UNIT 1

Introduction to Microprocessor

CHAPTER OUTLINE

INTRODUCTION TO THE MICROPROCESSOR

A processor is the logic circuitry that responds to and processes the basic instructions that drive a computer. The term processor has generally replaced the term central processing unit (CPU). The processor in a personal computer or embedded in small devices is often called a microprocessor. A microprocessor is designed to perform arithmetic and logic operations that make use of small number-holding areas called registers. Typical microprocessor operations include adding, subtracting, comparing two numbers, and fetching numbers from one area to another.

1.1 MICROPROCESSORS

A microprocessor is a computer processor on a microchip. It's sometimes called a logic chip. It is the "engine" that goes into motion when you turn your computer on. A microprocessor is designed to perform arithmetic and logic operations that make use of small number-holding areas called registers. Typical microprocessor operations include adding, subtracting, comparing two numbers, and fetching numbers from one area to another. These operations are the result of a set of instructions that are part of the

microprocessor design. When the computer is turned on, the microprocessor is designed to get the first instruction from the basic input/output system (BIOS) that comes with the computer as part of its memory. After that, either the BIOS, or the operating system that BIOS loads into computer memory, or an application program is "driving" the Microprocessor, giving it instructions to perform.

"A microprocessor is the brain of today's computer. It operates your computer by carrying out complex functions. A microprocessor, sometimes called a logic chip, is a computer processor on a microchip."

"A microprocessor functions as the central processing unit of a microcomputer ."

A microprocessor is a CPU that is in just one IC (chip). For example, the CPU in a PC is in a chip so it can also be referred to as microprocessor. The GPU core (of the graphics card) is also a chip so it could also be called a microprocessor (though the GPU is less flexible than the CPU). Microprocessors would also be the Northbridge, Southbridge, or any other chips that hold a CPU of some kind.

The microprocessor is a small VLSI (very large scale integration) chip with many pins. It processes information and manage the exchange between the Input/Output units and main memory. It is controlled by a sequence of instructions called microprocessor program, and the result of this program in sent to the appropriate peripheral (input and output).

1.1.1 Characteristics of Microprocessor

Three basic characteristics differentiate microprocessors:

- **Instruction set:** The set of instructions that the microprocessor can execute.
- **Bandwidth:** The number of bits processed in a single instruction.
- **Clock speed:** Given in megahertz (MHz), the clock speed determines how many instructions per second the processor can execute.

In this cases, the higher the value, the more powerful the CPU. For example, a 32-bit microprocessor that runs at 60 MHz is more powerful than a 16-bit microprocessor that runs at 30 MHz.

In according to bandwidth and clock speed, microprocessors are classified as being either **RISC** (reduced instruction set computer) or **CISC** (complex instruction set computer).

1.1.2 Speed of Microprocessor

The speed at which a microprocessor executes instructions, also called clock rate. The faster the clock, the more instructions the CPU can execute per second. The speed of microprocessor is measured in Megahertz or MHz. A single MHz is a calculation of 1 million cycles per second (or computer instructions), so if you have a processor running at 3000 MHz, then your computer is running at 3000,000,000 cycles per second, which in more basic terms is the amount of instructions your computer can carry out. Another important abbreviation is Gigahertz or GHz. A single GHz or 1 GHz is the same as 1000 MHz, so here is a simple conversion.

1000 MHz (Megahertz) = 1 GHz (Gigahertz)

= 1000,000,000 Cycles per second

(or computer instructions).

Now that we have covered the speeds, there is one more important subject to cover. Which processor? There are 3 competitors at present, the AMD Athlon, Intel Pentium and the Intel Celeron. They come in many guises, but basically the more cores they have and the higher the speed means better and faster

However, when trying to choose a microprocessor, it is difficult to understand what the designations mean. Should you choose an x64 processor or an x86 processor? Is a dual-core fast enough, or do you need a quad-core processor? Understanding the differences in microprocessor architecture will aid in the decision-making process.

X64 vs. X86

A microprocessor may be listed for sale as an “x86 processor” or an “x64 processor”. What is the difference, though?

An “x86” processor is a microprocessor that is capable of processing information in 32-bit pieces (called “instructions”). Each "bit" is a piece of information that the computer uses to transmit information, run computations and perform other such processes. A processor using x86 architecture is considered to be a successor technology to the original microprocessors used in the IBM PC. Since the original processor used in an IBM PC was based upon the Intel 8086 microprocessor, successive microprocessors using the same set of instructions to run have been named similarly—the 80286, 80386 and 80486, for example.

A microprocessor using x64 architecture is slightly different than the x86 processor. An x64 processor is capable of processing not only 32-bit instructions, but also 64-bit instructions as well. Because of the increased capability of the x64 microprocessor, a computer that utilizes an x64 microprocessor is also capable of utilizing more memory (128 GB maximum vs. 4 GB maximum) than a computer with an x86 microprocessor.

An x64 microprocessor, therefore, would be the better choice if you plan to use the computer for memory-intensive applications, or if you need better overall performance out of your computer system.

1.1.3 Latest Processor Version

Processors now come as dual core, triple core and quad core. These processors are the equivalent of running two CPU's (Dual core), three CPU's (Triple core) or four CPU's (Quad core).

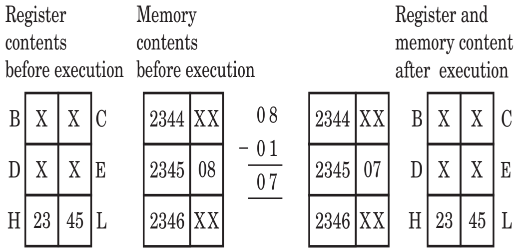
Single-core or traditional processor

In a single-core or traditional processor the central processing unit (CPU) is fed strings of instructions it must order, execute, then selectively store in its cache for quick retrieval. When data outside the cache is required, it is retrieved through the system bus from random access memory (RAM) or from storage devices. Accessing these slows down performance to the maximum speed the bus, RAM or storage device will allow, which is far slower than the speed of the CPU. The situation is compounded when multi-tasking. In this case the processor must switch back and forth between two or more sets of data streams and programs. CPU resources are depleted and performance suffers.

Dual core processor

In a dual core processor each core handles incoming data strings simultaneously to improve efficiency. Just as two heads are better than one, so are two hands.

Now when one is executing the other can be accessing the system bus or executing its own code.



Multi-core processors

Multi-core processors are the goal and as technology shrinks, there is more "real-estate" available on the die. In the fall of 2004 Bill Siu of Intel predicted that current accommodating motherboards would be here to stay until 4-core CPUs eventually force a changeover to incorporate a new memory controller that will be required for handling 4 or more cores.

Difference between dual core and multi-processor

A dual core processor is different from a multi-processor system. In the latter there are two separate CPUs with their own resources. In the former, resources are shared and the cores reside on the same chip. A multi-processor system is faster than a system with a dual core processor, while a dual core system is faster than a single-core system, all else being equal.

Dual-Core vs. Quad-Core Architecture

Many microprocessors made by Intel and AMD are multi-core processors. What this means is that within one microprocessor, there are two or more central processing units (cores) within one integrated circuit package.

As their names imply, "dual-core" multiprocessors have two CPU cores in the multiprocessor package, and "quad-core" multiprocessors have four CPU cores. Depending upon the software being used, the operating system the computer system has installed and the amount of memory available to the computer system, a quad-core system is capable of running more simultaneous processes than a dual-core system.

However, some older software or operating systems (such as Windows 95, Windows 98 or Windows Me) are not capable of utilizing multi-core microprocessors. If the software or operating system is incapable of utilizing the resources available, using a multi-core microprocessor is no different than using a single-core microprocessor.

1.1.4 Difference Between Microcontroller and Microprocessor

Microcontroller

A microcontroller is meant to be more self-contained and independent, and functions as a tiny, dedicated computer. Micro controller is a single chip. It Consists Memory, I/O ports. It used as single purpose. Microcontroller is more economical than microprocessor because by reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/ output devices.

"A microcontroller is a small computer on a single integrated circuit (IC) or chip consisting internally of a relatively simple CPU, clock, timers, I/O ports, and memory"

Where microcontroller used ?

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances power tools, and toys also used in display

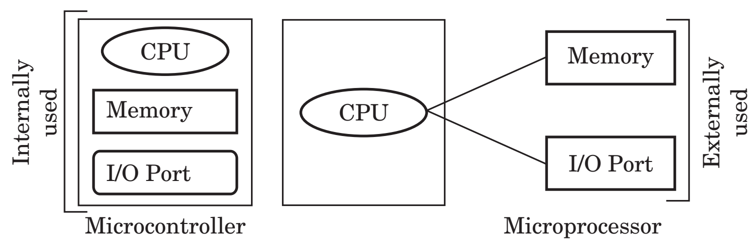
Example: Motorola's 6811, Intel's 8051

Microprocessor

But Microprocessor is a CPU. Memory, I/O Ports to be connected externally. Microprocessor is more expansive than microcontroller. It used as general purpose.

"An integrated circuit that contains the entire central processing unit of a computer on a single chip is known as microprocessor."

Example: *Intel's 8085, 8086 Motorola's 6800 etc.*



1.2 EVOLUTION OF MICROPROCESSOR

A common way of categorizing microprocessors is by the number of bits that their arithmetical logical unit (ALU) can work at a time. In other words:

"A microprocessor with a 4 bit ALU will be referred to as a 4 bit microprocessor, regardless of the number of address lines or the number of data bus lines that it has."

1.2.1 First Generation Microprocessors (1971–73)

The world's first microprocessor was the Intel 4004, produced in 1971. First microprocessor, the 4004, was co-developed by Busicom, a Japanese manufacturer of calculators, and Intel, a U.S. manufacturer of semiconductors. Intel 4004 microprocessor contained 2300 PMOS transistors. The next Microprocessor was the 8 bit 8008 microprocessor. It was developed by Intel in 1972 to perform complex functions in harmony with the 4004.

Technology

They were fabricated using PMOS technology which provided low cost, slow speed and low output currents.

1.2.2 Second Generation Microprocessors (1974–1978)

Intel 8080 is 8 bit processor, it can accept, process, or provide 8-bit data simultaneously. The Intel 8080 was widely used in control applications. Small computers were designed using the 8080 microprocessor as the CPU, this computer known as microcomputer. Intel 8080 used NMOS transistors, so Intel 8080 is operated much faster than the 8008 microprocessor. The 8080 is referred to as a second generation microprocessor.

After a few years Motorola 6800, the zilog Z80, and the Intel 8085 microprocessors were developed as improvements over the 8080. 8085 microprocessor is an 8-bit microprocessor *i.e.* it can accept, process, or provide 8-bit data Simultaneously. It operates on a single +5V power supply connected at Vcc.

Intel released the 8086 microprocessor in 1978. 8086 microprocessor is an 16-bit microprocessor *i.e.* it can accept, process, or provide 16-bit data. After a year, Intel released the 8088 microprocessor.

Technology

They were fabricated using NMOS technology. The NMOS technology offered fast speed than PMOS technology. The NMOS technology support high packing density than PMOS.

1.2.3 Third Generation Microprocessors (1979–80)

In this age, Intel introduced 16-bits microprocessors it can accept, process, or provide 16-bit data simultaneously.

Such as Intel's 8086/80186/80286 Other 16-bit microprocessors are Motorola's 68000/68010. They were designed using HMOS (High Density short channel MOS) technology. HMOS provides some advantages over NMOS as Speed-power-product of HMOS is four times better than that of NMOS. Circuit density is approximately is two times greater than that of NMOS.

Technology

They were designed using HMOS (High Density short channel MOS) technology.

