1. **Explain in detail the architecture of Intel 8085.**

Salient features of 8085 microprocessor:

- 40 pin DIP (Dual In-line package)
- Address bus- 16 bits
- Data Bus- 8 bits
- Memory of 64 KB

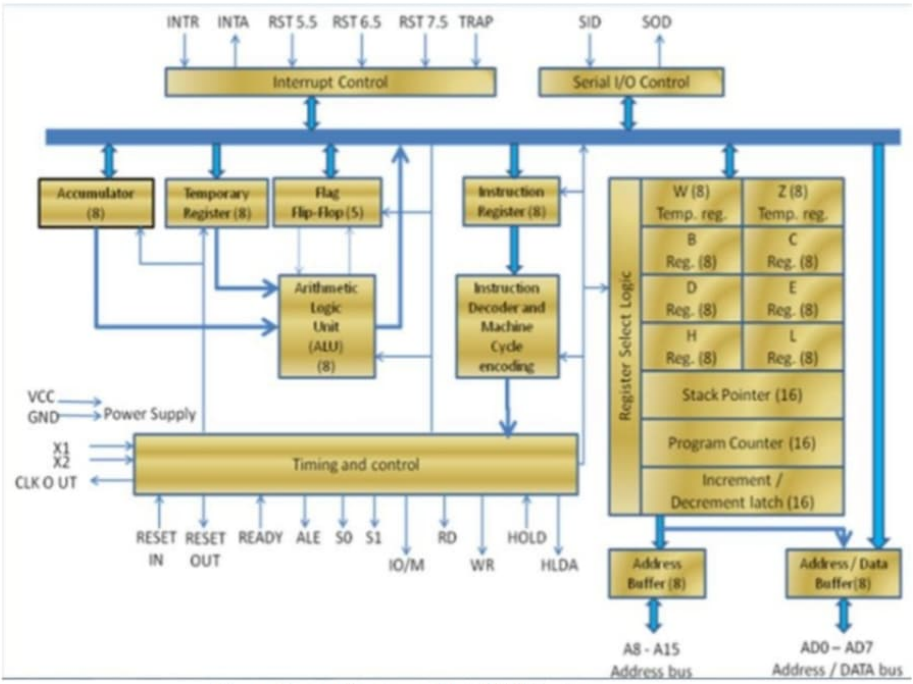


Figure 5: architecture of intel 8085 microprocessor

As shown in figure, an arithmetic logic unit (ALU), and its associated blocks. The whole system is controlled by the timing and control unit which synchronizes the activities of the processor with a central clock. All arithmetic and logic computations take place in the ALU. Instructions from memory are brought to an instruction register and decoded by the instruction decoder. Accordingly, the specified operation takes place in the ALU. The accumulator is an 8-bit register, called A, and its importance in 8085 is that it serves to hold one of the operands in most two-operand arithmetic and logical operations. The 8085 is an 'accumulator' based machine, which means that for many operations (mostly arithmetic, logical and I/O), it is mandatory to have the operand/one of the operands in the accumulator. In such cases, this register is implied and its name is not written in the instruction. It is notated as 'A' when used in other instructions.

There is a flag register which gives information about the result of arithmetic and logical computations. In such cases, 'flag bits' get altered as part of instruction execution.

All processors have registers which are used as temporary storage for operands. 'General-purpose registers' are used for data and they are 8-bit registers. The address registers are used for holding memory addresses and since addresses are 16-bit long, the address registers are 16 bits in size. The other 8-bit registers are B, C, D, E, H and L. There is the provision to combine them in pairs to make them 16-bit registers. The allowed combinations are BC, DE and HL where, in the 16-bit form, the first register holds the upper byte of the 16-bit number.

Program Counter (PC) This is a 16-bit register and it is used to 'sequence' the instructions being executed. The PC always points to the 'next instruction' to be executed.

Stack Pointer (SP) The stack pointer (SP) is a 16-bit register which points to a memory location in RAM which will hold temporary values in an area of RAM called the stack.

Arithmetic and Logical unit performs the arithmetic calculations and logical manipulations. It takes two operands, one from accumulator and the other from the temporary register that has value from any of the general purpose registers.

