

AN OVERVIEW OF MICROPROCESSORS AND ASSEMBLY LANGUAGE PROGRAMMING

¹Md. Abdullah Al Zaman, ²Nusrath Jahan Monira

^{1,2}Department of Physics, University of Chittagong, Chittagong-4331, Bangladesh

ABSTRACT

Microprocessor is a very useful tool for our modern communication. In fact the performance of any computer is vastly dependent on them. In this paper we have focused on the evolution of the microprocessors first, and then went for the categorization, organization, operation and some other fundamental things. Discussed about the several cycles that a microprocessor goes through and at last gave some ideas and aspects of assembly language programming.

1. INTRODUCTION

The term ‘micro’ means extremely small and ‘processor’ means the thing that accelerates tasks. So in general sense the term ‘microprocessor’ means an extremely small thing that can accelerate different tasks as ordered. But the actual definition of microprocessor is slightly different than this.

A microprocessor is a tiny electronic chip containing transistors found inside a computer’s central processing unit and other electronic devices. Its basic function is to take input, process it and then provide appropriate output.

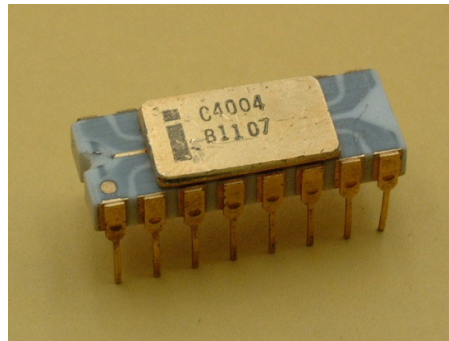


Fig-a: Intel-4004 (year-1971)

The most important technological invention of recent times is the microprocessor. The developments in the technology of the integrated circuits made it possible for the engineers of the “Intel” corporation of America to develop a micro-programmable computer on a chip in 1971. This device (“Intel 4004”) consist of about 2300 transistors on a chip which was fabricated using silicon-gate P-channel MOS technology. It was later named as microprocessor. In figure-a, we have shown a sample of a microprocessor (“Intel 4004”). It was a 4-bit microprocessor.

Microprocessors are usually made of silicon and often called ‘Logic chips’ or only ‘Processors’. The first generation microprocessor family was of 8-bit. But today we use both 32-bit and 64-bit microprocessors. In the last 44 years the number of transistors used in microprocessor has hugely increased. Now we have microprocessors containing more than six millions of transistors. The evolution of microprocessors with time is shown in figure-b.

On the surface, a microprocessor’s job may seem like an easy task, but modern microprocessors perform trillions of instructions per second. No technology is more incredible than the microprocessor. Every second, trillions of switch openings and closings occurs all within a thousandth of an inch below the surface.

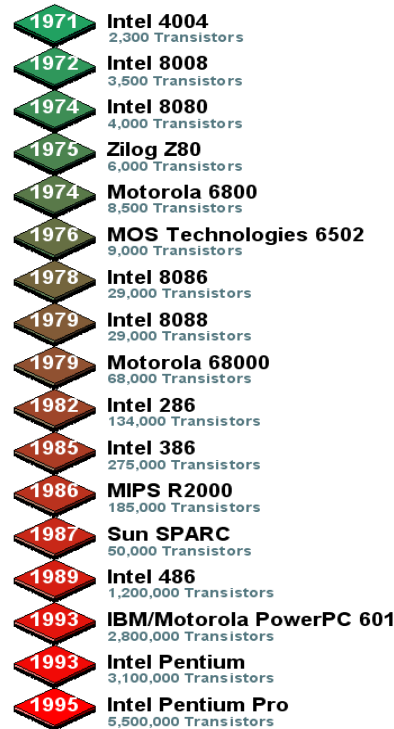


Fig-b: Evolution of microprocessors

From the time a computer is turned onto the time it is shut down, a microprocessor will have performed millions of logic and arithmetic operations. These operations utilize tiny number of holding spaces called register. Typical arithmetic operations include adding, subtracting and comparing two or more values. To perform the operations a microprocessor has to receive specific instructions as part of design. When a computer is started, the microprocessor receives its first set of instructions from the basic input-output system. A microprocessor’s speed is measured in Mega-Hertz (M-Hz).

