ICT 409

29 December 2020

RISC and CISC Processors

Outline

- Microprocessor
- Characteristics of microprocessor
- Classification of microprocessor
- RISC and CISC Architecture
- Example
- Difference

Microprocessor

A microprocessor is a **processing unit** on a *single chip*. It is an **integrated circuit** which performs the core functions of a computer CPU. It is a multipurpose programmable **silicon chip** constructed using Metal Oxide Semiconductor (MOS) technology which is clock driven and register based. It accepts binary data as input and provides output after processing it as per the specification of instructions stored in the memory. These microprocessors are **capable of processing 128 bits at a time** at the speed of one billion instructions per second.

Characterstics of microprocessor

Instruction Set: **Set of complete instructions** that the microprocessor executes is termed as the instruction set.

Word Length: The **number of bits** processed in a single instruction is called word length or word size. **Greater the word size**, larger the processing power of the CPU.

System Clock Speed: Clock speed determines **how fast a single instruction** can be executed in a processor. The **microprocessor’s pace** is controlled by the System Clock. Clock speeds are generally measured in million of cycles per second (**MHz**) and thousand million of cycles per second (**GHz**). Clock speed is considered to be a very important aspect of **predicting the performance of a processor**.

Classification of Microprocessors

Besides the classification based on the word length, the classification is also based on the **architecture i.e. instruction Set of the microprocessor**. These are categorized into RISC and CISC.

RISC: It stands for **Reduced Instruction Set Computer**. It is a type of microprocessor architecture that uses **a small set of instructions** of uniform length. These are **simple instructions** which are generally **executed in one clock cycle**. RISC chips are relatively **simple to design and inexpensive**. The setback of this design is that the computer has to **repeatedly perform simple operations to execute a larger program** having a large number of **processing operations**. **Examples:** SPARC, POWER PC etc.

Classification of Microprocessors

CISC: It stands for **Complex Instruction Set Computer**. These processors offer the users, **hundreds of instructions of variable sizes**. CISC architecture includes a **complete set of special purpose circuits** that carry out these instructions at a very high speed. These instructions interact with memory by using **complex addressing modes**. CISC processors **reduce the program size** and hence **lesser number of memory cycles** are required to execute the programs. This increases the overall speed of execution.

Examples: Intel architecture, AMD

RISC Characteristics

- **Simpler instruction**, hence simple instruction **decoding**.
- Instruction come under size of **one word**.
- Instruction take **single clock cycle** to get executed.
- More number of **general purpose register**.
- Simple **Addressing Modes**.
- Less **Data types**.
- **Pipelining** can be achieved.

CISC Characteristics

- **Complex instruction**, hence complex instruction **decoding**.
- Instructions are **larger than one word** size.
- Instructions may take **more than single clock cycle** to get executed.
- **Less number of general purpose register** as operation get performed in memory itself.
- **Complex Addressing Modes**.
- **More Data types**.

Example

Example – Suppose we have to add two 8-bit number:

CISC approach: There will be a single command or instruction for this like ADD which will perform the task.

RISC approach: Here programmer will write first load command to load data in registers then it will use suitable operator and then it will store result in desired location.

So, add operation is divided into parts, i.e., load, operate, store due to which RISC programs are longer and require more memory to get stored but require less transistors due to less complex command.

Difference

RISC	CISC
Very **fewer instructions** are present. The number of instructions are generally **less than 100**.	A **large number** of instructions are present in the architecture.
No instruction with a long execution time due to very **simple instruction set**. Some early RISC machines did not even have an integer multiply instruction, requiring compilers to implement **multiplication as a sequence of additions**.	Some instructions with **long execution times**. These include instructions that **copy an entire block** from one part of memory to another and others that **copy multiple registers** to and from memory.