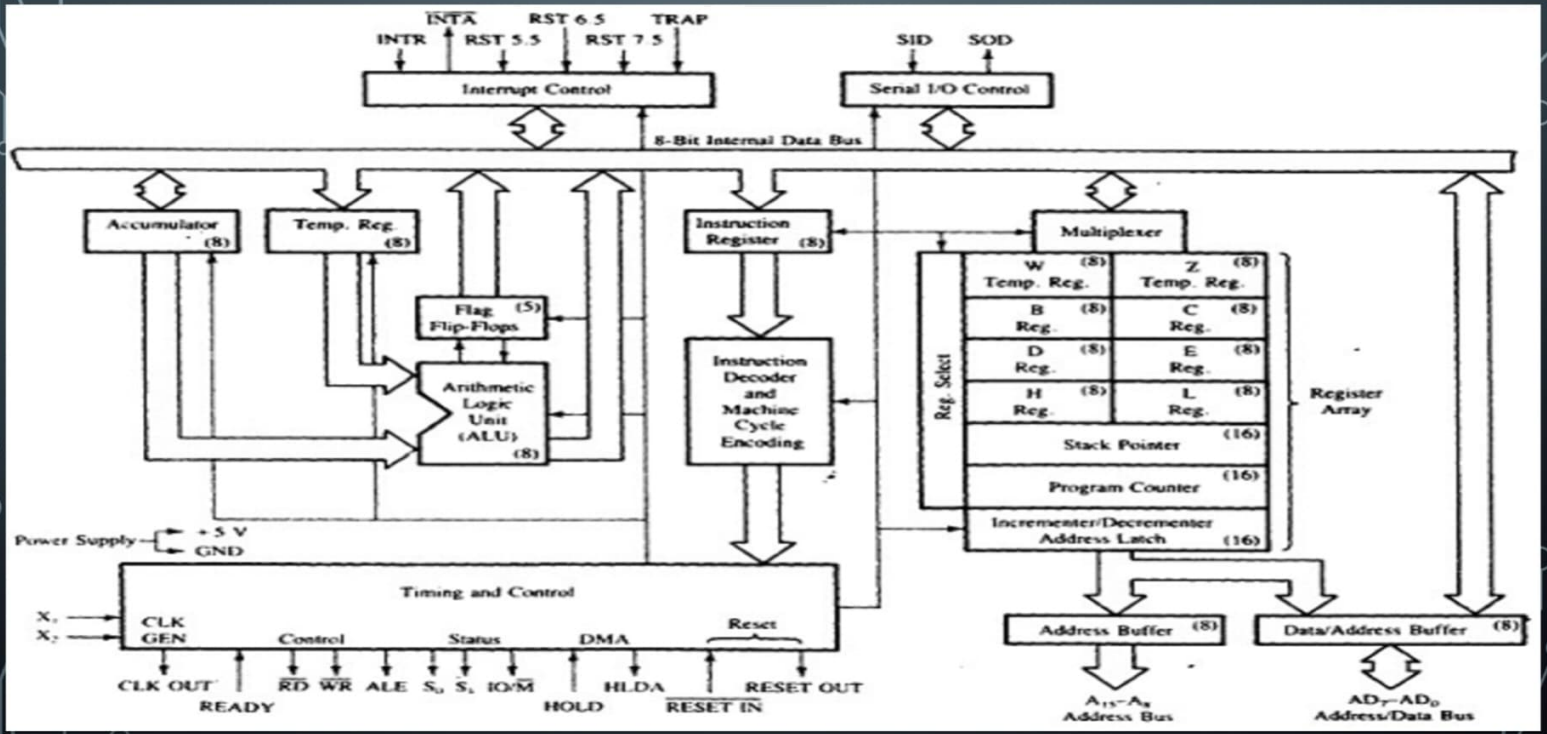
Interrupt control section is used to convey and manage interrupts to and from 8085 processor.

Serial communication section is used to manage serial communication from 8085 to other devices connected to it.