2. CLASSIFICATION

The very first microprocessor was of 4-bit. Then we got 8-bit, 16-bit, 32-bit, 64-bit microprocessors gradually with time and with the development of modern technology. We can classify it in many ways but the most commonly and popularly, microprocessors are classified into two categories, and they are:-

According to working capability:

- Dedicated microprocessors or Microcontroller
- General purpose microprocessors
- Bit-slice processors

According to characteristics:

- RISC processors
- CISC processors
- Special processors

2.1. According to working capability

2.1.1. Dedicated microprocessors or Microcontroller

A dedicated microprocessor or microcontroller is a computer present in a single integrated circuit which is dedicated to perform one task or one type of tasks. They are used in situations where limited computing functions are required. It has a dedicated input device and often has a display for output. It contains memory, programmable input/output peripherals as well as a processor.

Microcontrollers are mostly designed for embedded applications and are heavily used in automatically controlled electronic devices such as cell-phones, cameras, microwave ovens, washing machines etc. They are usually 4-bit and designed for low power consumption. This type of microprocessors is also known as Application specific integrated circuits (ASICs).

2.1.2. General purpose microprocessors (GPM)

This type of microprocessors is designed for various uses. They have a wide range of applications, not restricted to one function. They can be applied to all or most number of members of a category or group of application. The microprocessors of computers and Android cell-phones are the most common and popular example of it. They can run most of the applications under certain conditions and with the help of additional programming or applications.

2.1.2. Bit-slice processors

Some processors have limited working capacity. They are not useful in performing many difficult tasks, but they can be useful in making big processors by adding them one by one. That work is very expensive and time consuming. We hardly use this type of microprocessor.

We will discuss about the other type of microprocessor in the later-part of the article. Before going to that we need to understand the organization or internal architecture of the microprocessor. Here we will learn some important terms that will help to understand the other type of microprocessor more easily.

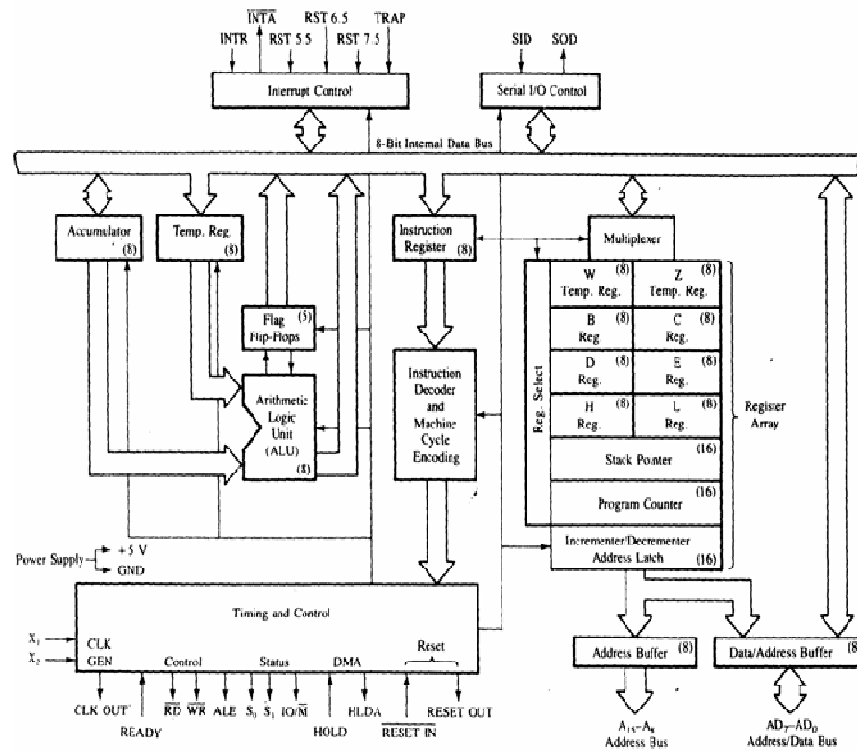
3. ORGANIZATION OF A MICROPROCESSOR

To explain the architecture or the organization of a microprocessor, we can take help from the intel-8085 microprocessor shown in figure c.



Fig-c: Intel-8085 microprocessor

It was an 8-bit microprocessor built in 1976. The internal architecture is shown above with the help of a functional diagram in fig-d.