1.2.4 Fourth Generation Microprocessors (1981–1995)

In this age, Intel introduced 32–bits microprocessors it can accept, process, or provide 32-bit data simultaneously. Intel released the 80386 microprocessors in 1985. Motorola also introduced 68020/68030 microprocessor. These microprocessor are also 32-bit. They were designed using HCMOS. HCMOS technology is low power version of HMOS. Motorola introduced 32-bit RISC (Reduced Instruction Set Computer) processors called MC88100.

Technology

They were fabricated using low-power version of the HMOS technology called HCMOS.

1.2.5 Fifth Generation Microprocessors (1995 – till date)

Microprocessors in their fifth generation, employed decoupled super scalar processing, and their design soon surpassed 10 million transistors. In this generation, PCs are a low-margin, high-volume-business dominated by a single microprocessor (Computer, 1996).

This age the emphasis is on introducing chips that carry on-chip functionalities and improvements in the speed of memory and I/O devices along with introduction of 64-bit microprocessors. Intel introduced with Pentium, Celeron and very recently dual and quad core processors working with up to 3.5 GHz speed.

Intel, Motorola and Zilog are most popular microprocessor brand in India. Some important microprocessors are

- 8-bit microprocessors are Intel 8085, Motorola 6809, Zilog Z 80, Zilog 800 etc.
- 16-bit microprocessors are Intel 8086, Intel 80286, Motorola 6800, Zilog 8000 etc.
- 32-bit microprocessors are Intel 80486, Pentium, Pentium Pro, Pentium II, Pentium III, Celeron, AMD K6 and K7, Power PC 601, 603, 604, 740, 750, Cyrix 686 etc.

64-bit microprocessors are DEC Alpha 21264, Power PC 620, MIPS-12000, HP PA8500 etc.

Intel Microprocessor

Processor	*Introduces year*	*Address Bus bit*	*Data Bus bit*	*Transistors*	*Clock frequency (speed)*
4004	1971	10	4	2300	108 kHz
8008	1972	14	8	3500	200 kHz
8080	1974	16	8	6000	2 MHz
8085	1976	16	8	6500	5 MHz
8086	1978	20	16	29000	5 MHz
80286	1982	16	24	134000	8 MHz
80386	1985	32	32	275000	16 MHz
80486	1989	32	32	1.2 M	25 MHz
Pentium	1993	32	32/64	3.1 M	60 MHz
Pentium Pro	1995	36	32/64	5.5 M	150 MHz
Pentium II	1997	36	64	8.8 M	233 MHz
Pentium III	1999	36	64	9.5 M	650 MHz
Pentium IV	2000	36	64	42 M	1.4 GHz

1.3 HISTORY OF COMPUTERS

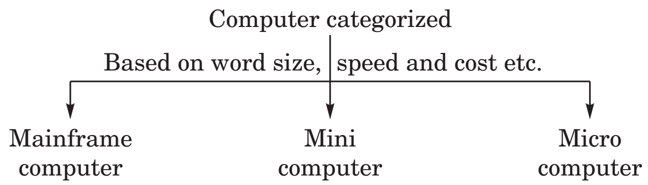
The development of the modern day computer was the result of advances in technologies and man's need to quantify. Computers were initially large machines that could fill entire rooms. Some were operated using large vacuum tubes that formed the basis of today's transistors. In order to operate such machines, punch cards were used. One of the first such examples of this was the Jacquard Loom.

The first general purpose, electronic computer system was developed in 1946. The first modern computer was called the ENIAC. ENIAC stood for Electrical Numerical Integrator and Computer. The ENIAC used over 17,000 of vacuum tubes and over 500 miles of wires. It was originally used to perform calculations for the hydrogen bomb.

The integrated circuit (IC)t led to the development of digital integrated circuit in the 1960. And the world's first microprocessor was the Intel 4004, produced in 1971.

The speed of the computer refers to the rate at which it executes the instructions. Based on these factors (word size, speed, cost etc.) the computers are categorized as

- Mainframe Computer
- Minicomputer Computer
- Micro Computer



1.3.1 Mainframe Computer

A very large and expensive computer capable of supporting hundreds, or even thousands, of users simultaneously. In the hierarchy that starts with a simple microprocessor (in watches, for example) at the bottom and moves to supercomputers at the top, mainframes are just below supercomputers. The largest and most powerful ones are often called mainframe computer.

Mainframes computer are powerful computers used mainly by large organizations for critical applications, typically bulk data processing such as census, industry and consumer statistics, and financial transaction processing.

In some ways, mainframes are more powerful than supercomputers because they support more simultaneous programs. But supercomputers can execute a single program faster than a mainframe. The distinction between small mainframes and minicomputers is vague, depending really on how the manufacturer wants to market its machines.

Mainframe computer are used for military defence control, for business data processing (for example an insurance company) and for creating computer graphics displays for science fiction movies.

1.3.2 Minicomputer

Minicomputers are scaled-down versions of mainframe computers. Minicomputers are computers that are somewhere in between a microcomputer and a mainframe computer. After some time past, the minicomputer was typically a stand alone device that was ideal for use by small and mid-sized businesses who needed more power and more memory than could be obtained with microcomputers, but less than mainframes computer. More recently, a minicomputer is thought of in terms of being a server that is part of a larger network.

A **minicomputer**, a term no longer much used, is a computer of a size intermediate between a microcomputer and a mainframe.

Examples of minicomputer type are Digital Equipment Corporation VAX 6360 and the Data General MV/8000II.

1.3.3 Microcomputer

A microcomputer is a computer with a microprocessor as its central processing unit. They are physically small compared to mainframe and minicomputers.

Micro computers have a microprocessor as their central processing unit (CPU), and are therefore relatively small in size. When they have a key board and monitor, they are used as personal computers, which is type of computer that you have in your home and office. Examples of microcomputers are Laptop computers, desktop computers.

1.4 8085 SYSTEM BUS

Bus means group of lines on which bits appear in parallel manner at a time.

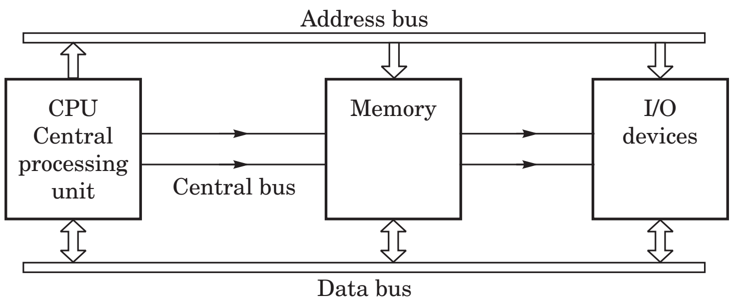
Or

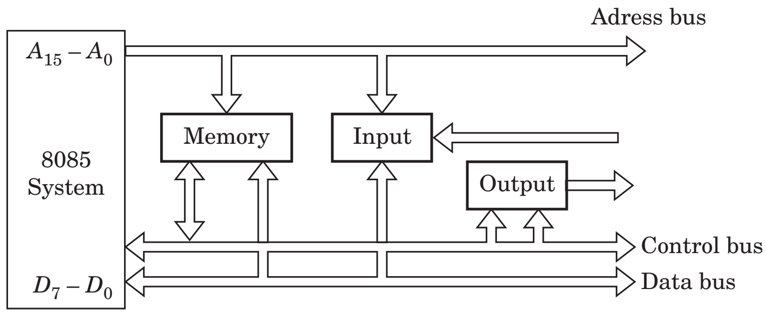
Bus means collection of wires, which transmit binary number (0 or 1), one bit per wire.

The system bus is used to transfer information between different element of microprocessor. Information like data, address and element of microprocessor like memory, I/O devices.

A typical microprocessor communicates with memory and other devices (input and output) using three busses:

1. Address Bus
2. Data Bus
3. Control Bus





8085 bus structure

1.4.1 Address Bus (Unidirectional Bus)

The *address bus* is the set of wire traces that is used to identify which address in memory the CPU is accessing) that is used to specify a physical address.

Or

An address bus is a computer bus (a series of lines connecting two or more devices) that is used to specify a physical address.

In 8085 address bus is 16-bits. One wire for each bit, therefore 16 bits = 16 wires. The Address Bus consists of 16 wires, therefore 16 bits. Its "width" is 16 bits. A 16 bit binary number allows 2^{16} different numbers, or 0000000000000000 (0000H) up to 1111111111111111 (FFFFH). Because memory consists of Memory cell, each with a unique address, the size of the address bus determines the size of memory, which can be used. To communicate with memory the microprocessor sends an address on the address bus, egg 0000000000000101 (5 in decimal), to the memory. The memory the selects memory cell number 5 for reading or writing data.

Address bus is unidirectional, *i.e.* address numbers only sent from microprocessor to memory.

"It is bus which carries the address of a memory location or I/O location. It is also called unidirectional bus. In 8085 the address bus is 16-bit (A0 — A15). The 8085 microprocessor has 16-bit address bus to access memory, hence it can access 2^{16} = 64KB memory locations."

1.4.2 Data Bus (Bi-directional Bus)

In 8085 data bus is 8-bits. The Data Bus typically consists of 8 wires one wire for each bit, therefore 8 bits=8 wires.

Data Bus: carries 'data', in binary form, between up and other external units, such as memory. Data Bus is bi-directional. Size of the data bus determines what arithmetic can be done. If only 8 bits wide then largest number is 11111111 (255 in decimal). Therefore, larger number have to be broken down into chunks of 255. Data bus used to transmit "data", *i.e.* information, results of arithmetic, etc, between memory and the microprocessor Data Bus also carries instructions from memory to the microprocessor. *Data bus* architectures use 16, 32 or 64 bit data paths.

"It is bus which used to transfer, information between microprocessor, I/0 devices and memory. It is also called bi-directional bus. The number of data lines used in the data bus is equal to the size of data word being written or read. Data bus is time multiplexed with lower order address bus."

1.4.3 Control Bus

Control bus is various lines which have specific functions for controlling microprocessor operations.

"It is bus which is necessary to control the direction of data flow on the bidirectional data bus and to differentiate between a memory address and an I/O address. Some of the control bus signals are Memory read, memory writer, I/O read and I/O writes.

Example: RD/WR line, single binary digit (1 or 0). Control whether memory is being 'written to' (data stored in memory) or 'read from' (data taken out of memory)

If RD/$\overline{\text{WR}}$ = 1 Read operation,

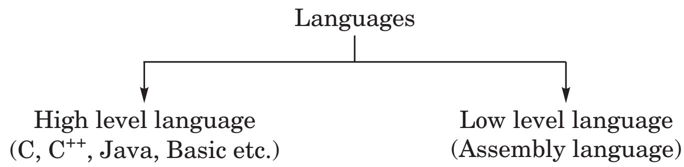
If RD/$\overline{\text{WR}}$ = 0 Write operation

Control bus also include clock line(s) for timing/synchronizing, 'interrupts', 'reset' etc. Typically microprocessor has 10 control lines. The Control Bus carries control signals partly unidirectional, partly bi-directional. Controls signals are things like "read or write". These controls signal tells memory that we are reading from a location, specified on the address bus, or **writing to** a location specified. Typically 16 or 32 bit busses, which allow larger number of instructions, more memory location, and faster arithmetic.

1.5 PROGRAMMING LANGUAGE

1.5.1 Machine Language

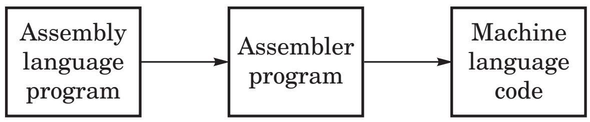
The software developed using 1's and 0's are called machine language, programs. Machine language is a sequence of instructions in the form of binary numbers. This binary coded instruction is called machine code. The first part of instruction is called the opcode code and second part is the operand.



