COMPUTER ORGANIZATION (IS F242)

LECT 06: COMPUTER ORGANIZATION

Driving force for CISC

- Increasingly complex high level languages
- Semantic gap Leads to:
  - Large instruction sets
  - More addressing modes
- Intension of CISC
  - Ease compiler writing
    - Compiler simplification?
    - Complex machine instructions harder to exploit
    - Optimization is more difficult
  - Improves execution efficiency

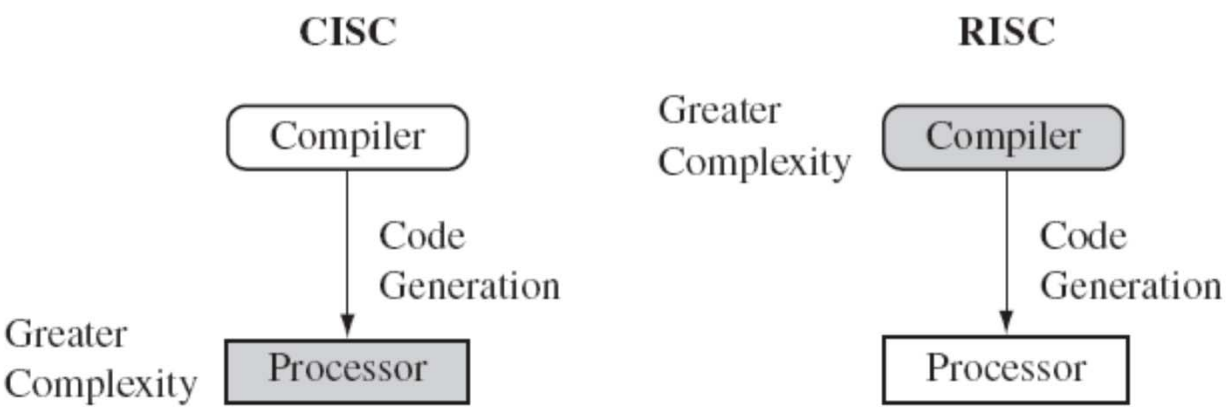
RISC–Reduced Instruction Set Computing

- Features
  - Large number of general purpose registers
    - Use compiler to optimize the register use
  - Limited and simple [but powerful] instruction set
  - Emphasis on optimizing the instruction pipeline
- Characteristics
  - One instruction per cycle [high clock speed]
  - Register to register operations
  - Few simple addressing modes & instruction formats
  - Hardwire (no micro code) control unit design
  - Fixed instruction format

Motivation for RISC Design

- Focuses on reducing the number and complexity of instructions
- Reduces the number of cycles needed per instruction
  - Goal: at least one instruction completed per clock cycle
- Designed with CPU instruction pipelinng in mind
- Fixed length instruction encoding
- Only load and store instructions access memory
- Simplified addressing modes
  - Usually limited to immediate, register indirect, register displacement and indexed
- Delayed loads and branches
- Prefetch and speculative execution

CISC Vs RISC



CISC vs. RISC. CISC emphasizes hardware complexity. RISC emphasizes compiler complexity.

Figure taken from ARM System Developer's Guide by Andrew N Sloss

CISC Vs RISC

- Not clear cut
- Many designs borrow from both philosophies
- Examples
  - PowerPC
  - Pentium II – IV
  - AMD Athlon

CISC	RISC
▪ Instructions are complex ▫ PUSHA, CALLP ▪ Motivated by high cost of memory ▪ Backward compatibility ▪ Large addressing modes ▪ Variable instruction format	▪ Instructions are simple ▪ Reduced number of cycles per instruction ▪ More general purpose registers ▪ Load – Store architecture ▪ Simplified addressing modes ▪ Fixed instruction format ▪ Pipelining easier ▪ Faster execution of instructions