Difference

RISC	CISC
Fixed-length encodings of the instructions are used. Example: In IA32, generally all instructions are encoded as **4 bytes**.	**Variable-length encodings** of the instructions. Example: IA32 instruction size can range from **1 to 15 bytes**.
Simple addressing formats are supported. Only base and displacement addressing is allowed.	**Multiple formats** are supported for specifying operands. A memory operand specifier can have many different combinations of displacement, base and index registers.

Difference

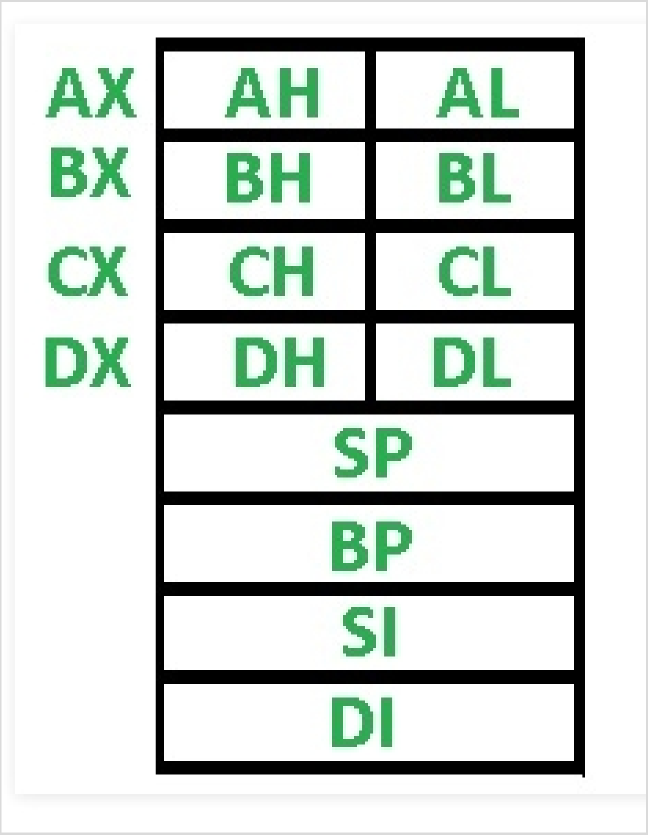
RISC	CISC
RISC does **not support array**.	CISC **supports** array.
Arithmetic and logical operations only use register operands. Memory referencing is only allowed by **load and store instructions**, i.e. reading from memory into a register and writing from a register to memory respectively.	Arithmetic and logical operations can be applied to both **memory and register operands**.

