Lec 13

Tuesday, June 7, 2022 12:00 PM

Addressing Modes

* Immediate addressing mode

  A ← 0X13

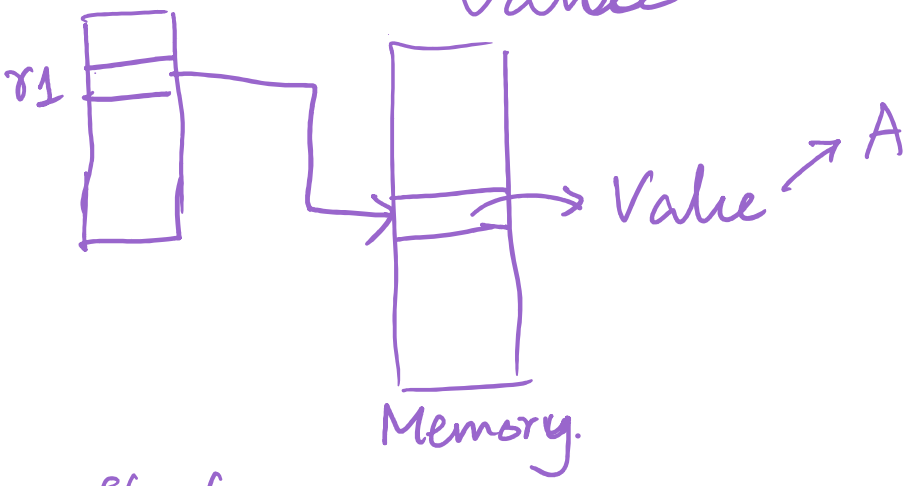
* Register direct " "

  A ← r1

* Register indirect " "

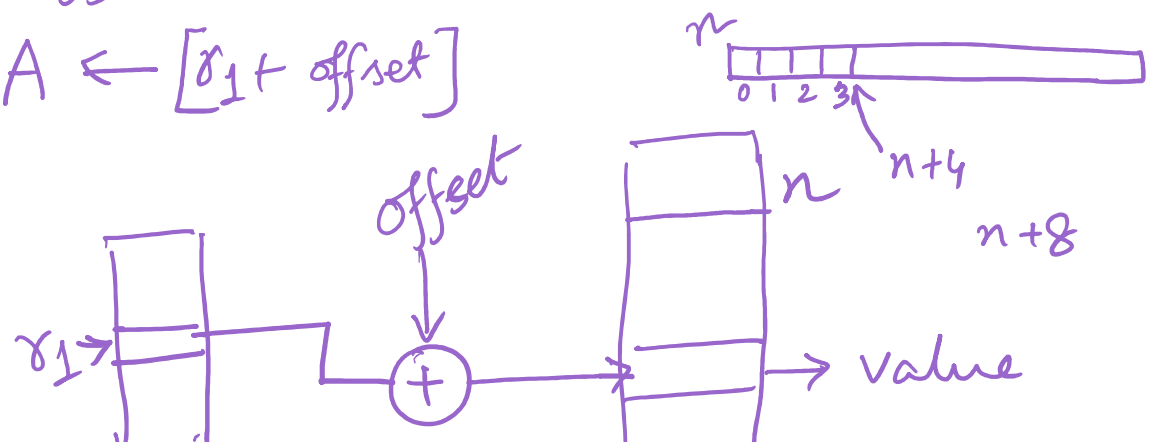
  A ← [r1]

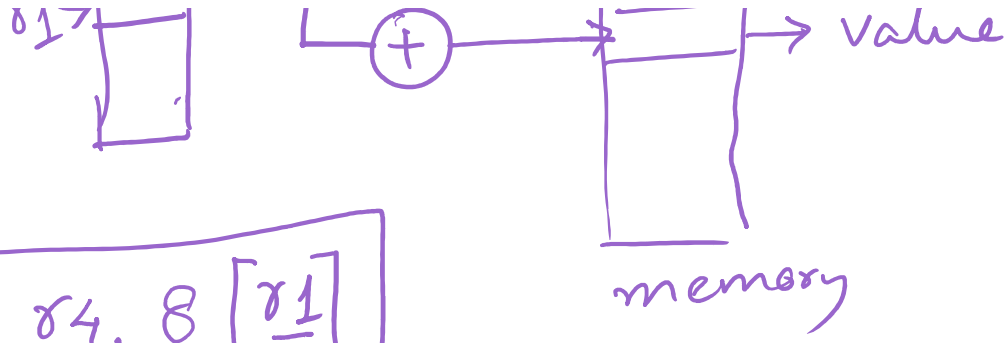
the register saves the address of the memory location that contain the value



Base-offset

A ← [r1 + offset]





st r4, 8 [r1]

SIMPLE RISC

16 register r0 ... r15

r14 ← stack pointer (sp)

r15 ← return address register (ra)

flags.E = 1 (equality), flags.GT = 1 (greater than)

mov

mov r1, r2 $r_1 \leftarrow r_2$

mov r1, 3 $r_1 \leftarrow 3$

16 bit immediates.

-2^{15} to $2^{15}-1$

Arithmatic / Logic Instⁿ

Arithmatic / Logic Inst

add, sub, mul, div, mod, cmp

add r_1, r_2, r_3 $\quad r_1 \leftarrow r_2 + r_3$

add $r_1, r_2, 10$

sub r_1, r_2, r_3 $\quad r_1 \leftarrow r_2 - r_3$

mul r_1, r_2, r_3 $\quad r_1 \leftarrow r_2 \times r_3$

div r_1, r_2, r_3 $\quad r_1 \leftarrow r_2 / r_3$ (quotient)

mod r_1, r_2, r_3 $\quad r_1 \leftarrow r_2 \bmod r_3$ (remainder)

cmp r_1, r_2 $\quad$ set flag

```
a = 3
b = 5
c = a+b
d = c-5
```

$a \leftrightarrow r_0$

$b = r_1$

$c = r_2$

$d = r_3$

```
mov r0, 3
mov r1, 5
add r2, r0, r1
```

sub $r_3, r_2, 5$

---

a = 3, b = 5 Compare 3 & 5

```
mov r0, 3
mov r1, 5
Cmp r0, r1
```

flags.E = 0

flags.GT = 0

Compute 31/29 - 50

```
mov r1, 31
mov r2, 29

div r3, r1, r2

sub r4, r3, 50
```

Branch Instⁿ

Unconditional Branch

b .foo → branch to .foo

add r_1, r_2, r_3

```
    b .foo
      .
      .
      .
.foo: sub r3, r2, r1
```

Conditional Branch Instⁿ

beq .foo → branch to .foo if flags.E = 1

bgt .foo → " " " if flags.GT = 1

* The flags are only set by the cmp instⁿ.