1.5.2 Assembly Language (Low Level Language)

The language in which the mnemonics are used to write a program is called assembly language. The manufacturers of microprocessor give the mnemonics. Mnemonics means short-hand form of instructions. Microprocessor understands Machine Language only.

Microprocessor cannot understand a program written in Assembly language. A program known as **Assembler** is used to convert a Assembly language program to machine language.



Machine language and Assembly language are both Microprocessor specific (**Machine dependent**) so they are called Low-level languages. The software developed using mnemonics are called assembly language programs.

1.5.3 High Level Language

Machine independent languages are called High-level languages. BASIC, PASCAL, C, C++, JAVA, etc. are High-Level languages.

A software called **Compiler** is required to convert a high-level language program to machine code

1.5.4 Disadvantages of Machine Language and Assembly Language

The machine language and assembly language programs are machine dependent. The programs developed using these languages for a particular machine cannot be directly run on another machine.

1.5.5 Advantages of Assembly Language

The advantages of assembly language in comparison with high level language are:

1. Assembly language is directly communicate with operating system of a computer. But high level language is difficult to communicate.
2. Assembly language programming is required small memory space. The high level programs require a large memory space.
3. Assembly language programming helps to learn about the internal architecture of the microprocessor.

1.6 MEMORY

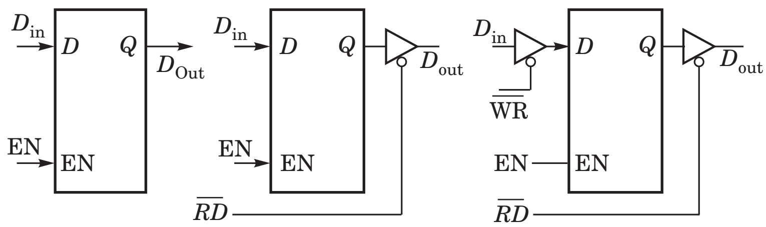
Memory is and essential component of microcomputer system; it stores binary instructions and data for the microprocessor. They can be classified in two groups:

1. Prime (or main) memory
2. Secondary or storage memory.

The R/W memory is made up of registers, and each register can use this memory to hold programs and store data. On the other hand, the ROM stores information permanently in the form of diodes.

1.6.1 Flip-Flop or Latch as a Storage Element

It is a circuit that can store bits. A flip-flop or latch is a basic element of memory. To write or store a bit in the latch, we need an input data bit and an enable signal.

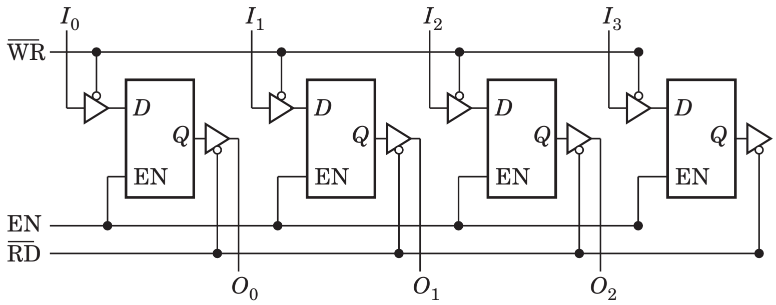


(*a*) Basic latch (*b*) Latch with one tristate buffers (*c*) Latch with two tristate buffers

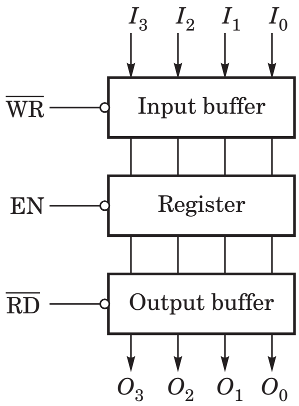
Where $\overline{\text{WR}}\cdot$, $\overline{\text{RD}}\cdot$ and EN are control signals if $\overline{\text{WR}}\cdot = 0$, $\overline{\text{RD}}\cdot = 1$ and EN = 1 then write data in latch and if $\overline{\text{WR}}\cdot = 1$, $\overline{\text{RD}}\cdot = 0$ and EN = 1 then read data from latch. This latch, which can store one binary bit, is called a memory cell.

1.6.2 4-Bit Register

Figure shows four latches as a 4 bit Register. This register has four input lines and four output lines and can store 4-bit digital data. Number of bits stored in a register is called a memory word.



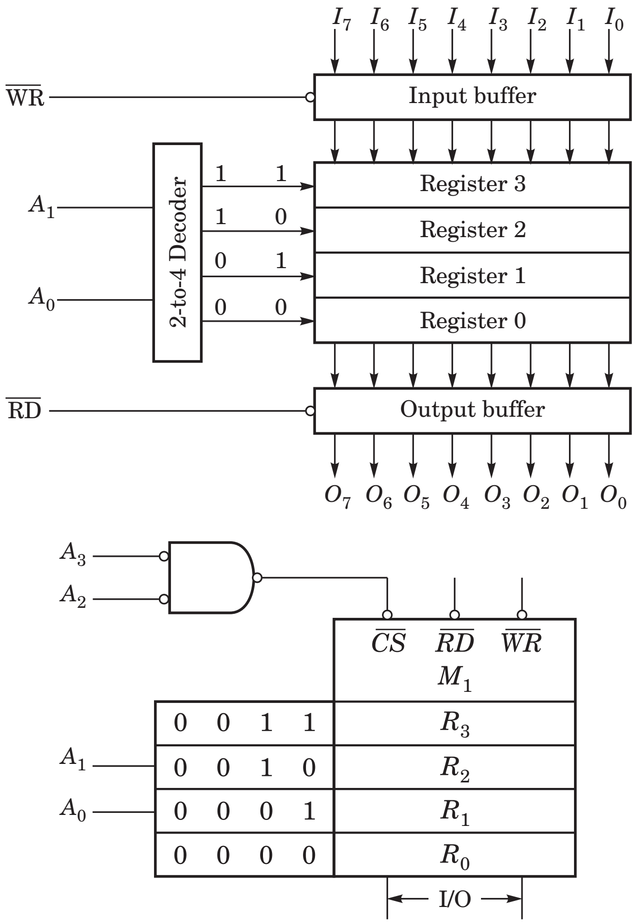
4-bit register using latch



Block diagram

1.6.3 4 × 8 Bit Register

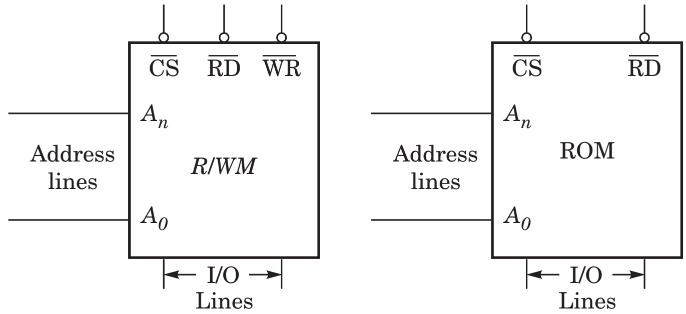
Figure shows latches as a 4 × 8 bit Register. It has four registers and each register has eight cells. Number of bits stored in a register is called a memory word. In this register the length of memory word are 8-bit. It has eight registers with four address lines.



Addressing registers

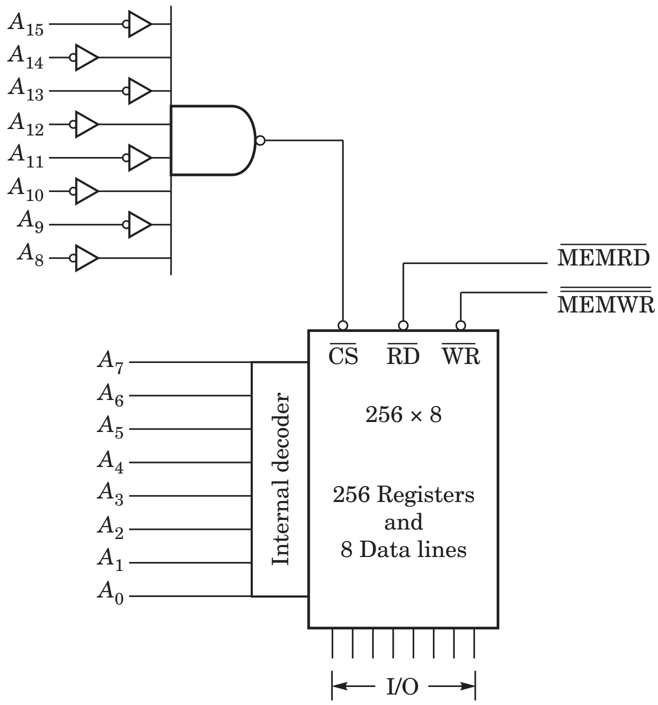
1.6.4 The Requirements of a Memory Chip

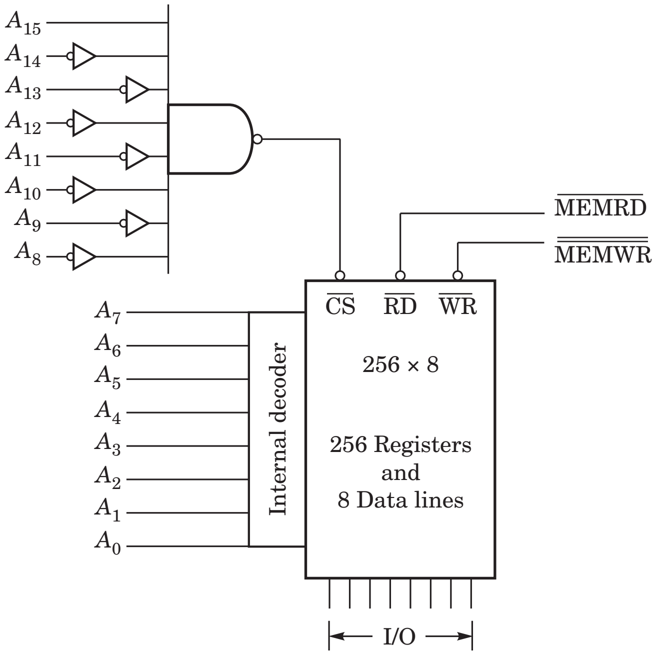
1. A memory chip requires address lines to identify a memory register, a chip select CS signal.
2. The number of address lines required is determined by the number of registers in a chip (2^n).
3. If additional address lines` are available in a system, chip select signal is used.
4. The control signal Read $(\overline{\text{RD}})$ enables the output buffer. The control signal Write $(\overline{\text{WR}})$ enables the input buffer.



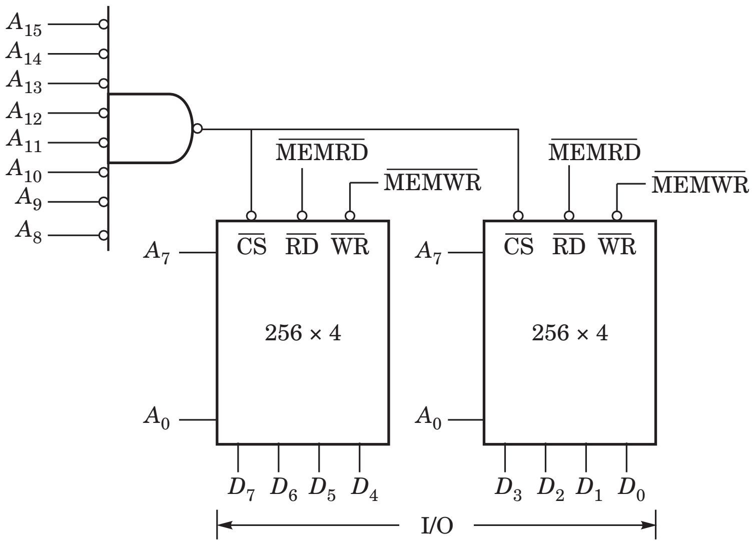
1.6.5 Memory Map and Address

Typically, in an 8-bit microprocessor system, 16 address lines are available for memory. It is capable of identifying 2^{16} (65,536) memory registers, each register with a 16 bit address. The entire memory addresses can range from 0000 to FFFF in Hex.





