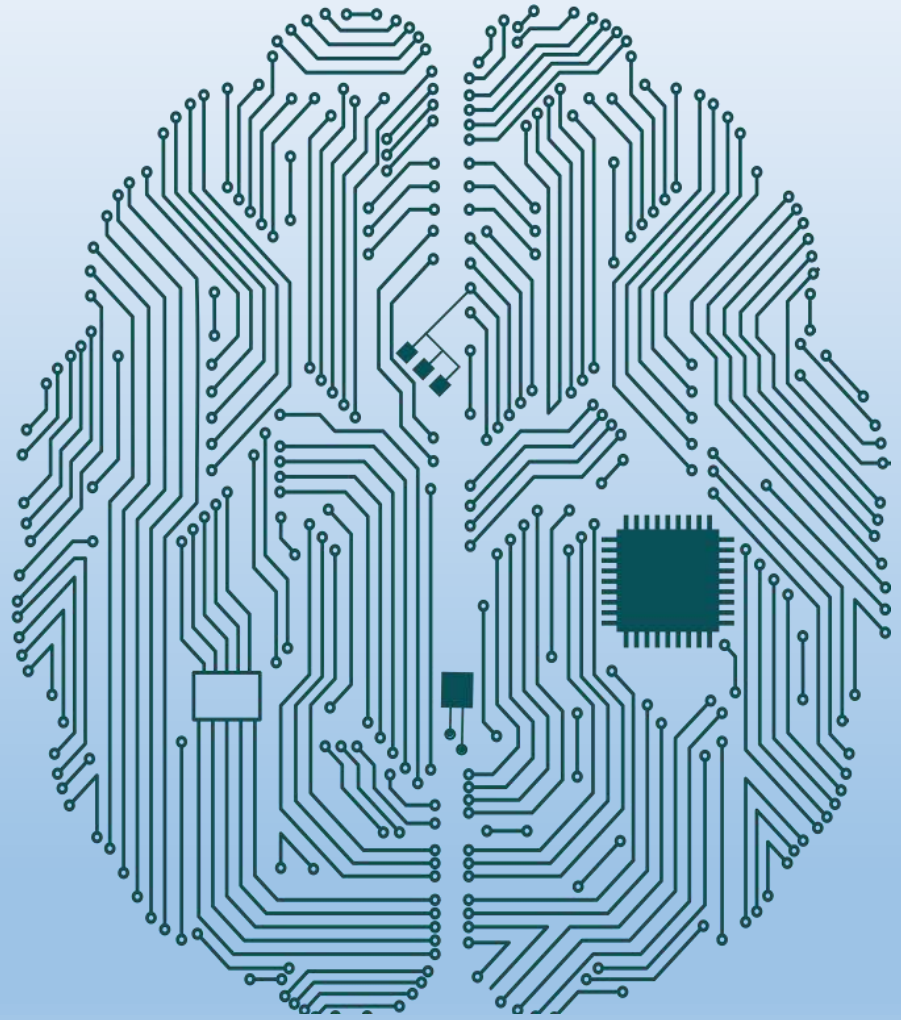


# 8085 Microprocessor

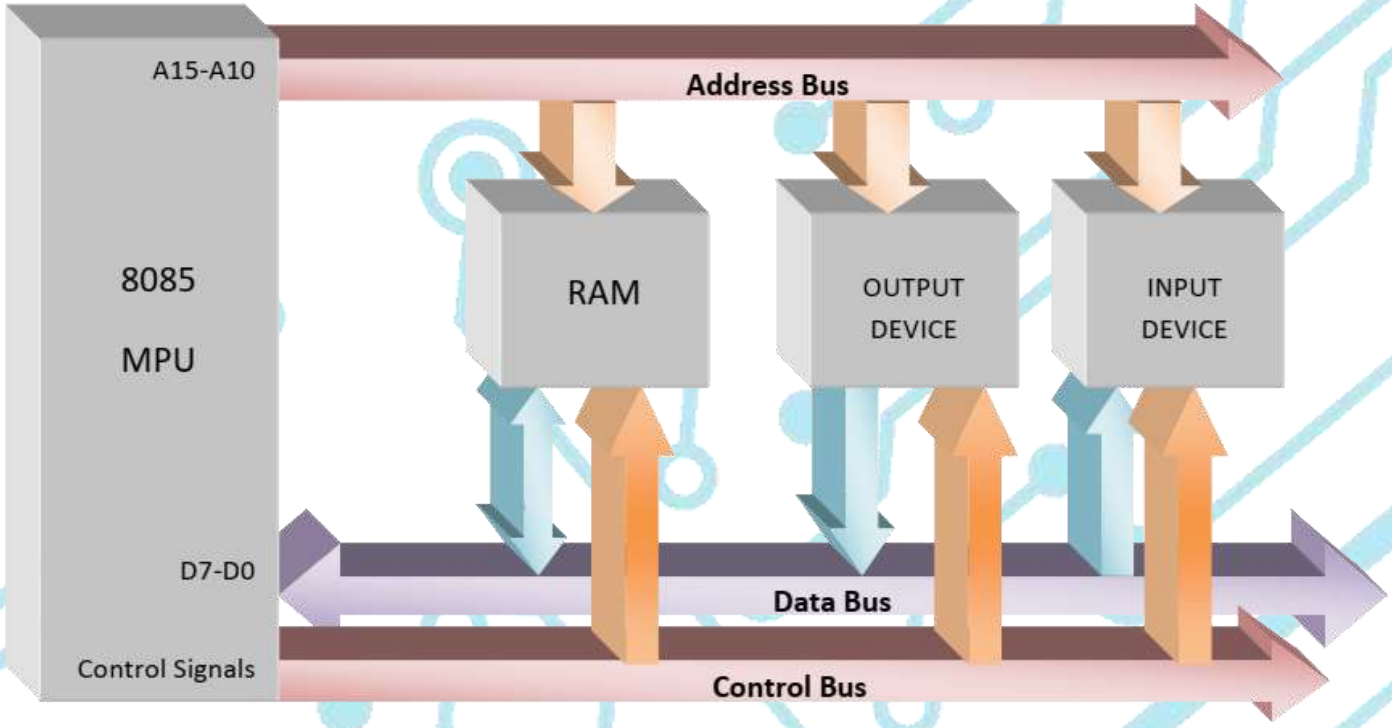
Unit - II

# Prerequisite

- Bus Organization
  - Address Bus
  - Data Bus
  - Control Bus

# Bus Organisation

- Bus: The medium that helps transmit information from one end to other
  - It is collection of wires, which transmit binary numbers, one bit per wire.
- Address Bus:
  - Carries the Address from Microprocessor to IO and Memory Devices
- Data Bus:
  - Transmit Data Between UP and IO –Memory Devices
- Control Bus:
  - Carries the control signals that helps system perform the IO and Memory operations.



# Address Bus

- Carries the Address of the data to be sent or fetch from.
- It's size indicates the size of primary memory it can address.
- Size of Primary memory $= 2^{Size\ of\ Address\ Bus}$

$2^2$ =

| | |
|---|---|
| 00 | 01 |
| 10 | 11 |

00 to 11
0 to 3

$2^3$ =

| | |
|---|---|
| 000 | 001 |
| 010 | 011 |
| 100 | 101 |
| 110 | 111 |

000 to 111
0 to 7

$2^4$ =

| | | | |
|---|---|---|---|
| 0000 | 0001 | 0010 | 0011 |
| 0100 | 0101 | 0110 | 0111 |
| 1000 | 1001 | 1010 | 1011 |
| 1100 | 1101 | 1110 | 1111 |

