Evolution of Microprocessor:-

The invention of transistors in 1947 laid the foundation stone for the development of the modern microprocessor through a series of evolutionary processes.

→ In the year 1959, Integrated IC were first developed by fabricating electronic circuits with more than one transistor, on a single chip. The development of IC technology started with the small scale integrated (SSI) circuits consisting of fewer than 10 gates per chip.

→ This led to the fabrication of medium scale Integrated (MSI) circuits with 10 to 100 gates on a chip.

→ Large scale Integrated (LSI) circuit with 100 to few thousand gates & very large scale integrated (VLSI) circuit with several thousand gates on a chip.

→ By the early seventies, IC technology rose to a height where it is possible to incorporate all the functional elements of the CPU, ALU, register section and the control unit on a single IC chip. such a device was named the microprocessor.

→ The first practical microprocessor IC, that was developed in 1971 was intel 4004. a 4-bit microprocessor

→ It was employed for manufacturing programmable calculators.

→ A number of contemporary IC manufactures started producing more & more sophisticated and powerful microprocessor. As a result, in little more than a decade, the microprocessor family progressed from the simplest 4-bit processor to the very powerful 32 bit processor having capability comparable to mainframe computers.

Scanned with CamScanner

Advantage of microprocessors

Modern microprocessors are available in different word sizes (8-bit, 16 bit, 32 bit)

Microprocessor is equivalent to the central processing unit (CPU) of a general purpose computer, But physically all of its functional elements. eg arithmetic logic unit, registers and control unit are fabricated on one chip. Thus basically, a microprocessor is a single chip CPU.

Low cost:

The most advantageous feature of a microprocessor is its low cost.

→ Microprocessors are of small size, as a result of improvement in VLSI fabrication technology

→ Low power consumption.

→ High versatility of microprocessor operation is achieved by its programmability.

→ Microprocessor based systems are extremely reliable in operation.

1.3 Applications

→ Single board microcomputer systems.

→ Terminals

→ Personal computers or PCs.

→ Superminis & CAD machines.

Scanned with CamScanner

Microcomputer :-

All microcomputer systems consist of some basic elements like an input device, a display device, a CPU, memory devices, address lines, data lines and controls states lines

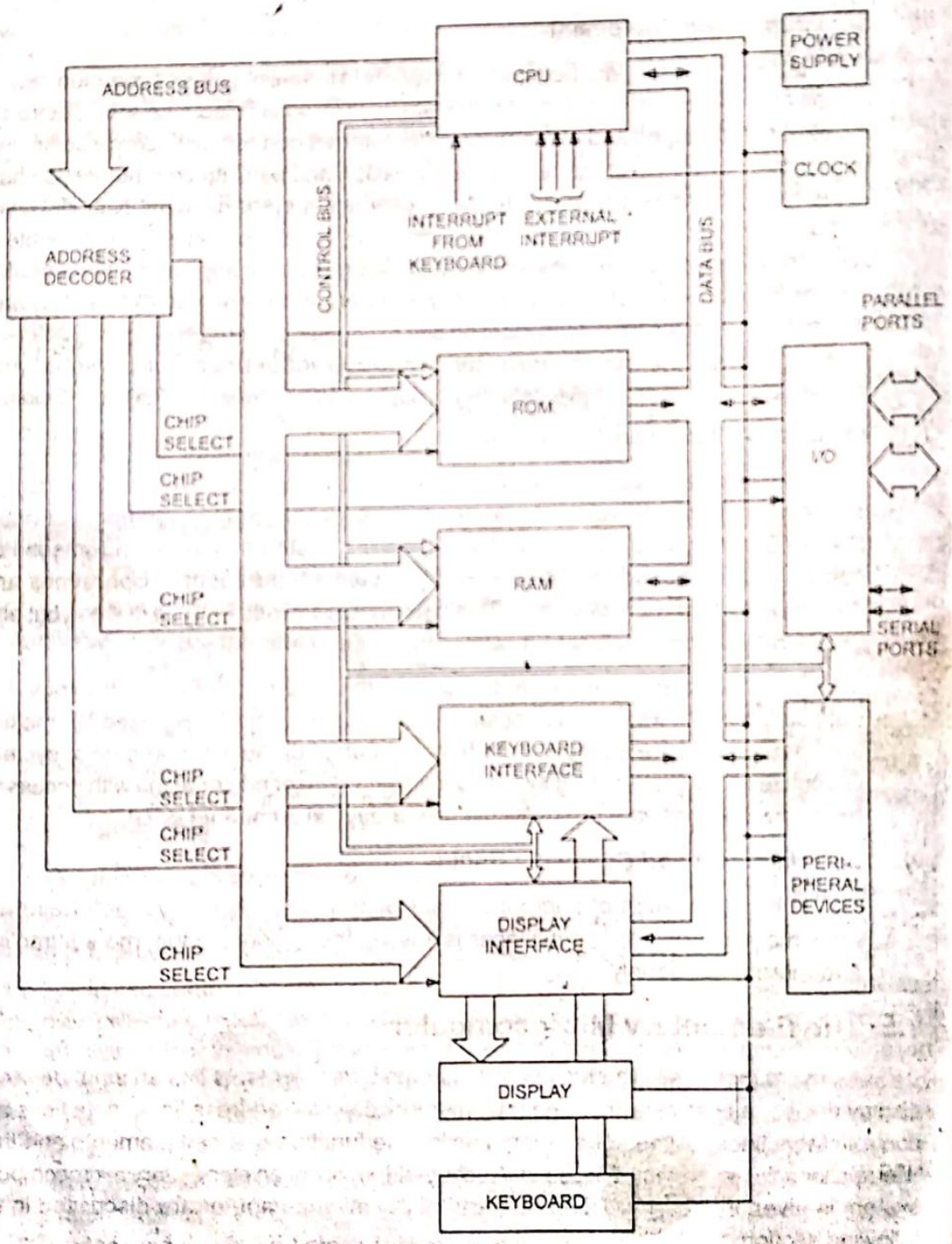
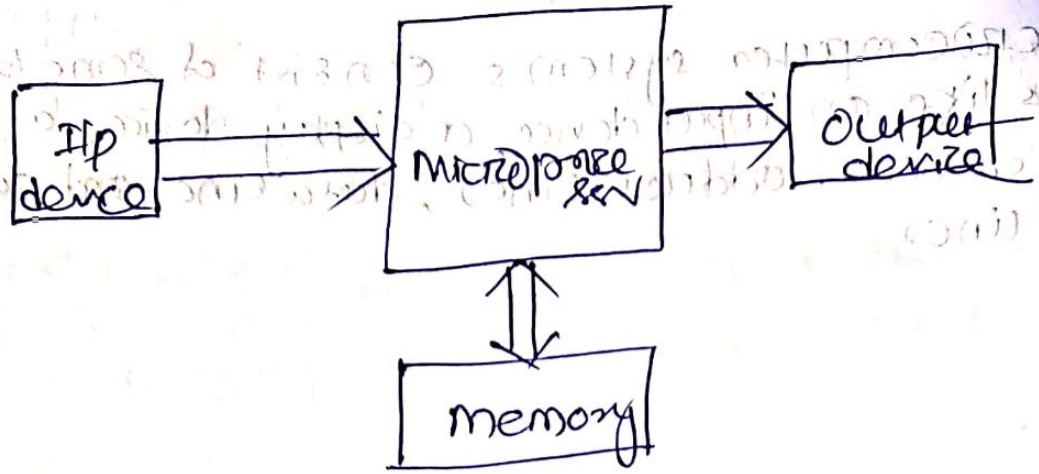


Fig.-1.1 Schematic Layout of an Elemental Microcomputer

Scanned with CamScanner



A digital computer in which one microprocessor has been provided to act as a CPU is called microcomputer.

