Chapter 04: Instruction Sets and the Processor organizations

Lesson 20:

RISC and converged Architecture

Objective

- Learn the RISC architecture
- Learn the Converged Architecture

Reduced Instruction Set Computer (RISC) and fixed length instructions

Reduced Instruction Set Computer (RISC)

- Fixed length encoding of instructions
- Each instruction executes in a single clock cycle by hardwired implementation of each instruction

Reduced Instruction Set Computer (RISC)

- Focus on reducing the number of instructions and working with simpler instruction sets having a limited number of addressing modes
- Allowing them to execute more instructions in the same amount of time

Reduced Instruction Set Computer (RISC)

- Programs written for RISC architectures tend to take more space in memory but the RISC processor's increased clock rate allows it to execute its programs in less time than a CISC processor takes to execute its programs (which require fewer instructions)

Reduced Instruction Set Computer (RISC) and differences from CISC

Major differences between RISC and CISC architectures

- The set of instructions that can access memory
- A related issue that affects both RISC and CISC architectures is the choice of which *addressing modes* the architecture supports

Architecture addressing modes

- Architecture addressing modes are the set of syntaxes and methods that instructions use to specify a memory address, either as the target address of a memory reference or as the address that a branch will jump to

Clear delineation between RISC and CISC architectures

- RISC architectures— load-store architectures, meaning that only load and store instructions may access the memory system
- The GPR architecture provides for load-store architecture plus several other instructions

Difference between load-store architectures and other architectures

- Difference between load-store architectures and other architectures that can merge memory references with other operations
- Because RISC architectures are implemented using the load-store model, an RISC processor would require several instructions to implement the single CISC ADD operation ADD (r1), (r2), (r3)

Difference with RISC

- RISC architecture may require many more operations to implement a function than CISC architecture, although ADD (r1), (r2), (r3) is an extreme example

Load and store architecture

- All of the inputs of an instruction must be loaded into the register file before the instruction can execute

Reduced Instruction Set Computer (RISC) Advantage

RISC processors advantage

- Breaking a complex CISC operation into multiple RISC operations can allow the compiler to schedule the RISC operations for better performance

RISC processors advantage Example

- If memory references take multiple cycles to execute (as they generally do), a compiler for RISC architecture can place other instructions between the LD instructions in the example and the ADD
- This gives the LD instructions time to complete before the ADD, preventing the ADD from having to wait for its inputs

Contrast to the CISC instruction

- No choice but to wait for its inputs to come back from the memory system, potentially delaying other instructions

RISC convergence with most used CISC instructions

Modern RISC Processor instructions

- Incorporates some of most useful complex instructions from CISC architectures, relying on their micro-architecture to implement these instructions with little impact on the clock cycle
- For example, modern RISC organizations allow arithmetic operations to reference memory

Modern CISC Processor instructions

- Modern CISCs also do not include complex instructions that are not used sufficiently often to justify their implementation (inclusion in the instruction set) and CISC have dropped arithmetic operations to reference memory

Modern CISC Processor instructions Examples

- For example, arithmetic operations and logical operations (AND, OR) generally take one or two inputs and generate one output, and the operations read their inputs from and write their outputs to the register file
- Some CISC (complex instruction set computer) organizations allow arithmetic operations to reference memory

Present Status

- Earlier when most computer programming was done in assembly language, instruction set architecture was considered the most important part of computer architecture, because it determined how difficult it was to obtain optimal performance from the system

Present Status

- Instruction set architecture has become less significant, for several reasons. First, most programming is now done in high-level languages, so the programmer never interacts with the instruction set

Need of compatibility between different generations of a computer system

- Second, and more significant, we have come to expect *compatibility*
- We expect programs that ran on their old system to run on their new system without changes

Need of compatibility between different generations of a computer system

- The instruction set of a new processor often required to be the same as the instruction set of the company's previous processor, sometimes with a few additional instructions, meaning that most of the processor design effort for a processor goes into improving the micro-architecture to increase performance

Summary

We learnt

- RISC single cycle and fixed length instructions
- Modern RISC have few most often required CISC instructions also
- RISC have Load, Compute and store architecture
- RISC performance is higher than CISC due to simpler machine design and ease in pipelining the instructions

End of Lesson 20 on
RISC and converged Architecture