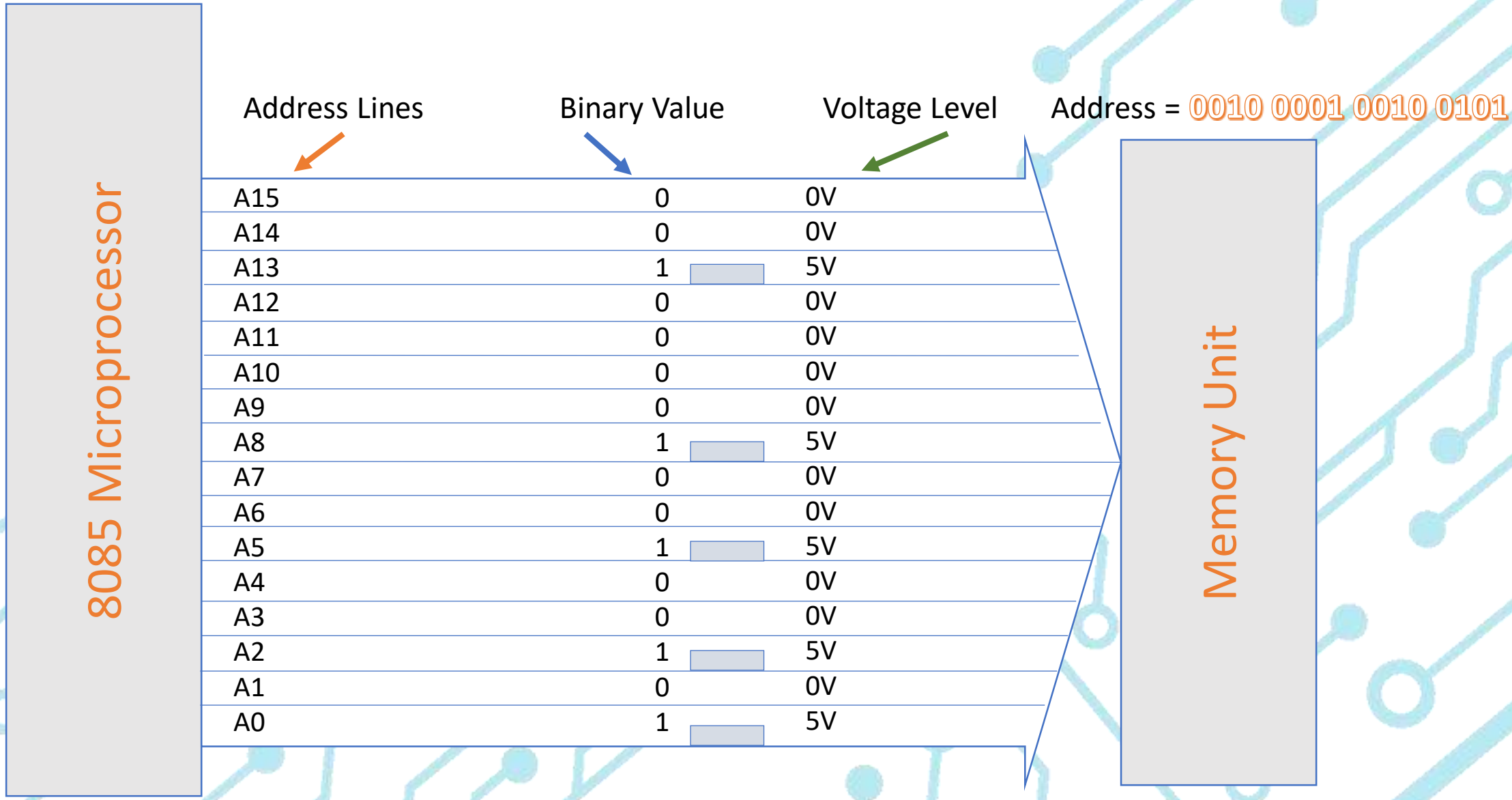
0000 to 1111
0 to 15

- In 8085, the size of Address bus is 16 bits.
- S= $2^{16} = 65536\ memory\ locations$.

**0000 0000 0000 0000 to 1111 1111 1111 1111**
**0 to 65535**

**Address Range**

# Address Bus


8085 Microprocessor
Address Lines
Binary Value
Voltage Level
Address = 0010 0001 0010 0101
A15 0 0V
A14 0 0V
A13 1 5V
A12 0 0V
A11 0 0V
A10 0 0V
A9 0 0V
A8 1 5V
A7 0 0V
A6 0 0V
A5 1 5V
A4 0 0V
A3 0 0V
A2 1 5V
A1 0 0V
A0 1 5V
Memory Unit

# Data Bus

- Data bus carries the data from transmitter to the receiver via medium.
- Used to carry Data parallelly in binary form.
- It's size indicate the largest value it can transmit.
- Largest Data value = $(2^{Size\ of\ Data\ Bus} - 1)$

1 bit data bus: $2^1$=2-1 = 1

| 0 | 1 |
|---|---|

2 bit data bus: $2^2$=4-1 = 3


3 bit data bus: $2^3$=8-1 = 7

| 000 | 001 |
|---|---|
| 010 | 011 |
| 100 | 101 |
| 110 | 111 |

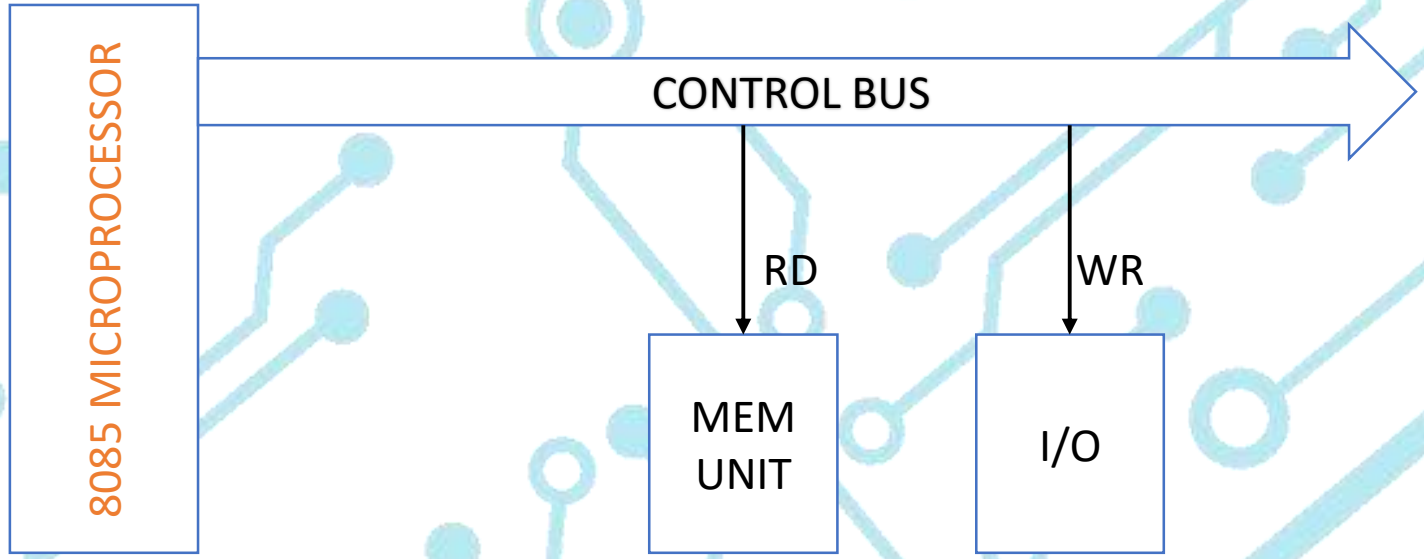
8 bit data bus: $2^8$=256-1 = 255

| 0000 0000 | 0 |
|---|---|
| 0000 0001 | 1 |
| 0000 0010 | 2 |
| ……. | …. |
| 1111 1111 | 255 |

With 8 bit data bus 255 is the largest number that can be transferred

# Control Bus

- Control bus is used to carry the control signals.
- The length of control bus is not fixed. It may vary as per the interfacing of the microprocessor.
- The control bus transmits control signals from microprocessor to other I/O or Memory devices or transmits signals from I/O or memory devices to microprocessor.
- No fix size of control bus.



# Topics to be covered...

- 8085 Pin Diagram & Pin Functions
- 8085 Microprocessor Architecture
- 8085 General Purpose Register
- 8085 Flag Register
- 8085 Instruction Execution
  - Fetch
  - Decode
  - Execute