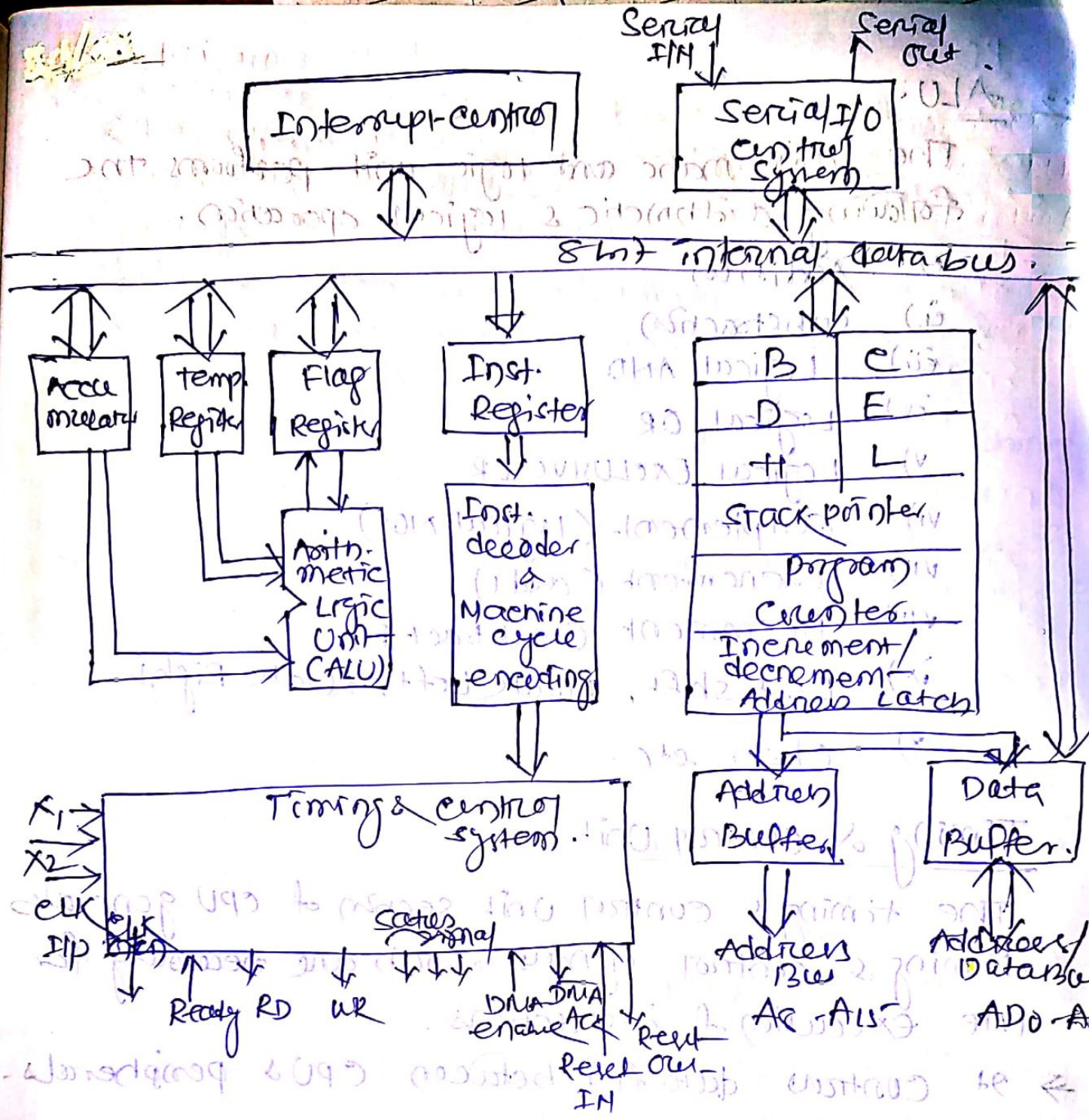
→ A desktop computer and portable computers like laptop, notebook, palmtop etc contain one microprocessor to act as a CPU and hence they come under the category of microcomputers.

→ The CPU of a largh powerful digital computer contains more than one microprocesor. high ender powerful servers, supercomputers, etc contain more than one microprocesor to act as CPU.

→ The microprocessor placed in a CPU of a largh powerful computer operate in parallel.

→ A computer where CPU contains more than one microprocessor is called a multiprocessor computer system.

Scanned with CamScanner



Intel 8085:

→ Intel 8085 is an 8-bit NMOS microprocessor

→ It has 40 pins IC package fabricated on a single LSI chip

→ Intel 8085 uses a single +5V dc power supply for its operation.

→ It's clock speed above 3MHz, the clock cycle is 320 ns.

→ It has 80 basic instruction & 246 opcodes.

Scanned with CamScanner

ALU:

The arithmatic and logic unit performs the following arithmetic & logical operation.

i) Addition
ii) Substraction
iii) Logical AND
iv) Logical OR
v) Logical EXCLUSIVE OR
vi) Complement (Logical NOT)
vii) Increment (add 1)
viii) Decrement (substract 1)
ix) Left shift, Rotate Left, Rotate Right
x) Clear etc.

Timing & control Unit

The timing & control unit section of CPU generate timing & control signals which are necessary for the execution of instructions.

→ It controls data flow between CPU & peripherals

→ It provides status, control and timing signals which are required for the operation of memory & I/O devices.

→ It controls the entire operations of the micro processor and peripherals connected to it.

⇒ Thus CPU acts as the brain of the computer system.

Scanned with CamScanner

Registers

1) One 8 bit accumulator i.e Register A
2) Six 8 bit GPR, B, C, D, E, H & L
3) One 16 bit stack pointer, SP.
4) One 16 bit program counter, PC
5) Instruction register
6) Temporary register

→ 8085 contains a set of five flip flops which serve as status flag

→ status flag indicates some condition which arises after the execution of an arithmetic or logical instruction.

Accumulator (A)

The register A of 8085 µp used to hold one of the operands of an arithmetic or logical operation.

→ It serves as one input to the ALU.

→ The other operand for an arithmetic & logical operation may be stored either in the memory or in one of the general purpose register.

→ The final result of an arithmetic or logical operation is placed in the accumulator.

GPR

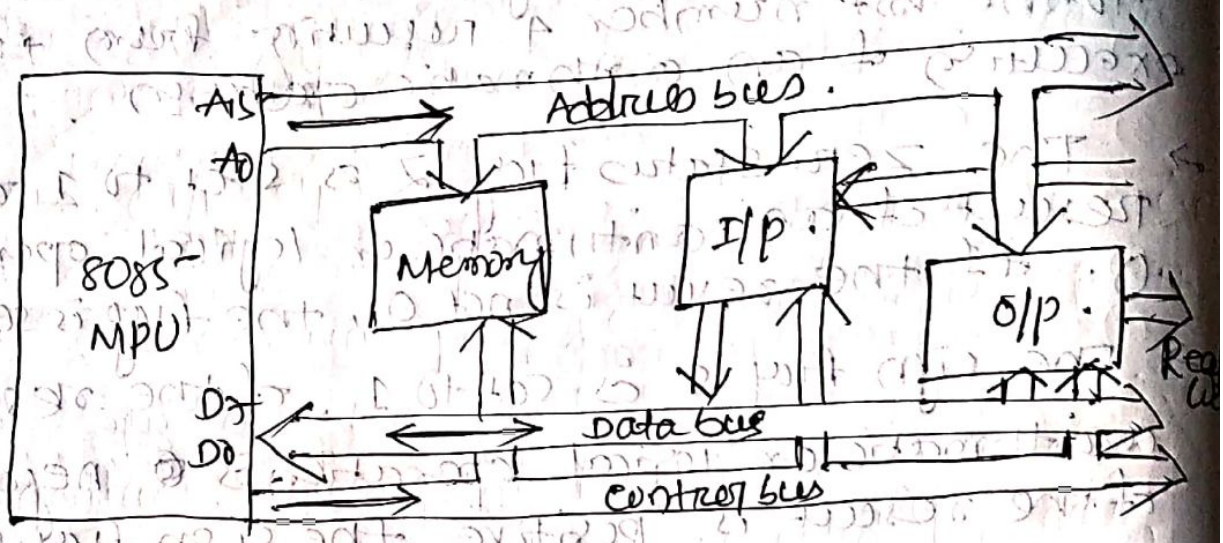
8085 µp has six general purpose register. They are B, C, D, E, H & L

→ To hold 16 bit data a combination of two 8-bit registers can be employed.

→ The combination of 8-bit register is known as register pair. The valid register pairs are BC, DE, HL

→ The HL pair is used to act as memory pointer and it holds the 16 bit address of memory location.

Scanned with CamScanner



Address bus:

→ The address bus is a group of 16 lines general identified as A0 to A15.

→ The address bus is unidirectional, bits flow in one direction - from the MPU to peripheral dev

→ In computer system each peripheral or memory location is identified by a binary number, call as address. and the address bus is used to car a 16 bit address.

→ 8085 MPU with its 16 address line is capable of addressing $2^{16} = 65536$ (64K) memory location.

Data bus:

The data bus is a group of eight lines used for dat flow. These lines are bidirectional

→ data flow in both directions between the MPU & memory & peripheral devices.

→ data bus is also used to transferring binary information.

→ The eight data lines enable the MPU to manipulate 8 bit data ranging from 00 to FF (2^8 = 256 numb

