FUNDAMENTAL OF MICROPROCESSOR

INTRODUCTION, APPLICATION AND TRENDS

UNIT OUTLINE

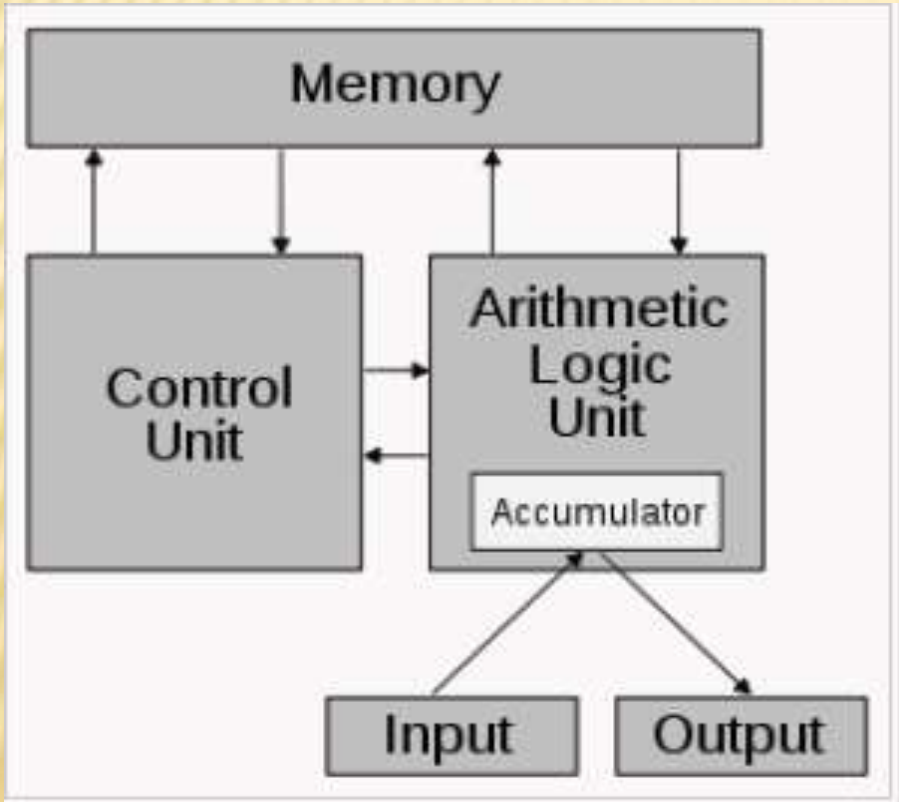
- FUNDAMENTAL OF MICROPROCESSOR
- EVOLUTION OF MICROPROCESSOR
- 8 BIT MICROPROCESSOR ARCHITECTURE
- DEFINITION OF EMBEDDED SYSTEM AND ITS CHARACTERISTICS
- ROLE OF MICROCONTROLLERS
- 8051 MICROCONTROLLERS AND ITS FAMILIES
- ARCHITECTURE OF 8 BIT MICROCONTROLLER

FUNDAMENTAL OF MICROPROCESSOR

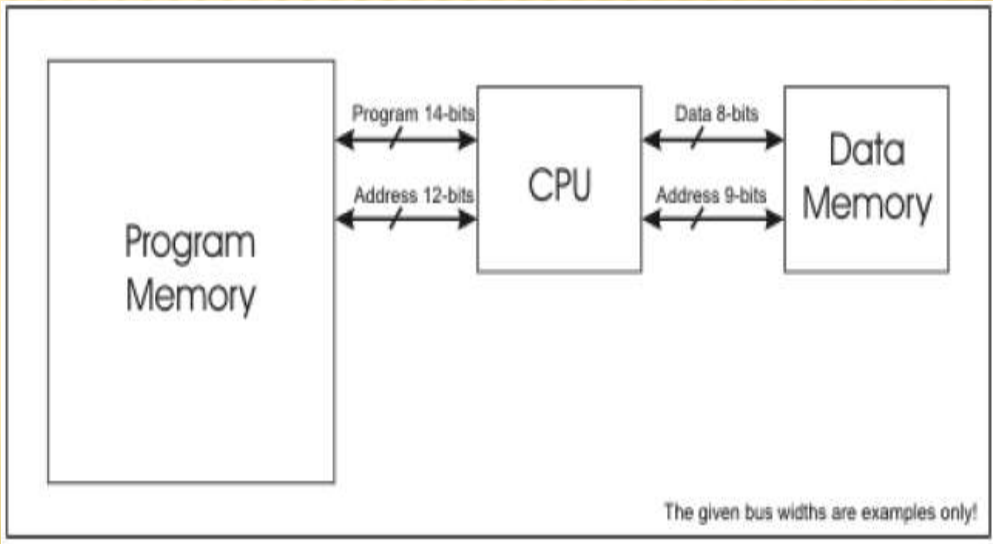
- The microprocessor is a multipurpose, clock driven, register based, digital-integrated circuit which accepts binary data as input, processes it according to instructions stored in its memory, and provides results as output."
- "Microprocessor is a computer Central Processing Unit (CPU) on a single chip that contains millions of transistors connected by wires."
- Microprocessor is a programmable integrated device that has computing and decision making capability, similar to CPU of a computer.
- Each μp has a fixed set of instructions in the form of binary patterns called machine language.
- The binary instructions are given abbreviated names, called mnemonics, form the assembly language.

FUNDAMENTAL OF MICROPROCESSOR

- Von Neumann architecture vs Harvard architecture



Design of the von Neumann architecture



FUNDAMENTAL OF MICROPROCESSOR

- Components of Microprocessor

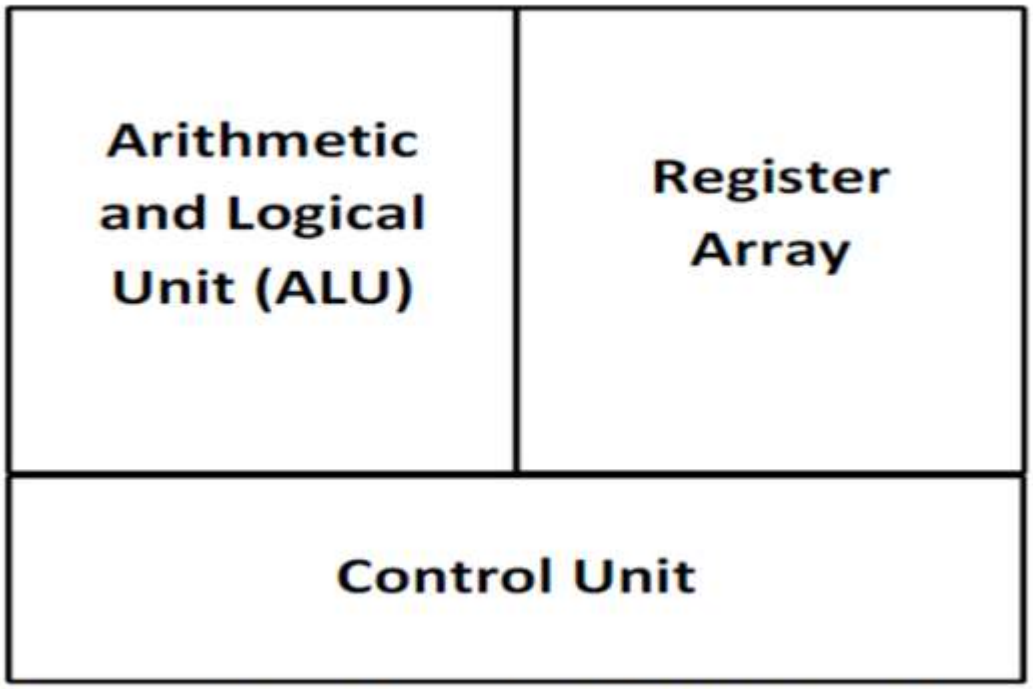


Figure: Components of Microprocessor

EVOLUTION OF PROCESSORS

- **Intel 4004**

  Year of introduction 1971

  - 4-bit microprocessor
  - 4 KB main memory
  - 45 instructions
  - PMOS technology
  - was first programmable device which was used in calculators

- **Intel 8008**

  Year of introduction 1972

  - 8-bit version of 4004
  - 16 KB main memory
  - 48 instructions
  - PMOS technology

EVOLUTION OF PROCESSORS

- **Intel 8080**

  Year of introduction 1973

  - 8-bit microprocessor
  - 64 KB main memory
  - 2 microseconds clock cycle time
  - 500,000 instructions/sec
  - 10X faster than 8008
  - NMOS technology
  - Drawback was that it needed three power supplies.
  - Small computers (Microcomputers) were designed in mid 1970's using 8080 as CPU

EVOLUTION OF PROCESSORS

- **Intel 8085**

  Year of introduction 1975

  - 8-bit microprocessor-upgraded version of 8080
  - 64 KB main memory
  - 1.3 microseconds clock cycle time
  - 246 instructions
  - Intel sold 100 million copies of this 8-bit microprocessor
  - uses only one +5v power supply.

EVOLUTION OF PROCESSORS

Intel 8086/8088

Year of introduction 1978 for 8086 and 1979 for 8088

- 16-bit microprocessors
- Data bus width of 8086 is 16 bit and 8 bit for 8088
- 1 MB main memory
- 400 nanoseconds clock cycle time
- 6 byte instruction cache for 8086 and 4 byte for 8088
- Other improvements included more registers and additional instructions
- In 1981 IBM decided to use 8088 in its personal computer

EVOLUTION OF PROCESSORS

Intel 80386

- Year of introduction 1986
  - Intel's first practical 32-bit microprocessor
  - 4 GB main memory
  - Improvements include page handling in virtual environment
  - Includes hardware circuitry for memory management and memory assignment
  - Memory paging and enhanced I/O permissions

EVOLUTION OF PROCESSORS

Pentium

Year of introduction 1993

- 32-bit microprocessor, 64-bit data bus and 32-bit address bus
- 4 GB main memory
- Double clocked 120 and 133MHz versions
- Fastest version is the 233MHz, Dual integer processor
- 16 KB L1 cache (split instruction and data: 8 KB each)

EVOLUTION OF PROCESSORS

Processor	Year Of Introduction	No. Of Transistors	Intial Clock Speed	Address Bus	Data Bus(in bit)	Addressable Memory
4004	1971	2300	108 kHz	10 bit	4	640 bytes
8008	1972	3500	200 kHz	14 bit	8	16 k
8080	1974	6000	2 MHz	16 bit	8	64 k
8085	1976	6500	5 MHz	16 bit	8	64 k
8086	1978	29000	5 MHz	20 bit	16	1 M
8088	1979	29000	5 MHz	20 bit	8	1 M
80286	1982	134000	8 MHz	24 bit	16	16 M
80386	1985	275000	16 MHz	32 bit	32	4 G
80486	1989	1.2 M	25 MHz	32 bit	32	4 G
Pentium	1993	3.1 M	60 MHz	32 bit	32/64	4 G
Pentium Pro	1995	5.5 M	150 MHz	36 bit	32/64	64 G
Pentium II	1997	8.8 M	233 MHz	36 bit	64	64 G
Pentium III	1999	9.5 M	650 MHz	36 bit	64	64 G
Pentium 4	2000	42 M	1.4 GHz	36 bit	64	64 G

COMPONENTS OF MICROPROCESSOR

- ALU
- The ALU performs the actual numerical and logic operation such as add, subtract, AND, OR, etc. Uses data from memory and from Accumulator to perform arithmetic. Always stores result of operation in Accumulator.
- It is not accessible by user.
- The word length of ALU depends upon of an internal data bus.
- IT is 8 bit. It is always controlled by timing and control circuits.
- It provides status or result on flag register.