# 8085 Microprocessor

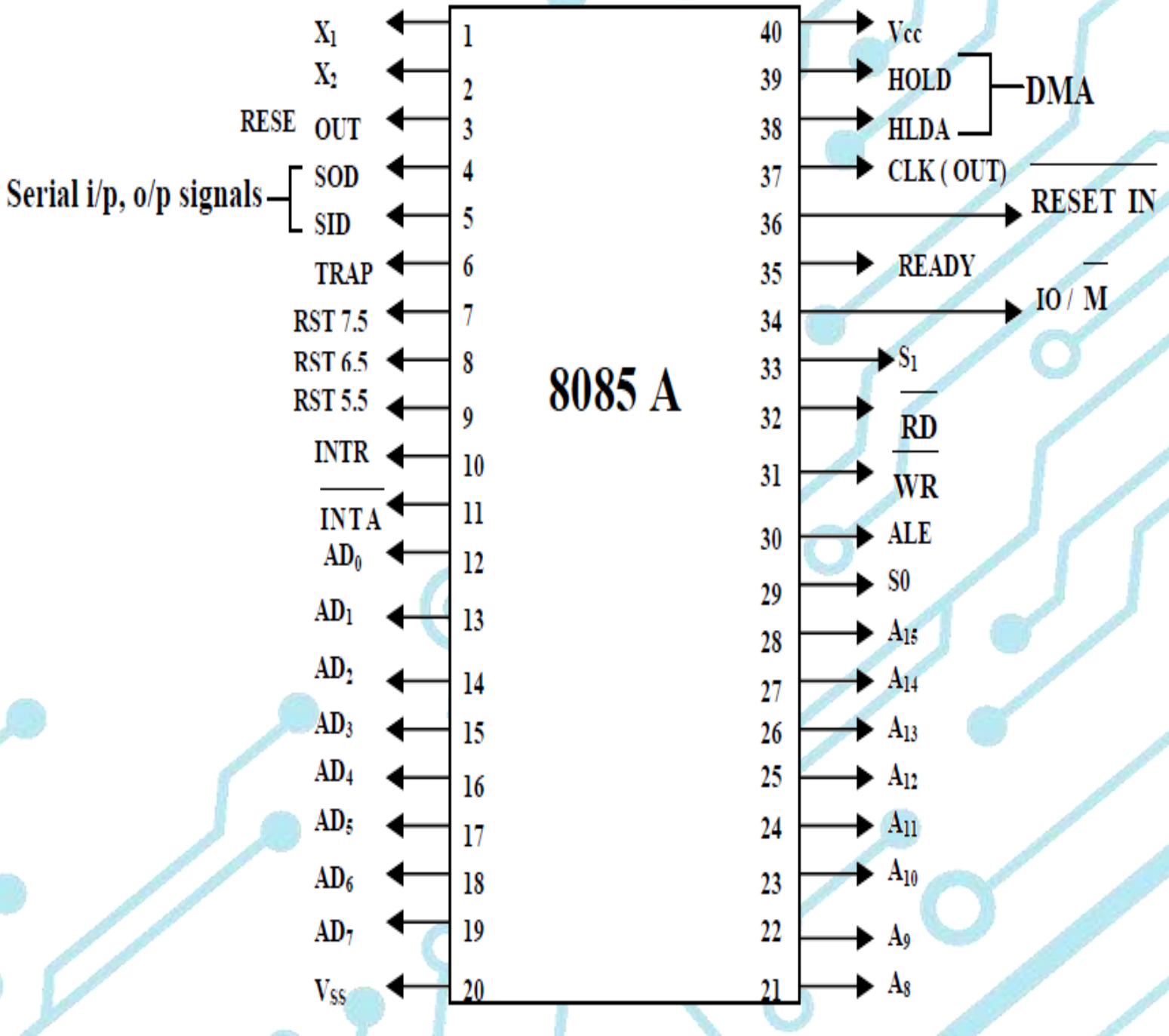
- The 8085 microprocessor is an 8-bit processor developed by Intel in 1976.
- It was a popular choice for early computers and embedded systems due to its relatively simple architecture and functionality.
- Key characteristics include an 8-bit data bus, a 16-bit address bus, and a maximum clock frequency of 3 MHz. This allows it to address up to 64KB of memory.

# Features of 8085 Microprocessor

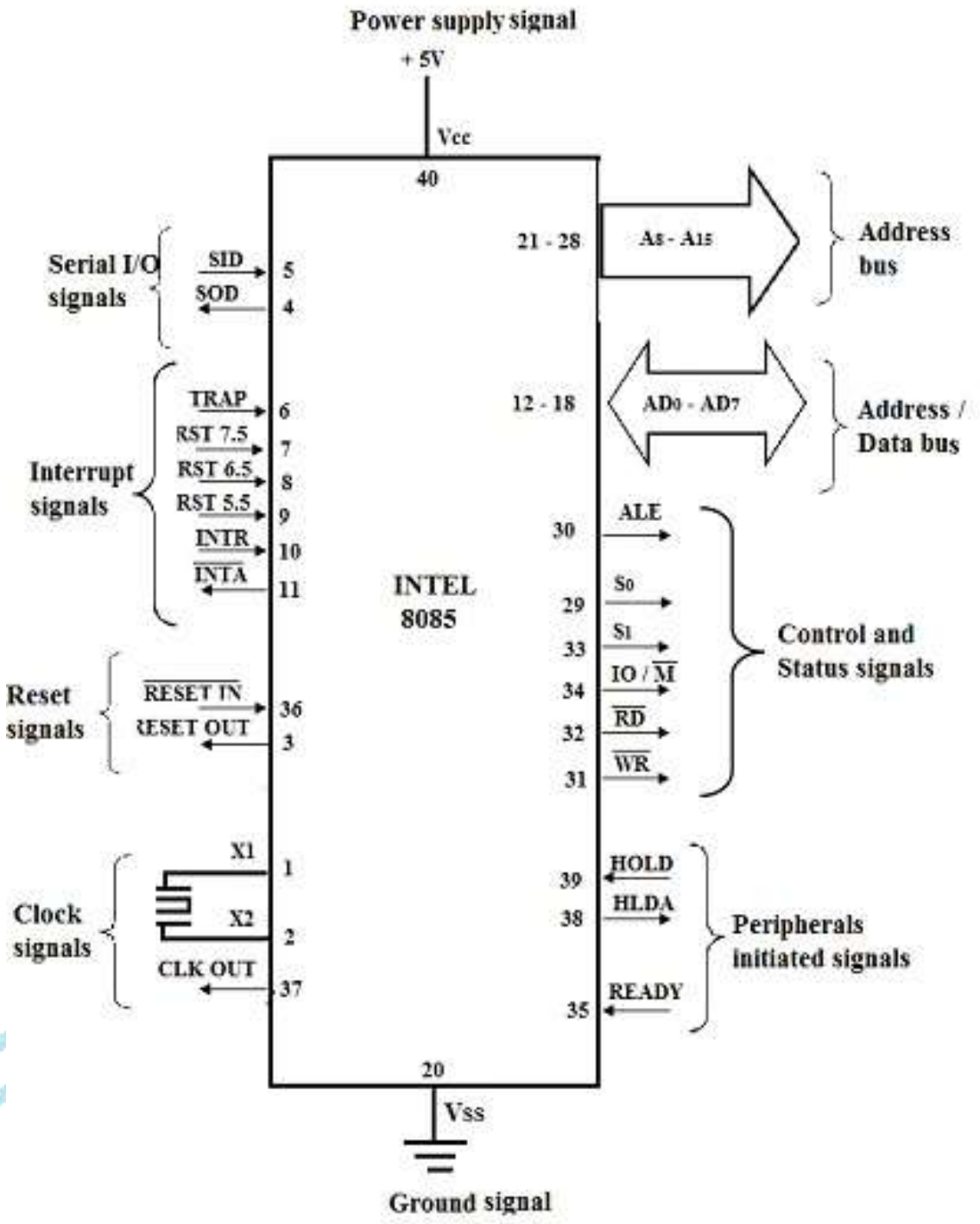
- **8-bit Architecture:**
- The 8085 processes data in 8-bit chunks, meaning it can handle 8 bits of data simultaneously.
- **16-bit Address Bus:**
- The 16-bit address bus allows the processor to access 2^16 (65,536) memory locations, which translates to 64KB of memory.
- **Clock Frequency:**
- The 8085 operates at a maximum clock frequency of 3 MHz, which determines the speed at which instructions are executed.
- **40-pin Package:**
- It's typically available as a 40-pin Dual Inline Package (DIP).
- **NMOS Technology:**
- The 8085 is built using NMOS (N-channel Metal-Oxide-Semiconductor) technology.
- **Registers:**
- It includes several registers for data storage and manipulation, including a set of general-purpose registers (B, C, D, E, H, and L), an accumulator, and a stack pointer.
- **Instruction Set:**
- The 8085 has a set of instructions that control its operations, including arithmetic, logic, data transfer, and control flow instructions.
- **Interrupt Handling:**
- It supports both hardware and software interrupts, allowing it to respond to external events and internal subroutine calls.

# 8085 Pin Diagram

- Address Bus
- Data Bus
- Control and Timing Signals
- Clock and Synchronisation
- Interrupt Signals
- Serial I/O Ports