Scanned with CamScanner

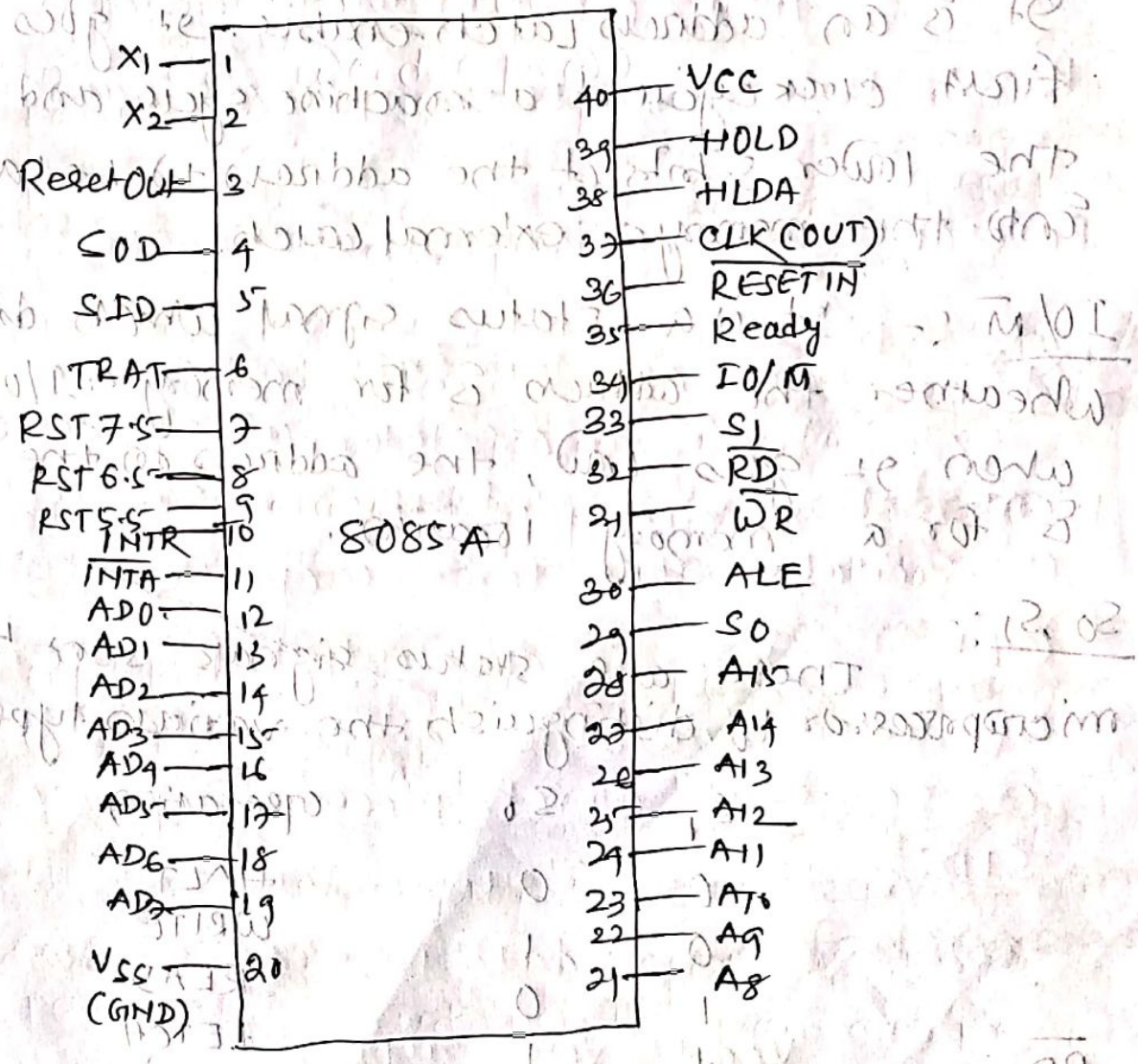
control bus:

The control bus is comprised of various single lines that carry synchronization signals.

→ The MPU generates specific control signals for every operation such as Memory Read or I/O write.

→ These signals are used to identify a device type with which the MPU intends to communicate.

2.6 pin diagram:-



Scanned with CamScanner

$A_8 - A_{15}$ (output): These are address bus and ... for the most significant bits of the memory addr... or 8 bits of I/O address.

$AD_0 - AD_7$:- (I/P & O/P)
These are time multiplexed address/data bus. i.e they serve dual purpose. They are used for the least significant bit (8) of the memory address or I/O address. during the first clock cycle of a machine cycle. Again they are used for data during second & third clock cycles.

ALE
It is an address latch enable. It goes high during first clock cycle of a machine cycle and enables the lower 8 bits of the address to be latched either into the memory or external latch.

IO/$\overline{M}$:- It is a status signal which distinguishes whether the address is for memory or I/O device. when it goes Low, the address on the address bus is for a memory location.

S_0, S_1 :-
These are status signals sent by the microprocessor to distinguish the various types of operations

S_1	S_0	operation
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

$\overline{RD}$:- when microprocessor reads data or codes from a memory location or an input device, it is called RD operation. $\overline{RD}$ is a signal sent by the

Scanned with CamScanner

to the memory/Input device to ...
→ when it goes low, the selected memory or I/p device is read.

WR :- when microprocessor sends data to a memory location or an output device to control write operation. when it goes low, the data which is on the data bus is written into the selected memory or sent to the output device.

Ready :-
It is a signal sent by an Input or output device to the microprocessor. The signal indicates that the Input or output device is Ready to send or receive data.

HOLD :-
when another device of a computer system, requests address and data bus for data transfer. It sends HOLD signal to the microprocessor
→ After receiving the HOLD request the microprocessor sends out a HLDA (HOLD acknowledge) signal to the device.

HLDA It is a HOLD acknowledge signal sent out by the microprocessor after receiving the HOLD signal.

INTR It is an Interrupt signal sent by an external device to the microprocessor. Through this line an external device informs microprocessor that, it is ready to transfer data or to initiate certain operation.

INTA :- It is an interrupt acknowledge signal issued by the microprocessor after receiving ~~acknowledge signal~~ INTR request from an external device. It is low active signal.

Scanned with CamScanner

RST 5.5, 6.5, 7.5, & TRAP

These are interrupts. when an interrupt is recognised the next instruction is executed from a fixed location in the memory.

TRAP	0024
RST 5.5	002C
RST 6.5	0034
RST 7.5	003C

The order of priority of interrupts is as follows.

TRAP (highest priority)
RST 7.5
RST 6.5
RST 5.5
INTR (lowest priority)

Reset In: It resets the program counter to zero.

Reset out: It indicates that the CPU is being reset.

X_1 X_2: These are terminals to be connected to an external crystal oscillator which drives an internal circuitry of the microprocessor to produce a suitable clock for the operation of microprocessor.

CLK: It is a clock output for user, which can be used for other digital ICs. Its frequency is same at which processor operates.

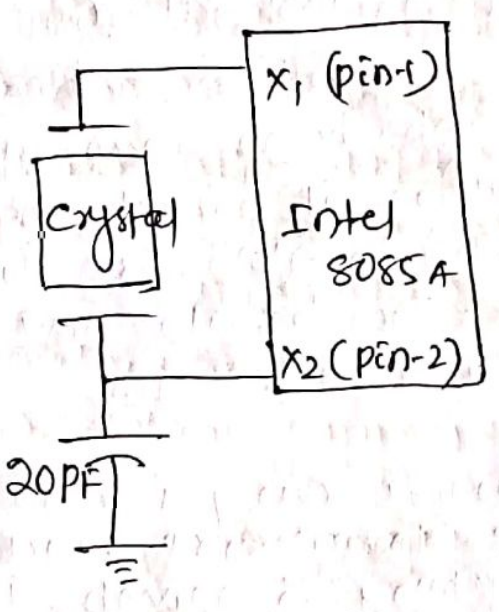
SID: It is data line for serial input.

SOD: It is a data line for serial output.

Vcc - +5 volts supply
Vss - Ground reference.

Scanned with CamScanner

2- Clock pulse generation

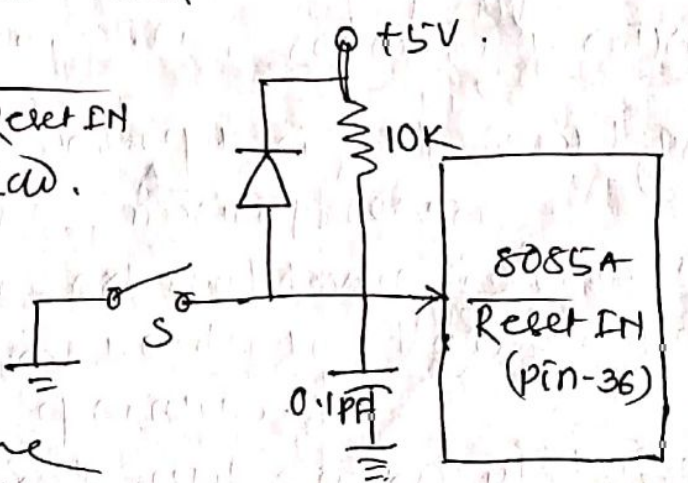


→ The 8085A microprocessor is provided with onchip clock generation circuit. The clock pulse are usually generated by connecting an external clock circuit between the X_1 & X_2 pins.

→ The required clock frequency can be achived by a crystal or LC tuned filter or an RC Network.

→ The 8085A can operate with a maximum clock frequency of 3.125MHz.

Reset circuit



→ When the active low $\overline{\text{Reset IN}}$ (pin-36) signal goes Low.

→ It causes the 8085A to execute the 1st Inst. at 0000H memory location.

→ Besides, It resets the Instruction register, Interrupt mask bits of RST 5.5, 6.5, 7.5 as well as other register.

→ This signal should be kept low for a duration of at least three clock cycle

→ When switch S is pressed, the $\overline{\text{Reset IN}}$ signal becomes Low with large time constant providing adequate time to reset the system.

Scanned with CamScanner

6.1.2 Instruction sets of 8085

The programmer can write a program in assembly language using these instructions. These instructions have been classified into the following groups:

1. Data transfer group
2. Arithmetic group.
3. Logical group.
4. Branch control group.
5. I/O & Machine control group.

Data transfer group:-

Instruction which are used to transfer data from one register to another register, from memory to register, or register to memory come under this group. Examples are MOV, MVI, LXI, LDA, STA etc.