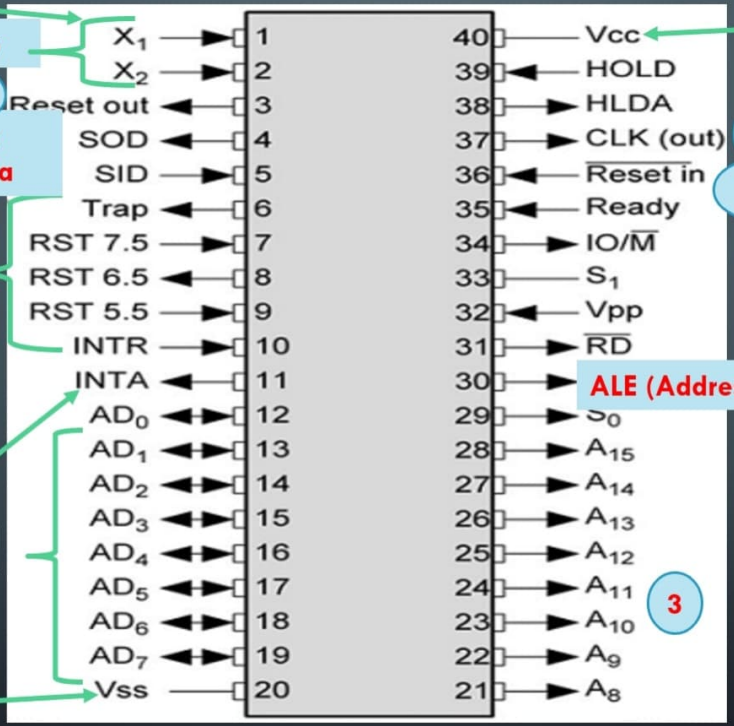
Timing and control unit is used to release control signals throughout the system to make operations run effectively.

Incrementor and Decrementor registers are used to keep track of increment and decrement operations.

ARCHITECTURE IN INTEL 8085



PIN CONFIGURATION



8085 PIN CONFIG.

→ Total pins - 40.
some pins are repetative.

1. X_1, X_2
~~→ maintains~~ frequency of 8085.
→ generates clockwise frequency of 8085.
→ set of 2 bits.

→ A_0 to A_{15} (16 bus)

- $A_8 - A_{15}$ → Address bus
  → carries address
- $AD_7 - AD_0$
  → can carry address or data.
  → depends on ALE.
  → If ALE is high - carries address
  If ALE is low - carries data

ALE (Address Enable Latch) is the control signal which is nothing but a positive going pulse generated when a new operation is started by microprocessor

3. RESET IN
   - resets the microprocessor

4. RESET OUT
   - used to
   - if microprocessor is reset, it will want other components to reset
   - Reset out is used to reset other components

5. SOD (Serial output Data) / SID (Serial Input Data)
   - Data sent input of the microprocessor is sent serially bit by bit.

6. TRAP

6. INTERUPT PINS -
   TRAP, RST 7.5, RST 6.5, RST 5.5, INTR
   → TRAP - highest priority interupt
   → every pins have diff. priority.
   TRAP > RST 7.5 > RST 6.5 > RST 5.5
   → INTR - sends interupt request (Interupt Req)

7. INTA - acknoledge interupt request (Interupt Acknowledge).

8. V_{SS} - Ground.

9. V_{CC} - Power supply.

10. HOLD - hold request

11. HLDA - tells us about the request at hold (acknoledges HOLD)

12. CLK (out) - execute time out → function as a synchronously to work other components

13. READY - shows that MP is ready to

14. RD - Read (processor is reading)

15. WR - write (processor is writing)

16. ALE - Address Latch Enable

17. S_0, S_1 and $IO/\overline{M}$ S_1-Status 1 S_0-Status 0

→ shows status of ~~signal~~ 8085.

$IO/\overline{M}$	S_0	S_1	STATUS
0	0	0	HALT
0	0	1	M-write
0	1	0	M-read
1	0	1	I/O read
1	1	0	I/O read
0	1	1	opcode fetch
1	1	1	INT-Ack (INTA)