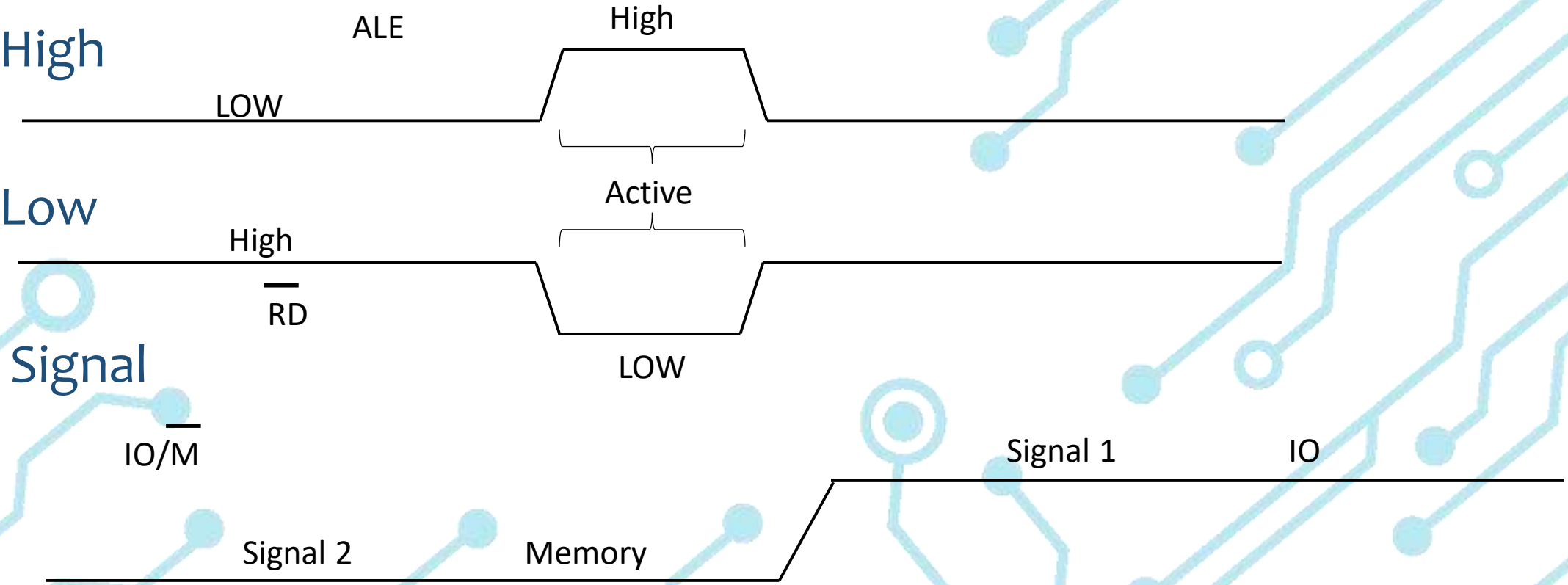


# 8085 Pin Function Diagram



# Signals type

- Active High
- Active Low
- Toggle Signal

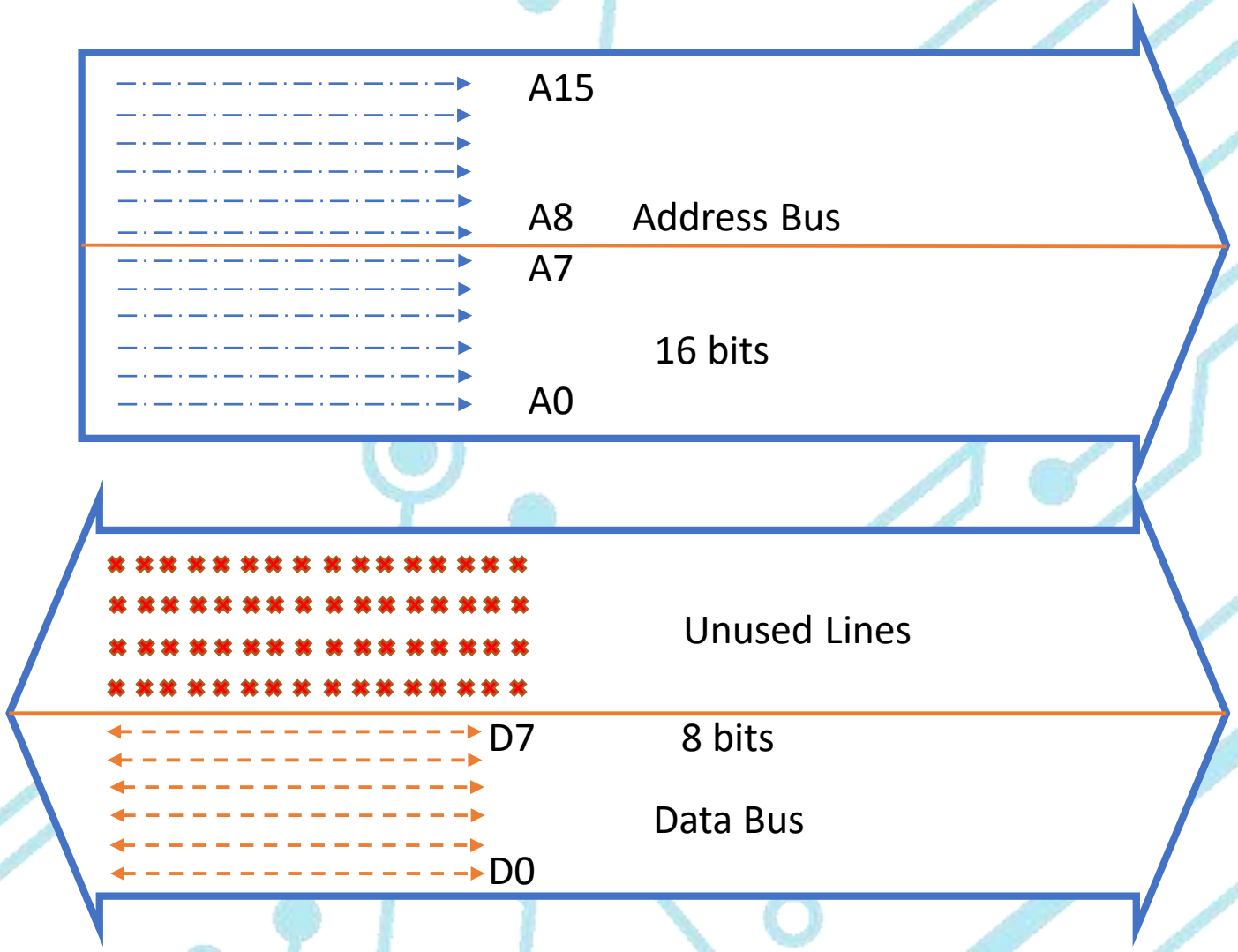


# Pin & Signal: Power & Clock Generator

- **VCC:** VCC is the power supply signal to operate the microprocessor. The signal provides +5 volt power supply.
- **VSS:** VSS is the ground signal.
- **X1, X2:** The crystal oscillator is connected to these two pins. The frequency is internally divided by two; therefore, to operate a system at 3 MHz, the crystal should have frequency of 6 MHz.
- **CLK (OUT) - Clock Output:** This signal can be used as the system clock for other devices.

# Pin & Signals: Address/Data Bus

- Address Data-Multiplex Bus (**$AD_0$ – $AD_7$ & $A_8$-$A_{15}$**)
- Works as both Address and Data
- Sends Address (A0 – A15) 16-bits
- Sends Data (D0- D7) 8-bits
- Only one of these needs to be done. So Multiplexed.
- Latch can be used to separate Address and Data.
- ALE signal is used to separate Address and Data.



# Pin & Signals: Control Signals

- **ALE** (Address Latch Enable): enables the Address Latching.
  - When signal goes high it indicates that the valid address is placed on Add bus.
- **$\overline{RD}$:**
  - The signal is used to indicate the Read Operation to be performed on I/O or Mem
- **$\overline{WR}$:**
  - The signal is used to indicate the Write Operation on I/O or Memory Devices
- **IO/$\overline{M}$:**
  - The signal is used to indicate one of two operations
  - If the signal is High it indicates I/O operations.
  - If the signal is Low it indicates Memory operation.

| $\overline{RD}$ | $\overline{WR}$ | IO/$\overline{M}$ | Function |
|---|---|---|---|
| 0 | 1 | 0 | Memory Read |
| 1 | 0 | 0 | Memory Write |
| 0 | 1 | 1 | I/O Read |
| 1 | 0 | 1 | I/O Write |

# Pin & Signal: Status Signals

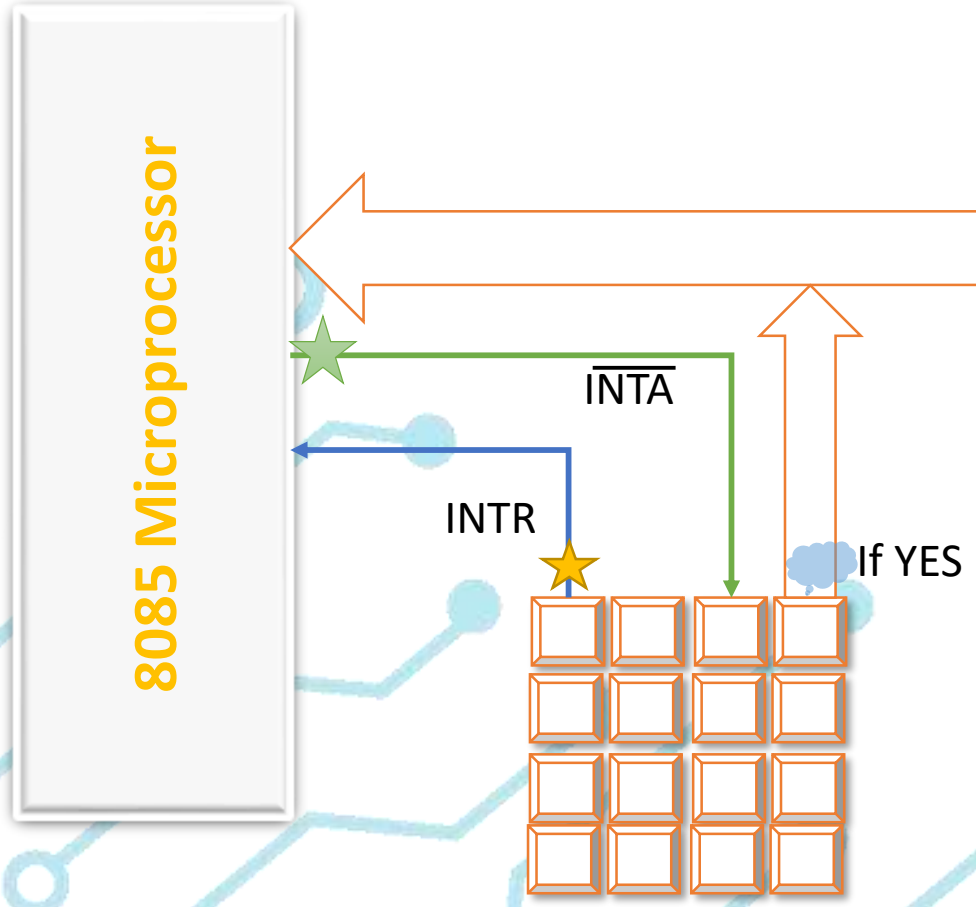
- IO/$\overline{M}$, S1, S0: status signal to indicate the status of the operation done.