Difference

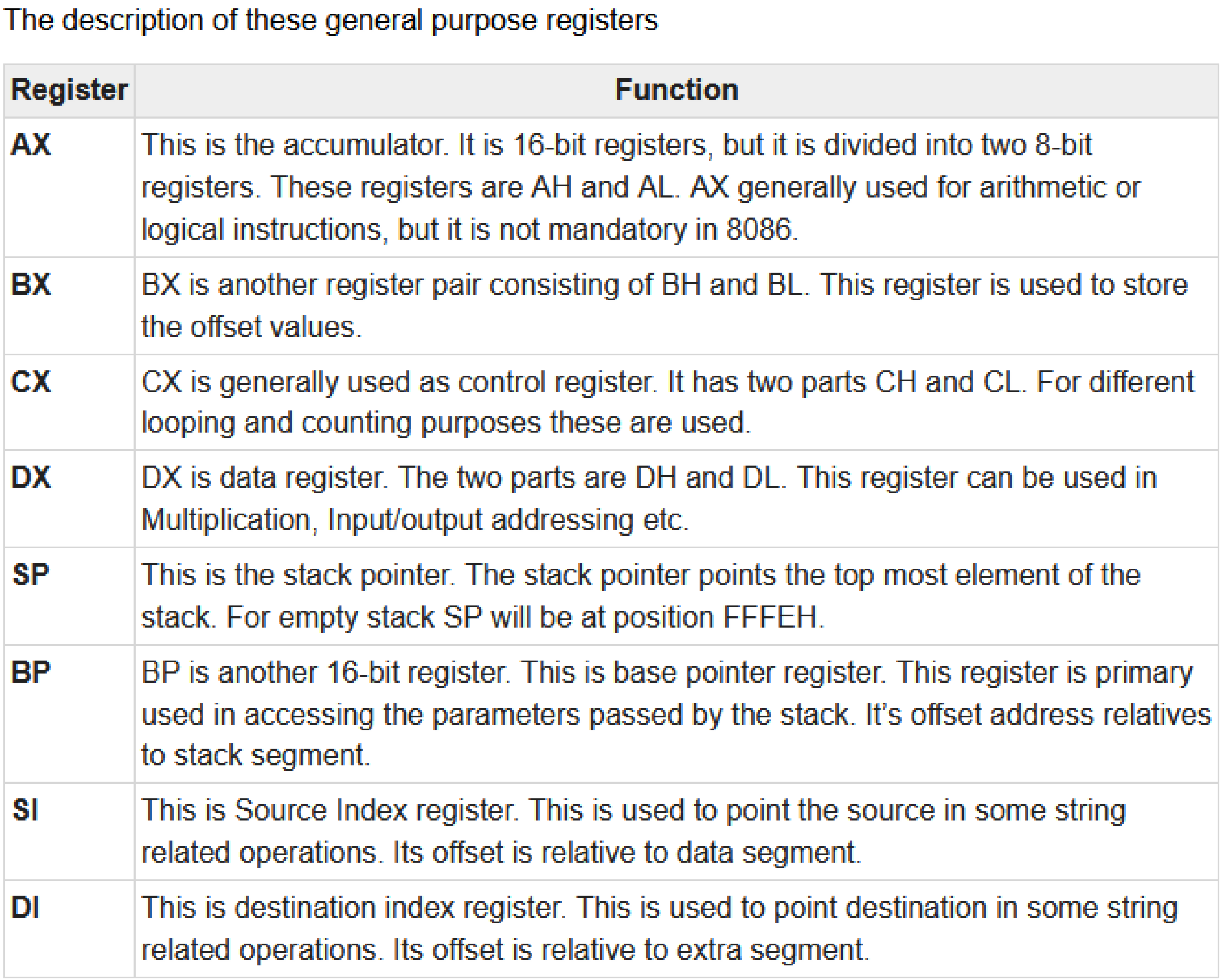
RISC	CISC
Focus on software	Focus on hardware
Transistors are used for more registers	Transistors are used for storing complex Instructions
Code size is large	Code size is small
An instruction is executed in single clock cycle.	Instructions take more than one clock cycle.
An instruction is fit in one word.	Instructions are larger than size of one word.

General purpose registers

The general purpose registers are used to store temporary data in the time of different operations in microprocessor. 8086 has eight general purpose registers.



General purpose registers



The description of these general purpose registers

Register	Function
AX	This is the accumulator. It is 16-bit registers, but it is divided into two 8-bit registers. These registers are AH and AL. AX generally used for arithmetic or logical instructions, but it is not mandatory in 8086.
BX	BX is another register pair consisting of BH and BL. This register is used to store the offset values.
CX	CX is generally used as control register. It has two parts CH and CL. For different looping and counting purposes these are used.
DX	DX is data register. The two parts are DH and DL. This register can be used in Multiplication, Input/output addressing etc.
SP	This is the stack pointer. The stack pointer points the top most element of the stack. For empty stack SP will be at position FFFEH.
BP	BP is another 16-bit register. This is base pointer register. This register is primary used in accessing the parameters passed by the stack. It's offset address relatives to stack segment.
SI	This is Source Index register. This is used to point the source in some string related operations. Its offset is relative to data segment.
DI	This is destination index register. This is used to point destination in some string related operations. Its offset is relative to extra segment.