- Arithmetic and Logical Section
- Instruction Decoder and Machine cycle Encoder
- Address Buffer
- Address/Data Buffer
- Incrementer/Decrementer Address Latch
- Interrupt Control
- Serial I/O Control Group
- Timing And Control

- Register Section
  - Temporary Register
  - General Purpose Register
  - Special Purpose Registers

- Temporary Registers

- In 8085 available temporary register are temporary data register and W and Z registers. These registers are not available to the programmer, but 8085 uses them internally to hold temporary data during execution of some instructions.
- W and Z are two 8-bit temporary registers, used to hold 8-bit

  data/address during execution of some instructions.

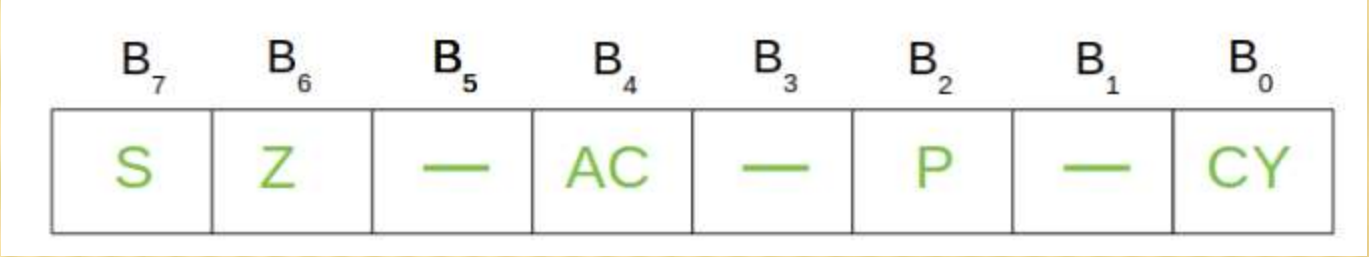
- General Purpose Registers
- B,C,D,E,H &L are used as general purpose register.
- Each 8 bit long .
- Apart from the above function these registers can also be used to work in pairs to hold 16-bit data.
- They can work in pairs such as B-C, D-E and H-L to store 16-bit data.
- The H-L pair works as a memory pointer.
- A memory pointer holds the address of a particular memory location.
- They can store 16-bit address as they work in pair.

- Special Purpose Register
- Accumulator
- Status or Flag Register
- Instruction Register
- Program Counter
- Stack Pointer

- Accumulator
- 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.
- The result of an operation is stored in the accumulator. The accumulator is also identified as register A.
- It also works as a via register for I/O accesses i.e. it reads data from input device and similarly can transfer data to output device.

Status or Flag Register

- Flag register is a group of flip flops used to give status of different operations result.
- The flag register is connected to ALU.
- Once an operation is performed by ALU the result is transferred on internal data bus and status of result will be stored in flip flops.
- They are called Zero (Z), Carry (CY), Sign (S), Parity (P), and Auxiliary Carry (AC) flags.



Flag Register…

- **Carry flag(CY):**
- If an operation performed in ALU generates the carry from D7 to next stage then CY flag is set, else it is reset.
- **Auxiliary carry(AC):**
- If an operation performed in ALU generates the carry from lower nibble (D0 to D3) to upper nibble (D4 to D7) AC flag is set, else it resets.
- **Zero flag(z):**
- If an operation performed in ALU results 0value of entire 8-bits then zero flag is set, else it resets
- **Sign flag(s):**
- If MSB bit =0 then the number is positive, else it is negative.
- **Parity flag(p):**
- If the result contains even no. of ones this flag is set and for odd no. of ones this flag is reset.

- Instruction Register
- Temporary store for the current instruction of a program. 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 register is 8-bit register just like every other register of microprocessor. Consider an instruction. The instruction may be anything like adding two data's, moving a data, copying a data etc. When such an instruction is fetched from memory, it is directed to Instruction register. So the instruction registers are specifically to store the instructions that are fetched from memory.

- There is an Instruction decoder which decodes the information's present in the Instruction register for further processing.

- Program Counter
- This 16-bit register deals with sequencing the execution of instructions. This register is a memory pointer. Memory locations have 16-bit addresses, and that is why this is a 16-bit register.
- The microprocessor uses this register to sequence the execution of the instructions. The function of the program counter is to point to the memory address from which the next byte is to be fetched. When a byte is being fetched, the program counter is incremented by one to point to the next memory location

- 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. The beginning of the stack is defined by loading 16-bit address in the stack pointer.
- Stack pointer is also a 16-bit register which is used as a memory pointer.
- A stack is nothing but the portion of RAM (Random access memory).
- Stack pointer maintains the address of the last byte that is entered into stack.
- Each time when the data is loaded into stack, Stack pointer gets decremented.
- Conversely it is incremented when data is retrieved from stack

INSTRUCTION DECODER AND MACHINE CYCLE ENCODER

- It accepts an OPCODE from instruction register, decodes it and gives the decoded information to control logic.
- The information includes what operation is to be performed, who is going to perform it, etc. It means it will understand the instruction in this block.
- The decoded information is given to the timing and control unit that provides control signals

ADDRESS BUFFER

- The contents of the stack pointer and program counter 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 which consists of 8-bits.
- The address data buffer can both send and receive data from internal data bus.
- The contents of the stack pointer and program counter 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 which consists of 8-bits. The address data buffer can both send and receive data from internal data bus.

INCREMENTER/DECREMENTER ADDRESS LATCH

- The 8-bit contents of a register or a memory location can be incremented or decremented by 1.
- This 16-bit register is used to increment or decrement the content of program counter and stack pointer register by 1.
- Increment or decrement can be performed on any register or a memory location.

INTERRUPT CONTROL

- Consider that a microprocessor is executing the main program. Now whenever the interrupt signal is enabled or requested the microprocessor shifts the control from main program to process the incoming request and after the completion of request, the control goes back to the main program.

- For example an Input/output device may send an interrupt signal to notify that the data is ready for input. The microprocessor temporarily stops the execution of main program and transfers control to I/O device. After collecting the input data the control is transferred back to main program.

SERIAL I/O CONTROL

- The data transferred on to data bus is parallel data, but under some conditions it is advantageous to use serial data transfer at that time this control group is brought into application.
- The input and output of serial data can be carried out using 2 instructions in 8085:
- 1. SID-Serial Input Data
- 2. SOD-Serial Output Data

Time & Control unit

- The control unit provides the necessary timing and control signals to all the operations in the microcomputer.
- It controls and executes the flow of data between the microprocessor, memory and peripherals.
- The control bus is bidirectional and assists the CPU in synchronizing control signals to internal devices and external components.
- This signal permits the CPU to receive or transmit data from main memory.

- It accepts data from instruction decoder and generates micro steps to perform it ,in addition it accepts clock inputs for synchronizing operations.

- This unit consists of an oscillator and controller sequencer which sends control signals needed for internal and external control of data and other units.

- 1. Control Signals: READY, RD(bar), WR(bar), ALE
- 2. Status Signals: S0, S1, IO/M(bar)
- 3. DMA Signals: HOLD, HLDA
- 4. RESET Signals: RESET IN(bar), RESET OUT.

OPERATIONS OF A MICROPROCESSOR

- Microprocessor Initiated operations
- Internal Operations
- Peripheral(Externally Initiated) operations
  - Microprocessor Initiated Operations
- Memory Read
- Memory Write
- I/O Read
- I/O Write

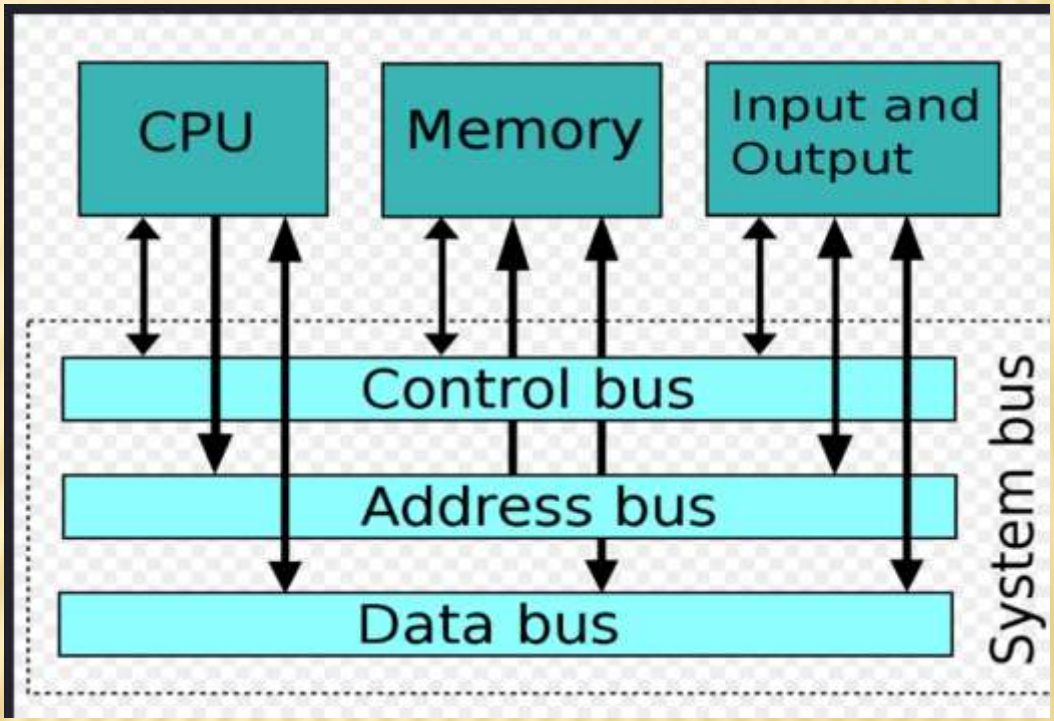
BUS ORGANIZATION

- **What is a Bus?**
- It is a set of pins, wires or signals having common functions as bus.

- **What is system Bus?**
- A system bus is a bundle of wires that are grouped together to serve a single purpose in microprocessor, generally there are three sets of communication lines that are called buses.
- It combines the functions of a data bus to carry information, an address bus to determine where it should be sent, and a control bus to determine its operation.