| IO/$\overline{M}$ | S1 | S0 | Status |
|---|---|---|---|
| 0 | 0 | 0 | Halt |
| 0 | 0 | 1 | Memory Write |
| 0 | 1 | 0 | Memory Read |
| 0 | 1 | 1 | Opcode Fetch |
| 1 | 0 | 0 | Unused |
| 1 | 0 | 1 | I/O Write |
| 1 | 1 | 0 | I/O Read |
| 1 | 1 | 1 | Interrupt Acknowledgement |

# Pin & Signal: Interrupt Signals

- **Interrupt:** Signal from I/O device to stop processor from doing it's normal task and start a special task as per the signal. After the completion of it resumes normal processor task.
  - Maskable & Non-maskable Interrupt
- INTR: Requests microprocessor of Interrupt signal
- RST 7.5, 6.5, 5.5: Additional Interrupt signals with RST 7.5 has highest priority, RST 5.5 has lowest of three priorities.
- $\overline{\text{INTA}}$: Acknowledges Response of interrupt request of I/O device.
- TRAP: Non-maskable interrupt to compulsorily perform task.

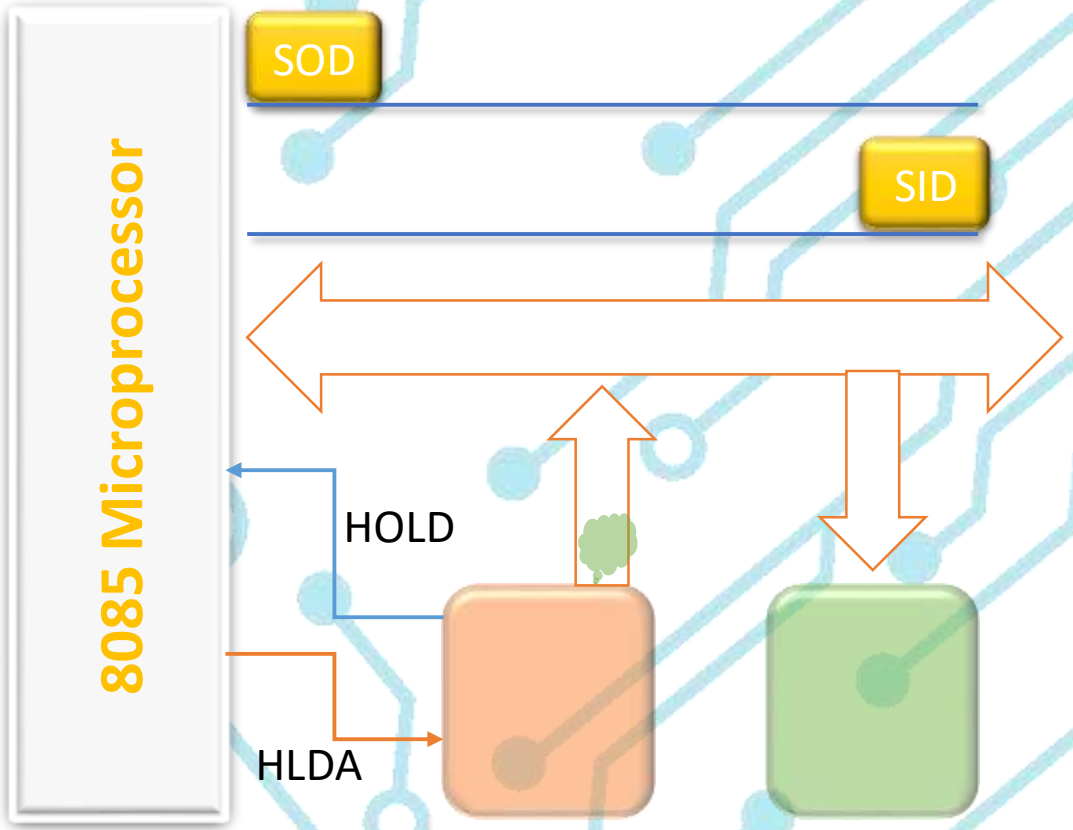
# Interrupt Process




- Processor is doing its normal task
- Event occurs on I/O device
- I/O Requests processor to perform task by sending INTR signal
- Processor identifies the importance
- Responses with $\overline{\text{INTA}}$ signal
- If negative response than I/O waits for processor to complete its task
- If positive response than I/O sends the data over data bus or performs necessary task.

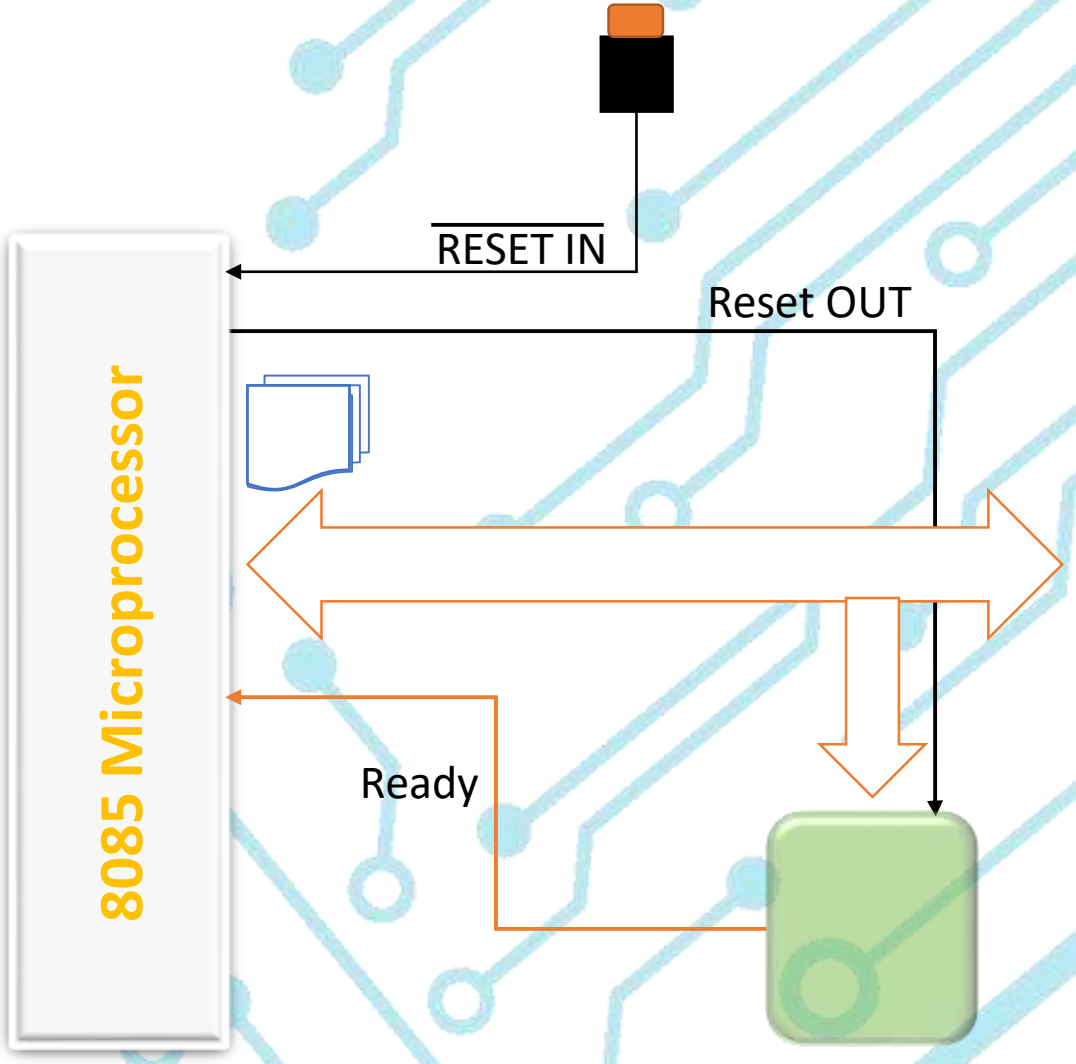
# Pin & Signal: Serial I/O & Bus Control

