Computer Architecture

Spring 2020

Hamed Farbeh

farbeh@aut.ac.ir

Department of Computer Engineering

Amirkabir University of Technology

Instruction Set Architecture (ISA)

- Instruction Set Architecture
  - A set of instructions used by a machine to run programs
  - Interface between hardware & software
  - Computer language vocabulary
  - Provides an abstraction of hardware implementation
    - Hardware implementation decides what and how instructions are implemented (microarchitecture)
  - ISA specifies
    - Instructions, Registers, Memory access, Input/output

Key ISA Decisions

- Instruction length?
- How many registers?
- Where operands reside?
  - Which instructions can access memory?
- Instruction format?
- Operand format?
  - How many? How big?

Where Operands Reside?

- **Stack** Machine
- **Accumulator** Machine
- **Register-Memory** Machine
- **Load-Store** Machine
  - aka Register-Register Machine

Where Operands Reside?

- **Stack Machine**
  - **“0-operand” ISA**
    - » **ALU operations (add, sub, and) don’t need any operands**
  - **“Push”**
    - » **Loads mem into 1st reg (“top of stack”),**
  - **“Pop”**
    - » **Does reverse**
  - **“Add”, “Sub”, “Mul”, and etc.**
    - » **Combines contents of first two regs**

Where Operands Reside?

- **Accumulator Machine**
  - **“1-operand” ISA**
  - **Only 1 register**
    - » **Called “accumulator”**
    - » **Stores intermediate arithmetic & logic results**
  - **Instructions include**
    - » **“Store”**
    - » **“Load”**
    - » **“acc ← acc + mem”**

Where Operands Reside?

- **Register-Memory Machine**
  - **Operands**
    - » **Register or memory**
  - **Arithmetic instructions can use data in registers and/or memory**

Where Operands Reside?

- **Register-Registor Machine (load-store machine)**
  - **Operands**
    - » **Register**
  - **Arithmetic instructions can use data in registers**
  - **No operation on memory**
    - » **Only read and write instruction (load and store)**

Where Operands Reside?

Code sequence for C = A + B

Stack	Accumulator	Register-Memory	Load-Store
Push A	Load A	Add C, A, B	Load R1,A
Push B	Add B		Load R2,B
Add	Store C		Add R3,R1,R2
Pop C			Store C,R3

Addressing Mode

- **Where the operands are located?**
- **Is specified by the instruction: Opcode**
- **Addressing Modes:**
  - **Implied mode**
    Complement ACC
  - **Immediate Mode**
    Operand is inside the instruction
    ADD R1, 105
  - **Register Mode**
    Operand is a register
    Add R1, R2, R3
  - **Register Indirect Mode**
    Add R1, R2, [R3]
  - **Auto-increment/decrement Mode:** Add R1, R2, [R3+4]

Addressing Mode

- **Addressing Modes:**
  - **Direct Mode**
  - **Indirect Mode**
  - **Relative Mode**
    Relative to PC
  - **Index Addressing Mode**
    - The content of an index reg. is added to address part of the instr: Add R1, R2, [R3+Const.]
  - **Base Register Mode**
    - The address part of the intr. Is added to the content of a base reg.
    - LW R1, 100(R3)

RISC vs. CISC

- **RISC:** Reduced Instruction Set Computer
- **CISC:** Complex Instruction Set Computer
  - Number of instructions
  - Addressing mode
  - Instruction length
  - Direct memory access
  - Hardware complexity
  - Number of registers
  - Cycles for instr. execution
  - Clock frequency
  - Code size