- They are address bus, the data bus and control bus

System bus (data, address and control bus)



ADDRESS BUS

- The bus over which the microprocessor sends out the address of a memory location or I/O location is called as the address bus.
- In 8085 microprocessor, Address bus is of 16 bits. This means that Microprocessor 8085 can transfer maximum 16 bit address which means it can address 65,536 different memory locations.
- This bus is multiplexed with 8 bit data bus. So, the most significant bits (MSB) of address goes through Address bus and LSB goes through multiplexed data bus.
- In 8085 address bus is 16 bit A0 – A15

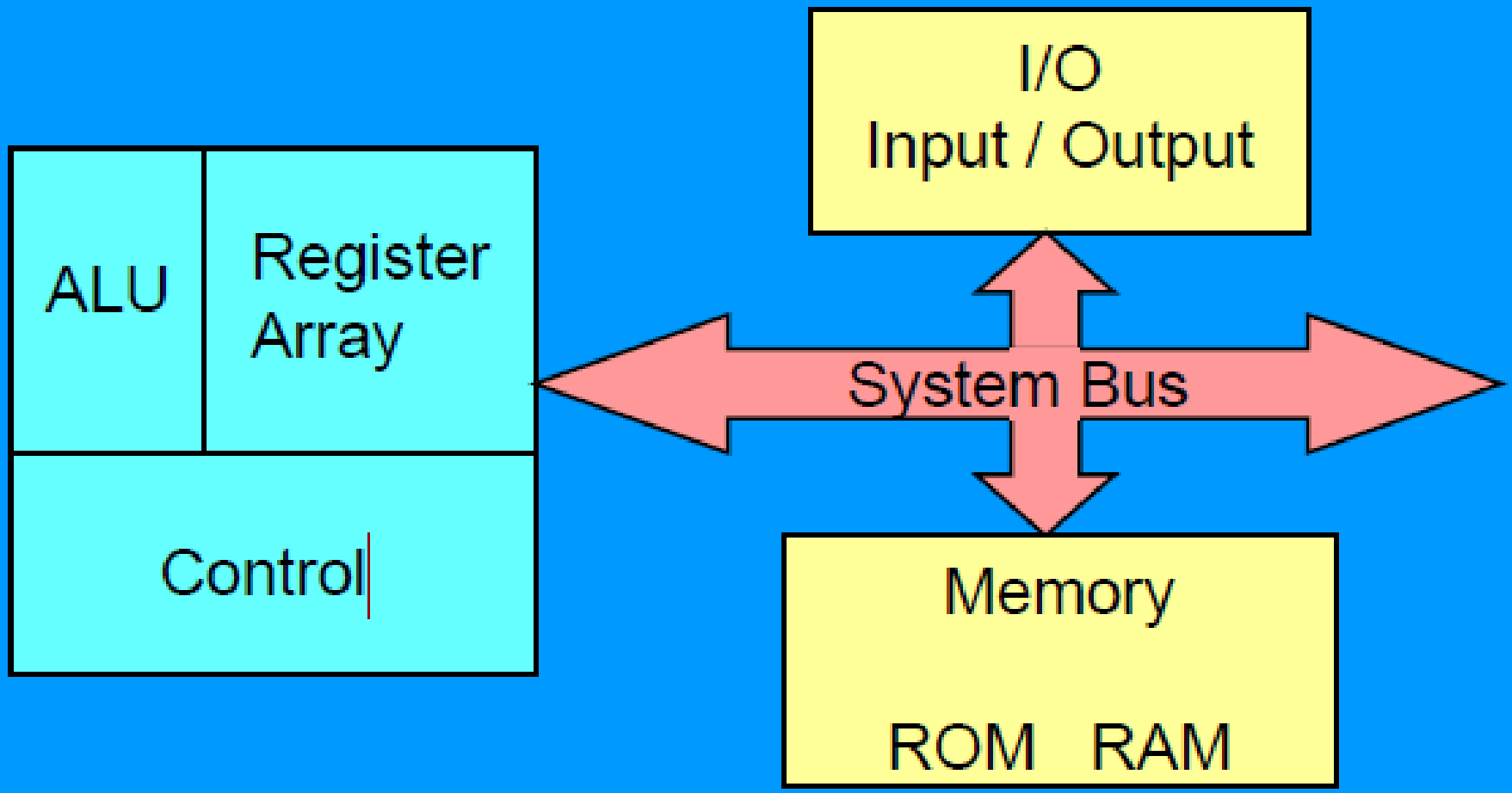
- Data Bus
- It is bi-directional as Microprocessor requires to send or receiver data. The data bus also works as bus in 8 bits long.
- A data bus simply carries data. Internal buses carry information within the processor, while external buses carry data between the processor and the memory. Typically, the same data bus is used for both read/write operations.
- When it is a write operation, the processor will put the data on to the data bus. When it is the read operation, the memory controller will get the data from the specific memory block and put it in to the data bus.
- The data bus in8085 is of 8 parallel lines $D_0 – D_7$

CONTROL BUS

- The control bus is used for sending control signals to the memory and I/O devices. The CPU sends control signal on the control bus to enable the outputs of addressed memory devices or I/O port devices.

- Some of the control bus signals are as follows:
- 1.Memory read
- 2.Memory write
- 3.I/O read
- 4.I/O write.

MICROPROCESSOR SYSTEMS WITH BUS ORGANIZATION



MEMORY

- The user enters its instructions in binary format into the memory.
- The microprocessor then reads these instructions and whatever data is needed from memory, executes the instructions and places the results either in memory or produces it on an output device.

MEMORY ORGANIZATION

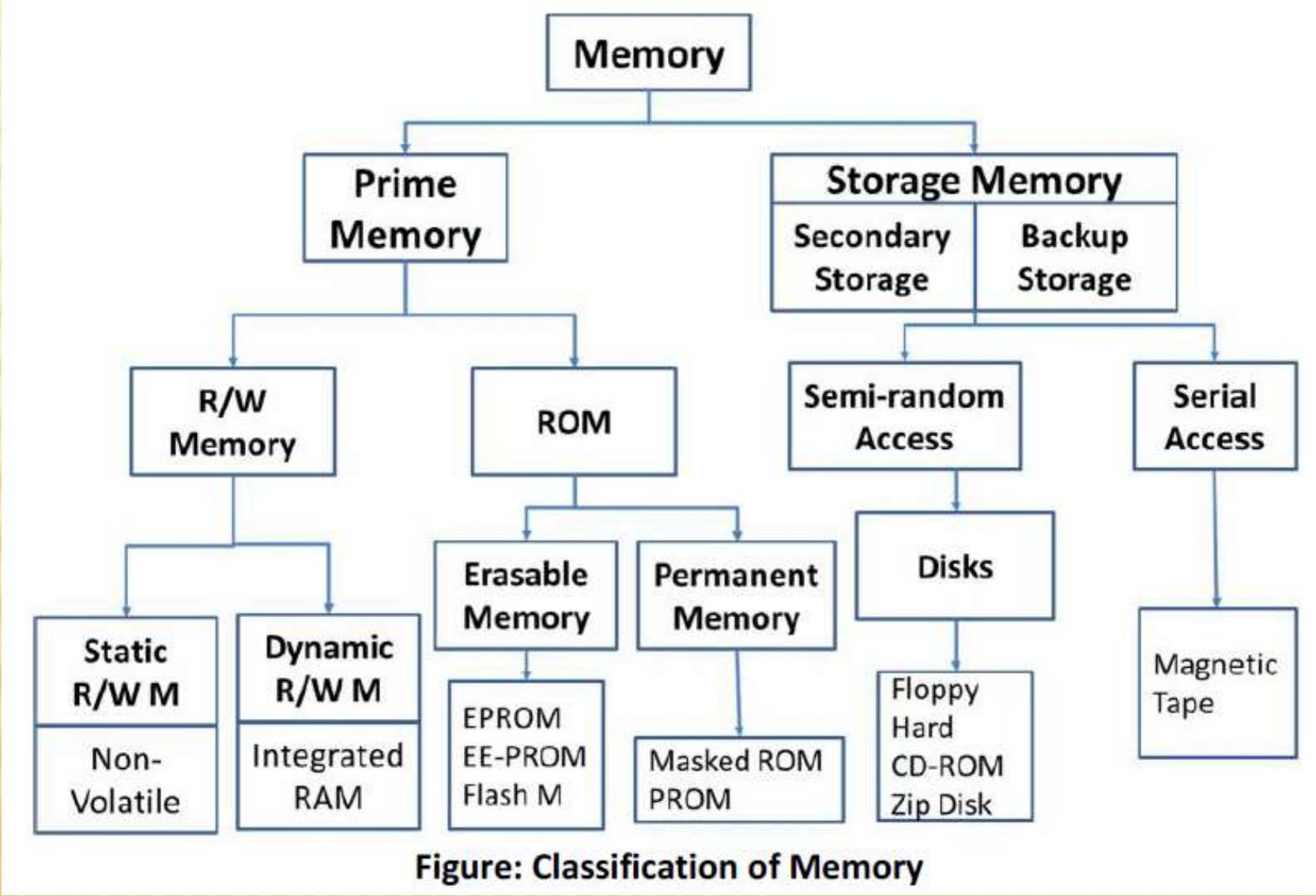


Figure: Classification of Memory

- ✓ Masked ROM: the program or data are permanently installed at the time of manufacturing as per requirement. The data cannot be altered. The process of permanent recording is expensive but economic for large quantities.
- ✓ PROM (Programmable Read Only Memory): The basic function is same as that of masked ROM. but in PROM, we have fuse links. Depending upon the bit pattern, the fuse can be burnt or kept intact.
- ✓ EPROM (Erasable Programmable Read Only Memory): The EPROM is programmable by the user. It uses MOS circuitry to store data. They store 1's and 0's in form of charge. The information stored can be erased by exposing the memory to ultraviolet light which erases the data stored in all memory locations.
- ✓ EEPROM: (Electrically erasable programmable read only memory): This is similar to EPROM except that the erasing is done by electrical signals instead of ultraviolet light. The main advantage is the memory location can be selectively erased and reprogrammed. But the manufacturing process is complex and expensive so do not commonly used.

+ **Classification of RAM memory**

- ✓ SRAM (Static RAM): SRAM consists of the flip-flop; using either transistor or MOS. for each bit we require one flip-flop. Bit status will remain as it is; unless and until you perform next write operation or power supply is switched off.

- Advantages of SRAM:
  - ✓ Fast memory (less access time)
  - ✓ Refreshing circuit is not required.
- Disadvantages of SRAM:
  - ✓ Low package density
  - ✓ Costly
- DRAM (Dynamic RAM): In this type of memory a data is stored in form of charge in capacitors. When data is 1, the capacitor will be charged and if data is 0, the capacitor will not be charged. Because of capacitor leakage currents, the data will not be held by these cells. So the DRAMs require refreshing of memory cells. It is a process in which same data is read and written after a fixed interval.
- Advantages of DRAM:
  - ✓ High package density
  - ✓ Low cost
- Disadvantages of DRAM:
  - ✓ Required refreshing circuit to maintain or refresh charge on the capacitor, every
- after few milliseconds.

INPUT / OUTPUT (I/O)

- MPU communicates with outside word through I/O device.
- There are 2 different methods by which MPU identifies and communicates With I/O devices these methods are:
  a. Direct I/O (Peripheral)
  b. Memory-Mapped I/O
- The methods differ in terms of the
- No. of address lines used in identifying an I/O device.
- Type of control lines used to enable the device.
- Instructions used for data transfer.

INPUT / OUTPUT (I/O)

- Direct I/O (Peripheral):-
- This method uses two instructions (IN & OUT) for data transfer.
- MPU uses 8 address lines to send the address of I/O device (can identify 256 input devices & 256 output devices).
- The (I/P & O/P devices) can be differentiated by control signals I/O Read (IOR) and I/O Write (IOW).
- The steps in communicating with an I/O device are similar to those in communicating with memory and can be summarized as follows:
  - 1 The MPU places an 8-bit device address on address bus then decoded.
  - 2 The MPU sends a control signal (IOR or IOW) to enable the I/O device.
  - 3 Data are placed on the data bus for transfer.

