
4.10 INTRODUCTION TO RISC AND CISC

CISC stands for Complex Instruction Set Computer. In Complex Instruction Set Computer, a single instruction can execute several low level operations (such as a load from memory, an arithmetic operation, and a memory store) and/ or capable of multi-step operations or addressing modes within single instructions.

The design constraints that lead to the development of CISC give CISC instructions set some common characteristics:

- A 2-operand format, where instructions have a source and a destination i.e., register to register, registers to memory, and memory to register commands. It also provides multiple addressing modes for memory, including specialized modes for indexing through arrays.
- Variable length instructions where the length often varies according to the addressing modes.
- Instructions which need multiple clock cycles to execute.
- Complex instruction decoding logic, driven by the need of a single instruction to support multiple addressing modes.
- A small number of general purpose registers and several special purpose registers.
- 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 condition occurs.

Let us have an example of a specific instruction called MULT. When executed this instruction, it loads the two values (or operand) into separate registers and multiplies the operand in the execution unit, and then stores the product in the appropriate register. Thus, the entire task of multiplying two numbers completed with one instruction: **MULT 2:3 5:2**

So, MULT is known as complex instruction. It operates directly on the computer's memory banks and does require the programmer to explicitly call any loading or storing functions. It closely resembled a command in a higher level language. For instance, if we let “a” represent the value of 2:3 and “b” represents the value of 5:2, then this command is identical to the C statement “a = a * b”.

The primary advantages of this approach is are-

- Compiler has to do very little work to translate a high-level language statement into assembly.

- As so many low level instructions are embedded into a single instruction, this approach needs little memory for storing the instructions.

RISC stands for Reduced Instruction Set Computer. RISC architecture based computer only use simple instructions that can be executed within one clock cycle. Thus the "MULT" command described above could be divided into three separate command-

1. LOAD- means moves data from memory bank to a register.
2. PROD- means finds the product of the two operands located within the registers.
3. STORE- means moves data from a register to the memory bank.

In order to make the exact series of the steps mentioned above, the programmer needs to code four lines of assembly code as follows-

LOAD A, 2:3

LOAD B, 5:2

PROD A, B

STORE 2:3, A

The following are the differences between CISC and RISC Architecture:

1. CISC architecture emphasis on hardware, but RISC architecture emphasis on software.
2. CISC architecture includes multi-clock complex instructions, but RISC architecture uses single clock reduced instruction only.
3. CISC architecture uses variable length instructions, but RISC architecture uses fixed length instructions.
4. CISC architecture uses many addressing modes, but RISC architecture few addressing modes.
5. In CISC architecture, complexity is in compiler, but in RISC architecture, complexity in micro-code.
6. In CISC architecture, many instructions can access memory, but in RISC architecture only LOAD/ STORE instructions can access memory.

As we know that following performance equation is commonly used for expressing a computer's performance ability-

Time per program execution is-

$$(\text{Time/ Cycle}) \times (\text{Cycles/ Instruction}) \times (\text{Instructions/ Program})$$

The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instructions. But RISC approach minimizes number of instructions per program, by sacrificing the number of cycles per instructions.



CHECK YOUR PROGRESS - 4

5. Fill in the blanks.

- (i) A _____ is an elementary CPU operation, performed during one clock pulse
- (ii) In hardwired scheme, implementing the control operation is through _____.
- (iii) RISC stands for _____ and CISC stands for _____.
- (iv) RISC approach minimizes the _____, by sacrificing the _____.

6. State whether the following statements are True or False

- (i) $\text{MAR} \leftarrow (\text{PC})$, means move the content of memory location specified by Memory Address register to Memory Buffer Register.
- (ii) $\text{MAR} \leftarrow (\text{IR (Address)})$, means move the address field of instruction to memory address register.
- (iii) In micro programmed implementation of control unit, speed of operation is high, but in hardwired implementation, speed of operation is low.
- (iv) CISC architecture emphasis on software, but RISC architecture emphasis on hardware.
- (v) RISC approach minimizes the number of number of instructions per program, by sacrificing the number of cycles per instructions.