

MIPS Assembly

part 2

E155

Outline

- ❑ Addressing modes
- ❑ Examples

How Do We Address Operands?

Register Only Addressing

- ❑ Operands found in registers
 - Example: `add $s0, $t2, $t3`
 - Example: `sub $t8, $s1, $0`

Immediate Addressing

- ❑ 16-bit immediate used as an operand
 - Example: `addi $s4, $t5, -73`
 - Example: `ori $t3, $t7, 0xFF`

How Do We Address Operands?

Base Addressing

□ Address of operand is:

base address + sign-extended immediate

– Example: `lw $s4, 72($0)`

• Address = $\$0 + 72$

– Example: `sw $t2, -25($t1)`

• Address = $\$t1 - 25$

How Do We Address Operands?

PC-Relative Addressing

```
0x10      beq    $t0, $0, else
0x14      addi   $v0, $0, 1
0x18      addi   $sp, $sp, i
0x1C      jr     $ra
0x20      else:  addi   $a0, $a0, -1
0x24      jal    factorial
```

Assembly Code

```
beq $t0, $0, else
(beq $t0, $0, 3)
```

Field Values

op	rs	rt	imm
4	8	0	3
6 bits	5 bits	5 bits	5 bits 5 bits 6 bits

How Do We Address Operands?

Pseudo-direct Addressing

0x0040005C jal sum

...

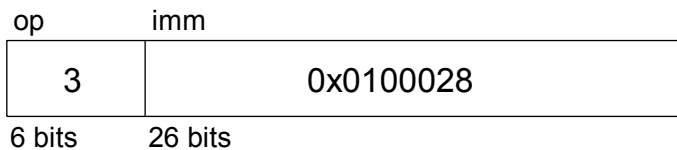
0x004000A0 sum: add \$v0, \$a0, \$a1

JTA 0000 0000 0100 0000 0000 0000 1010 0000 (0x004000A0)

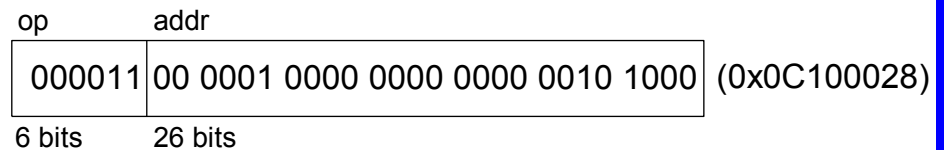
26-bit addr 0000 0000 0100 0000 0000 0000 1010 0000 (0x0100028)

0 1 0 0 0 2 8

Field Values



Machine Code



Examples

- ❑ Load \$t0 from address 0x40
- ❑ Load \$t1 from address 0x7000
- ❑ Add \$t0 to \$t1 and store result in \$t2 into address 0x2000

```
lw    $t0, 0x40($0)
lw    $t1, 0x7000($0)
add   $t2, $t0, $t1
sw    $t2, 0x2000($0)
```

Examples

- ❑ Load \$t0 from address 0x40
- ❑ Load \$t1 from address 0x7000
- ❑ Add \$t0 to \$t1 and store result in \$t3 into address 0xABCD8880

```
lui $t2, 0xABCD
ori $t2, $t2, 0x8880
lw $t0, 0x40($0)
lw $t1, 0x7000($0)
add $t3, $t0, $t1
sw $t3, 0($t2)
```


Examples

□ $s = a + b + c$

```
add $t0, $s0, $s1 # a=$s0, b=$s1
```

```
add $s7, $t0, $s2 # c=$s2, s=$s7
```

Examples

- ❑ if (x==5) y=4;
- ❑ else y=8;

```
        addi $t0, $0, 5
        bne  $t0, $s3, around  # x=$s3
        nop
        addi $s4, $0, 4 # y=$s4=4
        j   around2
around:  addi $s4, $0, 8 # y=$s4=8
around2: .....
```

Examples

- ❑ if ($x > 4$) $y=5$;
- ❑ else if ($x < 1$) $y=x$;

```
        addi $t2, $0, 4
        slt  $t0, $s3, $t2    # x=$s3
        bne $t0, $0, around
        nop
        addi $s4, $0, 5 # y=$s4=5
        j   around2
around:  addi $t2, $0, 1
        slt  $t0, $x3, t2
        bne $t0, $0, around1
        nop
        j   around2
around1: $s4, $0, $s3        # y=$s4=s3=x

around2: .....
```

Examples

- ❑ `m=0;`
- ❑ `for (i=0; i<5, i++) m=m+1;`

```
        addi $s5, $0, 0        # m=0=$s5
        addi $s6, $0, 0        # i=0=$s6
        addi $t0, $0, 5        # $t0=5
loop    slt $t1, $s6, $t0      # x=$s3
        bne $t1, $0, next
        nop
        j done
next:   addi $s5, $s5, 1
        addi $s6, $s6, 1
        j loop

done:   .....
```

Examples

❑ for (i=1, i<4, i++)

❑ for (j=i, j<5, j++) q=q+i-j;

```
                                # i=$s6, j=$s7 # q=$s8
                                # i=1
                                # t0=4
                                # t2=5
loop1:  slt $t1, $s6, $t0
        beq $t1, $0, done
        nop
        add $s7, $0, $s6        #j=i
loop2:  slt $t3, $s7, $t2        #j<5
        beq $t3, $0, done2      #exit inner
        nop
        add $t4, $s8, $s6        #t4=q+i
        sub $s8, $t4, $s7        #q=t4-j
        addi $s7, $s7, 1         #j++
        nop
done2:  addi, $s6, $s6, 1        #i++
        j loop1
done:   nop
```

Examples

- ❑ Int sum, i; # sum in \$s1, i in \$s6
- ❑ Int a[9]; # array start at \$gp
- ❑ for (i=0, i<9, i++) sum=sum+a[i];

```
        addi $s1, $0, 0        # sum=0
        addi $s6, $0, 0        # i=0
        addi $t0, $0, 9        # t0=9
loop:   slt $t1, $s6, $t0      # i<9
        beq $t1, $0, done      # exit loop
        nop
        sll $t3, $s6, 2        # i*4
        add $t4, $gp, $t3      # add of a[i]
        lw $t5, 0($t4)         # a[i]
        add $s1, $s1, $t5      # sum=sum+a[i]
        addi $s6, $s6, 1       # i++
        j loop
done:   nop
```