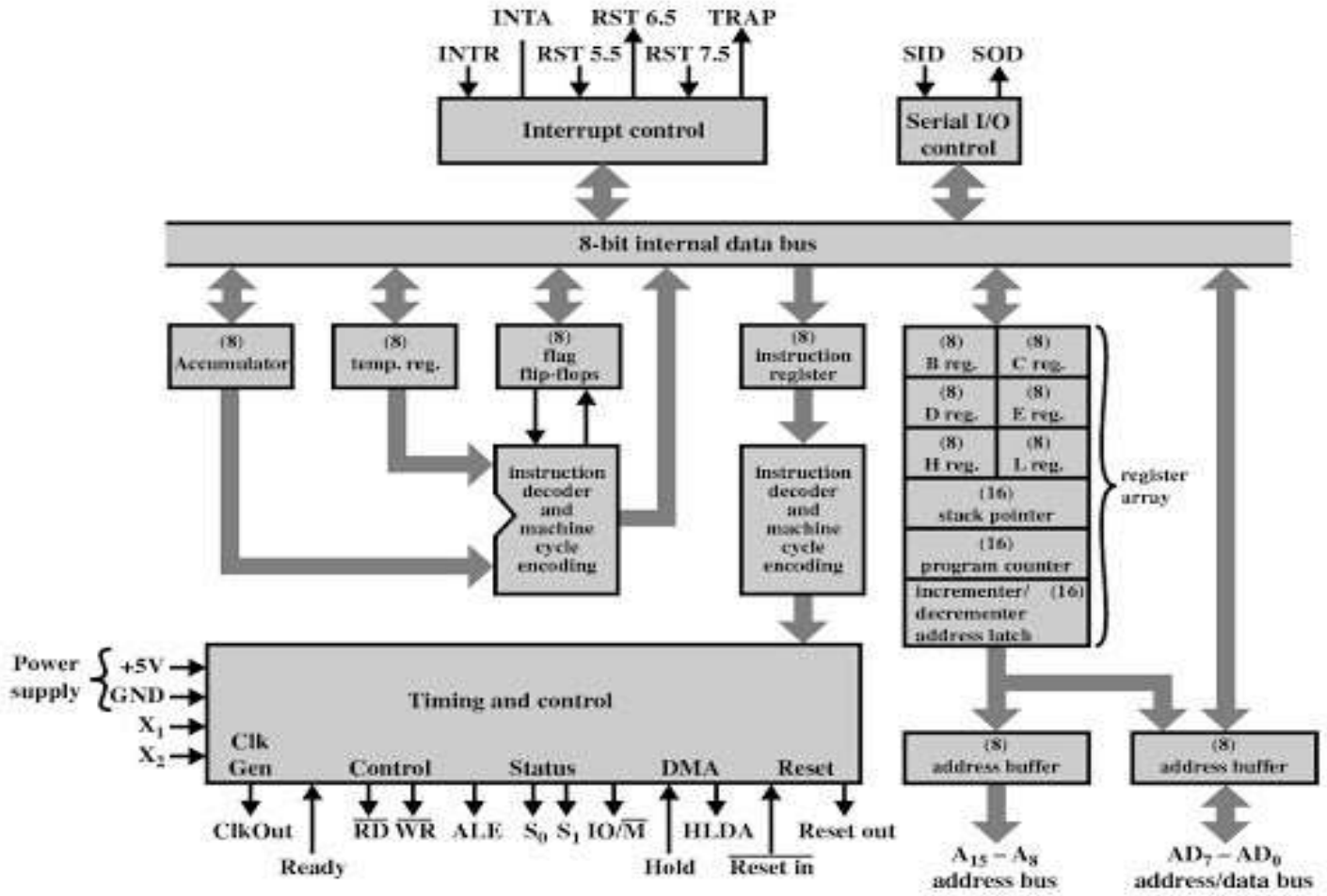
INPUT / OUTPUT (I/O)

- Memory-Mapped I/O:-
- The MPU uses 16 address lines to identify an I/O device.
- This is similar to communicating with a memory location.
- Use the same control signals (MEMR or MEMW) and instructions as those of memory.
- The MPU views these I/O devices as if they were memory locations.
- There are no special I/O instructions.
- It can identify 64k address shared between memory & I/O devices.

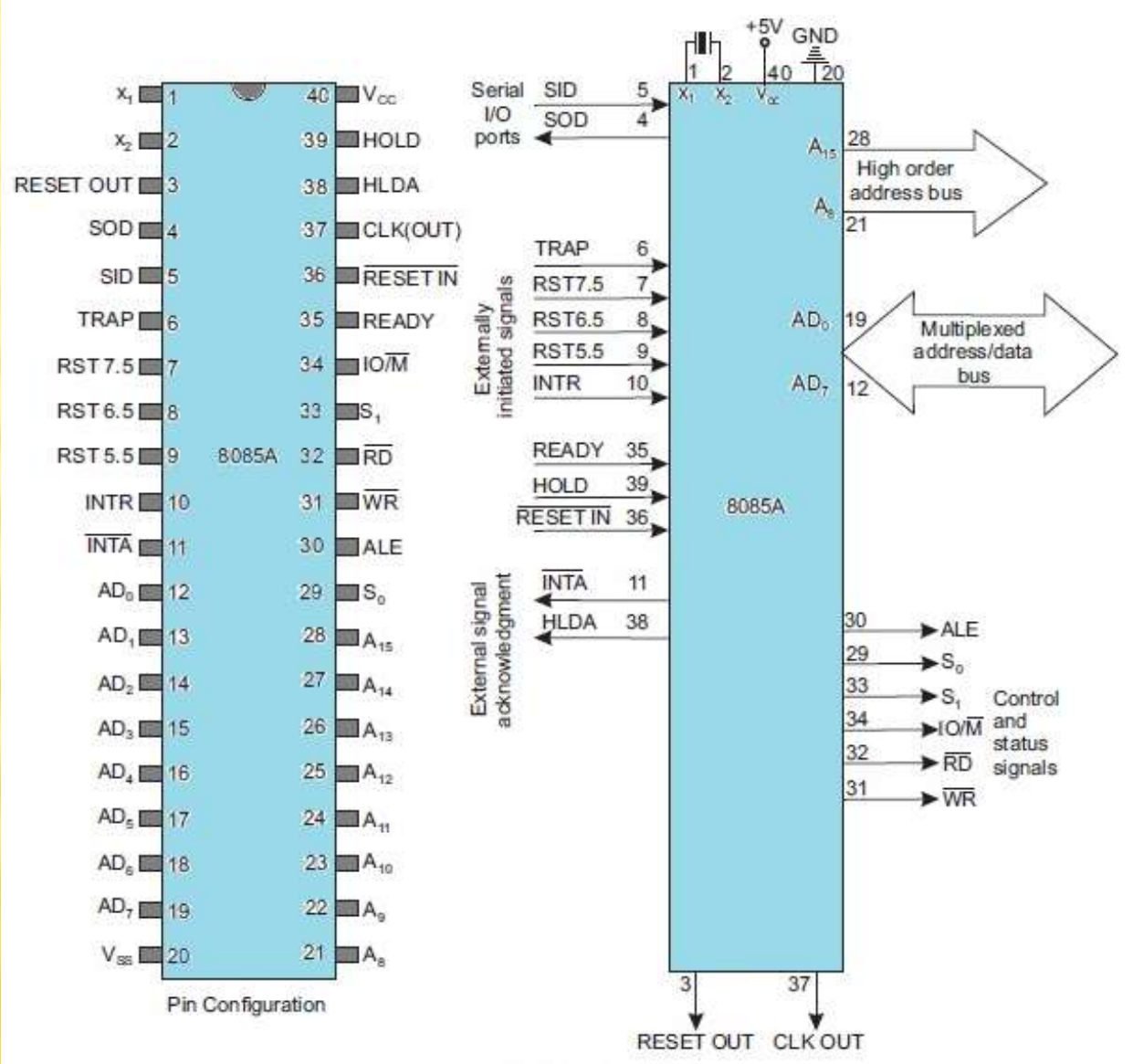
8085 MICROPROCESSOR

- It 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; power supply ground is connected to Vss.
- It operates on clock cycle with 50% duty cycle.
- It has on chip clock generator. This internal clock generator requires tuned circuit like LC, RC or crystal. The internal clock generator divides oscillator frequency by 2 and generates clock signal, which can be used for synchronizing external devices.
- It can operate with a 3 MHz clock frequency. The 8085A-2 version can operate at the maximum frequency of 5 MHz
- It has 16 address lines, hence it can access (216) 64 Kbytes of memory.
- It provides 8 bit I/O addresses to access (28) 256 I/O ports.
- In 8085, the lower 8-bit address bus (A0 – A7) and data bus (D0 – D7) are Multiplexed to reduce number of external pins. But due to this, external hardware (latch) is required to separate address lines and data lines.

FUNCTIONAL BLOCK DIAGRAM OF 8085



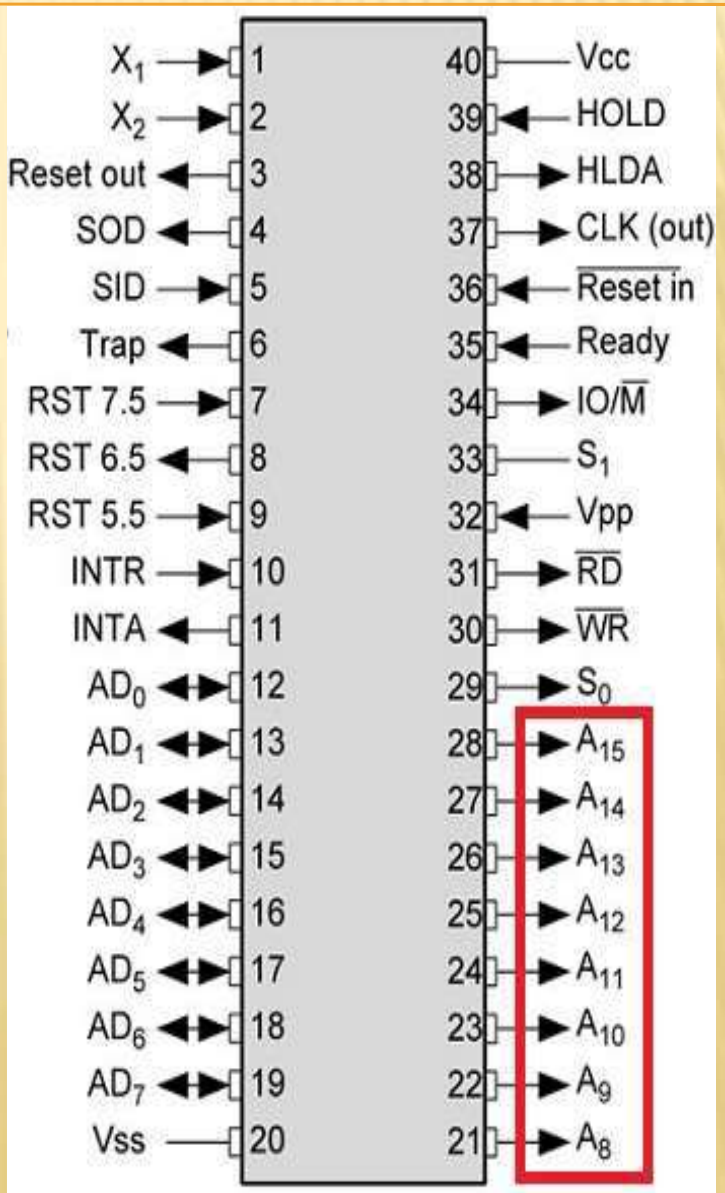
PIN DIAGRAM


X1 1
X2 2
RESET OUT 3
SOD 4
SID 5
TRAP 6
RST 7.5 7
RST 6.5 8
RST 5.5 9
INTR 10
INTA 11
AD0 12
AD1 13
AD2 14
AD3 15
AD4 16
AD5 17
AD6 18
AD7 19
VSS 20
8085A
40 VCC
39 HOLD
38 HLDA
37 CLK(OUT)
36 RESET IN
35 READY
34 IO/M
33 S1
32 RD
31 WR
30 ALE
29 S0
28 A15
27 A14
26 A13
25 A12
24 A11
23 A10
22 A9
21 A8
Pin Configuration
+5V
GND
1
2
40
20
Serial I/O ports
SID 5
SOD 4
A15 28
A8 21
High order address bus
TRAP 6
RST7.5 7
RST6.5 8
RST5.5 9
INTR 10
Externally initiated signals
AD0 19
AD7 12
Multiplexed address/data bus
READY 35
HOLD 39
RESET IN 36
8085A
INTA 11
HLDA 38
External signal acknowledgment
30 ALE
29 S0
33 S1
34 IO/M
32 RD
31 WR
Control and status signals
3
37
RESET OUT
CLK OUT

PIN DIAGRAM...

