Session 01: **Introduction to Machine Architecture**

Lesson 01

General Features of RISC and CISC

Objective —

- Understanding CISC (Complex Instruction Set Computer)
- Understanding RISC (Reduced Instruction Set Computer)
- Learn the Converged Architecture

CISC Architecture

Complex Instruction Set Computer (CISC)

- Complex instruction set,
- Variable length encoding of instructions and
- Instruction execution taking a varying number of clock cycles.

Complex Instruction Set Computer (CISC)

- Large number of addressing modes for the operations and instructions
- Generally requiring fewer instructions than RISC computers to perform the computation.

Meaning of Addressing modes

- 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.

Complex Instruction Set Computer (CISC)

- Programs written for CISC architectures tend to take less space in memory.

Complex Instruction Set Computer (CISC)

- Arithmetic and other instructions may also read the inputs from or write their outputs to the memory system, instead of the register file.

Complex Instruction Set Computer (CISC) Example

- Might allow an ADD operation of the form ADD (rl), (r2), (r3)
- The parentheses around a register name indicates that the register contains the address in memory where the operand can be found or the result should be placed.

CISC Control Logic

Complex processor Example

- Complex Instruction Set Computers (CISCs) large instruction sets
- IBM 360 had ~200 opcodes.
- An 8086 processor (used in IBM PC) many addressing modes and a large instruction set.

CISC control logic

- A large fraction of the processor hardware compared to the ALUs.

CISC control logic

- Employs a control memory, which stores the bits that defines the sequences of operations (and micro-operations for the instructions in a set).

RISC Architecture

Reduced Instruction Set Computer (RISC)

- Fixed length encoding of instructions and
- 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 for arithmetic and logic operations and
- Allowing execution more instructions in the same amount of time.

Reduced Instruction Set Computer (RISC)

- Programs tend to take more space in memory
- 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)

RISC Control Logic

ARM RISC processor Example

- Used in mobile phones
- Has fewer addressing modes for arithmetic and logic instructions

RISC control logic

- The control logic in a RISC simpler than in the processor hardware compared to the ALUs.

RISC control logic

- Does not employ the control memory to stores the bits that defines the sequences of operations and does not deploy micro-operations for the instructions in a set.
- Implement all operations in each instruction by hardwired logic

Major differences between RISC and CISC architectures

Major differences between RISC and CISC architectures

- The set of instructions that can access memory.
- Choice of which *addressing modes* the architecture supports.

Clear delineation between RISC and CISC architectures

- RISC load-store architectures, meaning that only load and store instructions may access the memory system.

Clear delineation between RISC and CISC architectures

- The RISC GPR architecture provides for load-store architecture plus several other instructions

Difference between load-store architectures and other architectures

- Because RISC architectures implemented using the load-store model, an RISC processor would require several instructions to implement the single CISC ADD operation, which references memory for operands.

The hardware required to implement the CISC processor

The hardware required to implement the CISC processor

- More complex
- To be able to fetch instruction operands from memory the CISC processor would probably have a longer cycle time (or would require more cycles to execute each instruction) than the RISC processor

Example

- In our GPR load-store architecture, how many instructions required to implement the same function as the CISC ADD (r1), (r2), (r3) operation?

Example of Load-Store Architecture

- Assume that the appropriate memory addresses present in r1, r2, and r3 at the start of the instruction sequence.

Solution

- Four instructions required:
- LD r4, (r2)
- LD r5, (r3)
- ADD r6, r4, r5
- ST (r1), r6

Difference with RISC

- CISC required ADD (r1), (r2), (r3) one instruction suffice
- RISC architecture may require many more operations to implement a function than CISC architecture in an extreme example.

Difference with RISC

- All of the inputs of an instruction must be loaded into the register file before the instruction can execute in RISC.
- RISC architectures generally require more registers to implement a function than CISC architectures

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.

RISC processors advantage Example

- 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.

Recent Innovations in Execution Unit design:
1. Converged Architectures

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.

Modern RISC Processor instructions

- For example, modern RISC organizations allow arithmetic operations to reference memory

Modern CISC Processor instructions

- Do not include complex instructions that are not used sufficiently often to justify their implementation (inclusion in the instruction set)
- 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.

Modern CISC Processor instructions Examples

- 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 less significant
- First, most programming 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, expected *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

Need of compatibility between different generations of a computer system

- 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
- CISC variable length instructions in multiple cycles
- Few CISC instructions also in Modern RISC

We learnt

- RISC Load, Compute and store architecture
- CISC memory reference instructions
- RISC performance higher than CISC due to simpler machine design and ease in pipelining the instructions

End of Lesson 01 on

General Features of RISC and CISC

THANK YOU