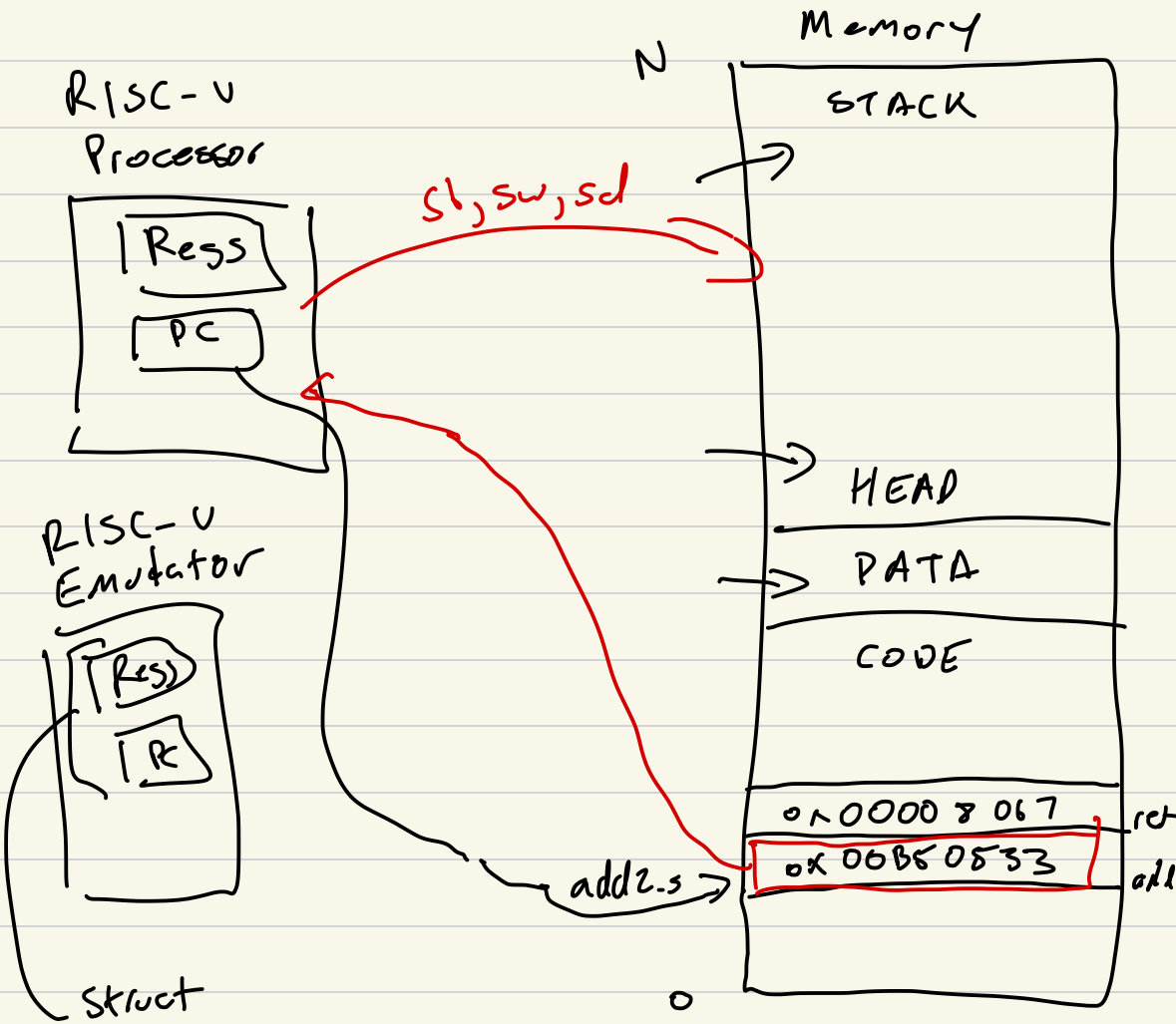
CS315-02 RISC-V Emulation



Processor State

Registers (32)

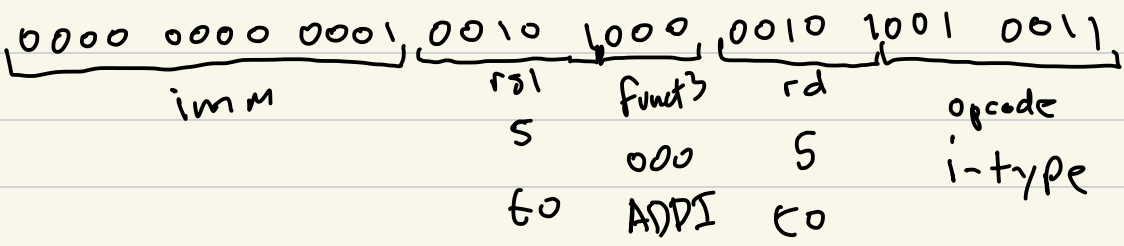
PC program counter

Memory

Implementation

Incremental approach

1. Identify instruction: ADDI
2. Identify instruction format
   i-type
3. Break down the iw
   addi t0, t0, 1 0x00128293



4. Implement oc add to a type function
5. Get all fields for the type

(6) Construct immediate value as needed

(7) Updating state

- update rd
- update memory
- update PC (PC = PC + 4)

