

# CS 264 Intro Systems: MIPS Instruction Encodings

February 19, 2020

## 1 Instruction Encoding

These notes use MIPS instruction encodings because it's the simplest one. For MIPS opcodes, see:

[https://en.wikibooks.org/wiki/MIPS\\_Assembly/Instruction\\_Formats](https://en.wikibooks.org/wiki/MIPS_Assembly/Instruction_Formats)

**Problem:** How do we tell the CPU what instruction to execute?

### R-Format

- Used for register-register operations.
- Mostly arithmetic ops.
- **funct** Field is an extension of the opcode.

Examples:

```
add rd,rs,rt      ; rd <- rs + rt
jr [rs]           ; PC <- rs
srl rd,rt,#shamt  ; rd <- rt << shamt
```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode				rs				rt				rd				shamt				func											

**I-Format** Used for register-immediate operations.

```
addi rt,rs,#0x123 ; rt <- rs + 0x123
sw [rs],rt        ; MEM(rs) <- rt
bne $2,$0,$L3     ; Branch to label $L3 if register $2 != 0
beq $2, $3, $L3   ; Branch to label $L3 if register $2 == $3
```

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode				rs				rt				imm16																			

### J-Format

jal func

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
opcode				imm26																											

## 2 Examples

### Example: listInsert

```
listInsert:
  0:  10000005      b      .L14 <listInsert+0x18>
  4:  8c820000      lw      v0,0(a0)
.L13:
  8:  10600005      beqz   v1,.L2 <listInsert+0x20>
  c:  00000000      nop
 10:  00402025      move   a0,v0
 14:  8c820000      lw      v0,0(a0)

.L14:
 18:  1440ffff      bnez   v0,.L13 <listInsert+0x8>
 1c:  0045182b      sltu   v1,v0,a1
.L2:
 20:  aca20000      sw     v0,0(a1)
 24:  aca40004      sw     a0,4(a1)
 28:  ac850000      sw     a1,0(a0)
 2c:  8ca20000      lw     v0,0(a1)
 30:  10400002      beqz   v0,.L15 <listInsert+0x3c>
 34:  00000000      nop
 38:  ac450004      sw     a1,4(v0)
.L15:
 3c:  03e00008      jr     ra
 40:  00000000      nop
```