Arithmetic group:-

The instructions of this group performs arithmetic operations such as addition, subtraction, increment, or decrement of the content of a register or memory. Examples are: ADD, SUB, INR, DAD etc

Logical group:-

The instructions under this group perform logical operation such as AND, OR, compare, rotate etc. Examples are - ANA, XRA, ORA, CMP, RAL etc.

Branch control group:- This group includes the instructions for conditional & unconditional jump, subroutine call and return & Restart. Examples are JMP, JC, JZ, CALL, CZ, RST. etc.

Scanned with CamScanner

I/O and Machine control group:

This group includes the instructions for Input/output ports, stack and machine control. Examples are IN, OUT, PUSH, POP, HLT, etc.

Symbols and abbreviations

The symbols and abbreviations which have been used while explaining Intel 8085 instructions are as follows.

Symbol/ Abbreviations	Meaning
addr	16-bit address of a memory location
data	8-bit data.
data 16	16-bit data.
r, r1, r2	One of the registers A, B, C, D, E, H, L
A, B, C, D, H, L	8-bit register.
A -	Accumulator.
H-L	Register pair. H-L
B-C	Register pair. B-C
D-E	Register pair D-E
PSW	Program status word.
M	Memory where address in the H-L pair.
H	Appearing at the end of a group of digits specifies Hexadecimal e.g - 2500H
rp	- one of the register pairs.
rh	The high order register of a register pair.
rl	The low order register of a register pair.
PC	16-bit program counter

Scanned with CamScanner

CS - Carry status.

[ ] - The contents of a register identified within bracket.

[[ ]] - The content of the memory location whose address is in the register pair identified within brackets.

∧ - AND operation.

∨ - OR operation.

∀ or ⊕ - Exclusive-OR

← - Move data in the direction of arrow.

⇔ - Exchange contents.

Addressing modes:

Each Inst. requires certain data on which it has to operate. It has already been explained that there are various techniques to specify data for instructions. These techniques are called addressing modes. Intel 8085 uses the following addressing mods:

1- Direct addressing
2- Register addressing
3- Register indirect addressing
4- Immediate addressing

Direct addressing:

In this mode of addressing, the address of the operand (data) is given in the inst. itself.

Ex: (32,00,24)

1) STA 2400H store the content of the accumulator in the memory location 2400H.

2) LDA 4002 (DB,02) Read the data from port c:

Scanned with CamScanner

Register addressing

In register addressing mode the operand is in one of the general purpose register. The opcode specifies the address of the registers in addition to the operation to be performed.

MOV A, B (78) Move the content of register B to register A

ADD B (80) Add the content of register B to the content of register A.

Register indirect addressing

In this mode of addressing the address of the operand is specified by a register pair

Ex:

1) LXI H, 2500H - Load HL pair with 2500H.
MOV A, M - Move the content of the memory location whose address is in HL pair (2500H) to the accumulator.
HLT. Halt.

2) LXI H, 2500H - Load HL pair with 2500H.
ADD M - Add the content of the memory location whose address is in HL pair to the content of the accumulator.
HLT. HLT.

4) Immediate addressing

In immediate addressing mode the operand is specified within the instruction itself.

1) MVI A, 05. (3E, 05) Move 05 in register A
2) ADI 06 (C6, 06) Add 06 to the content of the accumulator.

5) Implicit addressing

There are certain instructions which operate on the

contents of the accumulator. Such instruction do not require the address of the operand.
Examples:- CMA, RAL, RAR etc.

Grouping of Instructions

Data transfer Group: MOV r1, r2

Move data: (move the content of the one register to another)

MOV [r1] ← [r2], Register addressing, Machine cycle-1

Ex: MOV, A, B moves the content of register A to register B.

1- MOV, r, M — Move the content of memory to register.
[r] ← [[H-L]] Register Indirect addressing: M/C-2, states-7

Ex: LXI H, 2000H ← Load HL pair by 2000H
MOV, B, M, Move the contents of memory location 2000H to

2. MOV M, r - move the content of register to memory, states-7, Reg. Indirect/ M/C-2
[[H-L]] ← [r]

3- MVI r, data (Move immediate data to register)
[r] ← data, states-7, Immediate, M/C-2

4- MVI M, data (Move immediate data to memory)
[[H-L]] ← data, states-10, immediate/reg. indirect, M/C-3

Ex: LXI H, 2400H
MVI, M-08
HLT

Scanned with CamScanner

LXI rp, data 16 (Load register pair immediate)
[rp] ← data 16 bits [rh] ← MSBs [rL] ← 8 LSBs of data
States-10, immediate, M/c-3.

LDA addr (Load accumulator direct)
[A] ← [addr], states-13, direct, M/c-4.

STA addr (store accumulator direct)
[addr] ← [A]. states-13, direct, M/c-4.

LHLD addr (Load H-L pair direct)
[L] ← [addr], [H] ← [addr+1]. states-16, direct, M/c-5

LDAX rp (Load accumulator indirect)
[A] ← [[rp]] states: 7, register indirect, M/c-2

Arithmetic Group :-

ADD r. (Add register to accumulator)
[A] ← [A] + r states-4, Register, M/c-1

ADD M. (Add memory to Accumulator)
[A] ← [A] + [[H-L]] states-7, Reg. indirect M/c-2

ADC r (Add register with carry to accumulator)
[A] ← [A] + [r] + [CS] state-4, Register, M/c-1

ADC-M (Add memory with carry to Accumulator)
[A] ← [A] + [[H-L]] [CS] state-7, Reg. indirect, M/c-2

ADI data. (Add immediate data to accumulator)
[A] ← [A] + data state-7, immediate, M/c-2,3

INR r. increment register content.
state-4, Register, M/c-1

Scanned with CamScanner

Logical group

ANA, r. (AND register with accumulator)

ANA M (AND memory with accumulator)

ANI data (AND immediate with accumulator)

ORA r. (OR register with accumulator)

ORA M (OR memory with accumulator)

ORI data (OR immediate data with accumulator)

XRA r - (Exclusive OR with accumulator)

XRA M - (- do - memory with Accumulator)

XRI data (Exclusive OR immediate data with accumulator)

CMA - complement the accumulator.

Branch group

JMP addr (level) unconditional jump to the instr. specified

JZ addr (level)	Jump if the result is zero.
JNZ addr (level)	Jump if result is not zero.
JC addr (level)	Jump if there is carry
JNC addr (level)	Jump if there is no carry.
JP - do -	Jump if result is plus.
JM	Jump if result is minus.
JPM -	Jump if even parity
JPA -	Jump if odd parity.

Scanned with CamScanner

CAY addr (Level) call the subroutine identified by the address.

Stack, I/O & M/C Group

IN port-address (Input to accumulator from I/O port)

OUT port-address (output from accumulator to I/O port)

PUSH rp (push the content of register pair to stack)

PUSH PSW (push the program status word to the stack)

POP rp (copy two bytes from the top of the stack to the specified reg

HLT. HLT.

RIM (Read interrupt mask)

NOP - (NO operation)

SIM - Set interrupt mask

EI - Enable interrupt

DI - Disable interrupt

Scanned with CamScanner

Opcode & Operand

Each instruction contains two parts. operation code (opcode) and operand

→ The first part of the inst. which specifies the task to be performed by the comp. is called opcode.

→ The second part of the inst. is the data to be operated on. and it is called operand.

Instruction word size:

A digital computer understands instructions in binary codes (machine code). The machine codes of all insts are not of the same length. According to the word size the Intel 8085 instructions are classified into the following three types.

1) 1 byte instruction 2) 2-byte instruction
3) 3 byte instruction.

One byte instruction

MOV A, B - 78 H

ADD B

RAL

Two byte instructions:

In a two byte instr. the 1st byte of the inst. is its opcode and the 2nd byte is either data or address.

→ MVI B, 05. ⇒ 06-05

The 1st byte 06 is the opcode for MVI B, 2nd byte 05 is the data which is to be moved to register B.

→ IN 01 - DB, 01

Scanned with CamScanner

Three byte instruction:

In a three byte instruction the 1st byte of the instr. is its opcode and the 2nd & 3rd bytes are either 16 bit data or 16 bit address. Examples are:

1) LXI H, 2400H, 21, 00, 24
   L H

2) LDA A, 2500H. 3A, 00, 25

Instruction cycle:

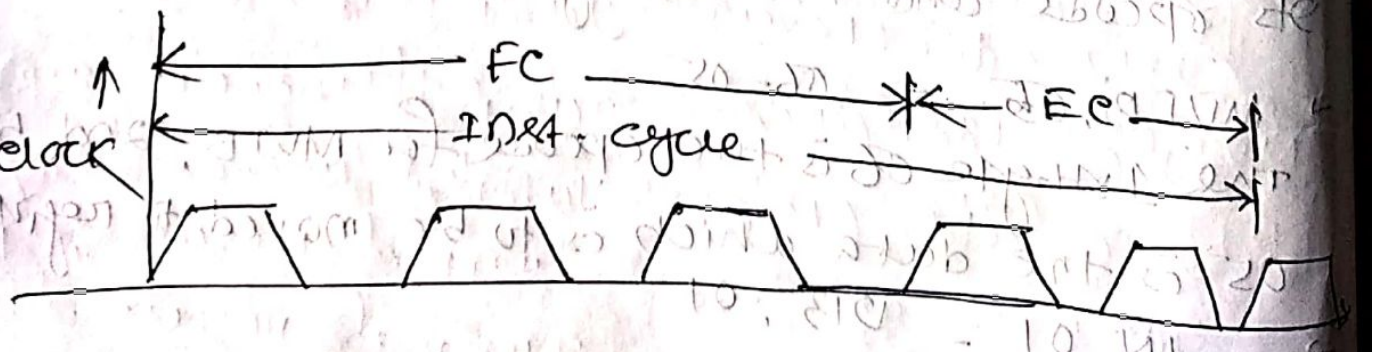
An inst. is a command given to the computer to perform a specific operation on given data. The necessary steps that a CPU carries out to fetch an instruction and necessary data from the memory and to execute it, constitute an inst. cycle.