(Fig -d: Block diagram of Intel 8085 microprocessor.Courtesy: Wikipedia)

So we have mainly the parts:-

- Arithmetic/logic unit
- Control unit
- Registers
- Bus unit

3.1 Arithmetic/ logic unit (ALU)

The ALU performs the actual computations or processing of data. ALU is the computer's calculator.

The functions of ALU are:-

- Arithmetic operations (addition, subtraction, multiplication and division)
- Logical operations (OR, AND, NOT etc)
- Decision making (ability of comparing two or more numbers or data).

3.2 Control unit

The control unit controls the movement of data and instructions into and out of the CPU. It also controls operation of the ALU. The control unit does not input, output, process, or store data; rather, it initiates and controls these operations. The control unit also communicates with input devices to begin the transfer of data or instructions into memory and with output devices to begin the transfer of results from memory.

3.3 Registers

The registers are special memory locations for storing temporary information much as a scratch pad does. Every register is made of more than one flip-flop. The ALU uses the data stored in the registers. Different types of registers are used. Among them 'accumulator register' and 'memory address register' or just 'address register' 'instruction register' and buffer register are mostly important for understanding the instruction execution. Accumulator register stores the data until an instruction is completed. The result of an arithmetic operation is also stored by it and it happens every time when an arithmetic operation occurs. The memory address register contains the address where the data is coming from and from which memory. The memory address register have the binary address where a data is stored somewhere in the memory. Instruction register gives the required instruction and the buffer register that stores the data being transferred to and from the immediate access store.

3.4 Bus unit

Buses carry digital information throughout the chip and computer; and local memory supports on-chip computation. There are so many bus connections like address bus, control bus, data bus memory bus etc.

More complex microprocessors often contain other sections—such as sections of specialized memory, called cache memory, to speed up access to external data-storage devices. We can also explain the organization of a microprocessor with the diagram given above (fig-e)

Microprocessor can't work automatically. It requires some data called instructions to perform a definite task. So while talking about microprocessors, we need to know their instructions and how they execute them.

4. INSTRUCTION EXECUTION

Instruction is the information or the way with which the required task can be performed. The instructions for the microprocessor in the storage must be in machine readable form. These instructions consist of two parts. The first one is 'an operation code' and 'one or more operand'. This is usually known as instruction format. We will discuss about it later.

Instruction execution is one of the major functions of the microprocessor. Instructions are executed one after one. Instruction execution is usually performed by two cycles and they are;

- Fetch cycle,
- Execute cycle.

4.1 Fetch cycle

At the beginning of the instruction execution, the central processing unit (CPU) fetches some data and instruction (program) from its main memory and stores them in its own internal temporary memory areas called 'registers'. The process of fetching data and instruction is possible through a path called bus. After that the CPU judges or makes sense of the instruction it has just fetched. This process is called 'Decode'. CPU decodes all instructions and prepares the areas within the microprocessor for the next step. The whole process or cycle is known as fetch cycle. [In many sources we got the 'decode' process as an individual process.]

4.2 Execute cycle

This step doesn't have many tasks to perform. In this step the decoded instruction is performed and the result is made. Then the result is preserved for further uses in the register. So the cycle which actually executes the given instructions is called execute cycle.

There is another kind cycle for explaining instruction execution, which is nothing but the combination of the fetch cycle and execute cycle. It is called "Instruction cycle". It is also known as 'fetch-execute cycle'

5. INSTRUCTION CYCLE

It is the cycle in which a single instruction is fetched, decoded and executed. The first half of the cycle transfers the instructions from the memory to the instruction register (accumulator) and decodes it. In the second half executes the instruction. The instruction cycle can be showed like the diagram (fig-f) below.

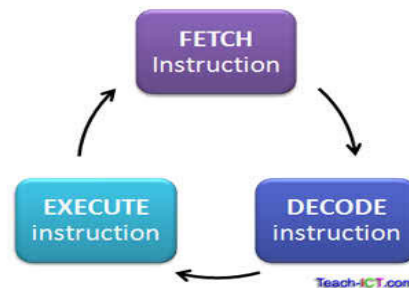


Figure-f: Diagram for instruction cycle

Or the operations are obtained for instruction cycle:

- Read an Instruction
- Decode the instruction
- Find the address of operand
- Retrieve an operand
- Perform desired operation
- Find the address of destination
- Store the result into the destination.

