Computer Organization

605.204

Module Three

Part Two

Language for the Machine

Module Three

- Part Two
- In this presentation, we are going to talk about :
- Language Design Goals
- Format
- Issues

Previously

- Previously we talked about:

- A Simple Machine Organization
- Stored Program Idea
- General Register Machine

- Now: Language for the Machine

Language for the Machine

- Design Goals
  - Simple easy to build Hardware
    easy to build Compiler
    easy to use
  - Maximum machine Performance
  - Minimum Cost
- Historically, computers were first programmed directly with machine code.

Language of the Machine

===VAX===

0101 0100 1101 0100 0101 0101 1011 0100

0101 0101 0110 0110 0100 0100 1010 0000

===MIPS===

1000 1110 0101 0001 0000 0000 0111 0100
0000 0010 0011 0010 0100 0000 0010 0000
1010 1110 0101 0001 0000 0000 0111 0100

Machine Code Format

- Bits - ones - zeros
- Formats give meaning to the bit patterns
- Fields subdivide the instruction
- ===VAX===

Operand	Operand	Operand	OpCode

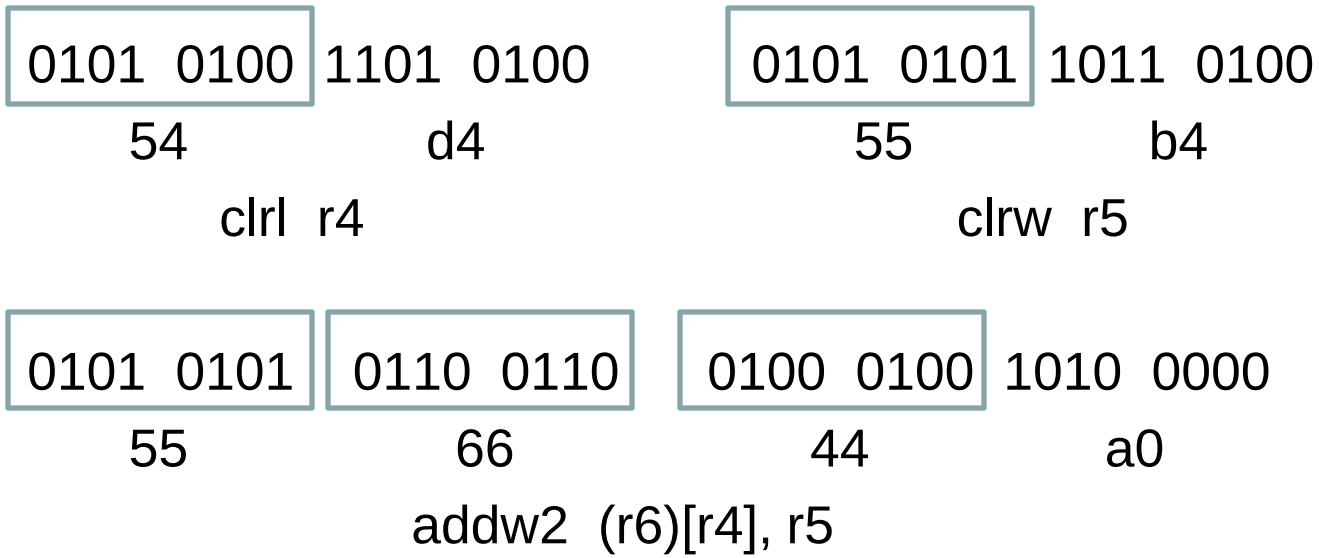
- ===MIPS===

OpCode	RegS	RegT	RegD	ShiftA	FnCode

Language for the Machine

===VAX===

operand specifier, operand specifier, operand specifier opcode



Language of the Machine
(more)

===MIPS===

100011	10010	10001	0000000001110100		
000000	10001	10010	01000	00000	100000
101011	10010	10001	0000000001110100		

```
LW      $s1, 116 ($s2)
ADD     $t0, $s1,$s2
SW      $s1, 116 ($s2)
```

Machine Code Format

===MIPS===

0000 0010 0011 0010 0100 0000 0010 0000

opcode,	reg.src1, rs	reg.src2, rt	reg.dst, rd	shiftamt,	function
000000	10001	10010	01000	00000	100000
0	17	18	8	0	32

ADD $t0, $s1, $s2

Hardware Instruction Format Issues

- Assembly language fields:

  opcode operands comments

- Machine code format problem:

  Limited number of bits (32) in the machine hardware word

  some bits needed for the op code

  3 operand address: 8 bits each - 256

  2 operand address: 12 bits each - 4096

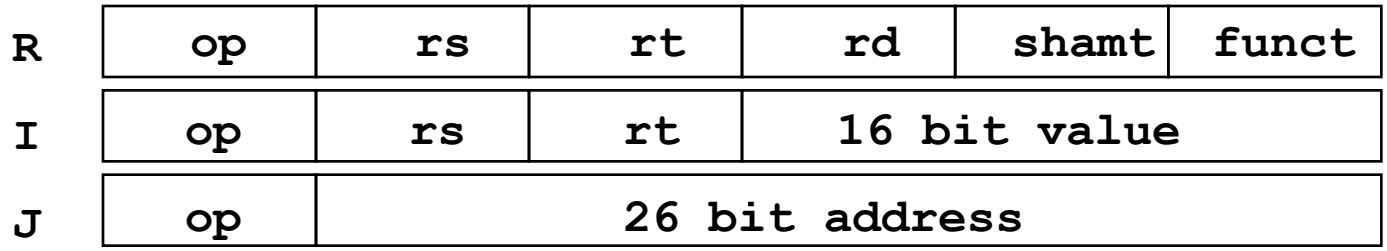
  1 operand address: 24 bits - 16 million

  0 operand address: implied to be stack

- Fixed or Variable length machine code

MIPS Machine Language

- Simple instructions, all 32 bits wide
- Very structured, no unnecessary baggage
- Only three instruction formats
- Rely on compiler to achieve performance goals



Summary

- Language of the Machine

- Next: An introduction to the MIPS Assembly Language