→ An inst. cycle consists of fetch cycle & execute cycle

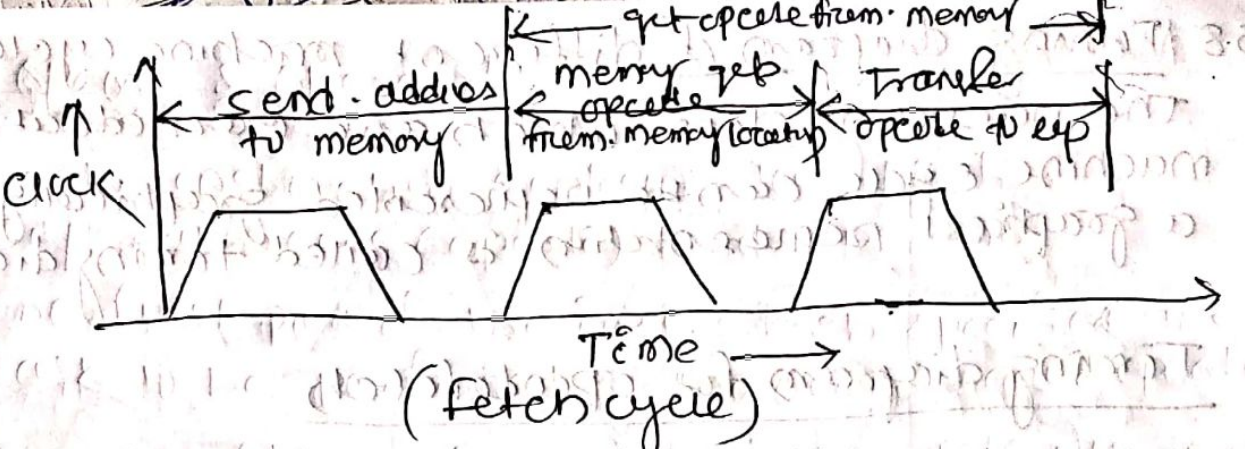
→ In fetch cycle a CPU fetches opcode (the machine code of an inst.) from the memory, constitute fetch cycle.

→ The necessary steps which are carried out to get data, if any from the memory and to perform the specific operation specified in an instruction, constitute an execute cycle.

IC = FC + EC

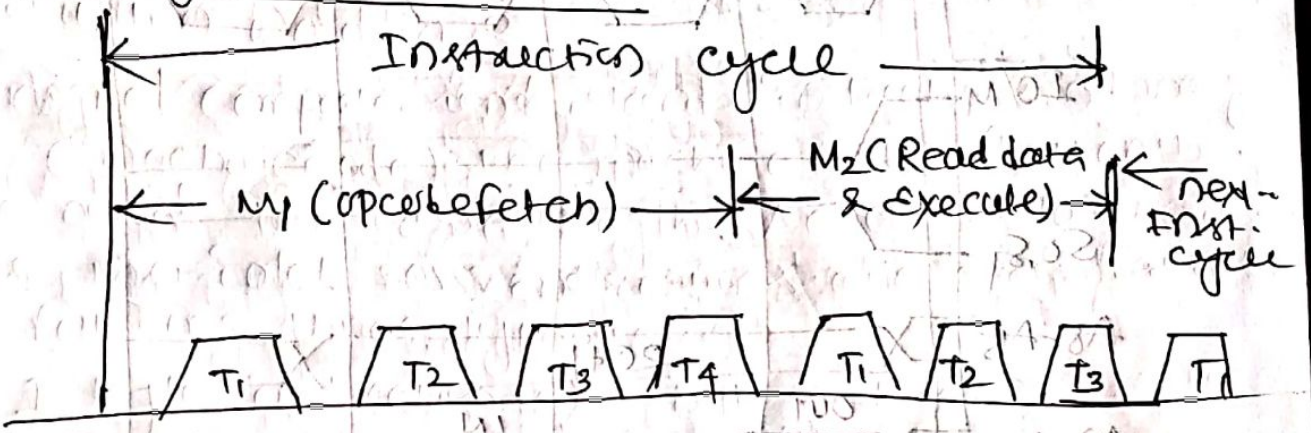


Scanned with CamScanner



(Fetch cycle)

Machine cycle and state



Typical inst. cycle.

→ In a machine cycle one basic operation such as opcode fetch, memory read, memory write, I/O read & I/O write is performed.

→ An inst. cycle consists of several machine cycle.

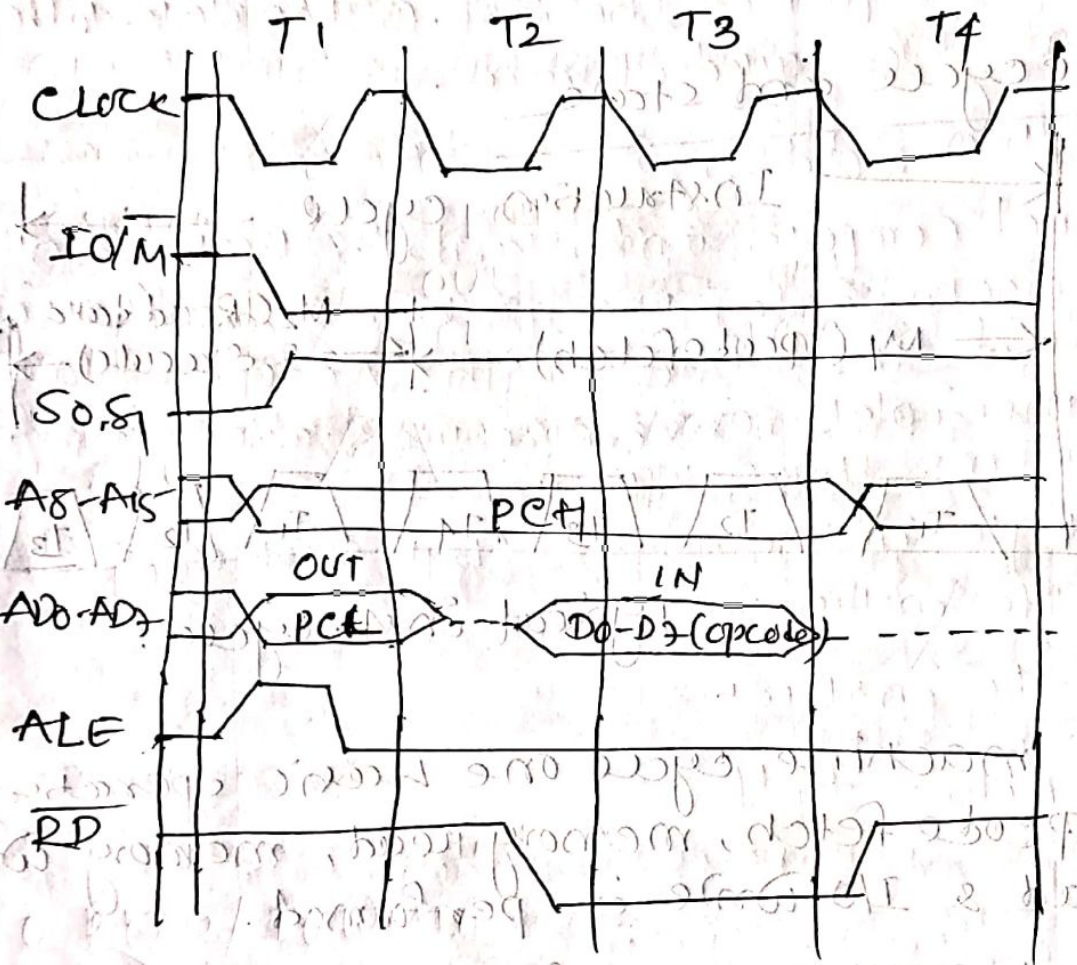
→ One subdivision of an operation performed in one clock cycle is called a state or T. state

Scanned with CamScanner

Timing diagram of different machine cycle.