6. CLOCK CYCLE

The speed of a computer microprocessor, or CPU, is determined by the clock cycle, which is the amount of time between two pulses of an oscillator. Generally speaking, the higher number of

pulses per second, the faster the computer processor will be able to process information. The clock speed is measured in Hz, typically either megahertz (MHz) or gigahertz (GHz). Computer microprocessors can execute one or more instructions per clock cycle, depending on the type of microprocessor. Early computer microprocessors and slower microprocessors can only execute one instruction per clock cycle, but faster, more advanced microprocessors can execute multiple instructions per clock cycle, processing data more efficiently.

7. INSTRUCTION OF THE MICROPROCESSOR

There is some definite things required for the execution of an instruction like the addressing modes, instruction set, operation codes etc.

7.1 Addressing modes

The term addressing modes refers to the way in which the operand of an instruction is specified. Information contained in the instruction code is the value of the operand or the address of the result/operand.

7.2 Instruction format

The set of simple tasks that the processor can perform are called the instruction set. The instruction set is usually composed of two parts: the first part is a mnemonic called the OPCODE, which tells what the microprocessor is supposed to do. The second part, which may be composed of one or two words, contains either data, or addresses where data manipulation is to take place. The actual values of these words depend upon the OPCODE. Usually, the data which each OPCODE operates on is called operand(s). The entire instruction set and its syntactic rules form a grammar that the CPU can understand and act upon. It is called assembly language.

7.3 Mnemonic

The word MNEMONIC means "A device such as a pattern of letters, ideas, or associations that assists in remembering something". So, it is usually used by assembly language programmers to remember the "OPERATIONS" a machine can do, like "ADD" and "MUL" and "MOV" etc.

7.4 Op code (operation code)

The OPCODE is part of an instruction word that is interpreted by the processor as representing the operation to perform, such as read, write, jump, add etc. Many instructions will also have OPERANDS that affect how the instruction performs, such as saying from where in memory to read or write, or where to jump.

Now we can talk about the remaining type of microprocessors that are based on their characteristics.

8. MICRO PROCESSORS ACCORDING TO CHARACTERISTICS

- RISC processors
- CISC processors
- Special processors.

8.1 RISC processors

RISC stands for '**Reduced instruction set computer**'. This microprocessor was designed to perform a smaller number of types of computer instructions so that it can operate at a higher speed. Characteristics of RISC processors are given below.

Relatively few instructions and addressing modes (what is meant by instructions and addressing mode, is explained later in this article)

- Memory access limited to load and store instructions.
- All operations done within the registers of CPU
- Fixed length, easily decoded instruction format
- Single cycle instruction execution i.e. required only one cycle to execute a definite instruction.
- The microprocessor is designed using hardwired control.
- It has several general purpose registers

Some popular RISC processors are:

- PA-RISC: HP700LC
- POWER PC- 601,604,615,620

8.2 CISC processors

CISC Stands for "**Complex Instruction Set Computing**." This is a type of microprocessor design. The CISC architecture contains a large set of computer instructions that range from very simple to very complex and specialized. The design was intended to compute complex instructions in the most efficient way.

Some characteristics of CISC processors are given below

- Large number of addressing modes and instructions
- Some instructions perform special task
- Variable length instruction format
- Instructions that manipulate control
- Several cycles may be required to execute one instruction.
- The microprocessor is designed using code control.
- It has small number of general purpose registers

Intel's processors of x86 Families are the best example of CISC processors.

We can see that the RISC and CISC processors can easily be differentiated by their own characteristics. Besides pipelining and superscalar architectures are the base methods to design RISC which is not true for CISC processors.

8.3 Special processors

There are several other processors, which are useful in special purposes. Some of these processors are briefly discussed below.

8.4 Coprocessor

Coprocessor is very similar to general purpose microprocessor. It is designed for a specific function. But coprocessor can handle its particular function many times faster than the ordinary general purpose microprocessor. The most well known coprocessor is math-coprocessor.