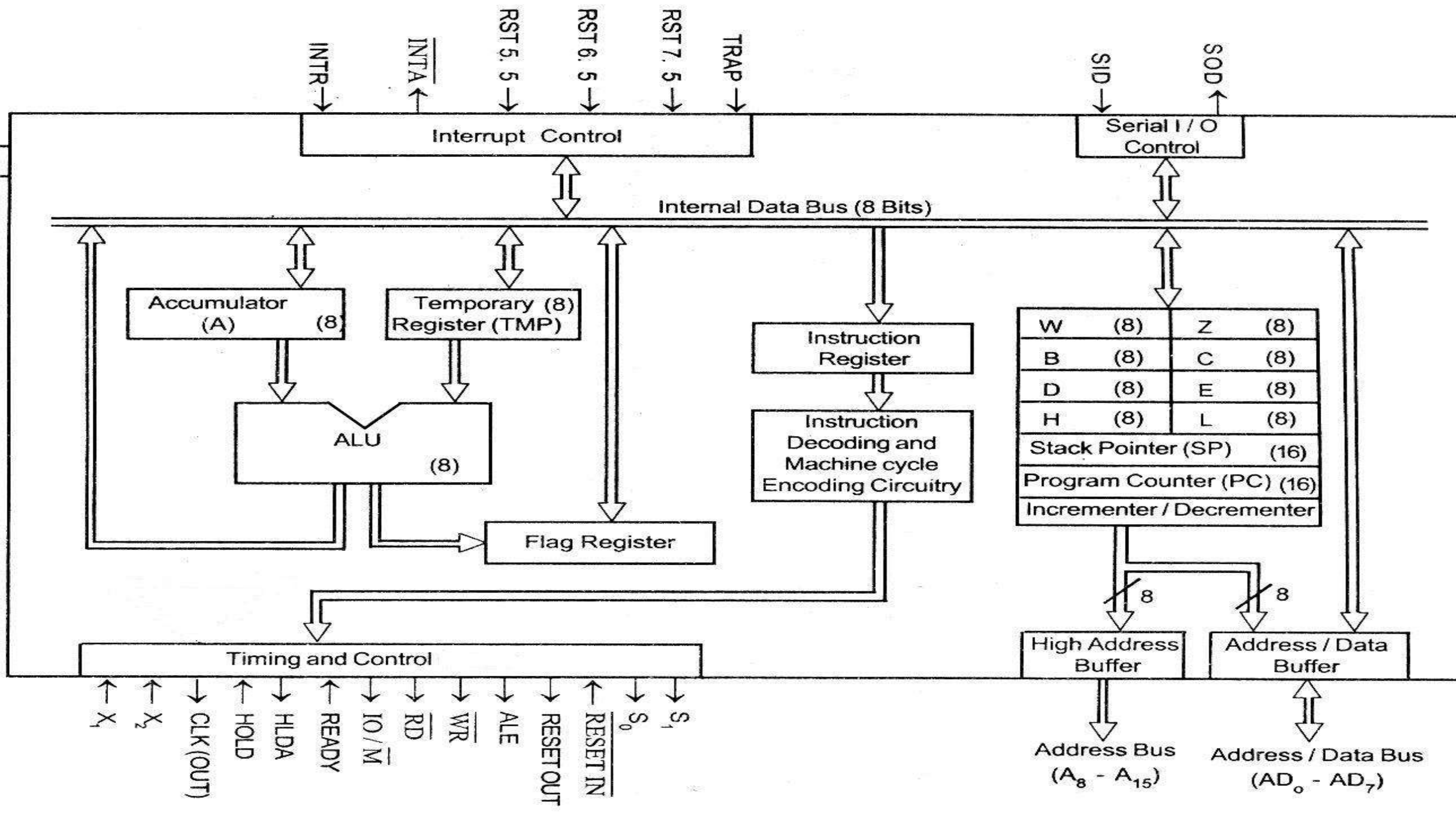
- **SOD:** Serial Output Data to send one bit of serial output data per T-State.
- **SID:** Serial Input Data to input one bit of serial input data per T-State.
- **HOLD:** Request by I/O or Memory devices to take control of bus for performing data transfer operation.
- **HLDA:** Acknowledgement signal in response to HOLD request by device.



# Pin & Signal: Synchronising

- **Ready:** Input signal from I/O device to indicate that the I/O device is ready to perform its next task. Processor waits until the device is ready
- **$\overline{\text{Reset IN}}$:** Signal indicates that user pressed Reset and processor needs to reset its operation.
- **Reset OUT:** Signal indicates to other devices that processor is going to reset its operations.



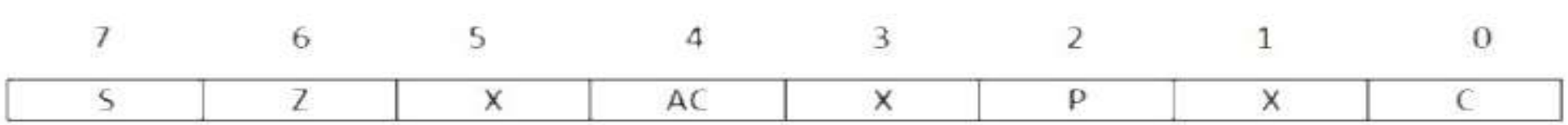

INTR
INTA
RST 5. 5
RST 6. 5
RST 7. 5
TRAP
SID
SOD
Interrupt Control
Serial I / O Control
Internal Data Bus (8 Bits)
Accumulator (A) (8)
Temporary (8) Register (TMP)
ALU (8)
Flag Register
Instruction Register
Instruction Decoding and Machine cycle Encoding Circuitry
W (8) Z (8)
B (8) C (8)
D (8) E (8)
H (8) L (8)
Stack Pointer (SP) (16)
Program Counter (PC) (16)
Incrementer / Decrementer
8
8
Timing and Control
High Address Buffer
Address / Data Buffer
X1
X2
CLK (OUT)
HOLD
HLDA
READY
IO / M
RD
WR
ALE
RESET OUT
RESET IN
S0
S1
Address Bus (A8 - A15)
Address / Data Bus (AD0 - AD7)

# Architecture of 8085

- The architecture of the 8085 microprocessor consists of several key components, including the accumulator, registers, program counter, stack pointer, instruction register, flags register, data bus, address bus, and control bus.
- The **accumulator** is an 8-bit register that is used to store arithmetic and logical results. It is the most commonly used register in the 8085 microprocessor and is used to perform arithmetic and logical operations such as addition, subtraction, and bitwise operations.
- The 8085 microprocessor has six general-purpose registers (B, C, D, E, H, and L), which can be paired as BC, DE, and HL for 16-bit operations. W and Z are reserved registers.
- The **program counter (PC)** is a 16-bit register that holds the address of the next instruction and increments after each instruction execution.
- The **stack pointer (SP)** is a 16-bit register that keeps track of the top of the stack, used for storing return addresses and temporary data.

- The **instruction register(IR)** is an 8-bit register that holds the instruction currently being executed.
- The **flags register** is an 8-bit register that stores status flags, including Carry (set when an arithmetic operation generates a carry), Zero (set when the result is zero), Sign (set when the result is negative), and Parity (set when the result has an even number of 1s).
- The **data bus** is an 8-bit bidirectional bus that transfers data between the microprocessor and memory or other devices.
- The **address bus** is a 16-bit unidirectional bus that specifies memory locations and devices the microprocessor accesses.
- The **control bus** consists of signals that manage microprocessor operations, including Read (to fetch data), Write (to store data), Interrupt (to signal an external event), and Reset (to restart the microprocessor)

- **Arithmetic and Logic Unit (ALU)**
- It is used to perform mathematical operations like addition, multiplication, subtraction, division, decrement, increment, etc. Different operations are carried out in ALU: *Logical operations, Bit-Shifting Operations, and Arithmetic Operations.*
- **Flag Register**
- It is an 8-bit register that stores either 0 or 1 depending upon which value is stored in the accumulator. Flag Register contains 8-bit out of which 5-bits are important and the rest of 3-bits are "don't Care conditions". The flag register is a dynamic register because after each operation to check whether the result is zero, positive or negative, whether there is any overflow occurred or not, or for comparison of two 8-bit numbers carry flag is checked.
- So we can say that the flag register is a status register and it is used to check the status of the current operation which is being carried out by ALU.