Memory address



Memory circuit using 256 × 4 memory chips

1.6.6 How the Microprocessor Unit Writes into and Reads from Memory

1. Places the 16 bit address on the address of the memory location where a byte is to be stored. The interfacing logic of the memory chip decodes the address and selects the memory register to be written into.
2. Places the byte on the data bus.
3. Send the control signal Memory. Write to enable the input buffer of the memory and then stores the byte.

To read from memory the step are similar to that of writing into memory, except the order of step 2 and 3.

1. The microprocessor unit send the control signal Memory Read to enable the output buffer of the memory chip.
2. The memory chip places the data byte on the data bus, and the MPU reads the data byte

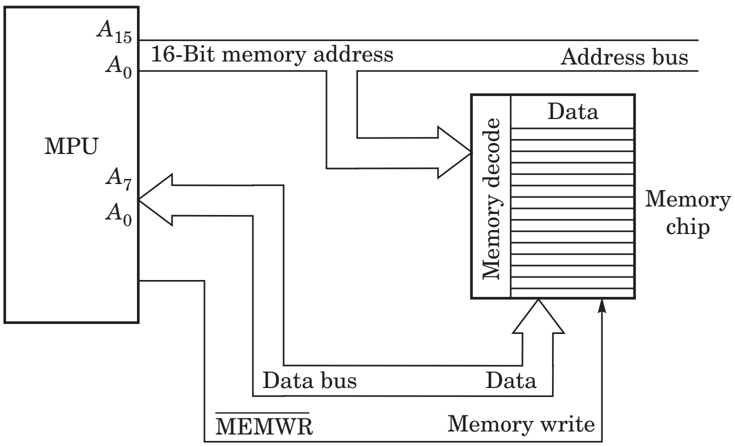


Fig. Memory write operation

1.7 INPUT AND OUTPUT (I/O) DEVICES

Input/output devices are the means through which the microprocessor unit communications with the outside world. The microprocessor unit accepts binary data (1 or 0) as input from input devices such as keyboards, mouse and A/D converters and sends data to output devices such as LEDs or printers etc.

There are two different methods by which an microprocessor unit can identify I/O devices:

1. 8-bit address
2. 16-bit address

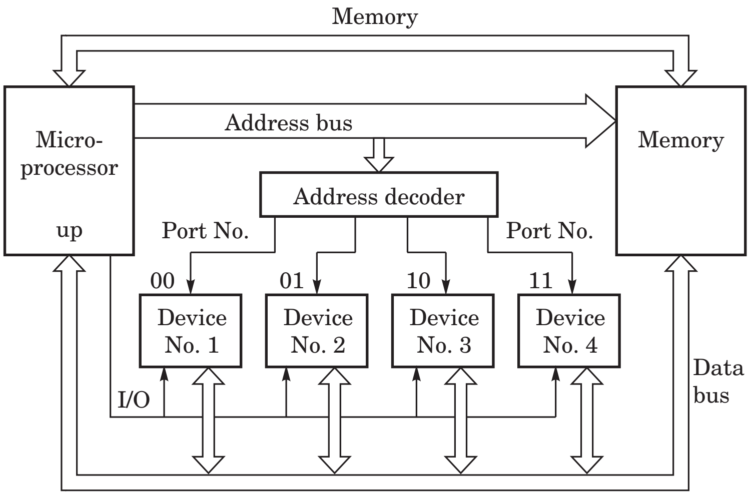
1.7.1 I/Os with 8-bit Addresses (Peripheral-Mapped I/O or I/O Mapped I/O)

The microprocessor unit uses eight address lines to identify an I/O device. The microprocessor unit can identify 256 input and 256 output devices with addresses ranging from $(00)_H$ to $(FF)_H$.

The following steps in communicating with I/O device can be summarized as follows:

1. The microprocessor unit places an 8-bit address on the address bus, which is decodes by the external decode logic.
2. The microprocessor unit sends a control signal (I/O Read or I/O Write) to enable the I/O device.
3. Data are transferred on the data bus.

"In this mapping scheme, an I/O device is treated as a part of memory addressing itself. No separate set of addresses are allocated for I/O devices."

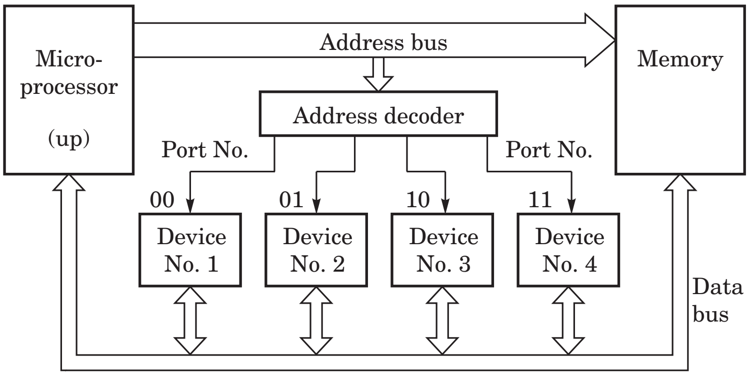


I/O mapped I/O

1.7.2 I/Os with 16 bit Addresses (Memory-Mapped I/O)

The microprocessor unit uses 16-address lines to identify an I/O device, an I/O is connected as if it is a memory register. In memory-mapped I/O, the microprocessor unit uses the same control signals (Memory Read or Memory Write) and instructions as there of memory and follows the same steps as when it is accessing a memory register.

"In this mapping scheme, separate address locations are allocated for I/O and memory."



Memory mapped I/O

1.7.3 Difference between Memory Mapping I/O Device and I/O Mapping I/O device

Memory Mapping I/O device	*I/O Mapping I/O device*
1. 16-bit port addresses are provided for I/O devices.	1. 8-bit port addresses are provided for I/O devices.
2. The I/O ports or peripherals can be used like memory location.	2. IN and Out instructions can be used for data transfer between I/O device and the processor.
3. In memory mapped ports the data can be transfer from any register to ports and vice versa.	3. In I/O mapped ports the data moved only between the ports and accumulator.
4. In memory mapping the devices are accessed by memory read or memory write cycles.	4. In I/O mapping the devices are accessed by I/O read or I/O write cycles. During these cycles the 8-bit address is available on address lines.
5. In memory mapped I/O devices, a large number of I/O ports can be interfaced.	5. In I/O mapped I/O device only 256 (2^8 = 256) ports can be interfaced.
6. For decoding of 16-bit address more hardware is needed.	6. For decoding of 8-bit address less hardware is needed.

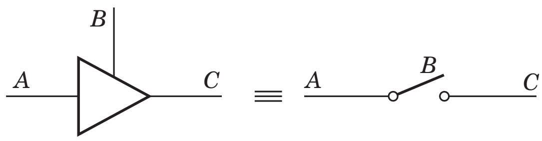
7. When memory mapping is used for I/O devices, then the full memory address space cannot be used for addressing memory. Hence it is useful only for small systems. Where the memory requirement is less.	7. When I/O mapping is used for I/O devices then the full memory address space can be used for addressing memory. Hence it is useful for large systems. This requires large memory capacity.
8. Signals IORD, IOWR and IO/M = 1 are used as control signals.	8. Signals MERD, MEWR and IO/M = 0 are used as control signals.
9. The execution of the IN and OUT instructions is required 10T states.	9. The execution of the STA and LDA instruction is required 13T states.

1.8 LOGIC DEVICES FOR INTERFACING

An I/O interface is required whenever the I/O device is driven by the processor. The interface must have necessary logic to interpret the device address generated by the processor. Many types of interfacing logic devices are used to interconnect the component of bus oriented system. The three state logic devices are necessary for proper functioning of the bus oriented system.

1.8.1 Three-state Logic Device

It is also know as Tri-state logic. In digital electronics three-state or tri-state, logic allows an output port to assume a high impedance state in addition to the 0 and 1 logic levels, effectively removing the output from the circuit. This allows multiple circuits to share the same output line or lines (such as a bus).



O/P C = A if B = ‘1’

High Impedance if B = ‘0’

A tri state buffer can be thought of as a switch. If B is on, the switch is closed. If B is off, the switch is open.

Three-state outputs are implemented in many registers, bus drivers, and flip-flops and also internally in many integrated circuits. Other typical uses are internal and external buses in microprocessors, memories, and peripherals. Many devices are controlled by an active-low input called $\overline{\text{OE}}$ (Output Enable) which dictates whether the outputs should be held in a high-impedance state or drive their respective loads (to either 0- or 1-level).

1.8.2 Digital Encoder

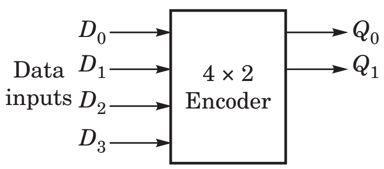
Digital Encoder more commonly called a **Binary Encoder** takes *all* its data inputs one at a time and then converts them into a single encoded output. So we can say that a binary encoder, is a multi-input combinational logic circuit that converts the logic level "1" data at its inputs into an equivalent binary code at its output. Generally, digital encoders produce outputs of 2-bit, 3-bit or 4-bit codes depending upon the number of data input lines. An "n-bit" binary encoder has 2^n input lines and n-bit output lines.

Encoders typically have 2^N inputs and N outputs.

These are called 2^N–to–N encoders.

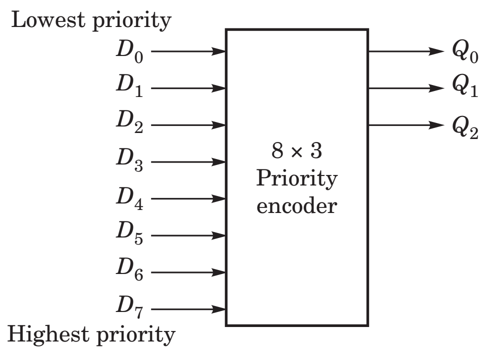
Typical examples include

- 4–to–2 encoders
- 8–to–3 encoders
- 16–to–4 encoders



Truth table of 4 × 2 encoder

Inputs				*Outputs*	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1



Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	D_2	D_1	D_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	X	X	X	X	X	1	0	1
0	1	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

From this truth table, the Boolean expression for the encoder above with inputs D_0 to D_7 and outputs Q_0, Q_1, Q_2 is given as

Output Q_0

$Q_0 = \Sigma\,(1, 3, 5, 7)$

$Q_0 = \Sigma(\overline{D}_7\overline{D}_6\overline{D}_5\overline{D}_4\overline{D}_3\overline{D}_2D_1 + \overline{D}_7\overline{D}_6\overline{D}_5\overline{D}_4D_3 + \overline{D}_7\overline{D}_6\overline{D}_5 + D_7)$

$Q_0 = \Sigma(\overline{D}_6\overline{D}_4\overline{D}_2D_1 + \overline{D}_6\overline{D}_4D_3 + \overline{D}_6D_5 + D_7)$

$Q_0 = \Sigma(\overline{D}_6\,(\overline{D}_4\overline{D}_2D_1 + \overline{D}_4D_3 + D_5) + D_7)$

Output Q_1

$Q_1 = \Sigma\,(2, 3, 6, 7)$

$Q_1 = \Sigma(\overline{D}_7\overline{D}_6\overline{D}_5\overline{D}_4\overline{D}_3D_2 + \overline{D}_7\overline{D}_6\overline{D}_5\overline{D}_4D_3 + \overline{D}_7D_6 + D_7)$

$Q_1 = \Sigma(\overline{D}_5\overline{D}_4D_2 + \overline{D}_5\overline{D}_4D_3 + D_6 + D_7)$

$Q_1 = \Sigma(\overline{D}_5\overline{D}_4(D_2 + D_3) + D_6 + D_7)$

Output Q_2

$Q_2 = \Sigma\,(4, 5, 6, 7)$

$Q_2 = \Sigma(\overline{D}_7\overline{D}_6\overline{D}_5D_4 + \overline{D}_7\overline{D}_6D_5 + \overline{D}_7D_6 + D_7)$

$Q_2 = \Sigma\,(D_4 + D_5 + D_6 + D_4)$

1.8.3 Encoder Applications

Keyboard Encoder

Priority encoders can be used to reduce the number of wires needed in a particular circuits or application that have multiple inputs.