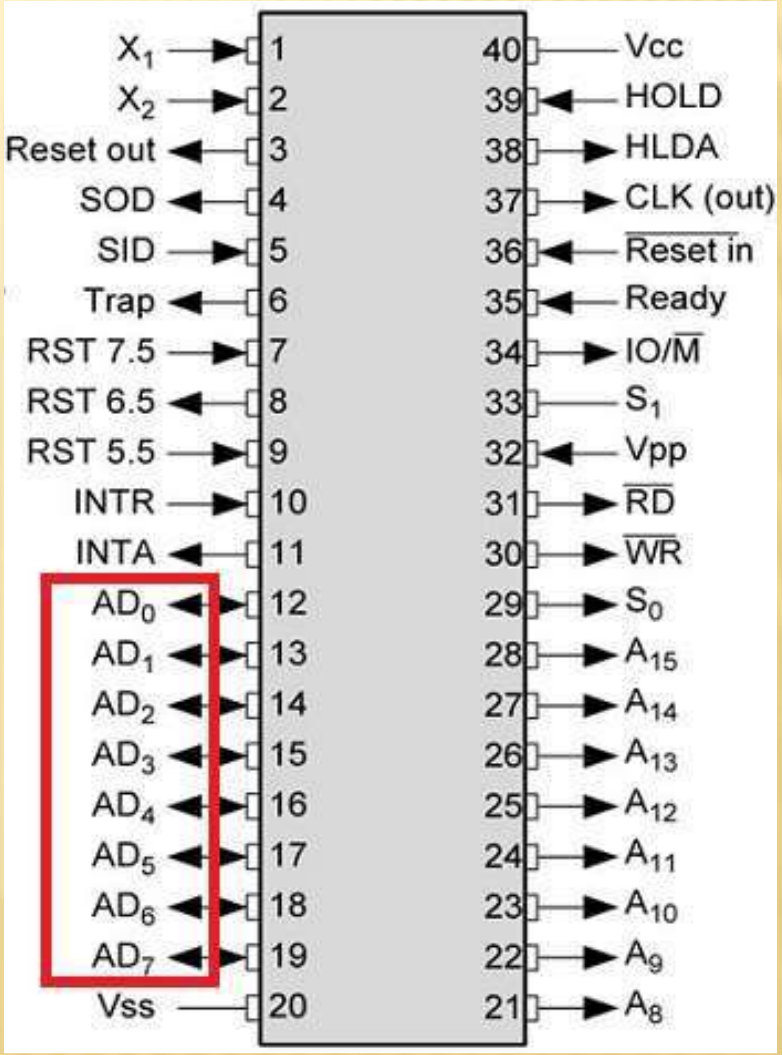
- Address Bus

- These pins carry the higher order of address bus.
- The address is sent from microprocessor to memory.
- A8 – A15. It carries the most significant 8-bit of memory I/O address.



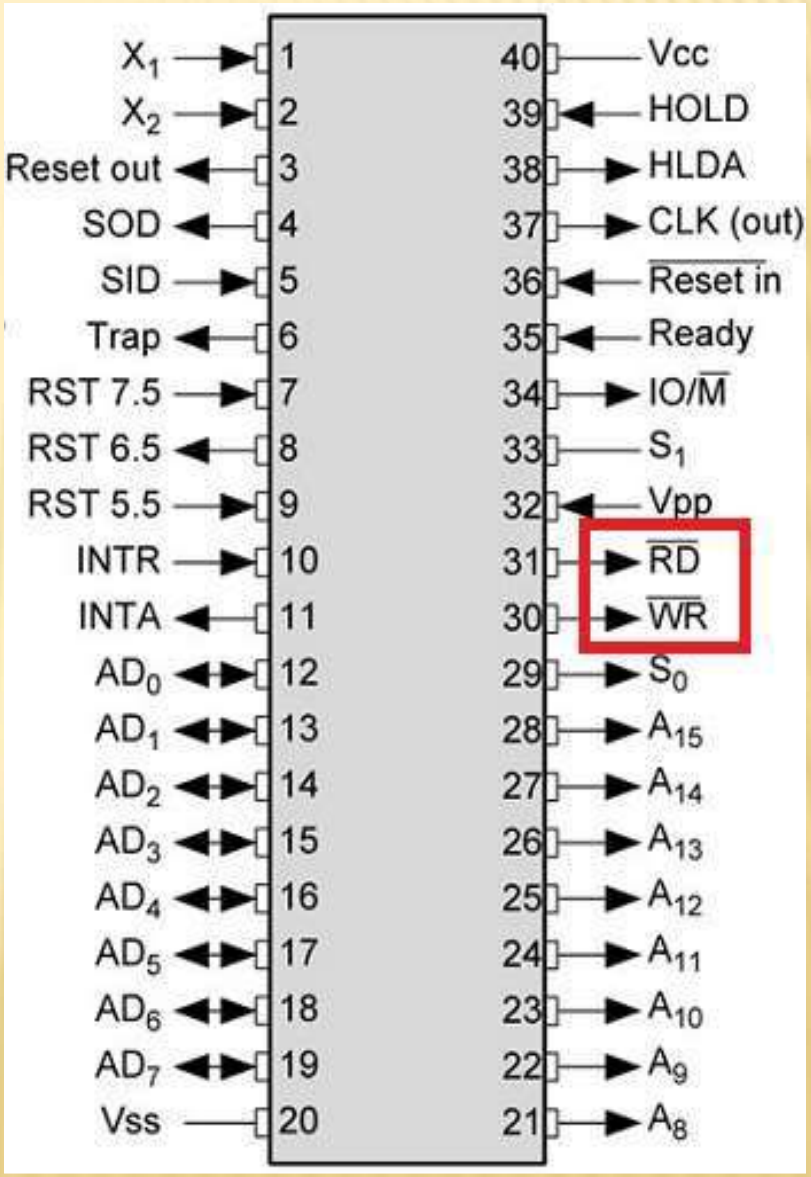
PIN DIAGRAM..

- Data Bus
  - ✓ Data bus is of 8 Bit.
  - ✓ It is used to transfer Data between microprocessor and memory.
  - ✓ AD0 – AD7. It carries the least significant 8-bit address and data bus.



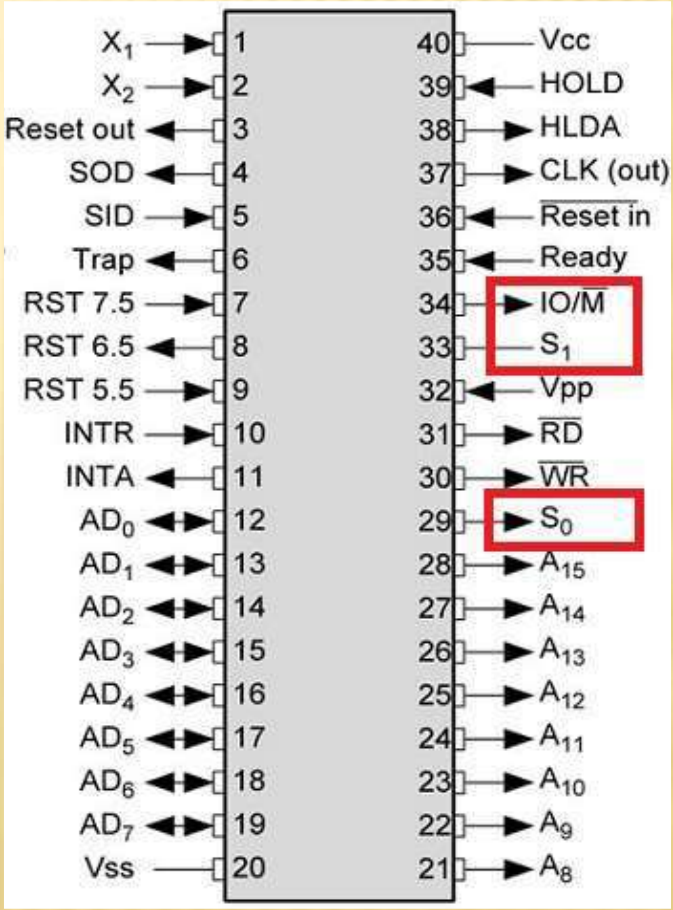
CONTROL SIGNAL

- **RD(Active Low Signal)**
- This signal indicates that the selected IO or memory device is to be read and is ready for accepting data available on the data bus.
- **WR(Active Low Signal)**
- This signal indicates that the data on the data bus is to be written into a selected memory or IO location.



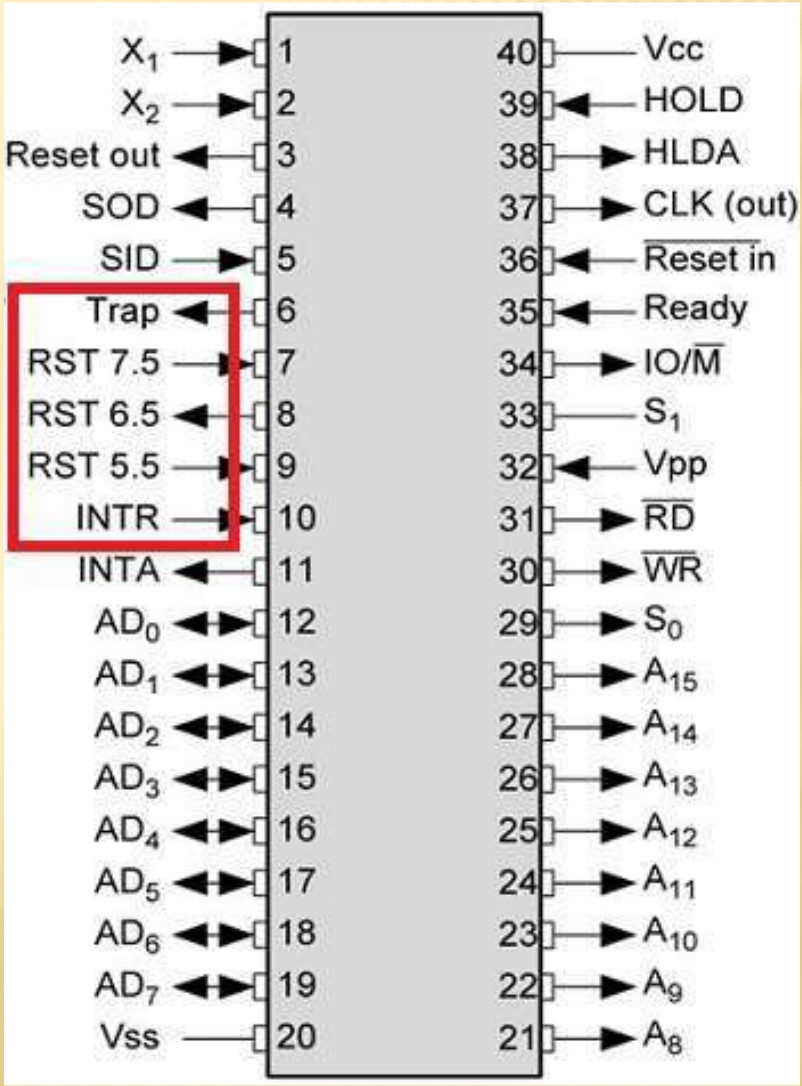
STATUS SIGNALS

- IO/M(Active low)
- This signal is used to differentiate between IO and Memory operations, i.e. when it is high indicates IO operation and when it is low then it indicates memory operation.
- **S0 & S1**
- These signals are used to identify the type of current operation.