If the result is zero than ZF = 1 else ZF=0

If carry Auxiliary Carry generated than AC=1

If Carry generated than CF=1

If Result is positive than SF = 0
Else SF=1

If Odd Parity than PE=0
else PE=1

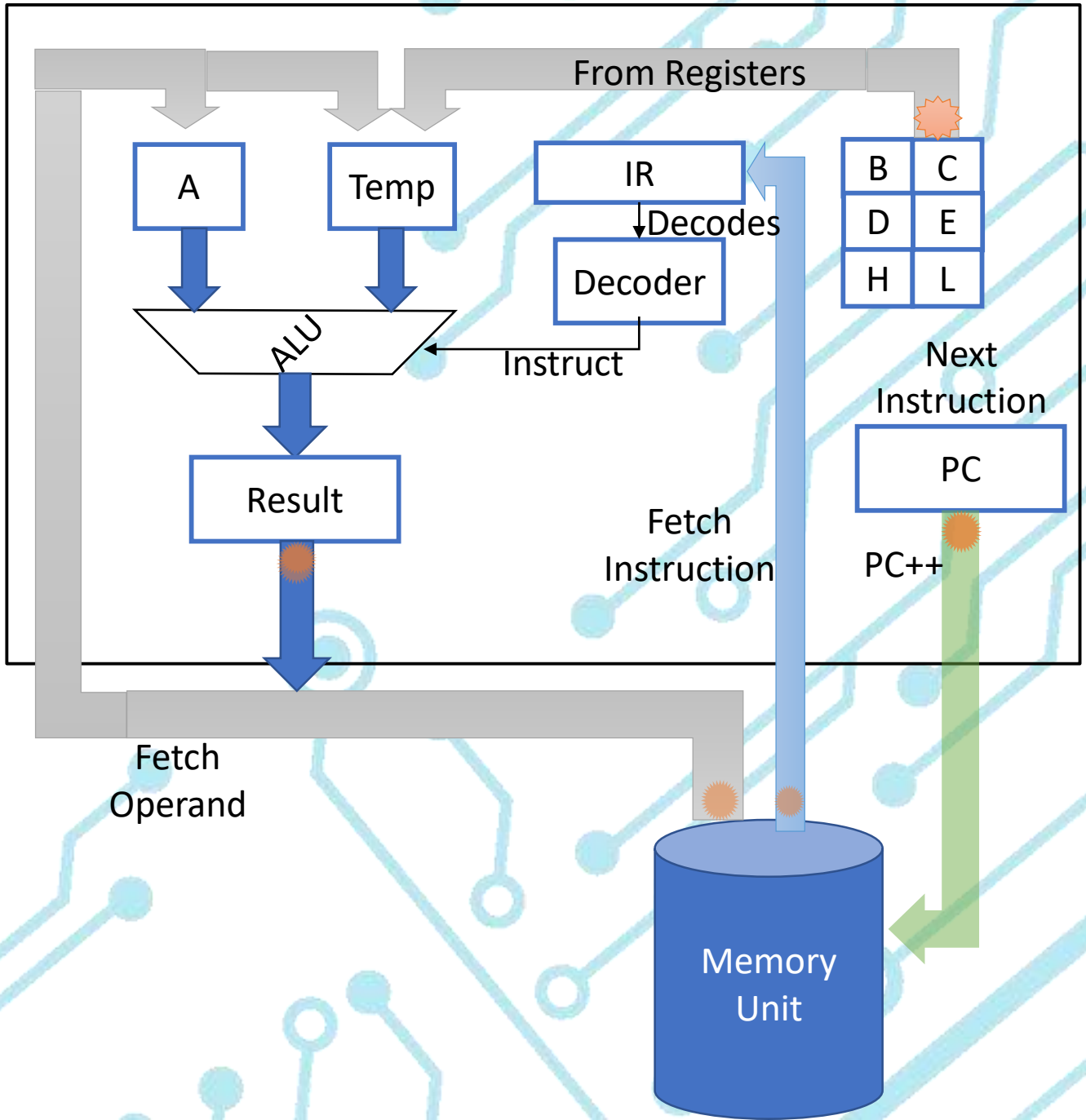
- **Sign flag(SF):** The sign flag indicates whether result generated is positive or negative. The sign flag is set to 1 whenever the result generated is negative. If the sign flag is set to 0 than the result generated is positive.
- **Zero Flag(ZF)**: The zero flag indicates the result generated is zero or non-zero. If the result generated is zero than the Zero Flag is set to 1. If the result is non-zero than the ZF=0.
- **Auxiliary Carry Flag (AC):** The Auxiliary Carry flag is set to 1 whenever there is a carry or borrow at nibble in the data. If the carry generated from 4th bit to 5th bit or borrow is taken from 5th bit to 4th bit than the Auxiliary Carry Flag is set to 1.
- **Parity Flag(PF):** The Parity Flag is set to 1 when even parity of 1s into the result. The Parity Flag is reset to 0 when odd parity of 1s into the result.
    - If the number of 1s into the result is ODD than the parity flag is reset to 0.
    - If the number of 1s into the result is EVEN than the parity flag is set to 1.
- **Carry Flag(CF):** The Carry flag is set to 1 whenever there is a carry from MSB or borrow to MSB.

- **Timing and Control Unit**
- The timing and control unit comes under the CPU section, and it controls the flow of data from the CPU to other devices. It is also used to control the operations performed by the microprocessor and the devices connected to it. There are certain timing and control signals like Control signals, DMA Signals, RESET signals and Status signals.
- **Interrupt Control**
- Whenever a microprocessor is executing the main program and if suddenly an interrupt occurs, the microprocessor shifts the control from the main program to process the incoming request. After the request is completed, the control goes back to the main program. There are 5 interrupt signals in 8085 microprocessors: INTR, TRAP, RST 7.5, RST 6.5, and RST 5.5.
- **Priorities of Interrupts:** TRAP > RST 7.5 > RST 6.5 > RST 5.5 > INTR

- **Serial Input/Output Control**
- It controls the serial data communication by using Serial input data and Serial output data.
- Serial Input/Output control in the 8085 microprocessor refers to the communication of data between the microprocessor and external devices in a serial manner, i.e., one bit at a time. The 8085 has a serial I/O port (SID/SOD) for serial communication. The SID pin is used for serial input and the SOD pin is used for serial output.
- **Timing and control unit**
- It provides timing and control signal to the microprocessor to perform operations.
- Control Signals: READY, RD, WR, ALE
- Status Signals: S0, S1, IO/M
- DMA Signals: HOLD, HLDA
- RESET Signals: RESET IN, RESET OUT

# Architecture of 8085

- Steps of Instruction Execution:
  - Fetch the instruction
  - Decode the instruction
  - Fetch the operand if required
  - Execute the instruction
  - Store the result