For example, assume that a microcomputer needs to read the 104 keys of a standard QWERTY keyboard where only one key would be pressed either "HIGH" or "LOW" at any one time. One way would be to connect all 104 wires from the keys directly to the computer but this would be impractical for a small home PC, but another better way would be to use a priority encoder. The 104 individual buttons or keys could be encoded into a standard ASCII code of only 7-bits (0 to 127 decimal) to represent each key or character of the keyboard and then inputted as a much smaller 7-bit B.C.D code directly to the computer

1.8.4 Decoders

Decoders are the opposite of encoders; they are N–to–2^N devices.

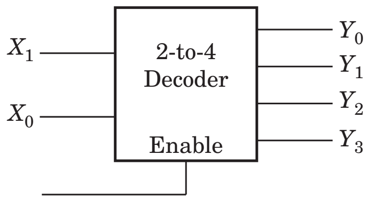
Typical examples include

2–to–4 decoders 3–to–8 decoders 4–to–16 decoders

N–to–2^N decoders have N inputs, labeled X0, X1, …., XN–1

2^N outputs, similarly labeled Y0, Y1, etc. optionally, an enable line.

Decoders come in two varieties: active high and active low



The truth table for an active–high 2–to–4 decoder that is enabled high follows.

Enable	X_1	X_0	Y_0	Y_1	Y_2	Y_3
0	*d*	*d*	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

The "*d*" indicates that when Enable = 0, all outputs are 0 independent of X_0, X_1

Active High vs. Active Low

Here are two decoders.

One is active high and one is active low. Attached to a LED, which illuminates when it is fed with a logic 1.

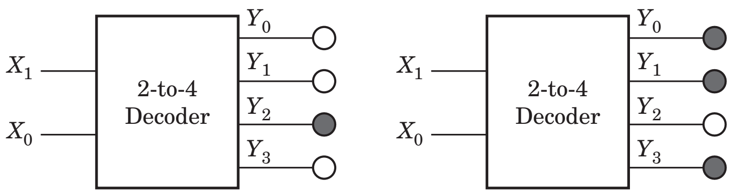
In the circuit at left, only the selected output illuminates its LED.

It is active high.

In the circuit at right, every output but the selected output illuminates its LED.

It is active low.

In each, output 2 has been selected



1.9 THE 8085 MICROPROCESSOR

The microprocessor is a semiconductor device, which has electronic logic circuits. It is also called as central processing unit (CPU). Which is made of very-large-scale integration (VLSI) technique. The microprocessor can perform various computing functions (addition, subtraction, multiplication etc.) and making decisions (logical operation).

Microprocessor can be divided into three blocks:

1. Arithmetic logic unit (ALU)
2. Register unit
3. Control unit.

1.9.1 Arithmetic/Logic Unit

The ALU unit performs such arithmetic operations as addition and subtraction, and such logic operations as AND, OR, and exclusive OR etc.

Functions of Arithmetic Logic Unit

(*a*) It performs arithmetic operations like ; addition, subtraction, increment, multiplication etc.

(*b*) It performs logical operations like ; AND ing, OR ing, X-OR ing, NOT etc.

(*c*) It accepts operands from accumulator and temporary register.

(*d*) It store the result in accumulator or temporary register.

(*e*) It provides states of result to the flag register.

(*f*) It looks after the branching decisions.

1.9.2 Register Unit

The microprocessor consists of various registers such as B, C, D, E, H and L. These registers are primarily used to store data temporarily during the execution of a program. These are general purpose register.

1.9.3 Control Unit

The control unit provides the necessary timing and control signals to all the operations in the microprocessor. It controls the flow of data between the microprocessor and memory and peripherals devices.

1.9.4 Features of 8085 Microprocessors

- The Intel 8085 is an 8-bit microprocessor introduced by Intel in 1977. It can accept, process, or provide 8-bit data simultaneously. The Intel 8085 microprocessor is an advance version of 8080 microprocessor. The 8080 processor was updated with Enable/Disable instruction pins and Interrupt pins to form the 8085 microprocessor.
- It operates on a single +5 V power supply connected at V_{cc}; power supply and ground is connected to V_{ss}.
- It is manufactured with N-MOS technology. The NMOS technology offered fast speed than PMOS technology. The NMOS technology support high packing density than PMOS.
- It has 16 bit address bus and so it can address upto 2^{16} = 65536 bytes (64 KB) memory locations through A_0–A_{15}.
- 8085 is an 8-bit microprocessor because data bus is a group of 8 lines (D_0-D_7).
- It provides 8 bit I/O addresses to access 256 I/O ports.
- It operates at 6.144 MHZ single phase clock. It is enclosed with 40 pins DIP (Dual in line package).
- In 8085, the lower 8-bit address bus (A_0–A_7) and data bus (D_0–D_7) are multiplexed to reduce number of external pins. But due to this, external hardware (latch) is required to separate address lines and data lines.
- It has 8-bit accumulator, flag register, instruction register, six 8-bit general purpose registers (B, C, D, E, H and L), and

two 16-bit registers (SP and PC). Getting the operand from the general-purpose registers is faster than from memory.

- It supports Six 8-bit general purpose register arranged in pairs: BC, DE, HL and a two 16 bit registers program counter (PC) and stack pointer (SP). It has 8-bit accumulator, flag register, instruction register.
- It provides 5 hardware interrupts: TRAP, RST 7.5. RST 6.5, RST 5.5 and INTR. It has serial I/O control which allows serial communication.
- The 8085 have an ability to share system bus with direct memory access (DMA) controller. This feature allows to transfer large amount of data from I/O devices to memory or from memory to I/O device with high speeds.

1.10 8085 MICROPROCESSOR ARCHITECTURE AND ITS OPERATIONS

8085 consists of various units and each unit performs its own specified functions. The various functional blocks are:

1. Arithmetic and logic unit (ALU)
2. Registers
3. Accumulator (A)
4. General purpose registers
5. Program counter (PC)
6. Stack pointer (SP)
7. Temporary register (W, Z)
8. Flags
9. Instruction register and decoder
10. Timing and control unit
11. Interrupt control
12. Serial input/output control
13. Buses address bus and data bus
14. Address buffer and address-data buffer

1.10.1 Arithmetic and Logic Unit (ALU)

The Arithmetic Logic Unit (ALU) of 8085 microprocessor can performs

(*a*) 8-bit binary addition with or without carry
(*b*) 16 bit binary addition
(*c*) 2 digit BCD addition

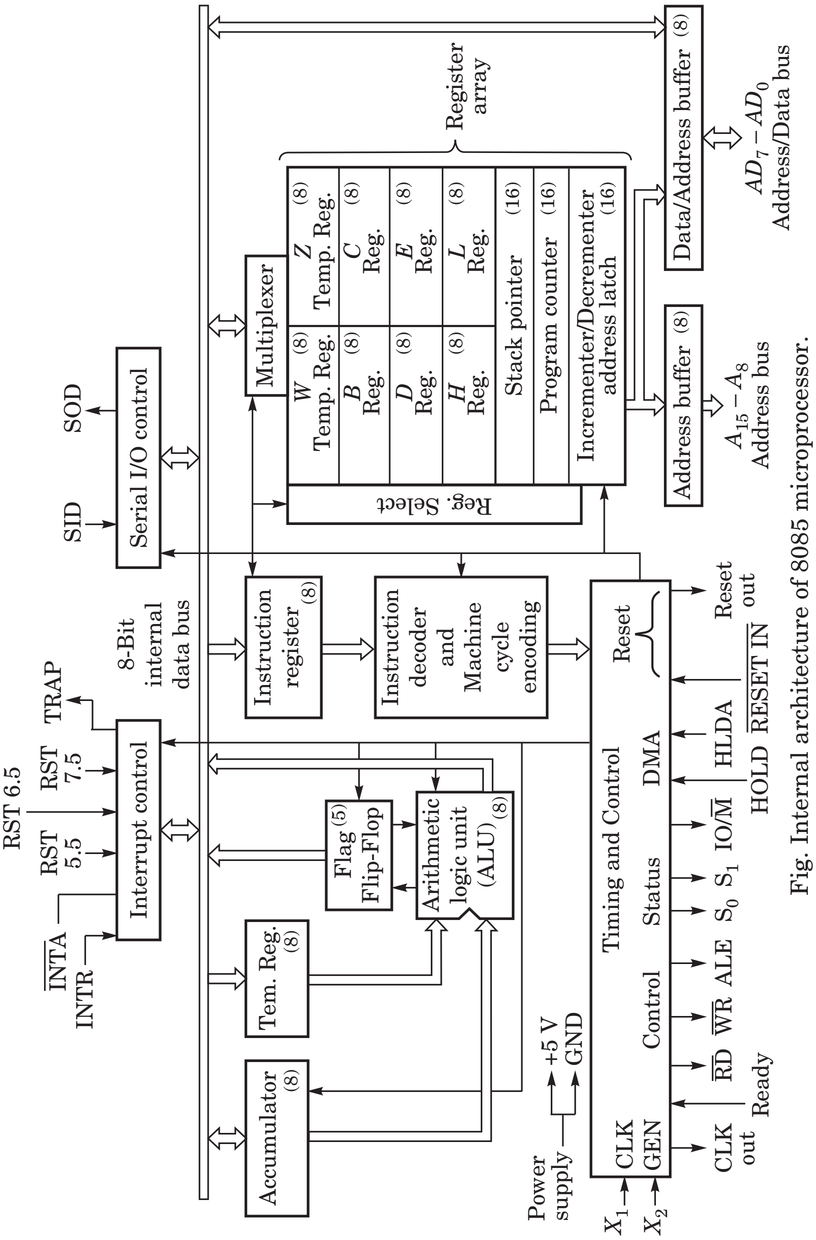


Fig. Internal architecture of 8085 microprocessor.

(*d*) 8-bit binary subtraction with or without borrow

(*e*) 8-bit logical AND, OR, EX-OR, Complement (NOT), and bit shift operations

8085 microprocessor ALU performs arithmetic and logical functions on eight bit variables. The arithmetic unit performs bitwise fundamental arithmetic operations such as addition, subtraction, multiplication, and division on 8 bit data.

The logic unit performs logical operations such as complement, AND, OR, NOT and EX-OR, as well as rotate and clear on 8 bit data.

ALU gets its input from accumulator and temporary register. After processing the necessary operations, the result is stored back in accumulator.

1.10.2 Register

Registers main function is storing data, some of them can be used by the programmer and some cannot, and used only by the processor.

The 8085 microprocessor has eight 8-bit registers: A, B, C, D, E, H, L. They can be combined as register pairs - BC, DE, and HL - to perform some 16-bit operations. It has two 16 bit registers program counter (PC) and stack pointer (SP). It has 8-bit accumulator, flag register, instruction register.