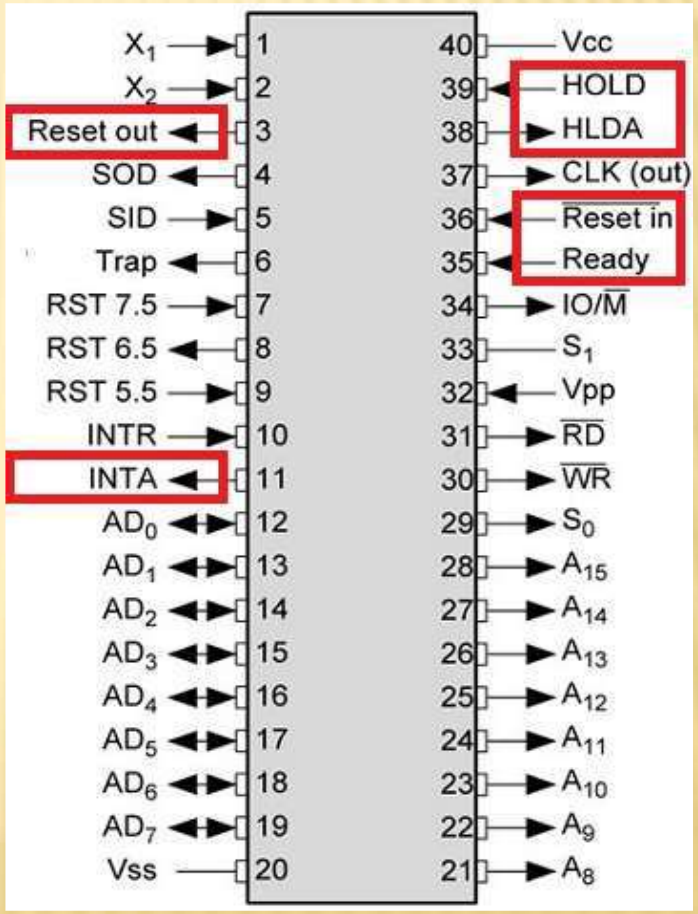
INTERRUPT SIGNALS

- TRAP is usually used for power failure
- and emergency shutoff.
- RST 7.5
- It is a maskable interrupt. It has the
- second highest priority.
- RST 6.5
- It is a maskable interrupt. It has the
- third highest priority.
- RST 5.5
- It is a maskable interrupt. It has the
- fourth highest priority.
- INTR
- It is a general-purpose interrupt. It is a
- maskable interrupt. It has the lowest
- priority.



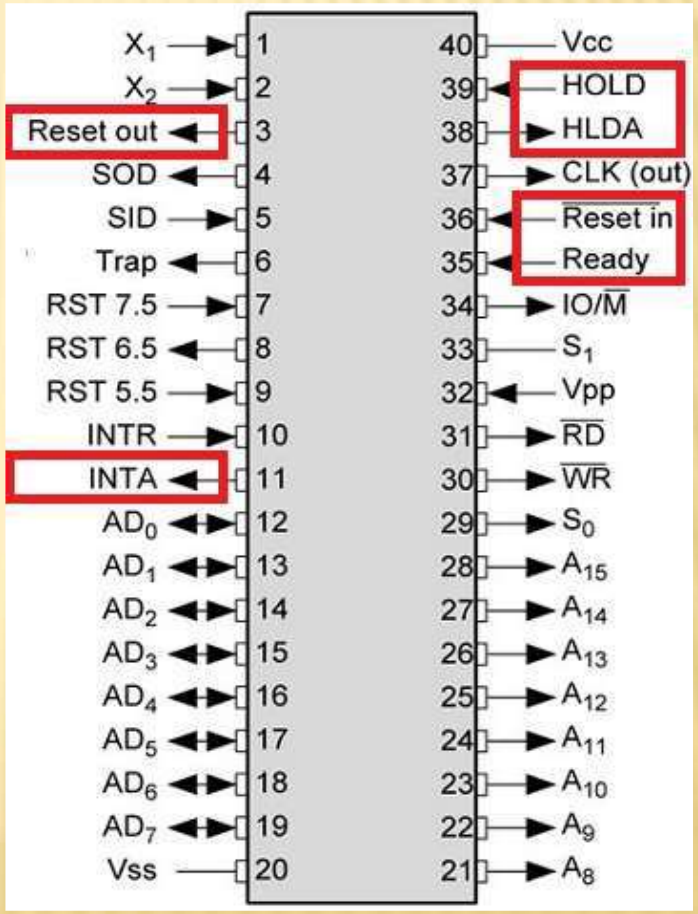
EXTERNALLY INITIATED SIGNALS

- INTA
- It is an interrupt acknowledgment signal.
- RESET IN
- This signal is used to reset the microprocessor by setting the program counter to zero.
- RESET OUT
- This signal is used to reset all the connected devices when the microprocessor is reset.



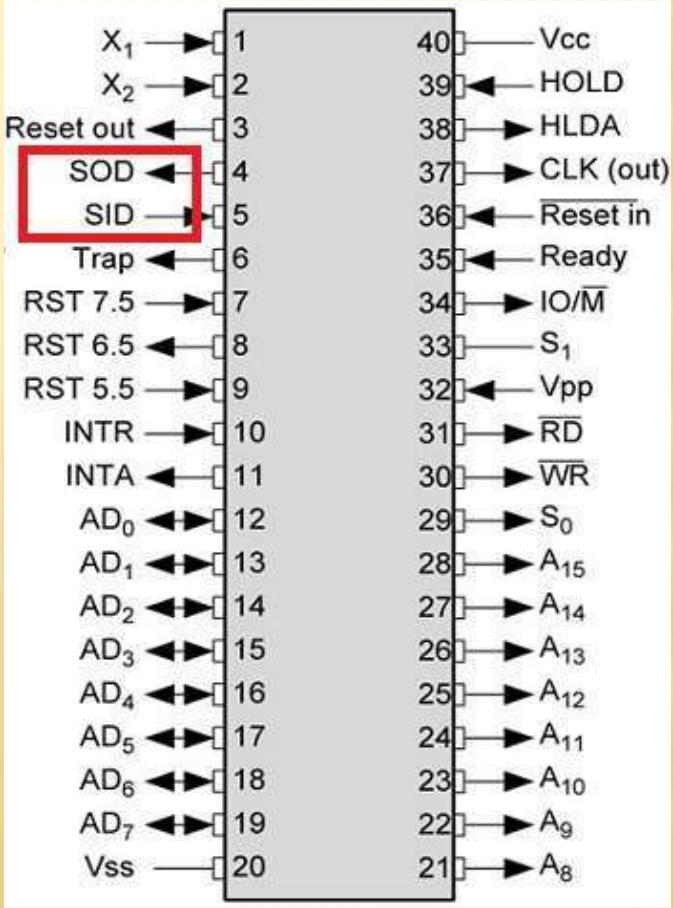
EXTERNALLY INITIATED SIGNALS

- Ready
- This signal indicates that the device is ready to send or receive data. If READY is low, then the CPU has to wait for READY to go high.
- HOLD
- This signal indicates that another master is requesting the use of the address and data buses.
- HLDA
- It indicates that the CPU has received the HOLD request and it will relinquish the bus in the next clock



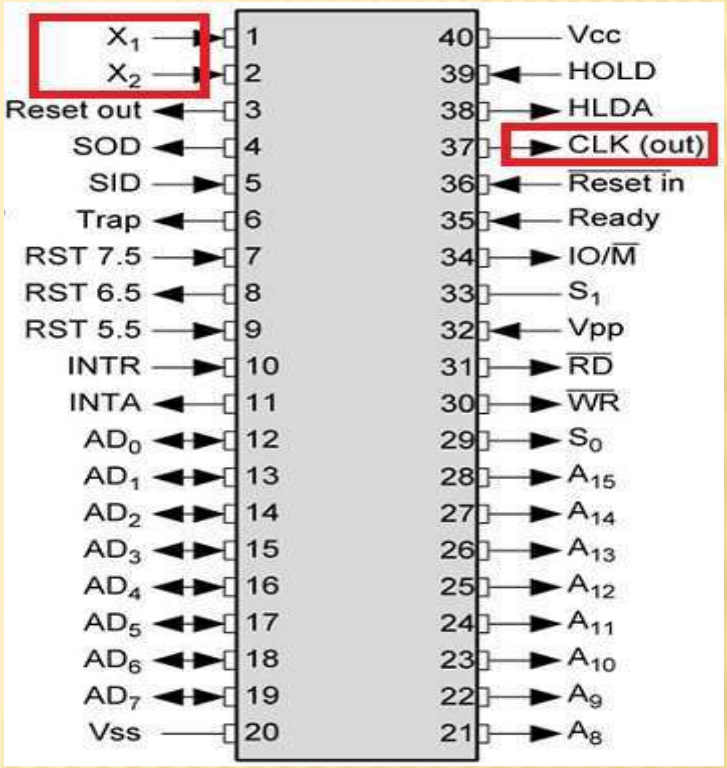
SERIAL I/O SIGNALS

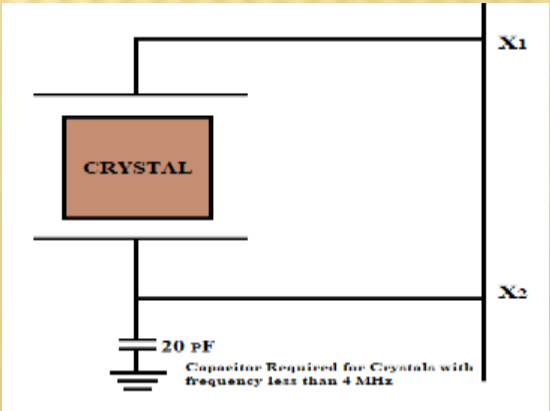
- SOD
- (Serial Output Data line) The output SOD is set/reset as specified by the SIM instruction.
- SID
- (Serial Input Data line) The data on this line is loaded into accumulator whenever a RIM instruction is executed.



CLOCK SIGNALS

- **X1, X2**
- A crystal (RC, LC N/W) is connected at these two pins and is used to set frequency of the internal clock generator. This frequency is internally divided by
- 2. To obtain 3.03 MHz, a clock source of 6.06 MHz must be connect to X1 and X2
- **CLK OUT**
- This signal is used as the system clock for devices connected with the microprocessor.

