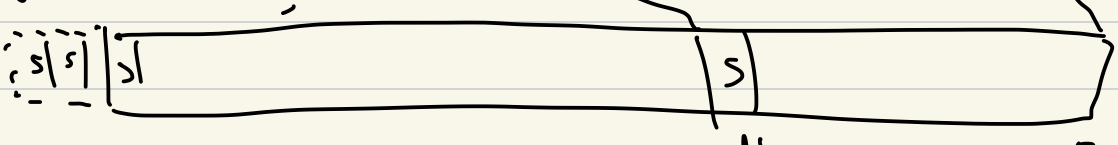
f3 rd opcode

i-type

000




uint64_t immv;



```
uint64_t immu = get_bits(iw, 20, 12);

int64_t imm = sign_extend(immu, 11);
```

---

Branches

B-type immediate ← branch offset

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

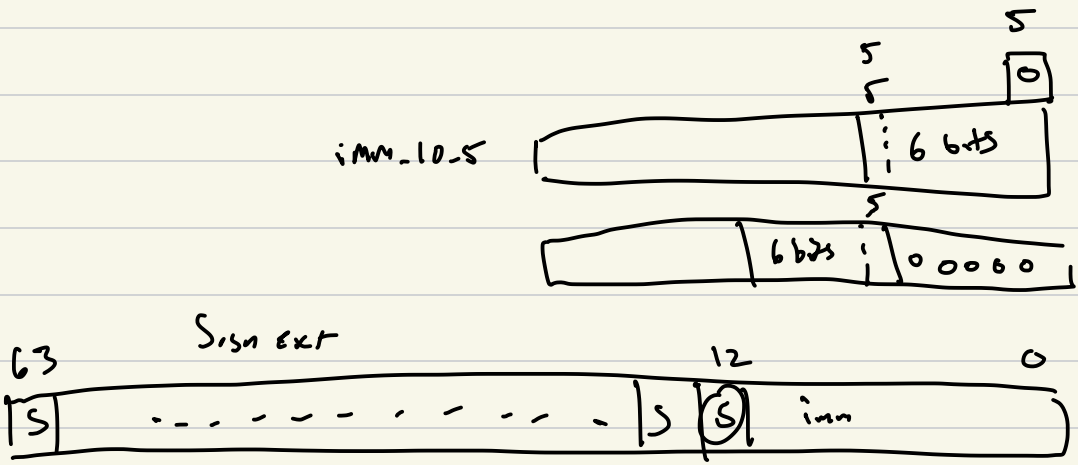
(1) get parts

```
uint32_t imm_12 = get_bits(iw, 31, 1);
uint32_t imm_11 = get_bits(iw, 7, 1);
uint32_t imm_10_5 = get_bits(iw, 25, 6);
uint32_t imm_4_1 = get_bits(iw, 8, 4);
```

② Combine parts

```
uint64_t immu;

immu = (imm_12 << 12) | (imm_11 << 11)
     | (imm_10_5 << 5) | (imm_4_1 << 1) ;
```




③ Sign extend

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

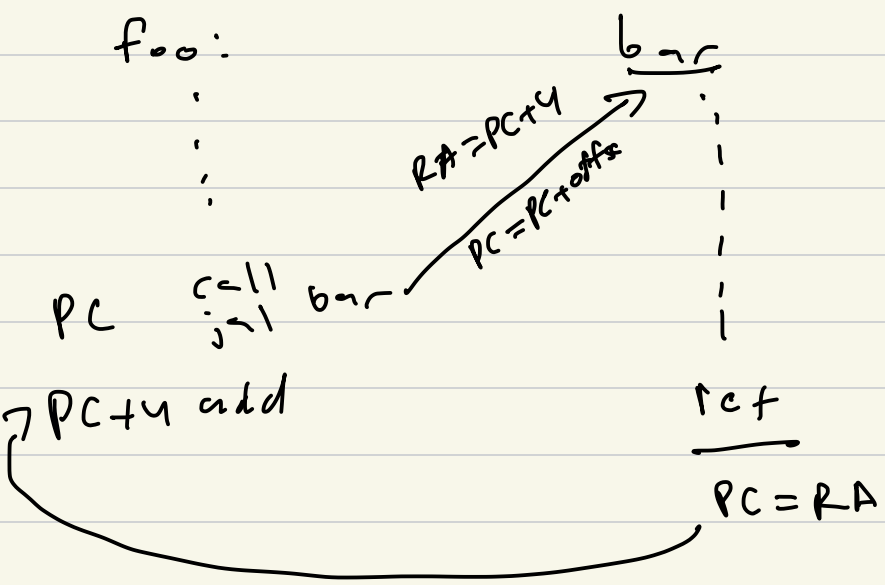
# JAL Jump and Link

JAL
- j
- jal/call

jal x0, offset
zero PC = PC + offset

jal ra, offset

[ RA = PC + 4
PC = PC + offset ]



# Loads and Stores

I-type

```
lw  a0, (a1)
lw  a0, 8(a1)
```

base address

```
lw  a0, offset(a1)
    rd  imm    rs1
```

a0 = *(a1 + offset)

Target address

uint64_t TA = a1 + offset

rd = *((uint32_t *) TA);

rsp->[rd] = *((uint32_t *)TA);

| | |
|---|---|
| lb | uint8_t * |
| lw | uint32_t * |
| ld | uint64_t * |

Stores S-type

sw a0, offset(a1)

TA = a1 + offset

*((uint32_t *)TA) = a0;