These registers can be classified as:

1. General Purpose Registers
2. Temporary Registers
   (*a*) Temporary Data Register   (*b*) W and Z registers
3. Special Purpose Registers
   (*a*) Accumulator   (*b*) Flag Registers
   (*c*) Instruction Register
4. Sixteen bit Registers
   (*a*) Program Counter (PC)   (*b*) Stack Pointer (SP)

1. General Purpose Registers

The 8085 has six 8-bit general purpose register B, C, D, E, H, L. This register is used to store 8-bit data or as 16-bit register pairs BC, DE, and HL. When used in register pair mode, the high order byte resides in the first register and the low order byte in the second. 8-bit B and 8-bit C registers can be used as one 16-bit BC register pair. Some instructions may use BC register as a data pointer. So these register can be combined as register pairs - BC, DE, and HL - to perform some 16-bit operations. These register are used to store or copy data into the registers by using data transfer instructions.

Register	*Register pair*
B (8 bit) and C (8 bit)	BC (16 bit)
D (8 bit) and E (8 bit)	DE (16 bit)
H (8 bit) and L (8 bit)	HL (16 bit)

General Purpose Register

Register	Register pair
B (8 bit) C (8 bit)	*BC* (16 bit)
D (8 bit) *E* (8 bit)	*DE* (16 bit)
H (8 bit) *L* (8 bit)	*HL* (16 bit)

Individual	B,	C,	D,	E,	H,	L
Combination	B & C,		D & E,		H & L	

2. Temporary register (W, Z)

Temporary register is an 8-bit register *i.e.* W, Z. These are not available for user. Unlike other registers, this temporary register can only be accessed by the microprocessor and it is completely inaccessible to Programmers.

The ALU has two inputs, one from the accumulator and other from temporary data register. They are used to hold 8-bit data during execution of some instructions and are not available for the programmer.

Use of temporary register (W and Z)

1. XCHG instruction exchanges the contents of register H with D and register L with E. At the time of exchange W and Z registers are used for temporary storage of data.

2. The CALL instruction is used to transfer program control to a subprogram or subroutine. This instruction pushes the current PC contents onto the stack and loads the given address into the PC. The given address is temporarily stored in the W and Z registers and placed on the bus for the fetch cycle. Thus the program control is transferred to the address given in the instruction.

3. Special Purpose Registers

Accumulator (A): The accumulator is an 8-bit register that is a part of arithmetic/logic unit (ALU). This register is used to store 8-bit data and to perform arithmetic and logical operations. Accumulator also known as A-register. It is used for storing the two quantities:

1. The data to be processed by arithmetic and logic unit is stored in accumulator.

2. It also stores the result of the operation carried out by the Arithmetic and Logic Unit (ALU).

The accumulator also acts as a general-purpose register, but it has some special capabilities not shared with the other registers. For example, the input/output instructions IN and OUT transfer data only between the accumulator and external I/O devices.

1.10.3 Status Register (Flag register)

The ALU contain five flip-flops, which are set or reset after an operation according to data conditions of the result in the accumulator and other registers. Status flags are also known as condition flag.

Flag Register has five 1-bit flags. Actually there are 8 bit flip flop out of 8 bit 5 bit are condition flags and 3 bit are used internally by the microprocessor. In another word the flags are affected by arithmetic and logic operations in the ALU, and the result is always stored in the accumulator.

S	Z		AC		P		CY
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Status Register

S	Z		AY		P		CY
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Sign flag	Zero flag	XX	Auxiliary carry flag	XX	Parity flag	XX	Carry flag

Carry flag (CY)

After an addition of two numbers, if the sum in the accumulator is larger than eight bits, the flip-flop uses to indicate a carry and the Carry flag (CY) is set to one (set); otherwise it is reset.

'This flag is set if there is an overflow out of bit 7. The carry flag also serves as a borrow flag for subtraction.'

Example:

		$D_7D_6D_5D_4$	$D_3D_2D_1D_0$
+	94H	1001	0100
	A2H	1010	0010
carry → 1		0011	0110

In this example out of 7 bit one carry is generated so carry flag is set to one (CY = 1).

Parity flag (P)

Parity flag set or reset according the number of ones present in the accumulator. After an arithmetic or logical operation if the result has an even number of ones *i.e.*, even parity, the parity flag is set (1). If the parity is odd, parity flag is reset (0).

Example:

	35H	0011	0101
+	C0H	1100	0000
		1111	0101

In this example total number of 1's is even so parity flag is set (PF = 1).

Auxiliary Carry Flag (AC): Auxiliary carry flag is set if there is an overflow out of bit 3 *i.e.*, carry from lower nibble to higher nibble. This flag is used for BCD operations and it is not available for the programmer.

Example:

4BH	=	0100	1011
2EH	=	0010	0110
71H	=	0111	0001

In this example, there is an overflow. Out of D_3 bit so auxiliary carry flag is set (AC = 1).

Zero flag (Z)

The zero flag sets if the result of operation in ALU is zero and flag resets if result is not-zero. This flag is modified by the results in the accumulator as well as in the other registers. The zero flag is also set if a certain register content becomes zero following an increment or decrement operation of that register.

Example:

	A7H	1010	0111
+	59H	0101	1001
		0000	0000

In this example the contents of the accumulator is zero so zero flag is set (ZF = 1)

Zero Flag = 1 (Set)

Sign flag (S)

Sign flag is set if the most significant bit (MSB) of the result is set or 1. After the execution of arithmetic or logical operations, if bit D_7 of the result is 1, the number is viewed as negative number, the sign flag is set or set to 1 one. If D_7 is 0, it will be considered as positive number the sign flag is reset or set to 0 (zero).

Example:

Sign flag

5OH =	5OH =	0	1	0	1	0	0	0	0
– B5H =	+ 4AH =	0	1	0	0	1	0	1	0
	MSB bit →	1	0	0	1	1	0	1	0

In this example MSB bit is 1 so sign flag is set (*i.e*. SE = 1)

1.10.4 Sixteen Bit Registers

Program Counter (PC)

Program counter is a special purpose 16-bit register. Point to the next instruction to be executed. Consider that an instruction is being executed by processor. As soon as the ALU finished executing the instruction, the processor waits for the next instruction to be executed. So, program counter is a necessity for holding the address of the next instruction to be executed in order to save time. The function of the program counter is to point to the memory address from which the next byte is to be fetched.

"*Program counter* (*PC*) *stores the address of the next instruction to be executed*."

Program counter acts as a pointer to the next instruction; for one byte instruction it increments program counter by one, for two byte instruction it increments program counter by two and for three byte instruction it increments the program counter by three such that program counter always points to the address of the next instruction.

Example: In case of JUMP and CALL instructions, address followed by JUMP and CALL instructions is placed in the program counter. The processor then fetches the next instruction from the new address specified by the JUMP or CALL instruction.

Stack Pointer (SP)

A stack is nothing but the portion of RAM (Random access memory). The beginning of the stack is defined by loading a 16-bit address in the stack pointer. The stack pointer is also a 16-bit register used as a memory pointer. It points to a memory location in R/W memory, called the stack. This register is always decremented /incremented by two during push and pop instructions. Each time when the data is loaded into stack (writing into stack), Stack pointer gets decremented. Conversely it is incremented when data is retrieved from stack (reading from stack).

"*Stack pointer is a special purpose 16-bit register in the Microprocessor, which holds the address of the top of the stack*."

1.10.5 Instruction Register & Instruction Decoder

Instruction register used to store the current instructions that are being executed in the microprocessor. Latest instruction sent here from memory prior to execution. Decoder then takes instruction and 'decodes' or interprets the instruction. Decoded instruction then passed to next stage.

Instruction Decoder

It decodes the instruction and accordingly gives the timing and control signals which control the register, the data buffers, ALU and external peripheral signals depending on the nature of the instruction.

Must remember:

Registers:

Registers main function is storing data, some of them can be used by the programmer and some can not, and used only by the processor.

Most important registers:

— *Accumulator:* Also known as A-register, is used for storing the results of mathematical operations.

— *Instruction register:* It is used to store the current instruction that are being executed in the microprocessor.

— *Program counter:* Its stores the address of the next instruction to be executed.

— *Buffer register:* Its stores data temporarily.

— *Status register:* Its stores the current state of the instruction that are being executed at the microprocessor.

— *Stack pointer:* Points to (stores the location of) the place in the main memory called stack

ALU

This unit executes the arithmetic and logical instructions.

1.10.6 Address Buffer and Address-Data Buffer

Address buffer is an 8-bit unidirectional buffer. It is used to drive multiplexed address/data bus *i.e.*, low order address bus (A_7-A_0) and Data bus (D_7–D_0).

The contents of the stack pointer (SP) and program counter (PC) are loaded into the address buffer and address-data buffer. These buffers are then used to drive the external address bus and address-data bus. As the memory and I/O chips are connected to these buses, the CPU can exchange desired data to the memory and I/O chips.

The address-data buffer is not only connected to the external data bus but also to the internal data bus. The address-data buffer is 8-bits long. The address data buffer can both send and receive data from internal data bus.

1.10.7 Timing and Control Circuitry

The timing and control circuitry in the 8085 processor is responsible for all operations. The control circuitry and its operations are synchronized with the help of a clock. It controls the flow of data between the microprocessor and memory and peripherals.

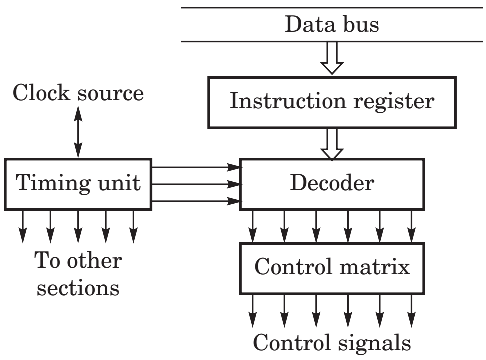
It is controls all internal and external circuits in the microprocessor system. It accepts information from instruction decoder and generates micro steps to perform it. This block accepts clock inputs, performs sequencing and synchronizing operations. The synchronization is required for communication between microprocessor and peripheral devices. To implement this it uses different status and control signals.

The operations of timing and control circuit are

— Control of fetching and decoding operations.

— Generating appropriate signals for instruction execution.

— Generating signals required for interfacing external devices to the processor.



