Computer Architecture

CSEN 3104

Lecture 4

Dr. Debranjan Sarkar

CISC vs. RISC Architecture

Complex Instruction Set Computer (CISC)

- More instructions for complex tasks
- Complex instructions require relatively complex processing circuits, and too much expensive
- More complex addressing modes (auto-increment, auto-decrement etc.)
- Small number of General Purpose Registers (GPR)
- Variable-length Instruction formats (spanning more than one word)
- Less suitable for pipelining

Reduced Instruction Set Computer (RISC)

- Reduced number of instructions
- This lowers processor cost, without much impact on performance.
- Simple addressing modes (Register, Register indirect etc.)
- Large register set
  - uniform (no distinction between e.g. address and data registers)
- All instructions have same length (one word)
- More suitable for pipelining

CISC vs. RISC Architecture

Complex Instruction Set Computer

(CISC)

- Operands for the arithmetic / logic operations may be in the register or in memory
- Memory-to-memory data transfer possible
- Fewer instructions executed per program
- Complex instructions reduce program size but does not necessarily translate into faster execution
- Not constrained to load/store architecture. Typically use two-operand instruction format, with at least one operand in a register

Reduced Instruction Set Computer

(RISC)

- Operands for the arithmetic / logic operations are always in the registers
- Memory-to-memory data transfer not possible
- More instructions executed per program
- Though the program size is more, overall execution is faster
- Instruction Set Architecture: Load/ store

CISC vs. RISC Architecture

Complex Instruction Set Computer

(CISC)

- As the number of memory access is more, the impact of von Neumann bottleneck is more
- Mostly micro-programmed control units
- Example of CISC (IBM 370/168, VAX 11/780, Intel x86, PDP-11, Motorola 68000 etc.)

Reduced Instruction Set Computer

(RISC)

- Reduces the impact of von Neumann bottleneck by reducing the total number of the memory access made by the CPU
- Mostly hardwired control unit
- Example of RISC (MIPS, SUN Sparc, Intel i860, Motorola 88000, IBM RS6000, PowerPC, ARM etc.)

MIPS ISA

Case Study

MIPS (Microprocessor without Interlocked Pipelined Stages)

- RISC Instruction Set Architecture
- Developed by MIPS Computer Systems -> MIPS Technologies -> Wave Computing (since December 2018)
- Multiple versions of MIPS
  - MIPS I, II, III, IV and V (32 bit)
  - MIPS32/64 (for 32- and 64-bit implementations, respectively) – six releases
- MIPS is a modular architecture supporting up to four coprocessors (CP0/1/2/3)

MIPS (Microprocessor without Interlocked Pipelined Stages)

- CP0 is the System Control Coprocessor (an essential part of the processor)
- CP1 is an optional floating point unit (FPU)
- CP2/3 are optional implementation-defined coprocessors
- For example, in the “PlayStation” video game console, CP2 is the Geometry Transformation Engine (GTE), which accelerates the processing of geometry in 3D computer graphics.
- Originally, MIPS was designed for general-purpose computing.
- During the 1980s and 1990s, MIPS processors were used for personal, workstation and server computers
- MIPS processors are used in embedded systems such as residential gateways and routers

Thank you