CH3: CISC and RISC Architecture

Dr. Moaath shatnawi

Instruction Format

What is CISC?

- CISC is an acronym for Complex Instruction Set Computer and are chips that are easy to program and which make efficient use of memory. Since the earliest machines were programmed in assembly language and memory was slow and expensive, the CISC philosophy made sense

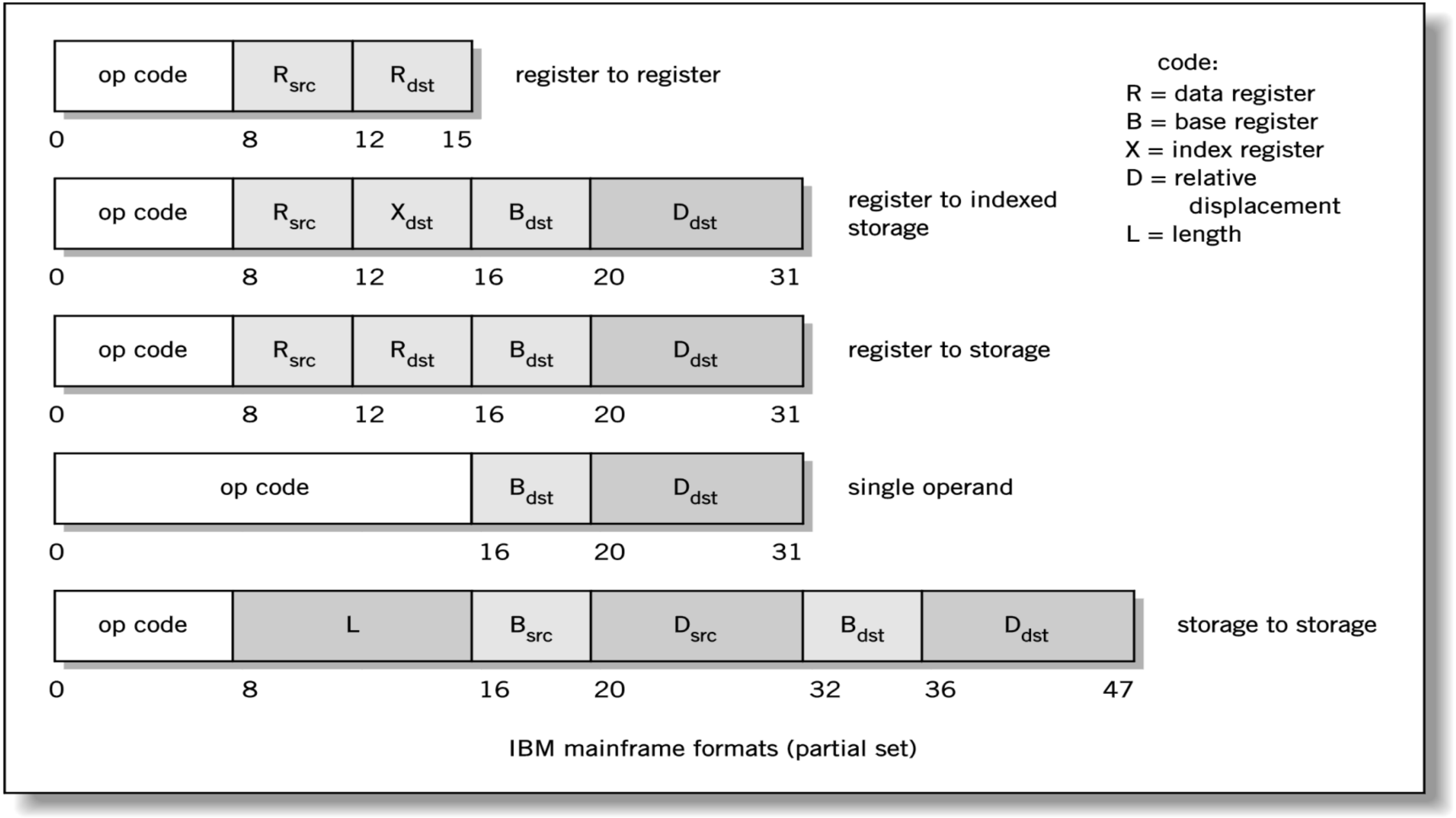
- Most common microprocessor designs such as the Intel 80x86 and Motorola 68K series followed the CISC philosophy.

