MIPS Instruction Set

Dr. Prasenjit Chanak

Department of Computer Science and Engineering
Indian Institute of Technology (BHU), Varanasi
UP, 221005

Control Instruction

- Decision making instructions - alter the control flow
- MIPS conditional branch instructions:

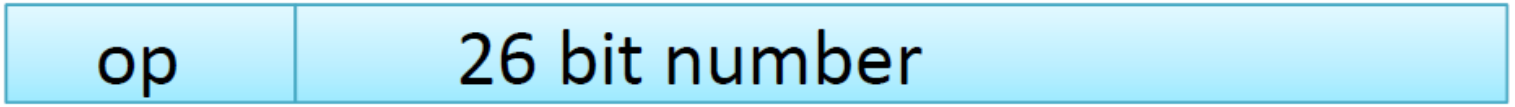
  bne and beq
- Example:

```
if (i == j)                bne $s0, $s1, Label
    h = i + j;             add $s3, $s0, $s1
                    Label: ....
```

Format of Instructions for Control

- beq, bne I - format

- j J - format (new)

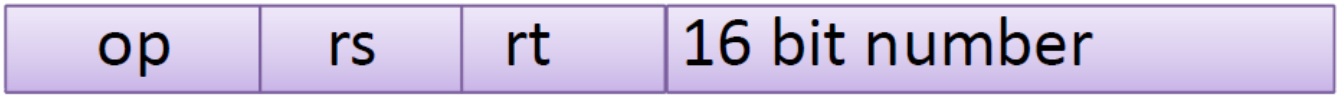


- slt R – format :set-if-less-than

Constants in MIPS instructions

```
addi $29, $29, 4 ‘i’ is for ‘immediate’
slti  $8, $18, 10
andi $29, $29, 6
ori   $29, $29, 4
```

I – Format is used



A special constant

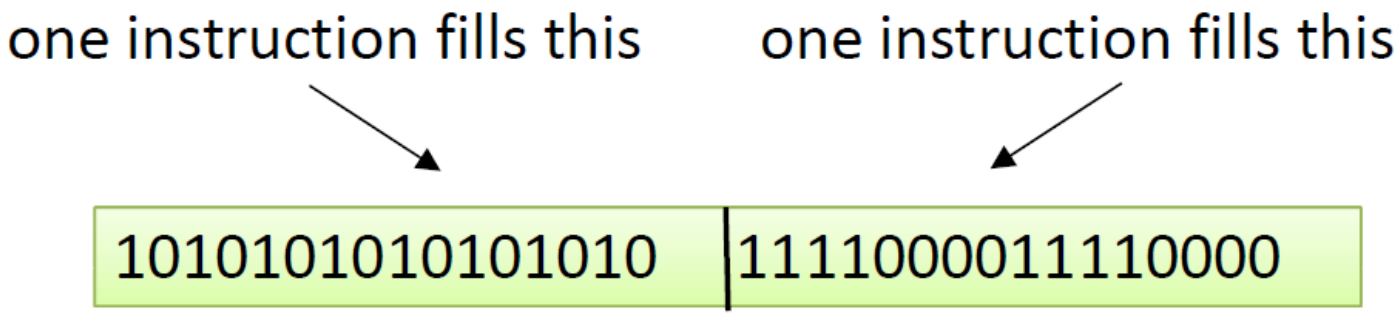
‘0’ is a special constant, hard-wired into register $0 (also called $zero)

add $s2, $s4, $zero

means move from $s4 to $s2

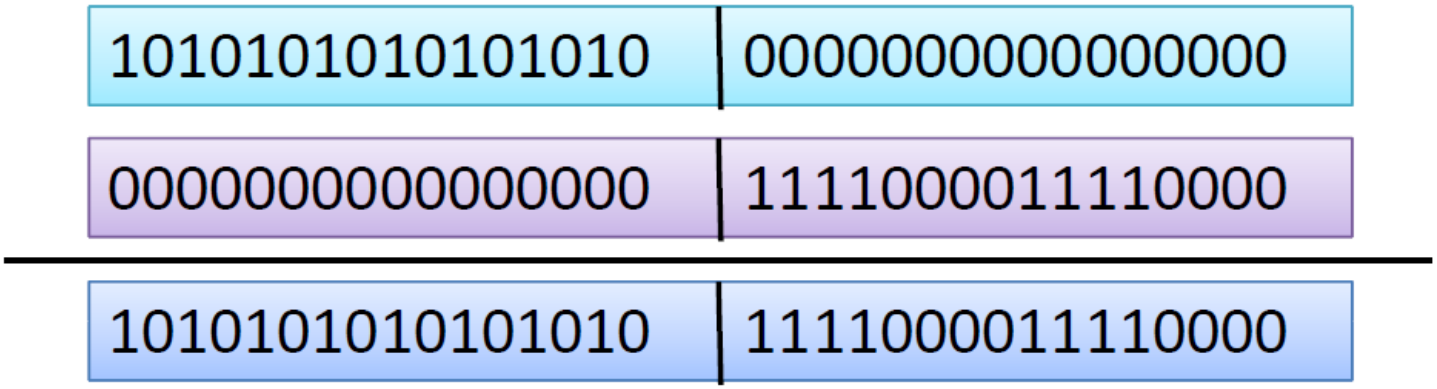
How about larger constants?

To load a 32 bit constant into a register, we use two instructions



Loading larger constants

- new "load upper immediate" instruction
  lui $t0, 1010101010101010
- then get the lower order bits right, i.e.,
  ori $t0, $t0, 1111000011110000



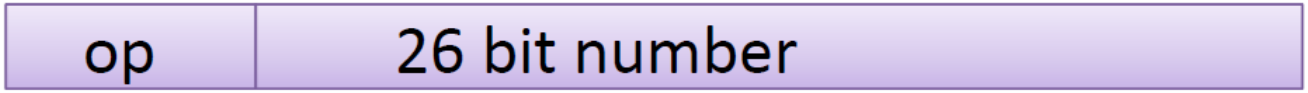
Summary of Instructions learnt

Instructions	Format
add, sub, addi, subi	R, I
and, or, andi, ori	R, I
slt, slti	R, I
beq, bne	I
j	J
lw, sw	I
lui	I

MIPS ISA features - encoding

- addi, lui, beq, bne, lw, sw I - format

- j, jal J - format



- add, jr R - format

MIPS ISA features - summary

- All instructions of same size
- Only 3 formats
- Fair number of GP registers
- Simple operations – either arith/logic or memory access or control transfer
- Limited addressing modes
- Separate fields for src1, src2 and dest

Location of operands – R/M

- R-R Both operands in registers
- R-M one operand in register and one in memory
- M-M Both operands in memory
- R+M Combines R-R, R-M and M-M

How many operand fields?

- 3 address machine — r1 = r2 + r3
- 2 address machine — r1 = r1 + r2
- 1 address machine — Acc = Acc + x

  Acc is implicit
- 0 address machine — add values on top of stack

Register organizations

- Register-less machine
- Accumulator based machine
- A few special purpose registers
- Several general purpose registers
- Large number of registers / register windows