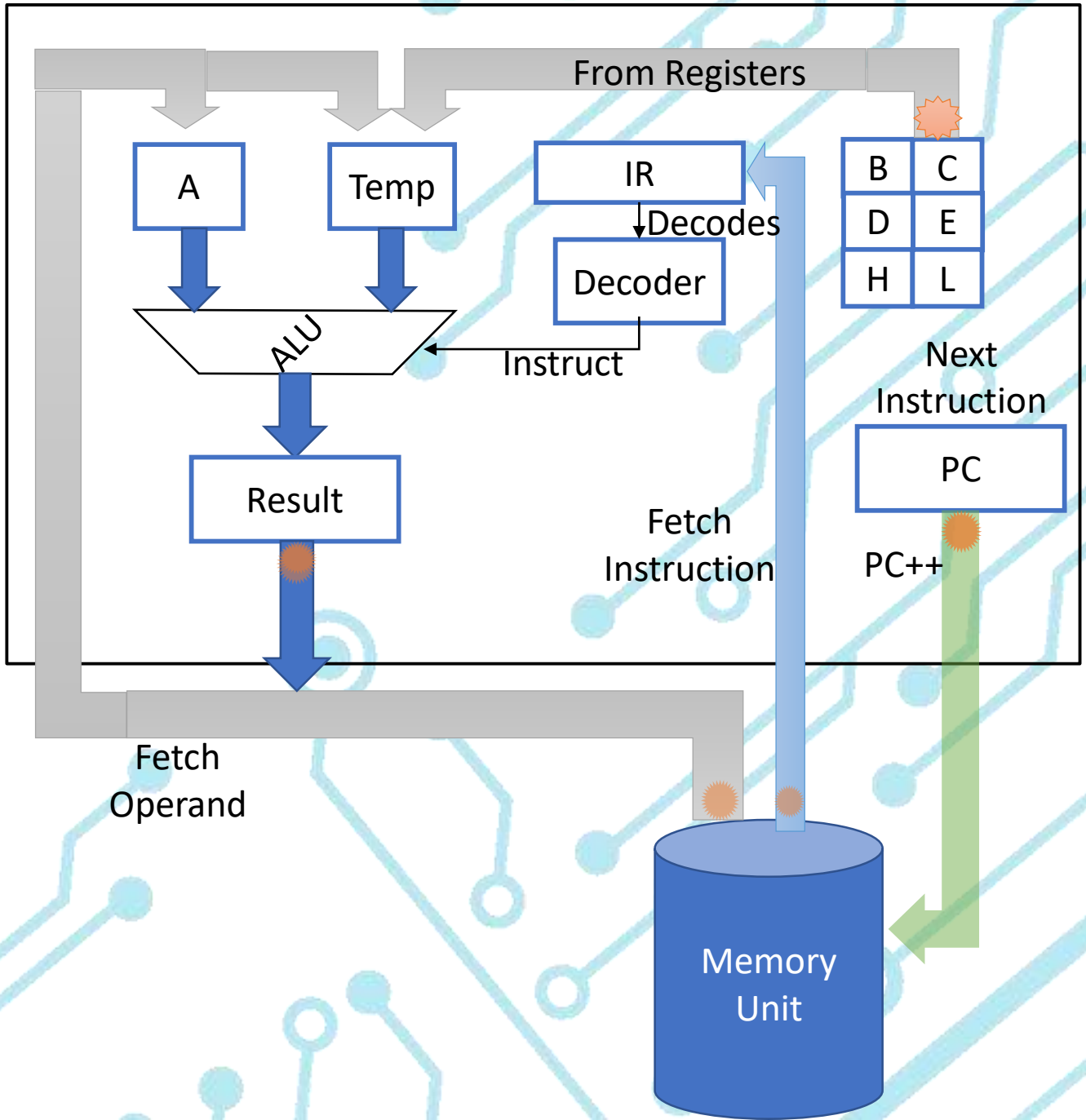


# Address Mapping

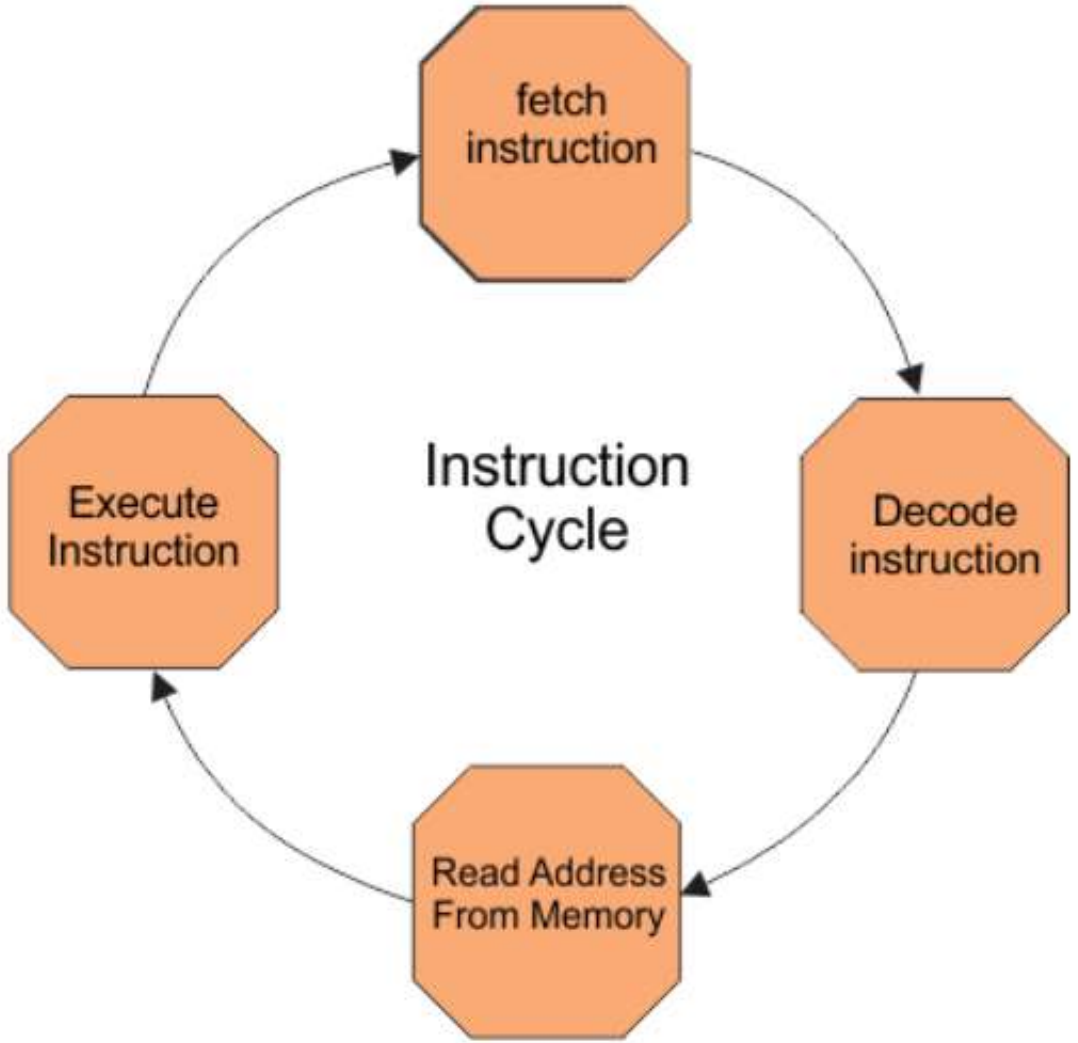
- Identifying the Memory-I/O device according to the address range.
- The starting-ending address assigned to Memory and I/O devices.
- I/O mapped I/O(Peripheral mapped I/O)
  - 8 bits used to identify the I/O device.
  - $2^8 = 256$ , $0\ to\ 255$ I/O devices can be connected with µP(microprocessor).
  - I/O Read and I/O Write signals used to indicate read and write operations
- Memory mapped I/O
  - 16 bits to identify the I/O device.
  - Treated as memory devices, used same Read and Write signals as memory units.
  - Entire $2^{16} = 65536$, 0 to 65535 range is used both by I/O and Memory

# 8085 Instruction Execution

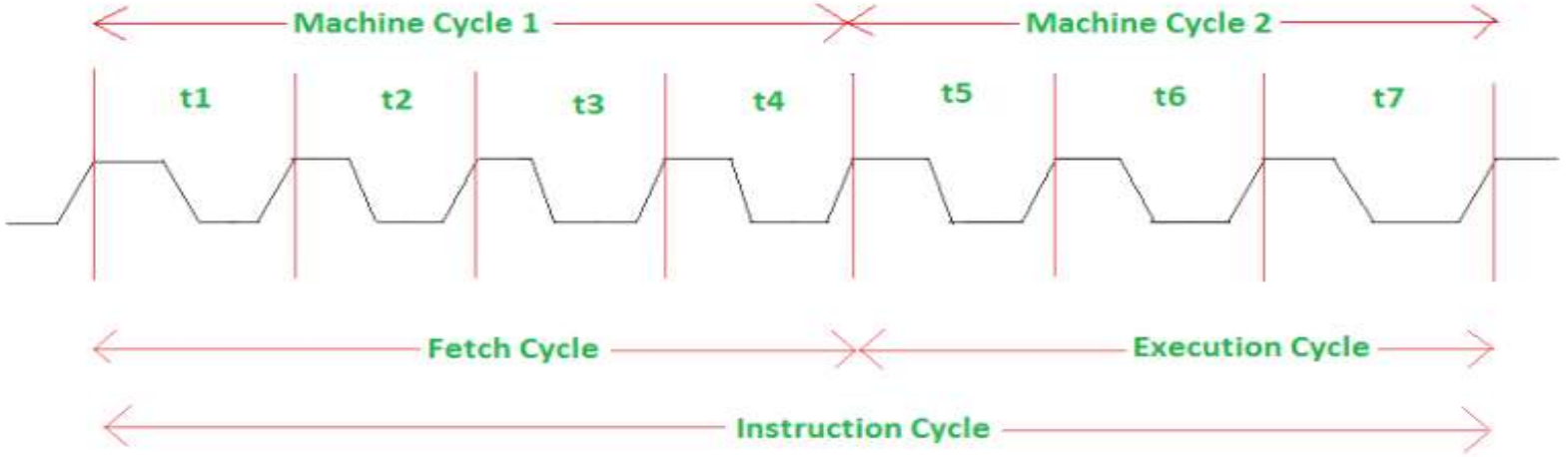
- Steps of Instruction Execution:
  - **Fetch** the instruction
  - **Decode** the instruction
  - Fetch the operand if required
  - **Execute** the instruction
  - **Store** the result



- The instruction cycle of the 8085 microprocessor consists of four basic steps, which are:
- **Fetch:** In this step, the microprocessor fetches the instruction from the memory location pointed to by the program counter (PC). The PC is incremented by one after the fetch operation.
- **Decode:** Once the instruction is fetched, the microprocessor decodes it to determine the operation to be performed and the operands involved.
- **Execute:** In this step, the microprocessor performs the operation specified by the instruction on the operands.
- **Store:** Finally, the result of the execution is stored in the appropriate memory location or register.
- Once the execution of an instruction is complete, the microprocessor returns to the fetch step to fetch the next instruction to be executed. This cycle repeats until the program is complete or interrupted.


fetch instruction
Decode instruction
Read Address From Memory
Execute Instruction
Instruction Cycle

- The time required by the microprocessor to complete an operation of accessing memory or input/output devices is called *machine cycle*. One time period of frequency of microprocessor is called ***t-state***. A t-state is measured from the falling edge of one clock pulse to the falling edge of the next clock pulse.
- **Fetch cycle takes four t-states**
- **execution cycle takes three t-states**




Instruction cycle in 8085 microprocessor