- But recent changes in software and hardware technology have forced a re-examination of CISC and many modern CISC processors are hybrids, implementing many RISC principles.
- CISC was developed to make compiler development simpler. It shifts most of the burden of generating machine instructions to the processor. For example, instead of having to make a compiler write long machine instructions to calculate a square-root, a CISC processor would have a built-in ability to do this.

Most CISC hardware architectures have several characteristics in common:

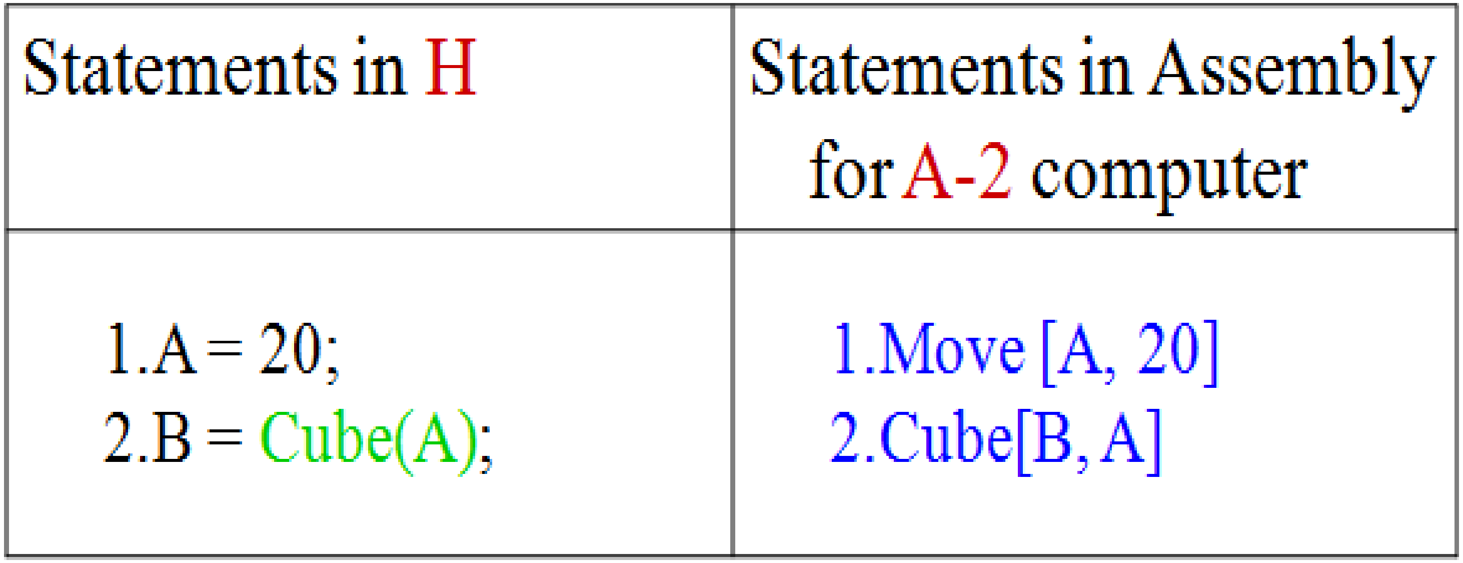
- Complex instruction-decoding logic, driven by the need for a single instruction to support multiple addressing modes.
- A small number of general purpose registers. This is the direct result of having instructions which can operate directly on memory and the limited amount of chip space not dedicated to instruction decoding, execution, and microcode storage.
- Several special purpose registers. Many CTSC designs set aside special registers for the stack pointer, interrupt handling, and so on. This can simplify the hardware design somewhat, at the expense of making the instruction set more complex.
- A 'Condition code" register which is set as a side-effect of most instructions. This register reflects whether the result of the last operation is less than, equal to, or greater than zero and records if certain error conditions occur.