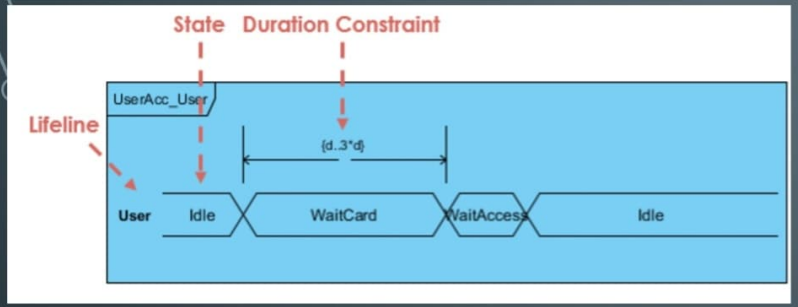
18. $IO/\overline{M}$

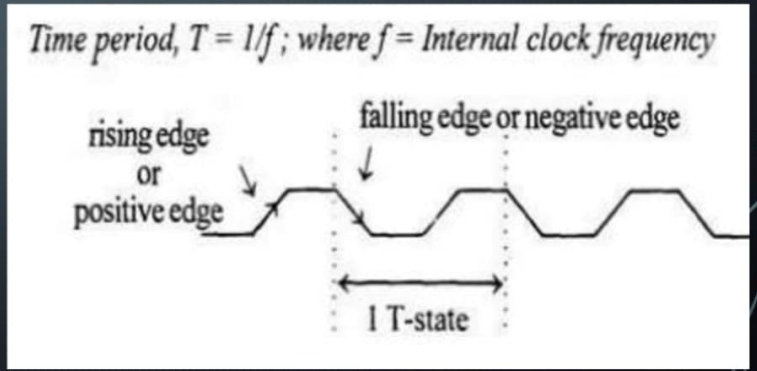
→ indicates I/O and memory oper.

→ high signal indicates i/o operation

→ low signal indicate memory operation.

TIMING DIAGRAM [1]

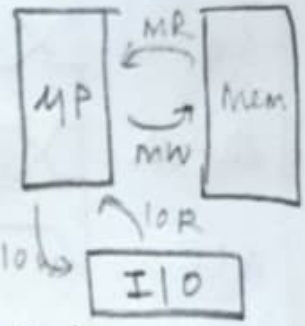




MACHINE CYCLE OF 8085

→ One data transfer that takes place in the system is called as one machine cycle.
→ one machine cycle transfers 1 byte (8 bits) of data.
→ Machine cycle operations -

1. Memory read cycle - (4T)
   transferring data from memory to up.
2. Memory write cycle (3T)
   transferring data from up to memory.
3. opcode fetch cycle - (3T)
4. I/O Read cycle (3T)
5. I/O write cycle. (3T)

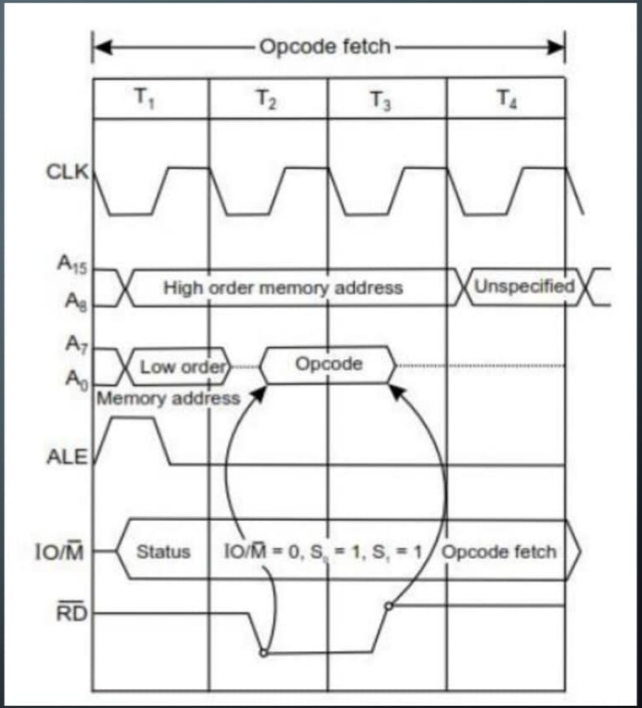


→ A portion of an operation carried out in one system clock period is called T-state. (Transition sta
→ One T-state is equal to the time period of internal clock signal of proce
→ T-states are a part of machine cycle and machine cycle are a part of instruction cycle.