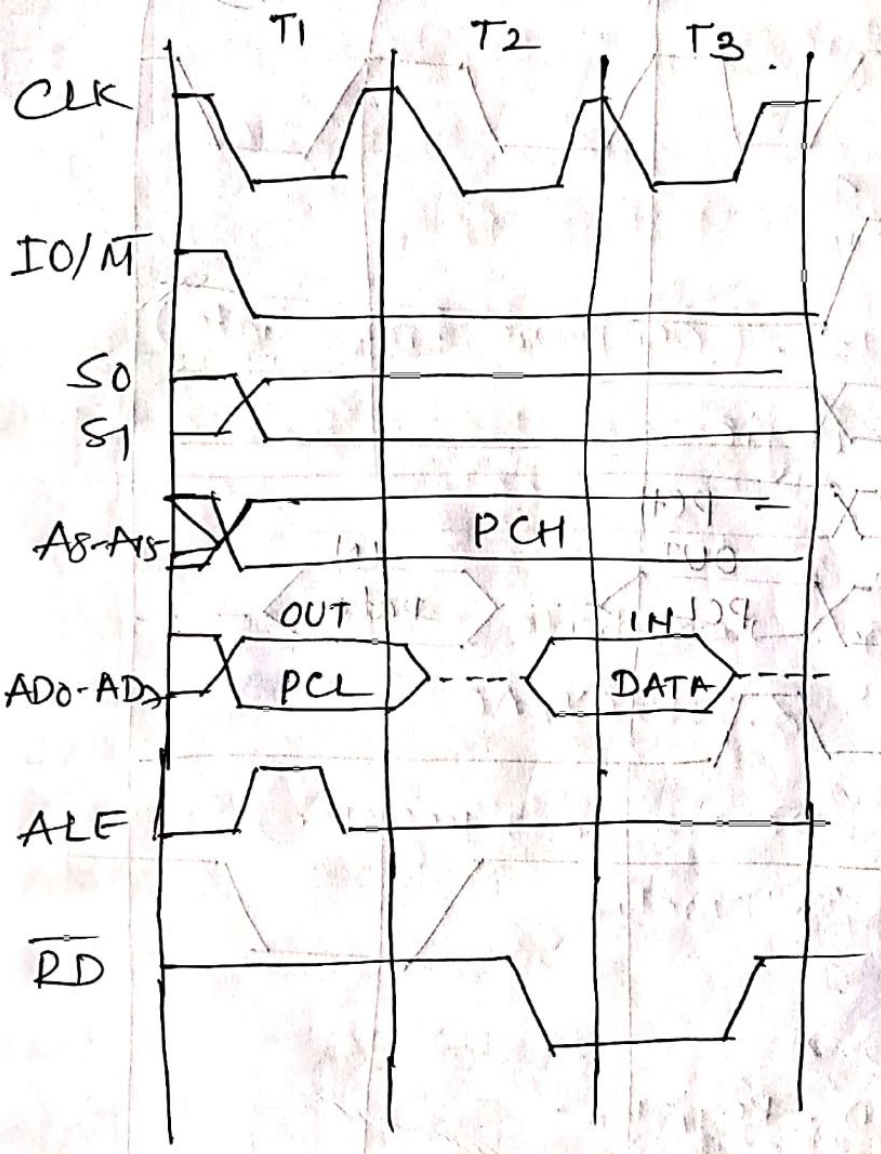
The necessary steps which are carried out in a machine cycle can be represented graphically and such a graphical representation is called timing diagram.

Timing diagram for opcode fetch



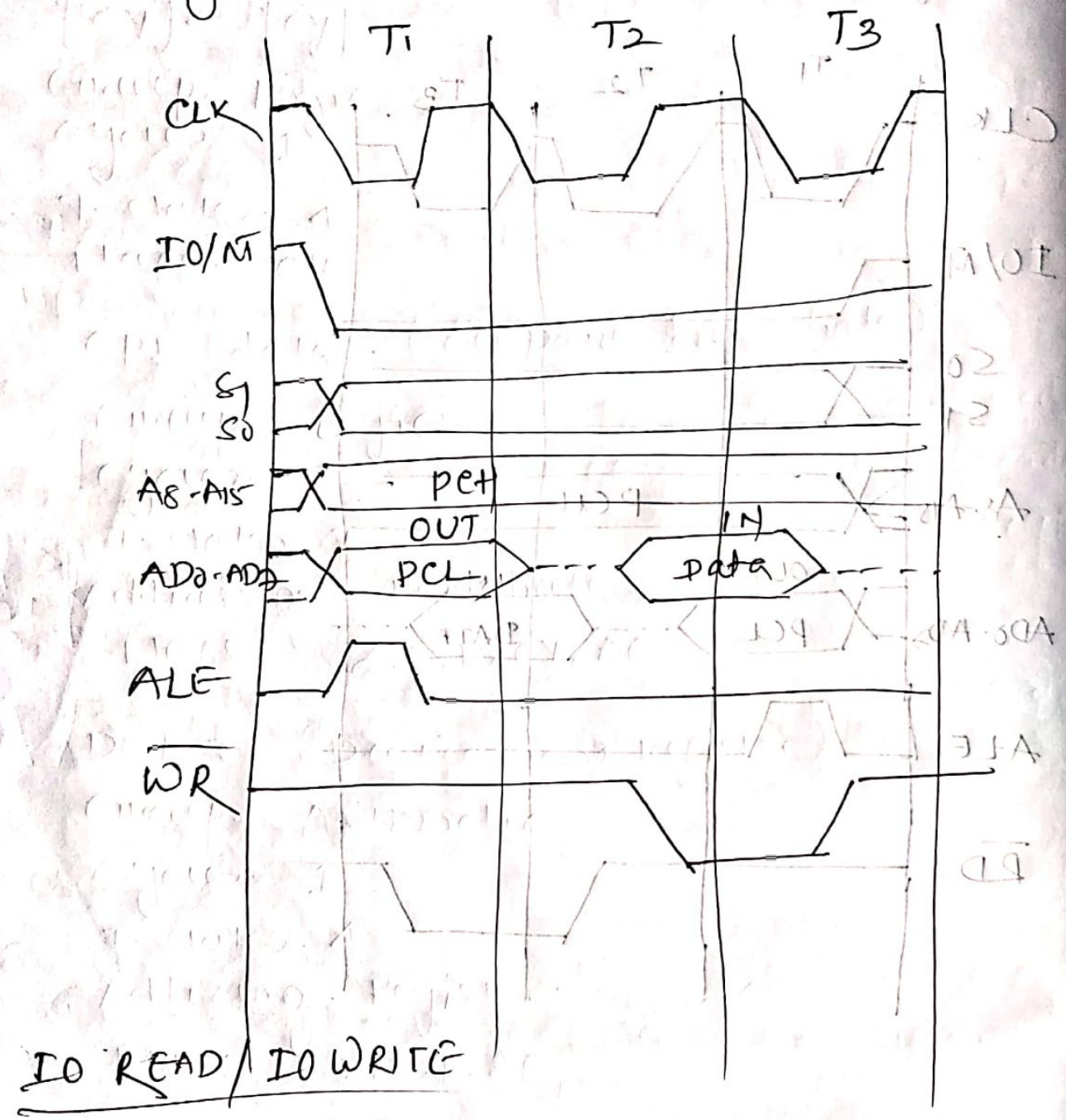
Scanned with CamScanner

Memory Read



Scanned with CamScanner

Memory Write



IO READ / IO WRITE

Scanned with CamScanner

Effect of addressing mode on executing timing & conditions flag

ACI data . Add immediate with carry.

[A] ← [A] + [Byte 2] + [CY].

Group. Arithmatic
cycles - 2.
T-states - 7
Addressing - Immediate
Flags. Z, S, P, CY, AC

→ ADD r. - Add register.

[A] ← [A] + [r].

Group. Arithmetic
cycle - 1
T-states - 4
Addressing. Register
flags. Z, S, P, CY, AC

→ ADD M - (Add memory)

[A] ← [A] + [M_HL]

Group. Arithmatic
cycle - 2.
T-states. 7
Addressing - Register Indirect
flags. Z, S, P, CY, AC

→ CMA - (complement accumulator)

[A] ← [$\bar{A}$]

Groups. Logic
cycle - 1
T-states. 4
flags. None

Scanned with CamScanner

→ CMC - (complement carry)

[CY] ← [CY]

Group - Logic

Cycles - 1

T-states - 4

Flags - CY

→ CPI data - (compare immediate)

Group - Logic

cycles - 2.

T-states - 7

Addressing - Immediate.

flags - Z, S, P, CY, AC

→ DCR r. (Decreament register)

Group - Arithmetic

cycles - 1

T-states - 4

Addressing - Register.

flags - Z, S, P, AC

→ DCR M (Decreament memory)

$[M]_{HL} \leftarrow [M_{HL}] - 1$

Group - Arithmetic

cycles - 3

T-states - 10.

Addressing - Register Indirect

flags - Z. S, P. AC

Scanned with CamScanner

→ MVI r, data - Move data to register immediately
Group - Data transfer.
Cycles - 2.
T-states - 7
Addressing - Immediate.
flags - None.

→ MVI M, data. (Move data to memory immediately)
Group - Data transfer
Cycles - 3.
T-states - 10.
Addressing - Immediate / Register indirect
flags - None.

Scanned with CamScanner

8085-A programming

The set of executable instructions through which a computer performs a specific task is known as a program and a set of such instructions is called software.

→ The codes which are understandable by the computer are collectively called programming language.

→ programming languages are of the following types.

i) Machine level language.
ii) Assembly language.
iii) High level language.

Assembly language

In assembly language, a programmer can comfortably write a program in specific alphanumeric symbols instead of binary codes.

→ ADD for addition, SUB for subtraction, CMP for comparison, LDA for Load accumulator, these symbols are called mnemonics.

→ The programs written with mnemonics are called assembly language programs.