Instruction Formats: CISC



IBM mainframe formats (partial set)

Instruction Formats: CISC example1



Statements in H	Statements in Assembly for A-2 computer

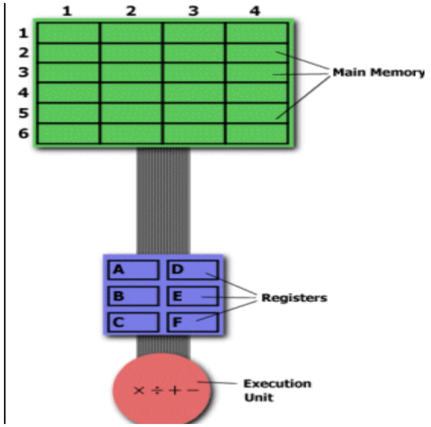
1.A = 20;
2.B = Cube(A);
	1.Move [A, 20]
	2.Cube[B, A]

Instruction Formats: CISC example2

- **The CISC Approach**
  The primary goal of CISC architecture is to complete a task in as few lines of assembly as possible. This is achieved by building processor hardware that is capable of understanding and executing a series of operations. For this particular task, a CISC processor would come prepared with a specific instruction (we'll call it "MULT"). When executed, this instruction loads the two values into separate registers, multiplies the operands in the execution unit, and then stores the product in the appropriate register. Thus, the entire task of multiplying two numbers can be completed with one instruction:

MULT 2:3, 5:2

MULT is what is known as a "complex instruction."



What is RISC?

- **RISC?**
  RISC, or *Reduced Instruction Set Computer*. is a type of microprocessor architecture that utilizes a small, highly-optimized set of instructions, rather than a more specialized set of instructions often found in other types of architectures.
- **History**
  The first RISC projects came from IBM, Stanford, and UC-Berkeley in the late 70s and early 80s. The IBM 801, Stanford MIPS, and Berkeley RISC 1 and 2 were all designed with a similar philosophy which has become known as RISC. Certain design features have been characteristic of most RISC processors:
  - *one cycle execution time*: RISC processors have a CPI (clock per instruction) of one cycle. This is due to the optimization of each instruction on the CPU and a technique called PIPELINING
  - *pipelining*: a techique that allows for simultaneous execution of parts, or stages, of instructions to more efficiently process instructions;
  - *large number of registers*: the RISC design philosophy generally incorporates a larger number of registers to prevent in large amounts of interactions with memory