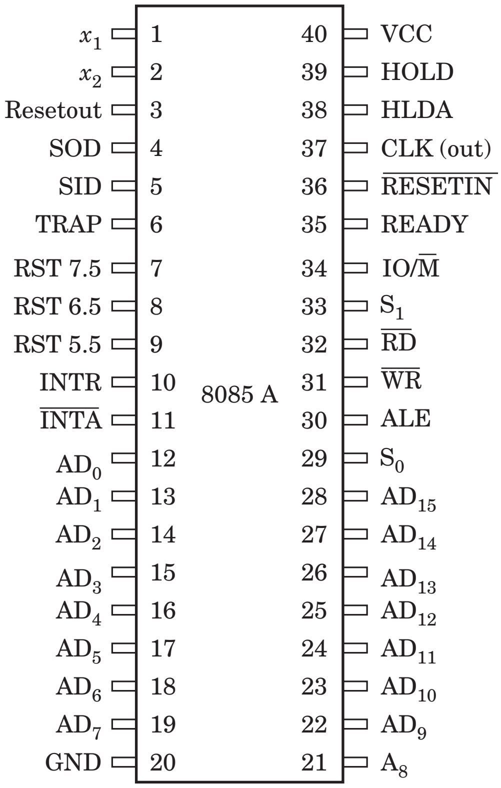
Control section of microprocessor

1.11 PIN DESCRIPTION OF 8085 MICROPROCESSOR

The 8085 is an 8-bit general-purpose microprocessor capable of addressing 64 K of memory because 8085 has 16 bit address bus. Figure shows the logic pin out of the 8085. 8085 microprocessor is a 40 pin IC, DIP package. The signals from the pins can be grouped as follows:

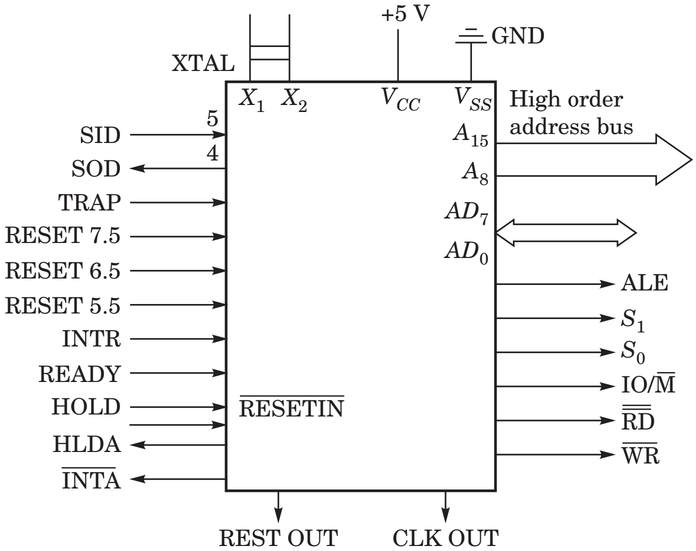
- Power supply and clock signals

- Address bus
- Data bus
- Control and status signals
- Interrupts and externally initiated signals
- Serial I/O ports



8085 Pinout

Pin diagram of 8085 microprocessor



Signal Group of 8085

1.11.1 Power Supply and Clock Frequency

The power supply and frequency signals are as follows:

V_{cc} — + 5 volt power supply

V_{ss} — Ground

X_1, X_2 - Crystal or RC network or LC network connections to set the frequency of internal clock generator. The frequency is internally divided by two; therefore to operate a system at 3 MHz, the crystal frequency should have a frequency of 6 MHz.

CLK (output)-Clock Output is used as the system clock for peripheral and devices interfaced with the microprocessor.

1.11.2 Address Bus/Data Bus

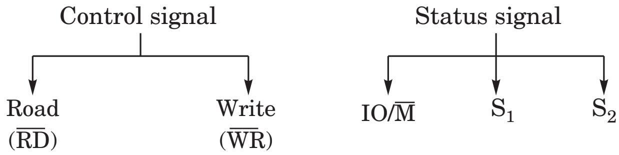
The 8085 has 16 signal lines that are used as the address bus. The address buses are divided into two segments: A_{15} –A_8 and AD_7–AD_0. The eight lines, A_{15}–A_8 are unidirectional and these lines used for the most significant bits (MSB), called high order address. The signal lines AD_7–AD_0 are bidirectional. The lines AD_7–AD_0 are used as the low-order address bus as well as the data bus. The signal lines AD_7–AD_0 are called multiplexed address/data bus. The CPU may read or write out data through these lines.

Low-order address bus : AD_7–AD_0 (bidirectional)

High-order address bus : A_{15}–A_8 (unidirectional)

1.11.3 Control and Status Signals

Control and status signals contain two control signals ($\overline{RD}$ and $\overline{WR}$), three status signals (IO/M, S_1 and S_2) to identify the nature of the operation, and one special signal (ALE) to indicate the beginning of the operation.



S_0, S_1 (output)

These status signals are similar to IO/M. But they are rarely used in small systems. Data Bus Status. Encoded status of the bus cycle:

S_0	S_1	*Operation*
0	0	Halt
0	1	Read
1	0	Write
1	1	Fetch

1.11.4 Address Latch Enable (ALE)

It occurs during the first clock cycle of a machine state.

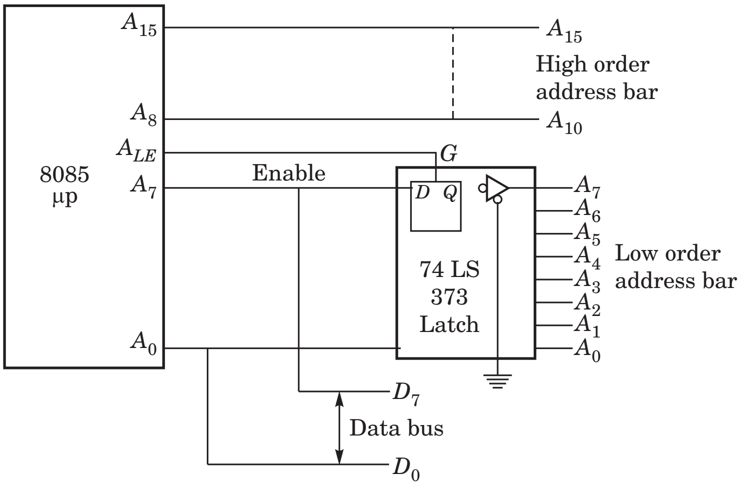
It is a positive going pulse generated every time the 8085 begins an operation (machine cycle). This signal helps to capture the lower order address presented on the multiplexed address/data bus and generate a separate set of eight address lines A_7–A_0. ALE can also be used to strobe the status information. It abbreviation denoted as ALE.

Why demultiplexing of AD_7–AD_0 is required.

Basically, ALE signal give us the information that which of the bus data bus or address bus we are using ALE signal is used to demultiplexing the bus

The bus AD_7 – AD_0 is connected as the input to latch 74LS373 ALE signal is attached to enable pin of the latch (4)

CC — output control ALE when high — latch is transparent output changes according to input data. To reduce the number of pins to make it compact one and easy to use data bus a” address bus is demultiplexed.



Data bus — 8 bits

Address bus —* 16 bits

So 8 bits of address bus and data bus are multiplex and them ALE (Address I enable) is used to demultiplexed them.

1.11.5 Read & Write Signals

Read ($\overline{\textbf{RD}}$)

This is a active low Read control signal. Read signal indicates the selected memory or I/O device is to be read and that the Data Bus is available for the data transfer.

Write ($\overline{\textbf{WR}}$)

This is also a active low write control signal. Write signal indicates the data on the Data Bus is to be written into the selected memory or I/O location.

1.11.6 IO/$\overline{\text{M}}$

This signal is used to differentiate between I/O and memory operations. When IO/$\overline{\text{M}}$ signal is high, it indicates an I/O operation. When IO/$\overline{\text{M}}$ signal is low, it indicates a memory operation.

When IO/$\overline{\text{M}}$ = 1 (high) = I/O operation

IO/$\overline{\text{M}}$ = 0 (Low) = Memory operation

"It's used for Select memory or an IO device".

1.11.7 READY

READY is used by the microprocessor to check whether a peripheral is ready to accept or transfer data. If Ready signal is high during a read or write cycle, it indicates that the memory or peripheral is ready to send or receive data. If Ready signal is low, the microprocessor waits until READY goes high.

1.11.8 DMA (Direct Memory Access) Signals

HOLD

HOLD signal is generated by the DMA controller circuit.

The **DMA controller** raises the **HOLD signal** to indicate that it wants the bus for a DMA transfer.

HOLD

Hold signal indicates if any other device is requesting the use of address and data bus. Consider two peripheral devices. One is the Printer and the other Analog to Digital converter (ADC). Suppose if analog to digital converter (ADC) is using the address and data bus and if printer requests the use of address and data bus by giving HOLD signal, then the microprocessor transfers the control to the printer as soon as the current cycle is over. After the printer process is over, the control is transferred back to analog and digital converter.

HLDA

HLDA is the acknowledgement signal for HOLD. It indicates whether the HOLD signal is received or not. After the execution of HOLD request, HLDA goes low. After the HLDA signal the DMA controller starts the direct transfer of data with the help of DMA controller.

1.12 INTERRUPTS

Interrupt is signals send by an external device to the processor, to request the processor to perform a particular specific task or work. When a program receives an interrupt signal, it takes a specified action. Interrupts are used for data transfer between the peripheral and the microprocessor. The microprocessor will check the interrupts always at the 2nd T-state of last machine cycle.

1.12.1 Classification of Interrupts

Interrupts can be classified into two types:

1. Maskable–Interrupts (this type of interrupt can be delayed or rejected)
2. Non-**Maskable** Interrupts (this type of interrupt cannot be delayed or Rejected)

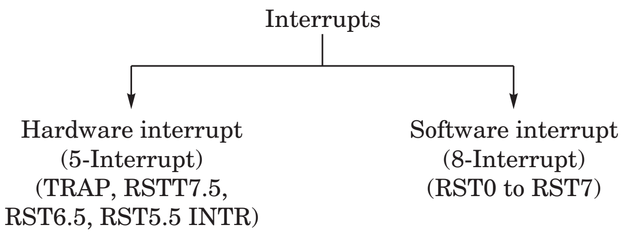
Interrupts can also be classified into:

1. Vectored interrupt (the address of the service routine is hard-wired)

2. Non-vectored interrupt (the address of the service routine needs to be supplied externally by the device)

8085 microprocessor supports two types of interrupts.

- Software interrupts
- hardware interrupts



1.12.2 Software Interrupts

Interrupt signals initiated by programs are called *software interrupts.* The software interrupts are program instructions. These instructions are inserted at desired locations in a program. A software interrupt is also called a *trap* or an *exception.* The 8085 has eight software interrupts from RST 0 to RST 7.

How can calculate vector address?

Vector address= Interrupt number * 8

For example calculate vector address for RST3

$$3 * 8 = 24 = 18H$$

Vector address for interrupt RST 3 is 0018H

Software Interrupt	**Vector address**
RST0	$(0000)_H$
RST1	$(0008)_H$
RST2	$(0010)_H$
RST3	$(0018)_H$
RST4	$(0020)_H$
RST5	$(0028)_H$
RST6	$(0030)_H$
RST7	$(0038)_H$

1.12.3 Hardware Interrupts

The hardware interrupts signals initiated by an external device and placing an appropriate signal at the interrupt pin of the microprocessor.

Hardware interrupt signals present in 8085 are

(*i*) TRAP (*ii*) RST 7.5 (*iii*) RST 6.5

(*iv*) RST 5.5 (*v*) INTR

How can calculate vector address?

Vector address = Interrupt number * 8

For example calculate vector address for RST7.5

7.5 * 8 = 60 = 3CH

Vector address for interrupt RST7.5 is 003CH

Hardware Interrupt	Vector Address
RST7.5	$(003C)_H$
RST6.5	$(0034)_H$
RST5.5	$(002C)_H$
TRAP	$(0024)_H$

TRAP

— It has the highest priority.

— TRAP interrupt is edge and level triggered. This means hat the TRAP must go high and remain high until it is acknowledged.

— Among the interrupts of 8085 microprocessor, TRAP is the only non-maskable interrupt.

— It cannot be enabled or disabled using a program.

— TRAP interrupt can be clear by two ways:

1. By giving a high TRAP ACKNOWLEDGE (Internal signal)
2. By resetting microprocessor (External signal)

— When TRAP interrupt is received the processor saves the contents of the program counter (PC) register into stack and branches to 24H (Vector address) address.

— The HOLD signal can be overrides the TRAP. If the processor receives HOLD and TRAP at the same time then HOLD is recognized first and then TRAP is recognized.

RST7.5

— It has the second highest priority.

— RST7.5 interrupt is edge sensitive. Input goes to high and no need to maintain high state until it recognized.

— RST7.5 is maskable interrupt. RST7.5 can be enabled or disabled using programs. It is enabled by EI instruction and it is disabled by three ways:

1. DI instruction
2. System or processor reset.
3. After reorganization of interrupt.

— When this interrupt is received the processor saves the contents of the PC register into stack and branches to 3CH (Vector address) address.

RST6.5 and RST5.5

— RST6.5 has the third highest priority and RST5.5 has the fourth priority.