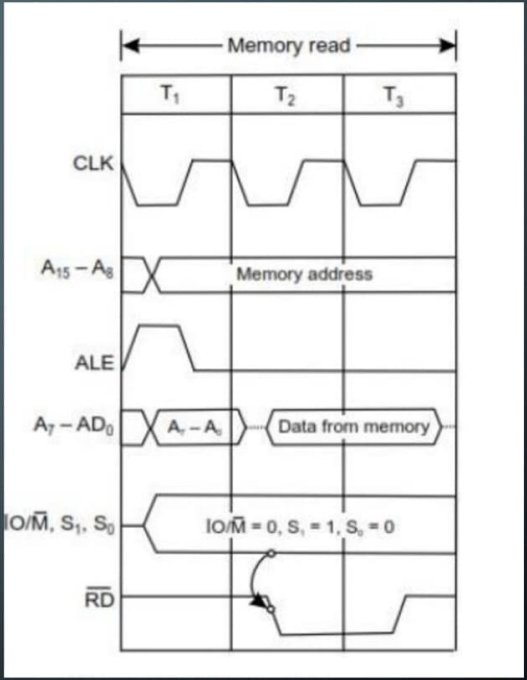
Machine cycle.

Machine cycle	Status		control signals		
	IO/$\overline{M}$	S1	S0	$\overline{RD}$	$\overline{WR}$
opcode fetch	0	1	1	0	1
Memory Read	0	1	0	0	1
Memory write	0	0	1	1	0
I/O Read	1	1	0	0	1
I/O write	1	0	1	1	0

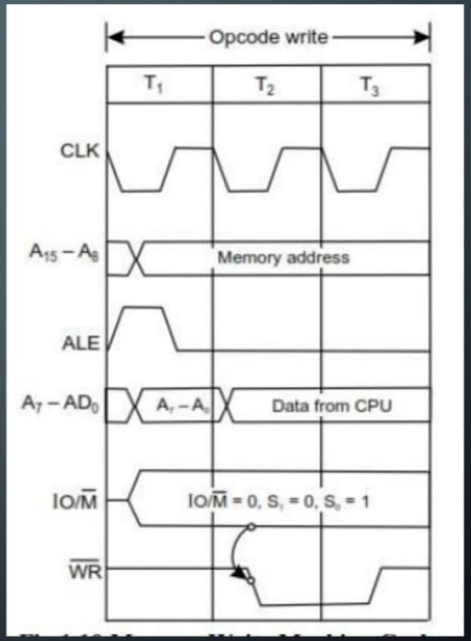
OPCODE FETCH CYCLE:



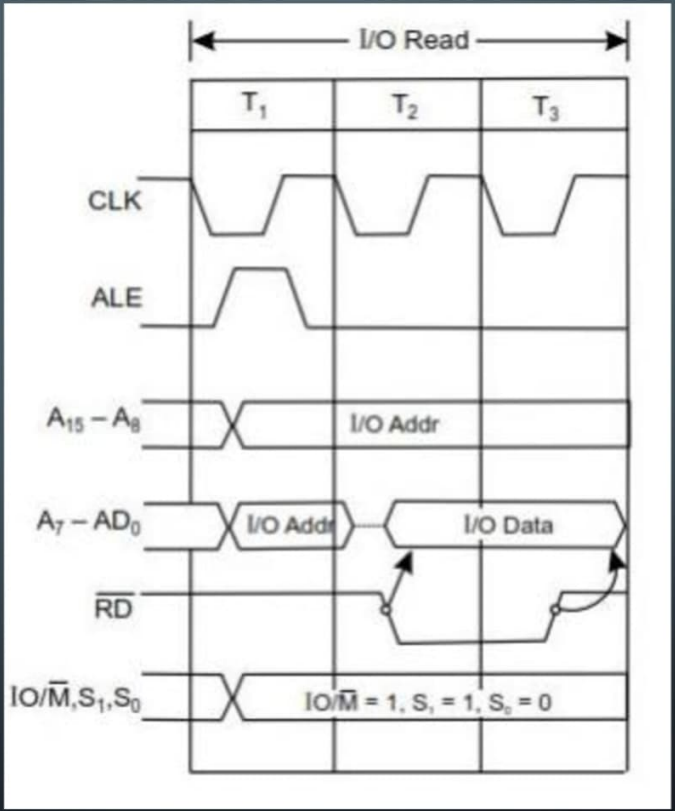
MEMORY READ MACHINE CYCLE



MEMORY WRITE MACHINE CYCLE



I/O READ


I/O Read
T1
T2
T3
CLK
ALE
A15 – A8
I/O Addr
A7 – AD0
I/O Addr
I/O Data
RD
IO/M,S1,S0
IO/M = 1, S1 = 1, S0 = 0