RISC Attributes

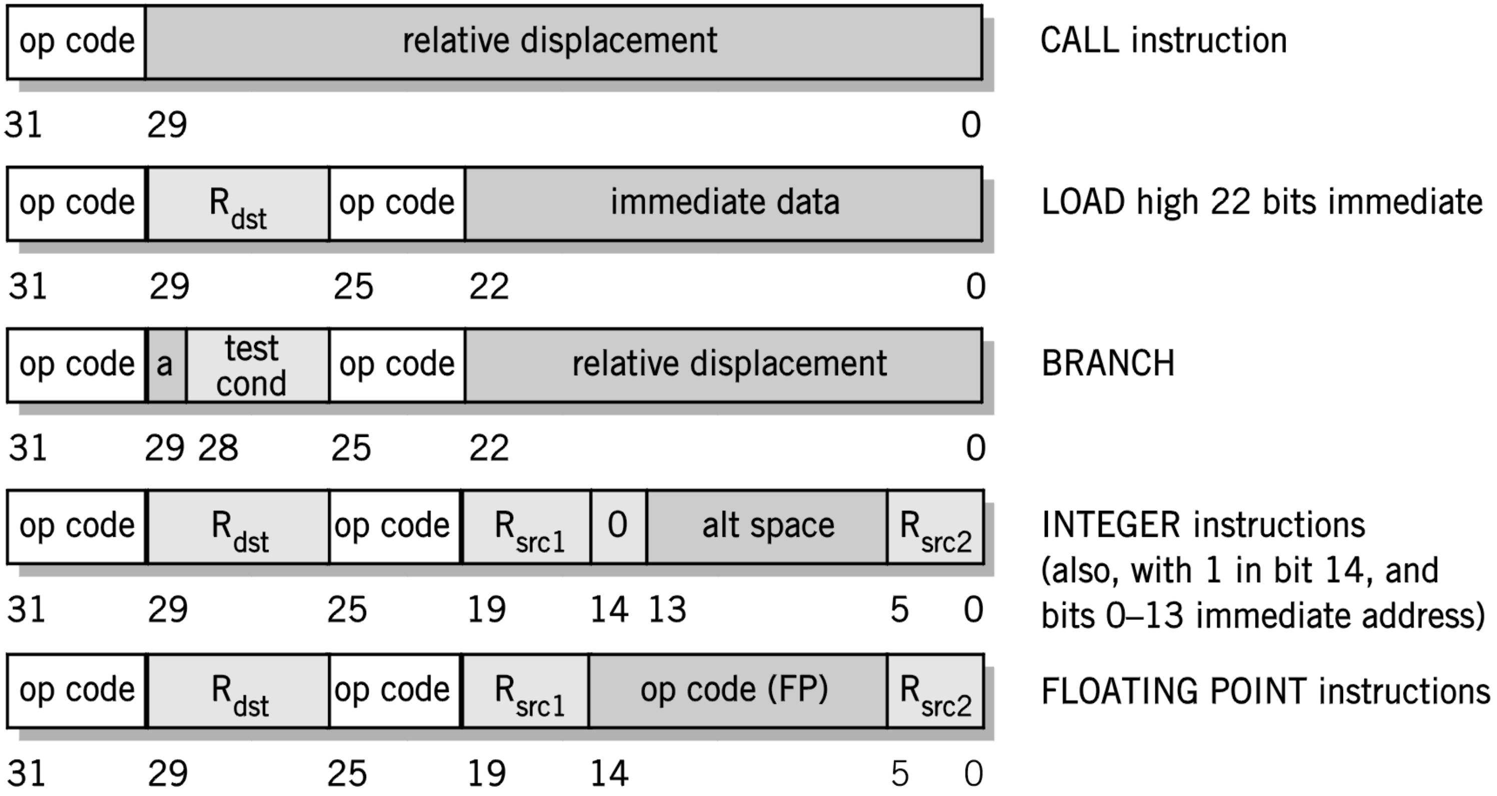
The main characteristics of CISC microprocessors are:

- Extensive instructions.
- Complex and efficient machine instructions.
- Microencoding of the machine instructions.
- **Extensive addressing capabilities** for memory operations.
- Relatively few registers.

In comparison, RISC processors are more or less the opposite of the above:

- Reduced instruction set.
- Less complex, simple instructions.
- Hardwired control unit and machine instructions.
- **Few addressing schemes** for memory operands with only two basic instructions, LOAD and STORE.
- Many symmetric registers which are organized into a register file.

Instruction Formats: RISC



op code | relative displacement — CALL instruction
31 29 0

op code | R_{dst} | op code | immediate data — LOAD high 22 bits immediate
31 29 25 22 0