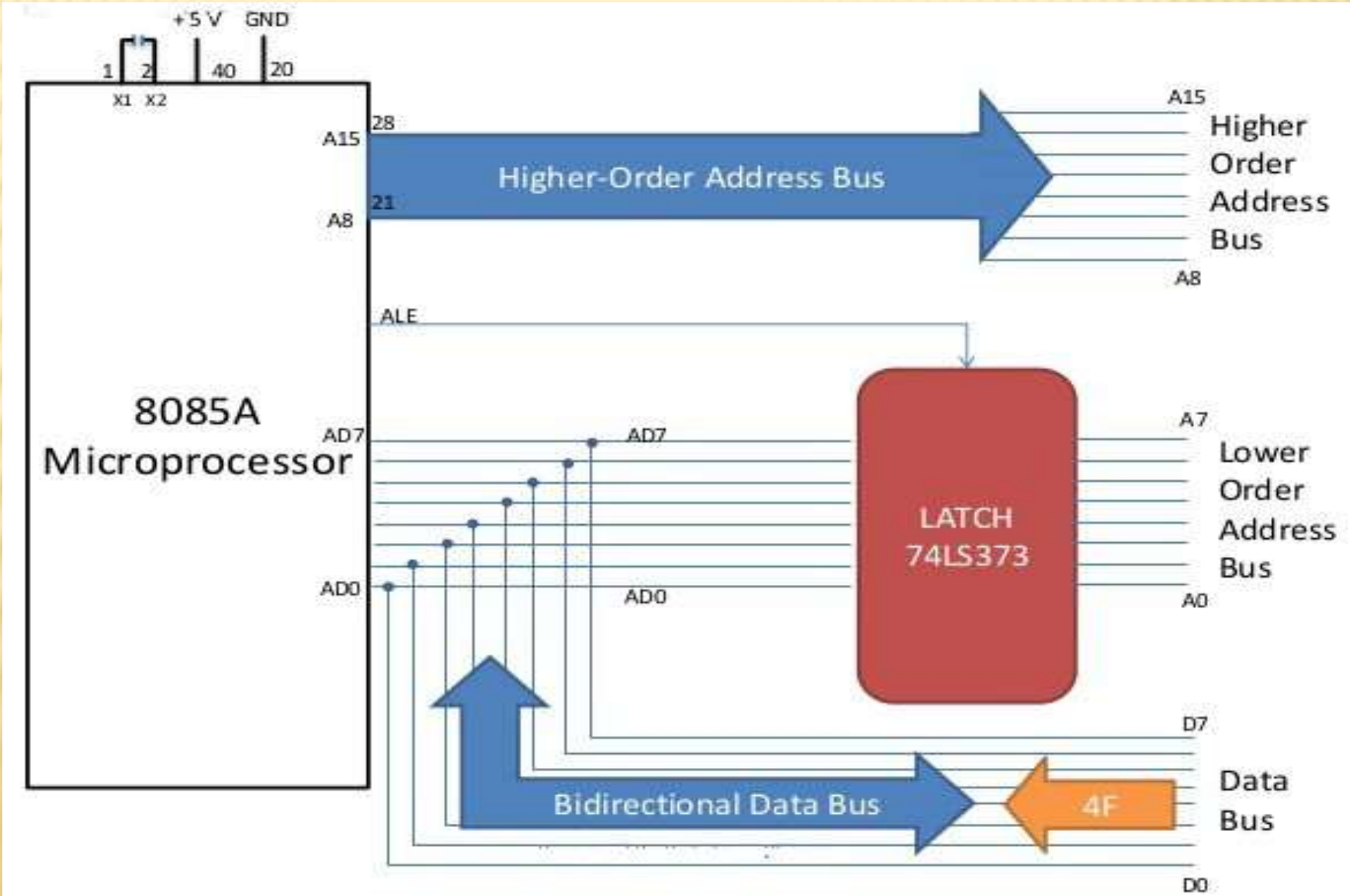




ADDRESS BUS

- The address bus has 8 signal lines A8 – A15 which are unidirectional.
- The other 8 address bits are multiplexed(time shared) with the 8 data bits. So, the bits AD0 –AD7are bi-directional and serve as A0 –A7and D0 –D7at the same time.
- The AD7–AD0 lines are serving a dual purpose and that they need to be demultiplexed to get all the information.
- The high order bits of the address remain on the bus for three clock periods. However, the low order bits remain for only one clock period and they would be lost if they are not saved externally. Also, notice that the low order bits of the address disappear when they are needed most.

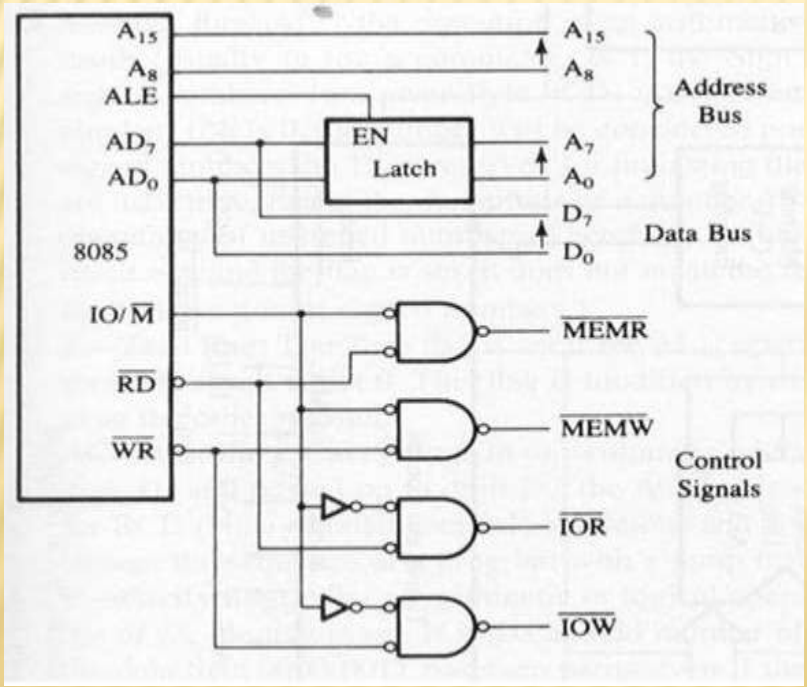
DEMULTIPLEXING OF BUSES



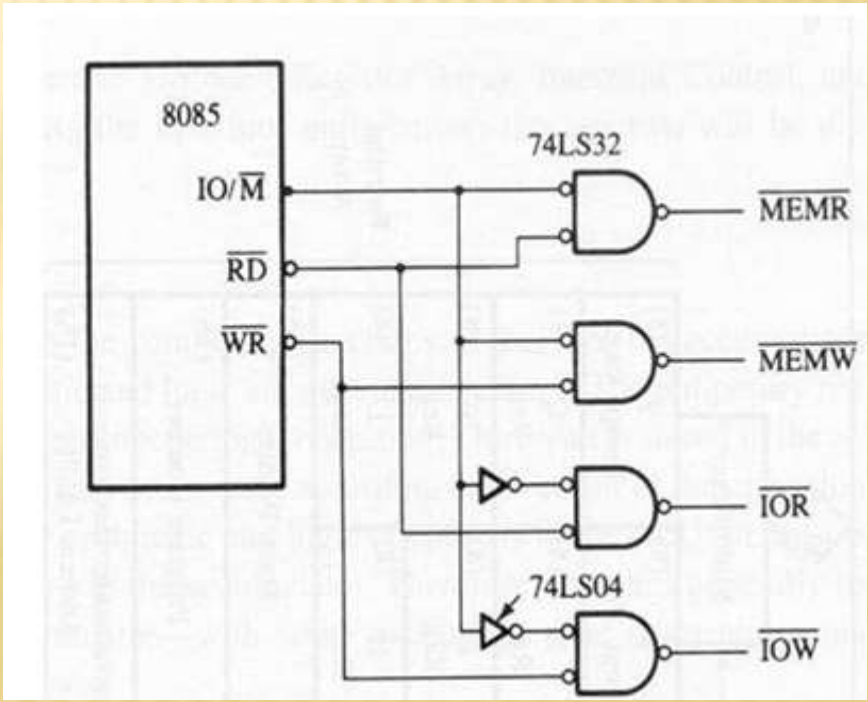
GENERATION OF CONTROL SIGNALS

- Signals are used both for memory and I/O related operations. So four different control signals are generated by combining the signals RD, WR and IO/M.

- 8085 De-multiplexed address and databus with Control Signal

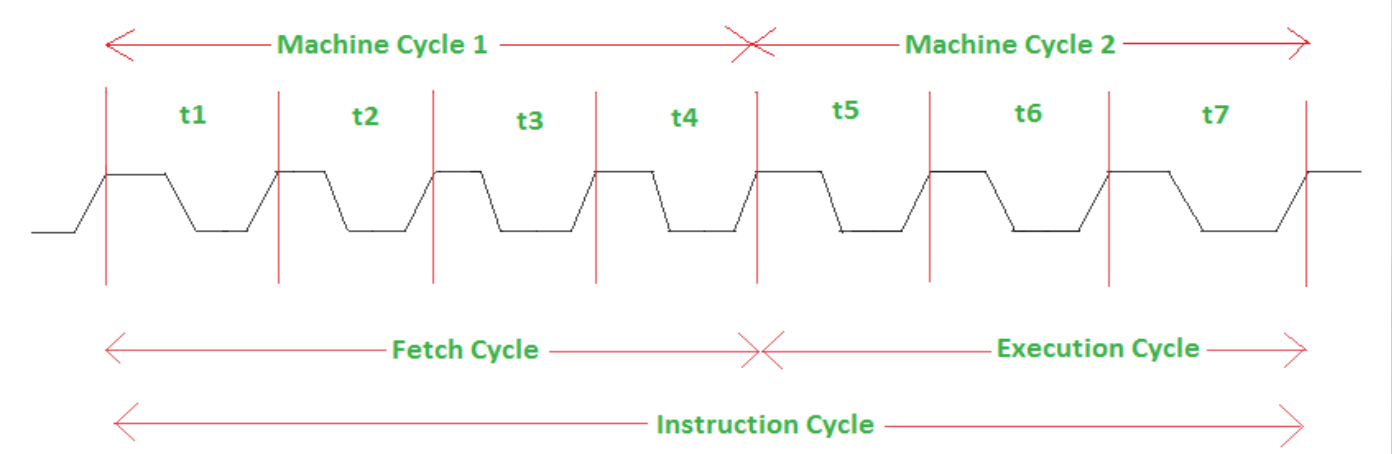


Control Signal Generation



INSTRUCTION CYCLE

- The time required to complete the execution of an instruction.
- In the 8085, an instruction cycle may consist of 1 to 6 machine cycles.
- The function of the microprocessor is divided into two cycle of the instruction
  a. Fetch (b)Execute
- Number of instructions are stored in the memory in sequence.
- In the normal process of operation, the microprocessor fetches (receives or reads) and executes one instruction at a time in the sequence until it executes the halt (HLT) instruction.
- Thus, an instruction cycle is defined as the time required to fetch and execute an instruction. Instruction Cycle (IC) = Fetch cycle (FC) + Execute Cycle (EC)



MACHINE CYCLE

- The time required to complete one operation of accessing memory, I/O, or acknowledging an external request. This cycle may consist of 3 to 6 T-states.
- The 8085 microprocessor has 7 basic machine cycles. They are
- 1. Opcode fetch cycle (4T)
- 2. Memory read cycle or operand fetch(3 T)
- 3. Memory write cycle (3 T)
- 4. I/O read cycle (3 T)
- 5. I/O write cycle (3 T)
- 6. Interrupt Acknowledge
- 7. Bus Idle cycle

OPCODE FETCH CYCLE

- The first step of executing any instruction is the Opcode fetch cycle. In this cycle, the microprocessor brings in the instruction's Opcode from memory.
- To differentiate this machine cycle from the very similar "memory read" cycle, the control & status signals are set as follows
- IO/M=0, s0 and s1 are both 1.
- This machine cycle has four T-states.
- The 8085 uses the first 3 T-states to fetch the opcode.
- T4 is used to decode and execute it.
- It is also possible for an instruction to have 6 T-states in an opcode fetch machine cycle.

WHAT IS EMBEDDED SYSTEM?

- Embedded systems are devices used to control, monitor or assist the operation of equipment, machinery or plant. "Embedded" reflects the fact that they are an integral part of the system
- An embedded system is one that has computer hardware with software embedded in it as one of its components.
- We can define an embedded system as "**A microprocessor based system that does not look like a computer**"
- An embedded system is some combination of computer hardware and software, either fixed in capability or programmable, that is specifically designed for a particular function.
- An embedded system is a special-purpose computer system designed to perform certain dedicated functions. It is usually embedded as part of a complete device including hardware and mechanical parts.

CLASSIFICATION OF EMBEDDED SYSTEM?