— RST6.5 and RST5.5 are maskable interrupt. RST6.5 and RST5.5 can be enabled or disabled using programs. It is enabled by EI instruction and it is disabled by three ways:

1. DI instruction
2. System or processor reset.
3. After reorganization of interrupt.

— RST6.5 and RST5.5 are level triggered. Inputs goes to high and stay high until it recognized.

— When RST6.5 interrupt is received the processor saves the contents of the PC register into stack and branches to 34H (Vector address) address.

— When RST5.5 interrupt is received the processor saves the contents of the PC register into stack and branches to 2CH (Vector address) address.

INTR

— It has lowest priority. It is a level sensitive interrupts. Input goes to high and it is necessary to maintain high state until it recognized.

INTR is maskable interrupt. It is enabled by EI instruction and it is disabled by three ways:

1. DI, SIM instruction
2. System or processor reset.
3. After reorganization of interrupt.

Interrupt time	*Maskable interrupt*	*Vectored interrupt*
RST 5.5	Yes	Yes
RST 6.5	Yes	Yes
RST 7.5	Yes	Yes
TRAP	No	Yes
INTR	Yes	No

1.13 THE 8085 ADDRESSING MODES

The instructions are to copy data from a source into a destination. In these instructions the source can be a register, an input port, or an 8-bit number (00H to FFH). Similarly, a destination can be a register or an output port. The sources and destination are operands. The various formats for specifying operands are called the ADDRESSING MODES.

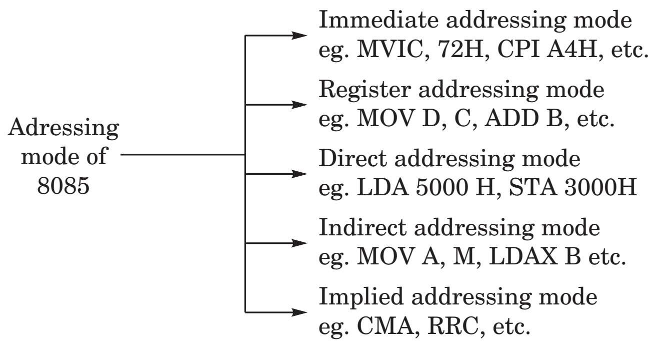
"The data are specified in different modes in the instructions. The various ways of specifying data are called addressing modes."

Or

"Addressing modes are the ways how microprocessors specify the address of an object they want to access."

The 8085 microprocessor has five types of addressing modes

1. Immediate Addressing mode
2. Register Addressing mode
3. Direct Addressing mode
4. Indirect Addressing mode
5. Implied Addressing mode



1.13.1 Immediate Addressing Mode

When the data is directly specified in the instruction, Which is called immediate addressing mode. (MVI Reg, Data). In immediate addressing mode, the data is specified in the instruction itself. The data will be a part of the program instruction. Load the immediate data to the destination provided.

Example: MVI C, 72 H

Intermediate data

In this example move the data immediate data *i.e.* 72H given in the instruction itself to C register.

More examples LXI SP, 5000H, ADI 40H, ANI 75H, CPI A4H etc are immediate addressing mode.

1.13.2 Register Addressing Mode

When the data is stored in the register and if the register is specified in the instruction, then it is called register addressing mode. (MOV Rd., Rs). In register addressing mode, the instruction specifies the name of the register in which the data is available. Data is provided through the registers.

Example: MOV D, C

Data stored in register

In this example Move the content of C register to D register.

More examples ADD B. SUB C, ANA D, ORA B etc. are register addressing mode.

1.13.3 Direct Addressing Mode

In direct addressing mode, the address of the data is specified in the instruction. The data will be in memory. When the memory Address specified with in the instruction, then it is called direct addressing mode.

Example: LDA 5000H

In this example Load the data available in memory location 5000H in to accumulator.

More examples SHLD 4000H, STA 3000H etc are direct addressing mode.

1.13.4 Indirect Addressing Mode

Load the data, to the accumulator from the memory. The data which is pointed by the memory pointer (HL reg), it is called indirect addressing mode.

Example: MOV A, M

In this example the memory data addressed by H L pair is moved to A register.

More examples LDAX D, STAX B etc. are direct addressing mode.

1.13.5 Implied Addressing Mode

When the instruction itself specifies the data to be operated, then it is called Implied addressing mode.

Example:

CMA - Complement the content of accumulator.

RRC - rotate accumulator content with out carry.

RAR, RAL RLC etc.

REVIEW QUESTIONS

1. What is microprocessor ? What is the difference between a microprocessor and a central processing unit (CPU).
2. Explain the difference between a microprocessor and a micro-controller.
3. Discuss about evolution of microprocessor. (*GBTU 2009-10*)
4. What do you understand by machine language?
5. What are low and high level languages?
6. What is an assembler explain in brief.
7. Explain difference between dual core and multi-processor.
8. What are the advantages of an assembly language in comparison with high-level languages?
9. Explain the 8085 based microcomputer system.
10. Draw the Pin Diagram of 8085 and explain the function of various signals.
11. What are the advantages of an assembly language in comparison with high level languages?
12. Draw and explain internal architecture of 8085. (*GBTU 2009-10*)
13. What is the need for ALE signal in 8085 microprocessor?
14. What is the need for interfacing?
15. What are the different addressing modes of 8085?
16. Describe the functional block diagram of 8085.

17. What is a bus?

18. Why is the data bus bidirectional?

19. Define bit, byte, word, double word, quad word and instruction.

20. Explain the interrupt structure of 8085.

21. What is the function of the accumulator.

22. Explain the use of Stack Pointer (SP) during PUSH and POP operation. (*UPTU 2009-2010*)

23. If the memory chip size is 4096 × 8 bits, how many chips are required to make up 16K-byte memory.

24. Explain the functions of the ALE and IO/M signals of the 8085 microprocessor.

25. What are the various registers in 8085?

26. Calculate the address lines required for an 8K-byte. (*GBTU 2010-2011*)

27. Calculate the number of memory chips needed to deign 8K-byte memory if the memory chip size is 1024 × 1. (*GBTU 2010-2011*)

MULTIPLE CHOICE QUESTIONS

1. Who invented the microprocessor?

(*a*) Herman H. Goldstein (*b*) Marcian E. Huff

(*c*) Joseph Jacquard (*d*) All of above

2. An integrated circuit (IC) is

(*a*) A complicated circuit

(*b*) Fabricated on a tiny silicon chip

(*c*) Much costlier than a single transistor

(*d*) An integrating device

3. Which is used for manufacturing chips?

(*a*) Bus (*b*) Control unit

(*c*) Semiconductors (*d*) (*a*) and (*b*) only

4. The first machine to successfully perform a long series of arithmetic and logical operations was:

(*a*) ENIAC (*b*) Mark-I

(*c*) Analytic Engine (*d*) UNIVAC-1

5. The processes of starting or restarting a computer system by loading instructions from a secondary storage device into the completer memory is called

(*a*) Duping (*b*) Booting

(*c*) Padding (*d*) All of above

6. EBCDIC can code up to how many different characters?

(*a*) 256 (*b*) 16

(*c*) 32 (*d*) 64

7. The memory which is programmed at the time it is manufactured
(*a*) EPROM (*b*) PROM
(*c*) RAM (*d*) ROM

8. Which of the following memory medium is not used as main memory system?
(*a*) Magnetic core (*b*) Magnetic tape
(*c*) Semiconductor (*d*) Both (*a*) and (*b*)

9. Registers, which are partially visible to users and used to hold conditional, are known as
(*a*) PC (*b*) Memory address registers
(*c*) General purpose register (*d*) Flags

10. One of the main feature that distinguish microprocessors from micro-computers is
(*a*) Microprocessor does not contain I/O devices
(*b*) Words are shorter in microprocessors
(*c*) Words are usually larger in microprocessors
(*d*) Exactly the same as the machine cycle time

11. The least significant bit (LSB) of the binary number, which is equivalent to any odd decimal number, is
(*a*) 1 (*b*) 0
(*c*) 1 or 0 (*d*) 3

12. When was the world's first laptop computer introduced in the market and by whom?
(*a*) Hewlett-Packard (*b*) Epson, 1981
(*c*) Laplink travelling software Inc. 1982
(*d*) Tandy model-2000, 1985

13. What type of control pins are needed in a microprocessor to regulate traffic on the bus, in order to prevent two devices from trying to use it at the same time?
(*a*) Bus control (*b*) Interrupts
(*c*) Bus arbitration (*d*) Status

Hint: In a single bus architecture when more than one device requests the bus, a controller called bus arbiter decides who gets the bus, this is called the bus arbitration. Arbitration is mostly done in favour of a master micro processor with the highest priority.

14. The first microprocessor built by the Intel Corporation was known
(*a*) 8008 (*b*) 8080
(*c*) 8800 (*d*) 4004

15. Which of the following memories must be refreshed many times per second?
(*a*) Dynamic RAM (*b*) Static RAM
(*c*) EPROM (*d*) ROM

16. To locate a data item for storage is

(*a*) Field (*b*) encode

(*c*) Fetch (*d*) Database

17. A digital computer did not score over an analog computer in terms of

(*a*) Speed (*b*) Accuracy

(*c*) Reliability (*d*) Cost

18. How many address lines are needed to address each memory location in a 2048 × 4 memory chip?

(*a*) 10 (*b*) 11

(*c*) 8 (*d*) 12

19. Which American Computer Company is called big blue?

(*a*) IBM (*b*) Compaq Corp

(*c*) Microsoft (*d*) Tandy Sevenson

20. When did IBM introduced the 80286 based PC/AT?

(*a*) 1982 (*b*) 1989

(*c*) 1985 (*d*) 1984

21. Which of the following required large computer memory?

(*a*) Imaging (*b*) Voice

(*c*) Graphics (*d*) All of above

22. The computer code for the interchange of information between terminals is

(*a*) BCD (*b*) ASCII

(*c*) EBCDIC (*d*) All of above

23. A bit consists of

(*a*) One bit (*b*) Four bits

(*c*) Eight bits (*d*) Sixteen bits

24. A nibble consists of

(*a*) One bit (*b*) Four bits

(*c*) Eight bits (*d*) Sixteen bits

25. A byte consists of

(*a*) One bit (*b*) Four bits

(*c*) Eight bits (*d*) Sixteen bits

26. Which was the world's first minicomputer and when was it introduced?

(*a*) PDP-I, 1958 (*b*) PDP-II, 1961

(*c*) IBM System/36, 1960 (*d*) VAX 11/780, 1962

27. Address line for RST4 is?

(*a*) 0020H (*b*) 0028H

(*c*) 0018H (*d*) 0030

28. A computer which CPU speed around 100 million instruction per second and with the word length of around 64 bits is known as

(*a*) Super computer (*b*) Mini computer

(*c*) Micro computer (*d*) Macro computer

29. The digital computer was developed primarily in

(*a*) USSR (*b*) USA

(*c*) Japan (*d*) UK

30. Why 8085 processor is called an 8 bit processor?

(*a*) Because 8085 processor has 8 bit ALU

(*b*) Because 8085 processor has 8 bit data bus

(*c*) (*a*) and (*b*)

(*d*) none of these

ANSWERS

1. (*b*)	**2.** (*b*)	**3.** (*c*)	**4.** (*b*)	**5.** (*b*)
6. (*a*)	**7.** (*d*)	**8.** (*d*)	**9.** (*c*)	**10.** (*a*)
11. (*a*)	**12.** (*b*)	**13.** (*c*)	**14.** (*d*)	**15.** (*a*)
16. (*c*)	**17.** (*b*)	**18.** (*b*)	**19.** (*a*)	**20.** (*d*)
21. (*d*)	**22.** (*b*)	**23.** (*a*)	**24.** (*b*)	**25.** (*c*)
26. (*a*)	**27.** (*a*)	**28.** (*a*)	**29.** (*b*)	**30.** (*a*)