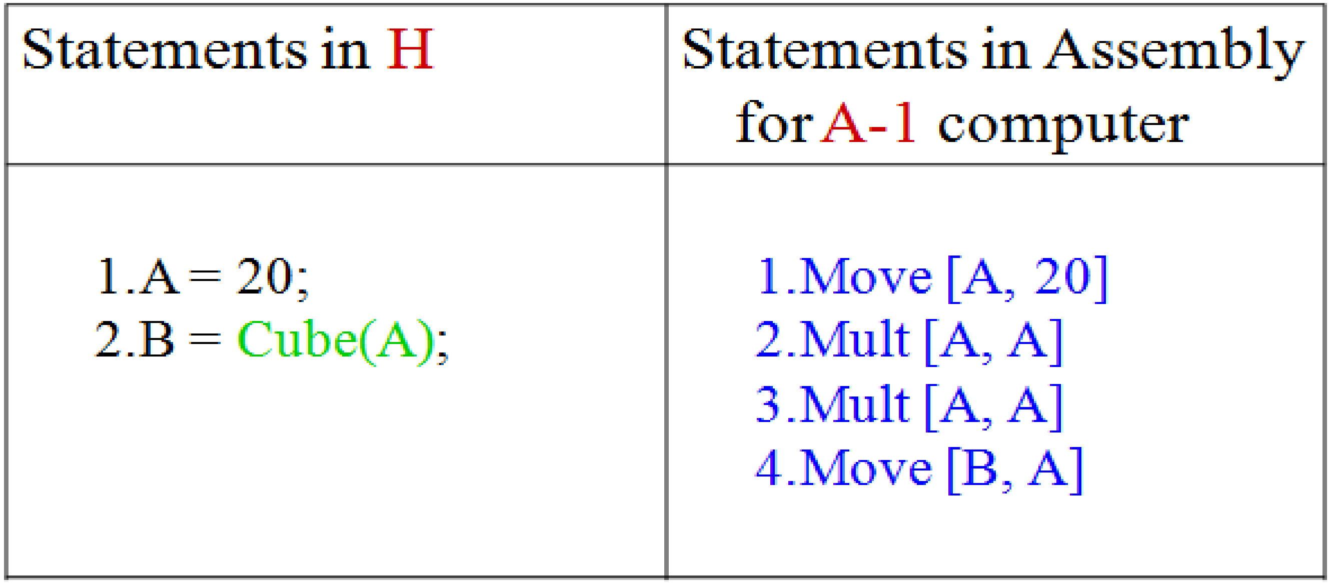
op code | a | test cond | op code | relative displacement — BRANCH
31 29 28 25 22 0

op code | R_{dst} | op code | R_{src1} | 0 | alt space | R_{src2} — INTEGER instructions (also, with 1 in bit 14, and bits 0–13 immediate address)
31 29 25 19 14 13 5 0

op code | R_{dst} | op code | R_{src1} | op code (FP) | R_{src2} — FLOATING POINT instructions
31 29 25 19 14 5 0

SPARC RISC formats (complete set)

Instruction Formats: RISC example1



Statements in H	Statements in Assembly for A-1 computer

1.A = 20;
2.B = Cube(A);
	1.Move [A, 20]
	2.Mult [A, A]
	3.Mult [A, A]
	4.Move [B, A]

Instruction Formats: RISC example2

RISC processors only use simple instructions that can be executed within one clock cycle. Thus, the "MULT" command described above could be divided into three separate commands: "LOAD," which moves data from the memory bank to a register, "PROD," which finds the product of two operands located within the registers, and "STORE," which moves data from a register to the memory banks. In order to perform the exact series of steps described in the CISC approach, a programmer would need to code four lines of assembly:

```
LOAD A, 2:3
LOAD B, 5:2
PROD A, B
STORE 2:3, A
```

CISC versus RISC

CISC	**RISC**
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	Register to register: "LOAD" and "STORE" are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers

Chip Types

- Complex Instruction Set Computer (CISC)
  - Many forms of instructions (some special purpose, but chip must support all of them)
  - x86, Pentium
- Reduced Instruction Set Computer (RISC)
  - Specific list of supported instructions (Want to do something else? Find a combination of instructions to accomplish the task)
  - Power PC (Motorola), Alpha, IBM RISC System/6000, Sun SPARC, MIPS
- Difference is speed of execution: RISC is faster, but may require longer instruction combinations to accomplish the same task as a CISC