- Types of Embedded System
  - Based on Performance and Functional Requirements
    - MOBILE NETWORKED
    - STAND ALONE
    - REAL TIME
  - Based on Performance of Microcontroller
    - Medium Scale
    - Small Scale
    - Sophisticated

BASED ON PERFORMANCE AND FUNCTIONAL REQUIREMENT

1. REAL TIME EMBEDDED SYSTEM

Real-time embedded systems are defined as those systems in which the correctness of the system depends not only on the logical result of computation, but also on the time at which the results are produced.

- Hard real-time systems (e.g., Avionic control).
- Firm real-time systems (e.g., Banking).
- Soft real-time systems (e.g., Video on demand).

BASED ON PERFORMANCE AND FUNCTIONAL REQUIREMENT

2. STAND ALONE EMBEDDED SYSTEM

A standalone device is able to function independently of other hardware. This means it is not integrated into another device

It takes the input from the input ports either analog or digital and processes, calculates and converts the data and gives the resulting data through the connected device-Which either controls, drives and displays the connected devices

For example, a TiVo box that can record television programs , mp3 players are standalone devices

BASED ON PERFORMANCE AND FUNCTIONAL REQUIREMENT

3.NETWORKED EMBEDDED SYSTEM

These types of embedded systems are related to a network to access the resources.

The connected network can be LAN, WAN or the internet. The connection can be any wired or wireless.

This type of embedded system is the fastest growing area in embedded system applications. .

BASED ON PERFORMANCE AND FUNCTIONAL REQUIREMENT

4. MOBILE EMBEDDED SYSTEMS

Mobile embedded systems are used in portable embedded devices like cell phones, mobiles, digital cameras, mp3 players and personal digital assistants, etc.

The basic limitation of these devices is the other resources and limitation of memory

BASED ON PERFORMANCE OF MICROCONTROLLER

- Small Scale Embedded Systems
- These types of embedded systems are designed with a single 8 or 16- bit microcontroller, that may even be activated by a battery.
- For developing embedded software for small scale embedded systems, the main programming tools are an editor, assembler, cross assembler and integrated development environment (IDE).
- Medium Scale Embedded Systems
- These types of embedded systems design with a single or 16 or 32 bit microcontroller, RISCs or DSPs.
- These types of embedded systems have both hardware and software complexities.
- For developing embedded software for medium scale embedded systems, the main programming tools are C, C++, JAVA, Visual C++, RTOS, debugger, source code engineering tool, simulator and IDE.

BASED ON PERFORMANCE OF MICROCONTROLLER

- Sophisticated Embedded Systems
- These types of embedded systems have enormous hardware and software complexities, that may need ASIPs, IPs, PLAs, scalable or configurable processors.
- They are used for cutting-edge applications that need hardware and software Co-design and components which have to assemble in the final system.

APPLICATIONS OF EMBEDDED SYSTEMS

- Industrial machines, automobiles, medical equipment, cameras, household appliances, airplanes, vending machines and toys (as well as the more obvious cellular phone and PDA) are among the myriad possible hosts of an embedded system.

WHY DO WE NEED TO LEARN MICROCONTROLLERS?

- The microprocessor is the core of computer systems.
- Nowadays many communication, digital entertainment, portable devices, are controlled by them.
- A designer should know what types of components he needs, ways to reduce production costs and product reliable.

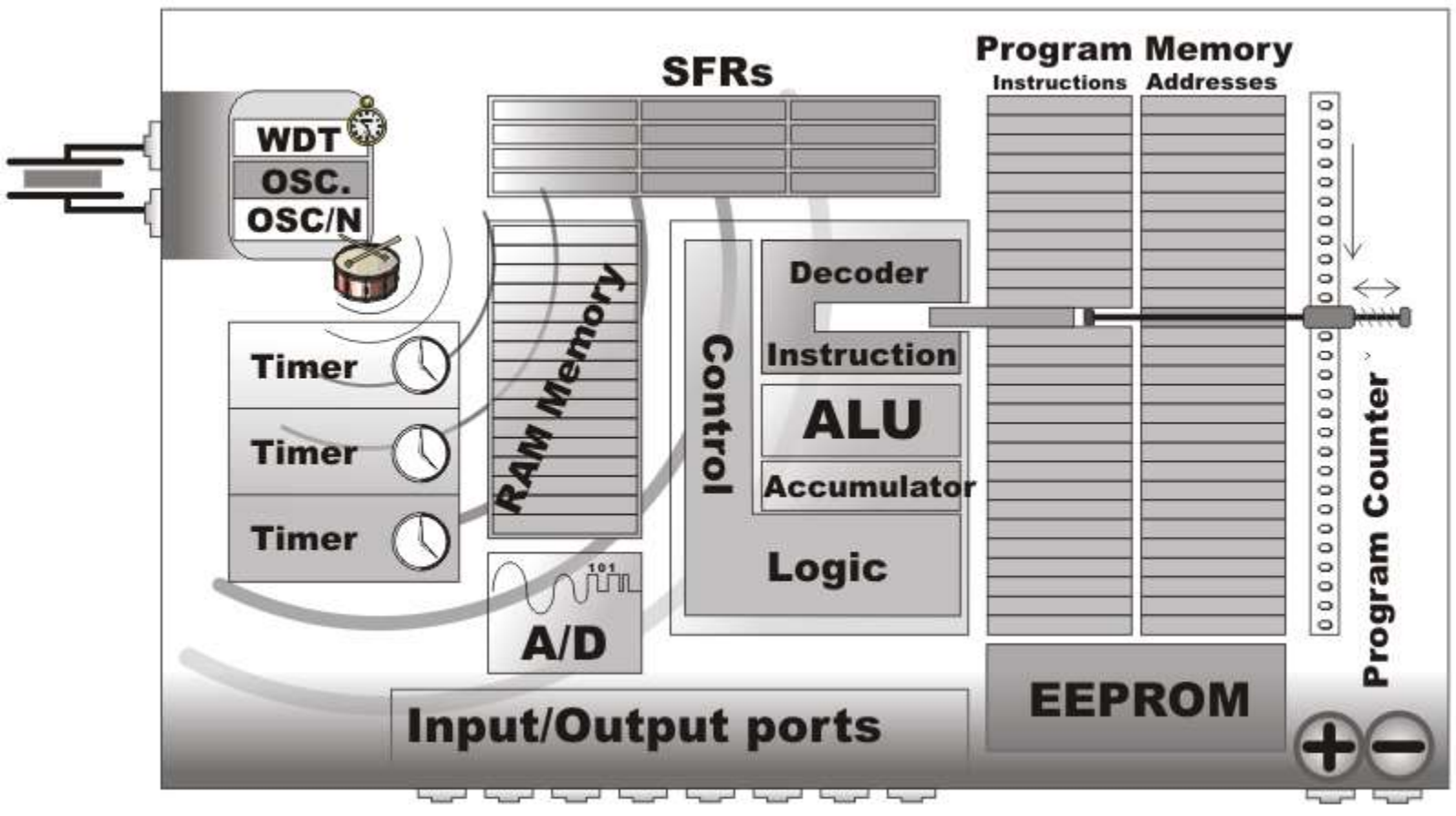
- Two modes of controllers:
- Hardware :Interface to the real world

- Software :order how to deal with inputs

THE NECESSARY TOOLS FOR A CONTROLLER

- CPU: Central Processing Unit
- I/O: Input /Output
- Bus: Address bus & Data bus
- Memory: RAM & ROM
- Timer
- Interrupt
- Serial Port
- Parallel Port

MICROCONTROLLER BLOCK DIAGRAM



THREE CRITERIA IN CHOOSING A MICROCONTROLLER

1. meeting the computing needs of the task efficiently and cost effectively
   - speed, the amount of ROM and RAM, the number of I/O ports and timers, size, packaging, power consumption
   - easy to upgrade
   - cost per unit
2. availability of software development tools
   - assemblers, debuggers, C compilers, emulator, simulator, technical support
3. wide availability and reliable sources of the microcontrollers.

MICROPROCESSOR VS. MICROCONTROLLER

Microprocessor

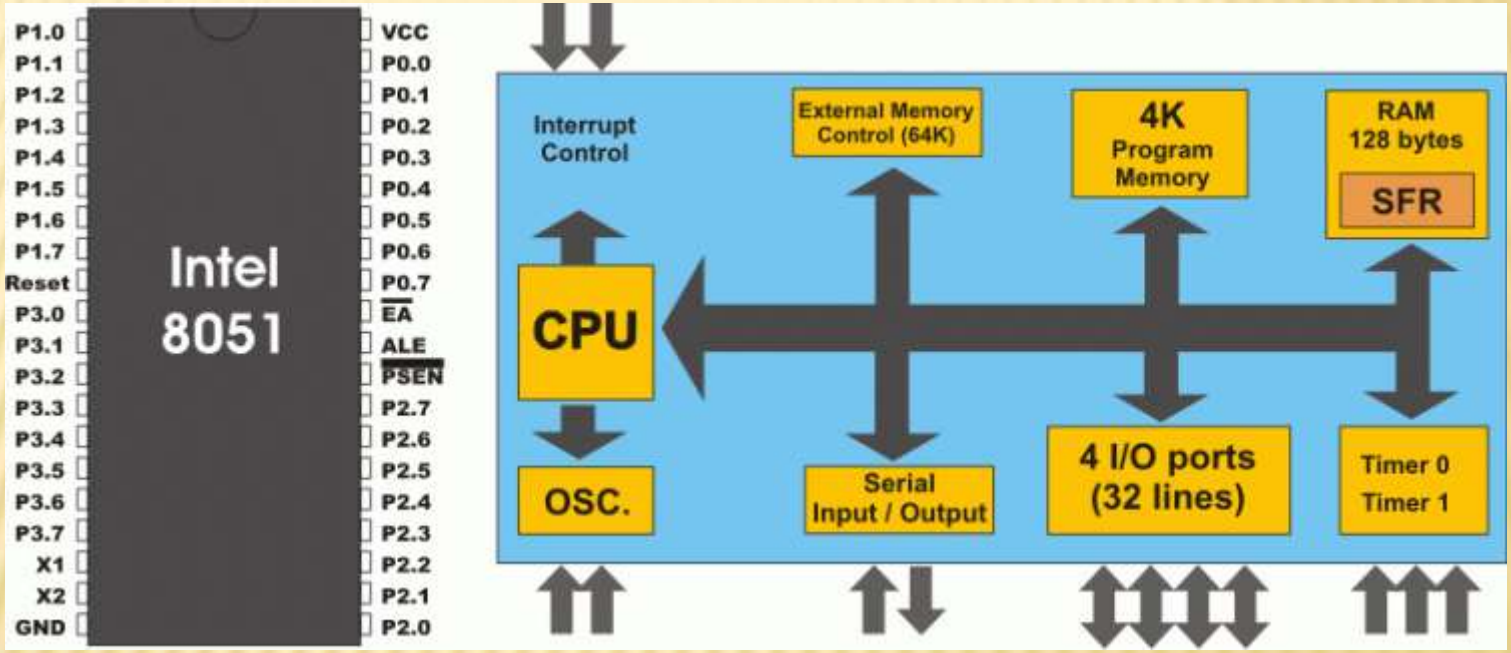
- CPU is stand-alone, RAM, ROM, I/O, timer are separate
- designer can decide on the amount of ROM, RAM and I/O ports.
- Expansive
- general-purpose

- **Microcontroller**
- **CPU, RAM, ROM, I/O and timer are all on a single chip**
- **fix amount of on-chip ROM, RAM, I/O ports**
- **for applications in which cost, power and space are critical**
- **single-purpose**

COMPARISON OF THE 8051 FAMILY MEMBERS

Feature	8051	8052
ROM (program space in bytes)	4K	8K
RAM (bytes)	128	256
Timers	2	3
I/O pins	32	32
Serial port	1	1
Interrupt sources	6	8

8051 ARCHITECTURE



8051 ARCHITECTURE