8.5 Input/output processor

This is one of the most important microprocessors. With this processor a large set of I/O device can be controlled, with minimum CPU involvement. The I/O processors take care of most of the tasks involved in controlling the terminals. The common examples of I/O processors are keyboard/mouse controller, graphic display controller etc.

8.6 Transputer (Transistor computer)

It is a high performance microprocessor designed to facilitate inter process and inter processor communications and targeted at the efficient exploitation of very large scale integration. The most important feature of transputer is its external links, which enables it to be used as a building block in the construction of low cost, high performance multiprocessing system.

8.7 DSP (Digital signal processor)

This processor is specially designed to handle real world analog signals that have been converted into digital representation. Modem function, speech recognition, 2D and 3D graphics acceleration, audio and video compression etc are the applications of DSP.

9. ASSEMBLY LANGUAGE PROGRAMMING

Assembly language programming is writing machine instructions in mnemonic form, using an assembler to convert these mnemonics into actual processor instructions and associated data. An assembly language is a low-level programming language for microprocessors and other programmable devices.

9.1 Features of assembly language programming

Assembly language is the most basic programming language available for any processor. Assembly languages generally lack high-level conveniences such as variables and functions. It has the same structures and set of commands as machine language, but it allows a programmer to use names instead of numbers. This language is still useful for programmers when speed is necessary or when they need to carry out an operation that is not possible in high-level languages.

Some important features of assembly language programming are given below:

- Allows the programmer to use mnemonics when writing source code programs, like 'ADD' (addition), 'SUB' (subtraction), JMP (jump) etc.
- Variables are represented by symbolic names, not as memory locations, like MOV A, here 'A' is the variable.
- Symbolic code,
- error checking,
- Changes can be quickly and easily incorporated with a re-assembly,
- Programming aids are included for relocation and expression evaluation.

9.2 Advantages of assembly language programming

- Easy to understand and use
- Easy to locate and correct errors
- Easy to modify
- No worry to address.
- Efficient than machine language programming

9.3 Disadvantages of assembly language programming

- The programmer requires knowledge of the processor architecture and instruction set.
- Machine language coding
- Many instructions are required to achieve small tasks
- Source programs tend to be large and difficult to follow

9.4 Some instructions set of Intel 8085 microprocessor according to assembly language programming

- ADD r:- add
- ADD M:- Add to Memory
- ADC r:-add with carry
- ADC M:-add with carry to memory
- LDI :- This instruction stands for load immediate and it means load' (put) into a particular named register a certain value named in the instruction
- LDA a:- load accumulator direct
- SLA: -means shift the contents of a register one place to the left.
- MOV:-means to move data from one place to desired place
- MOV M A:- means to move the result from memory to register A
- STA:- Store on register(accumulator or others)
- STA a: - store to accumulator.
- JMP- Jump to somewhere
- SUB r :- subtract
- SUB M:- subtract memory
- CY:-carry

REFERENCES

1. Nichols, A. J. (1976). An overview of microprocessor applications. *Proceedings of the IEEE*, 64(6), 951-953.
2. Shriver, B. D., & Smith, B. (1998). *The Anatomy of a High Performance Microprocessor (Interactive Book/CD-ROM): A Systems Perspective with Cdrom*. IEEE Computer Society Press.
3. Spruth, W. (2012). *The design of a microprocessor*. Springer Science & Business Media.
4. Orlando, R. V., & Anderson, T. L. (1981). An overview of the 9900 microprocessor family. *IEEE Micro*, 1(3), 38-44.
5. Imani, Mahdi, and Ulisses Braga-Neto. "Multiple model adaptive controller for partially-observed Boolean dynamical systems." *Proceedings of the 2017 American Control Conference (ACC2017)*, Seattle, WA. 2017.
6. https://www.tutorialspoint.com/microprocessor/microprocessor_overview.htm

7. <http://www.worldofcomputing.net/processor/microprocessor.html>
8. <http://www.computerhistory.org/microprocessors/>
9. <http://spectrum.ieee.org/computing/hardware/the-surprising-story-of-the-first-microprocessors>
10. https://en.wikibooks.org/wiki/Microprocessor_Design/Computer_Architecture
11. <http://beams.ulb.ac.be/courses/ma1/microprocessor-architectures>
12. <http://www.brightengineering.com/diy-electronics-devices/51225-architecture-of-8085-microprocessors-part-one/>