4.2 Hand Assembler

This is an assembler where the whole process of assembling is done manually. firstly, the assembly language program is written with mnemonics and necessary labels for looping back and jumping forward. Now the equivalent hexadecimal op-codes of the

Scanned with CamScanner

mnemonics are written usually with corresponding memory locations. where the machine language program will be written.

→ The sequence is maintained identifying the memory location corresponding to the labels for forward and backward looping.

→ There hexadecimal opcodes are entered into the target memory locations through hexadecimal keyboards and the program is run through run command.

→ No editing facility is available in hand assembly program development & debugging become very tedious.

Cross assembler

The cross assembler helps to develop the equivalent executable hexadecimal codes corresponding to an assembly language program.

→ A microprocessor ICE is connected to a PC through serial port.

→ Interfacing between a microprocessor and a PC with which the ICE is connected.

→ Any hexadecimal program code can be downloaded from secondary storage device eg HDD, FDD to the primary memory of the processor.

Scanned with CamScanner

→ In this case during developing assembly language the editing, modifying and debugging can be done through cross assembler

One pass assembler:

→ An assembler which compiles as assembly language program only once is known as one pass assembler.

→ These assemblers normally have some techniques to take the forward, backward loop reference into account.

→ One pass assembler is definitely faster than two pass assembler as it compiles the program only once.

Two pass assembler:

An assembler which passes through an assembly language program twice during compilation is called two pass assembler.

→ This assembler is free from any difficulty with forward and backward loop references.

→ During the first pass it identifies all labels, during the second pass it generates the machine codes for each instruction and assigns address to each of the labels by counting their position from the starting address.

Scanned with CamScanner

Advantage of assembly language

1) The computation time is less

2) Assembly language program can produce result much more faster than equivalent high level language programs.

Advantage of high level language

1- Instructions and syntaxes are normally standard and easy to understand.

2- The work of program is high level language is comparatively much more easier and faster than that in assembly language

3 - High level language programs are generally portable i.e they can run on any computer or microcomputer.

4- Program documentation becomes easier.

5- High level language is problem oriented

Operating system software:

→ Microcomputer can perform through its hardware as per the program guideline as directed through software.

→ The operating system software is a collection of programs with the help of which the programmer or user can access the computer hardware to run his developed program.

→ The operating system s/w includes operating system BIOS, assembler, compiler, editor, etc.

⇒ It controls the overall operation of a computer and provides necessary interface with the programmers.

Modular programming:

→ When a program becomes very long, it is divided into smaller programs or modules. These smaller programmes can be written, tested, & debugged separately.

→ This technique of programming is called Modular programming.

Advantage

→ It is easy to write, test & debug a module.

→ Modules of common nature are prepared, which can be used at many places.

→ The programmer can divide tasks and use the previously written programs.

→ If a change is to be made, it can be made in particular module and the entire program is not affected.

Scanned with CamScanner

Disadvantage:

1) combining modules together may become difficult.
2) It needs careful documentation as it may affect the other parts of the program.
3) Modular programming requires extra time & memory space.

Structured programming:

→ Structured programming is technique developed for writing a program using specially defined program logics.

→ These technique was previously adopted for writing very long programs exceeding 1000 lines.

→ But this method can also be used for writing shorter programs for microprocessors.

→ A program can be written using three types of basic structures as described below

1- Sequential structure
2- A conditional structure
3 - A loop structure.

Micro programming:

The CPU performs the following tasks to execute an instruction.

1- It places the content of the program counter on the address bus to fetch the opcode of the instruction from the memory.
2- It reads the opcode from the memory and places it on the instruction register.

Scanned with CamScanner

3. It decodes the instruction.
4. It takes other data, if necessary from the memory or other register to execute the instruction.
5. It executes the instruction.
6. It finally incremented the content of the program counter so that it will be able to fetch the next instruction.

Stack & subroutines

During the execution of a program sometimes it may be necessary to store the contents of certain registers because the registers may be required for some other operation in subsequent steps.

→ Afterwards the data may be required by the registers for further processing.

→ This can be achived by saving data on the 16 bit registers in memory stack using PUSH commands and retrieving data back from memory stack to 16-bit registers using POP command.

→ Data saving & retreiving from stack takes place in the form of first out (LIFO) principle.

Subroutines

→ In intel 8085 A programming language several mathematical fuctions are not available.

→ Small programs can be written for performing certain functions. These functional programs can be used by main programs several times as per necessity of situations by using the concept of subroutines.

Scanned with CamScanner

→ These smaller functional programs are called subroutines.

→ Subroutines are used to avoid the repetition smaller functional programs are called Subroutines.

→ Subroutines are used to avoid repetitive smaller program blocks in a main program.

→ The instructions CALL & RET are used with inbuilt PUSH-POP operations for accessing a subroutine and return back to the main program after finishing the subroutine job.

Scanned with CamScanner

chapter-05

Memory & I/O Interface

Primary memory:

The primary memory is that memory which can be directly accessed by the CPU of the processor

→ Therefore primary memory can be defined as that memory which forms the address space of the CPU.
→ If a processor having n number of address lines can access 2^n memory locations.
→ The total primary memory is divided into two separate sections.
i) The ROM area ii) The RAM (Read/Write) area.

RAM

→ During computation, it is sometimes required to store intermediate results for later use.
→ The storage space available in the general purpose registers of the CPU being inadequate.
→ The processor stores majority of the results outside the CPU in a special type of memory locations called Read/write memory or Random access memory.
→ RAM is a volatile memory i.e the contents are lost when the power is switched off.
→ RAM are of two types. i) Static RAM
ii) Dynamic RAM.
→ In static RAM, the data once written in the memory location remains unchanged unless new data is written in to the same memory location.
→ The storage elements used in static RAM are flip-flops.

Scanned with CamScanner

→ In dynamic RAM, the basic storage element are capacitors.

→ The storage elements in a dynamic RAM have to be refreshed from time to time.

Advantage

low power consumption.

disadvantage

special circuitry is needed to precharge refresh the memory.

ROM

→ The non-volatile memory is generally called Read only memory

→ It is not possible to overwrite the data stored in ROM. It can be read hence Read only

→ ROM stored permanent information.

PROM:

→ It can be programmed by the user with the help of a PROM programmer.

→ It generally consists of fusible links which can be burned or melted by a "PROM burner circuit".

→ A fused link corresponds to a 0 or Low and an open link corresponds to a 1 or high.

Scanned with CamScanner

EPROM

The programs information stored in it can be erased by exposing the memory to ultraviolet light through a window on the chip which is covered during normal operations.

→ EPROM uses MOS charge storage technology for storing information.

EEPROM :-

In EEPROM the information is erased by applying electrical signals instead of ultra violet light.

→ The data in memory locations can be selectively erased for EEPROM.

Secondary memory :-

When it is necessary to store more data than the maximum capacity of primary memory secondary memory is employed.

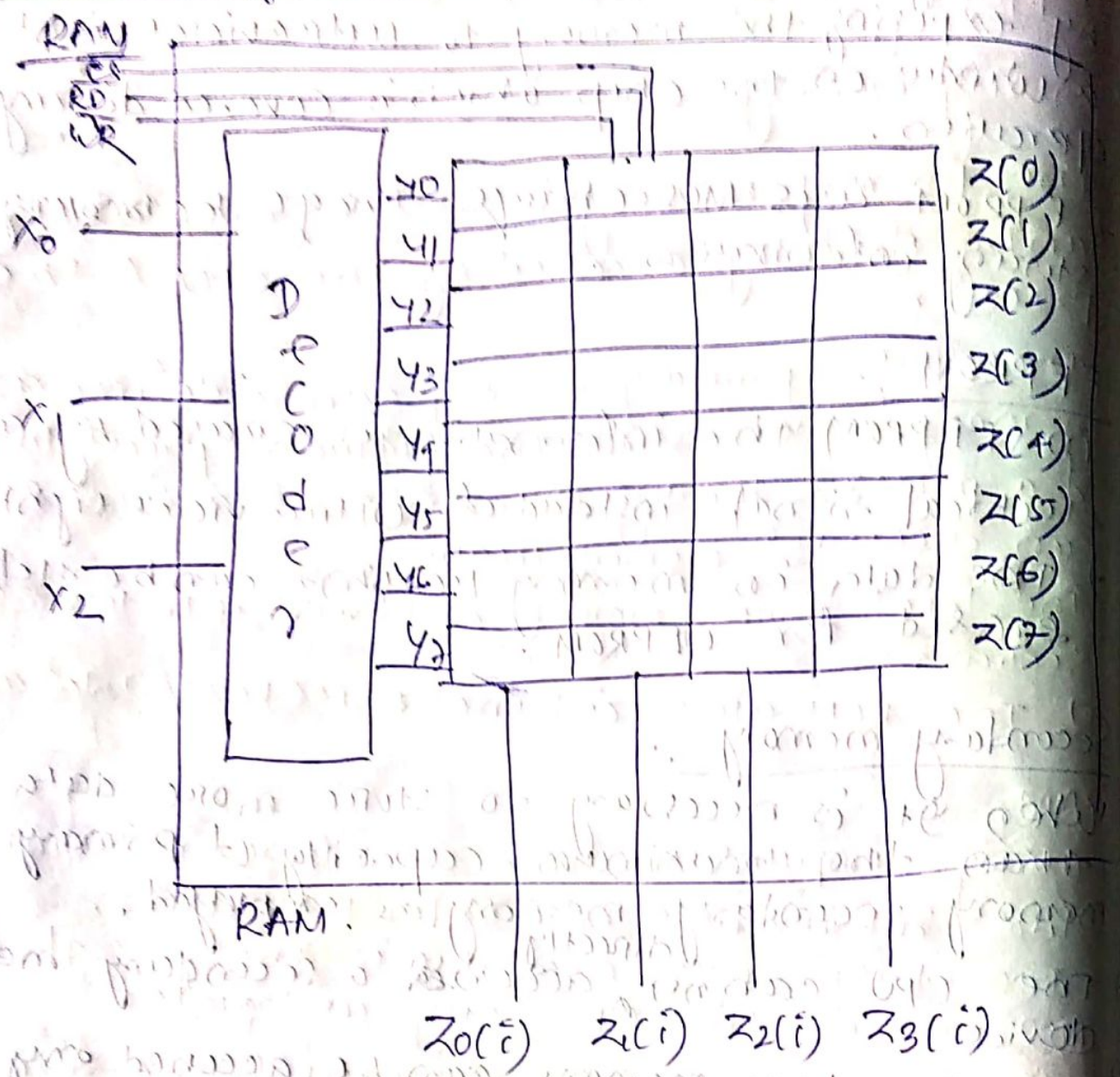
→ The CPU cannot accessed a secondary memory directly device.