(8) ret

Dealing with bits

```
uint32_t get_bits(uint64_t num,
                  uint32_t start,
                  uint32_t count)
```

```
rsl = (iw >> 15) & 0b11111;
rsl = get_bits(iw, 15, 5);
```

```
x = num >> 15;
mask = (0b1 << 5) - 1;
x = x & mask;
```

0b1 << 5

0b1

0 0 0 0 0

1 0 0 0 0 0

Dec 1000

\- 1

0999

100000 (32)

\- 1

011111 (31)

```
uint32_t get_bit(uint64_t num,
                 uint32_t which);
```

ADDI

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

regs[rd] = regs[rs1] + immu

addi t0, t0, -3

0x FFD2 8293

imm

1111 1111 1101 0010 1000 0010 1001 0011

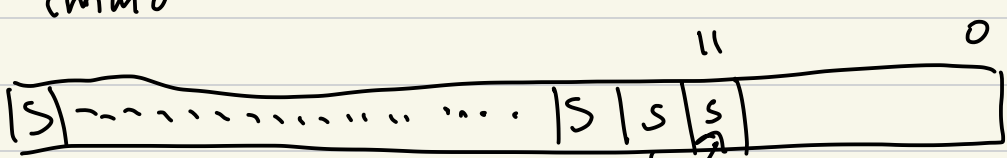
0000 0000 0010

\+ 1

0000 0000 0011 (3)

int64_t sign_extend (uint64 num,
uint32_t start)

immu



immu<<

imm

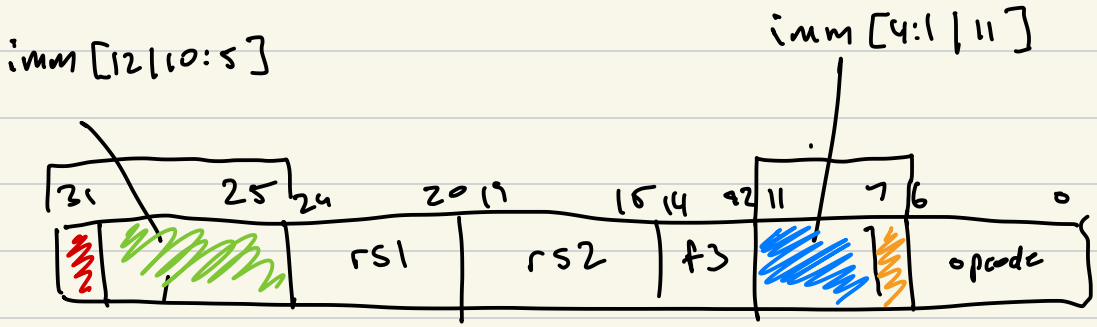
imm

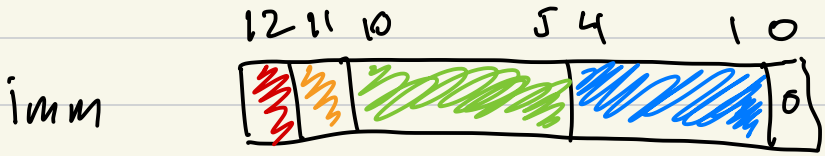
int64_t
imm

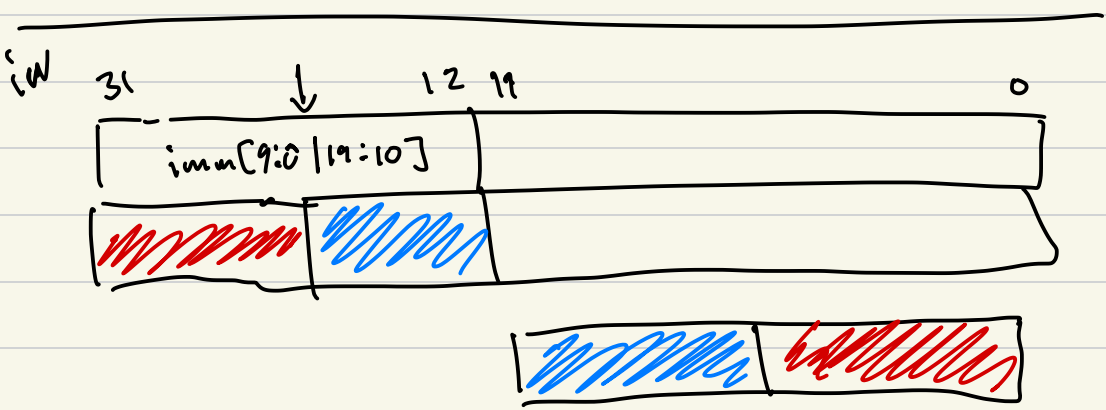
>>

B-type Branches

beq t0, t1, done







```
imm9_0 = get_bits(iw, 22, 10);
imm19_0 = get_bits(iw, 12, 10);

imm = (imm19 << 10) | imm9_0;
```