CISC Architecture

Characteristics

□ Few general purpose registers

□ Many addressing modes

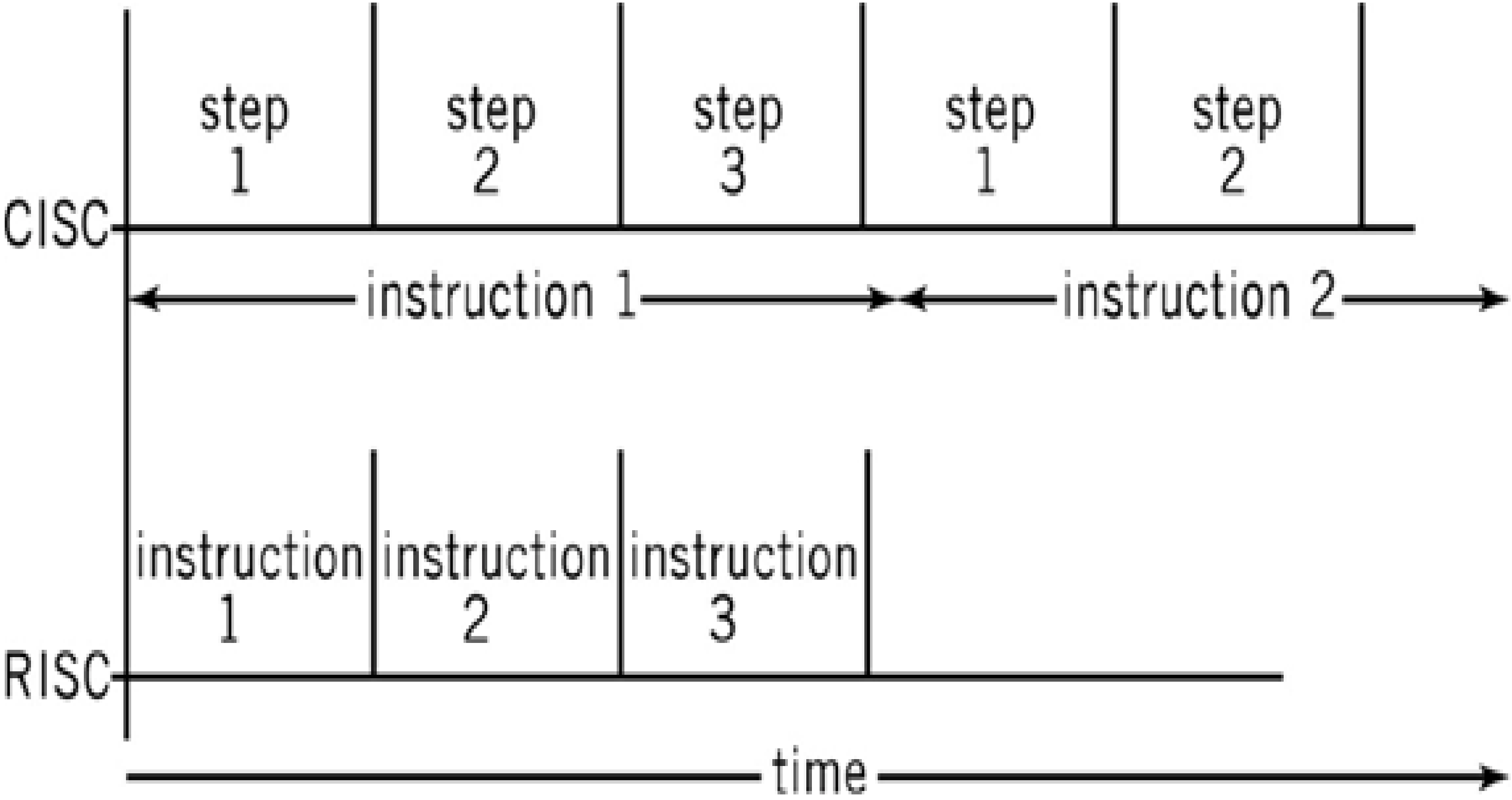
□ Large number of specialized, complex instructions

□ Instructions are of varying sizes

RISC Features

- Limited and simple instruction set
- Fixed length, fixed format instruction words
  - Enable pipelining, parallel fetches and executions
- Limited addressing modes
  - Reduce complicated hardware
- Register-oriented instruction set
  - Reduce memory accesses
- Large bank of registers
  - Reduce memory accesses
  - Efficient procedure calls

CISC vs. RISC Processing


step 1
step 2
step 3
step 1
step 2
CISC
instruction 1
instruction 2
instruction 1
instruction 2
instruction 3
RISC
time