→ The secondary memory can be accessed only through I/O ports or in a serial format using proper software & Hardware

→ Magnetic tapes, floppy disks, CD-ROM, hard disk etc. are common examples of secondary memory

Scanned with CamScanner

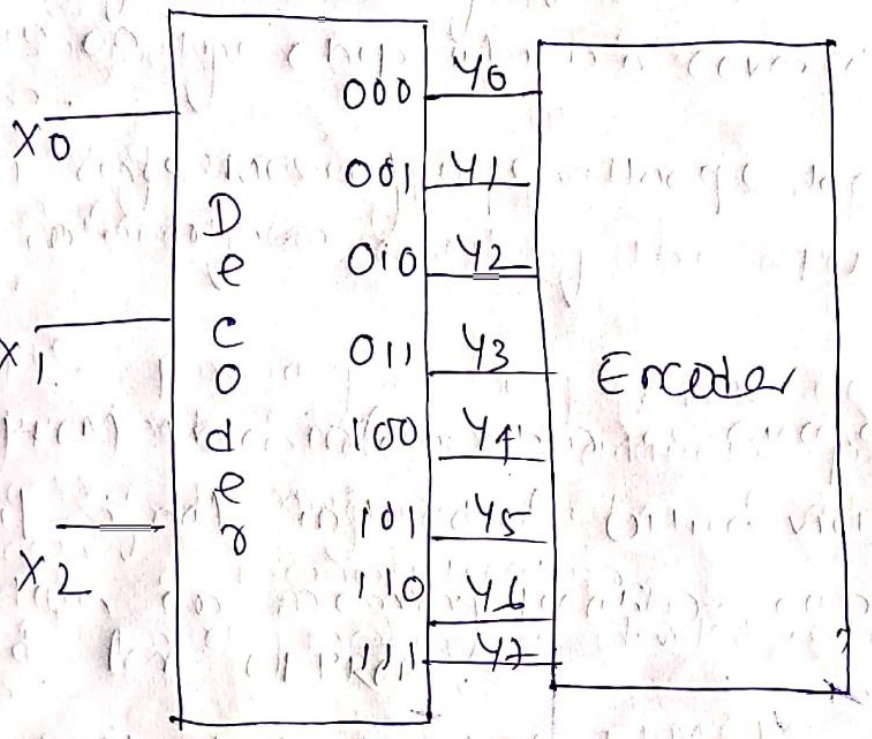
Internal organization of RAM & ROM



→ RAM requires an active low WR which has to be activated when data is written into the RAM.

→ A 4-bit eight word RAM is shown in fig 6-6. Each word can be assumed to be a 4-bit register.

ROM

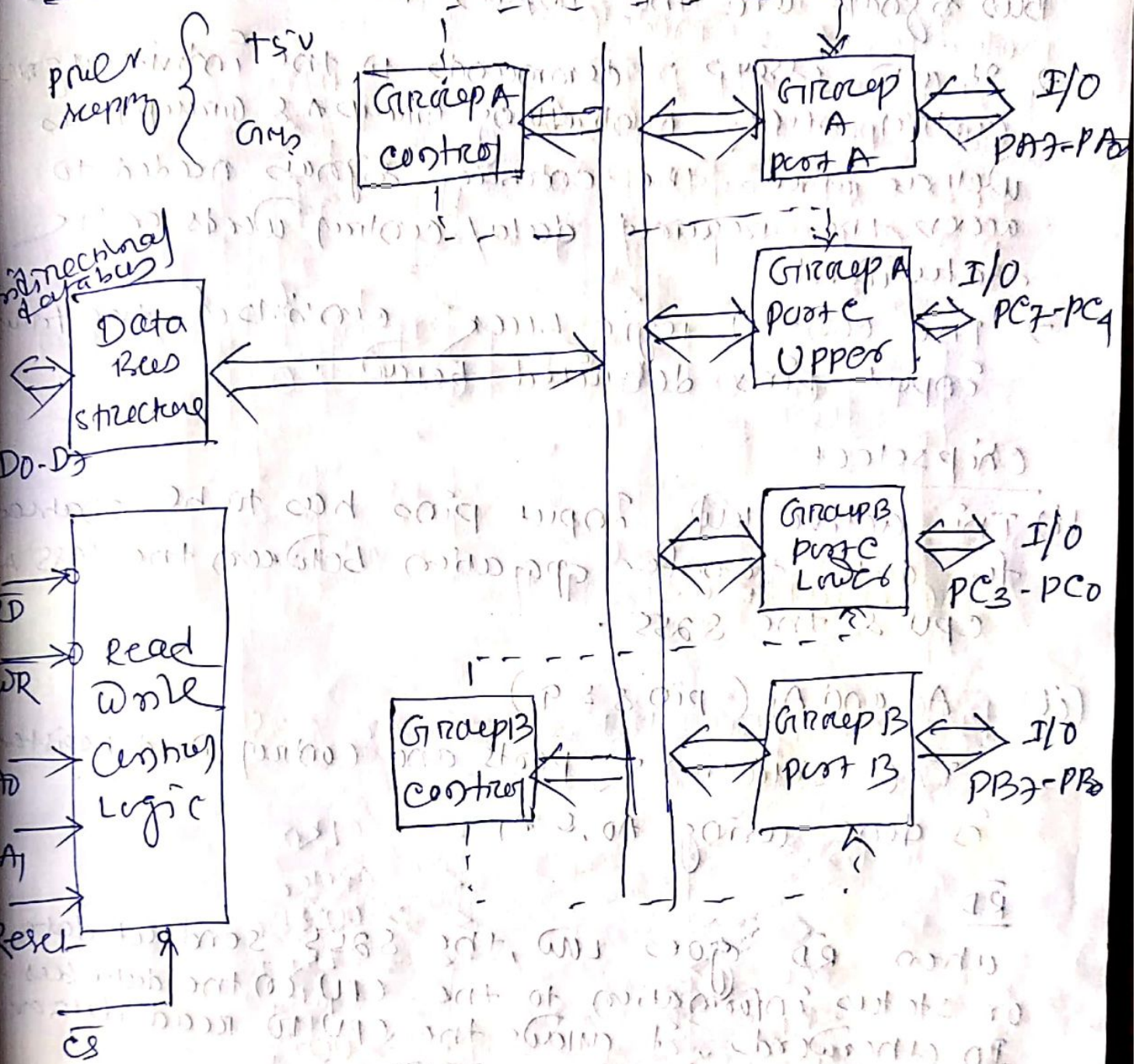


Decoded 8 byte-ROM.

Scanned with CamScanner

PPI Intel - 8255

Functional Block diagram:



→ Intel 8255 chip is used as a general purpose device for interfacing a peripheral device to the microprocessor system bus.

Data bus buffer

The interface bidirectional data bus buffer is used to interface the 8255 to the system data bus.

→ data can be read from and written into the buffer through IN & OUT instructions of the CPU.

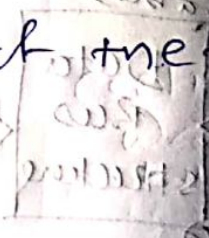
Scanned with CamScanner

Control Logic

The control logic block is used to accept control bus signals and the inputs from the address bus.

→ It also issues commands to the individual group control blocks controlling Group A & Group B as well as issues the control signals needed to access the required data/control words or the status word.

→ The control logic block consists of the following input pins described below.



Chip select

(i) This active low input pins has to be enabled for data transfer operation between the 8085A CPU & the 8255.

(ii) A0 and A1 (pin 8 & 9)

The selection of ports and control word register is done using A0 & A1

RD

When RD goes low, the 8255 sends out data or status information to the CPU on the data bus. In otherwords, it allows the CPU to read data or status information from 8255.

WR

When WR goes low, the CPU writes data or control word into 8255.

→ The CPU writes data into the output port of 8255 and the control word into the control word register.

Scanned with CamScanner

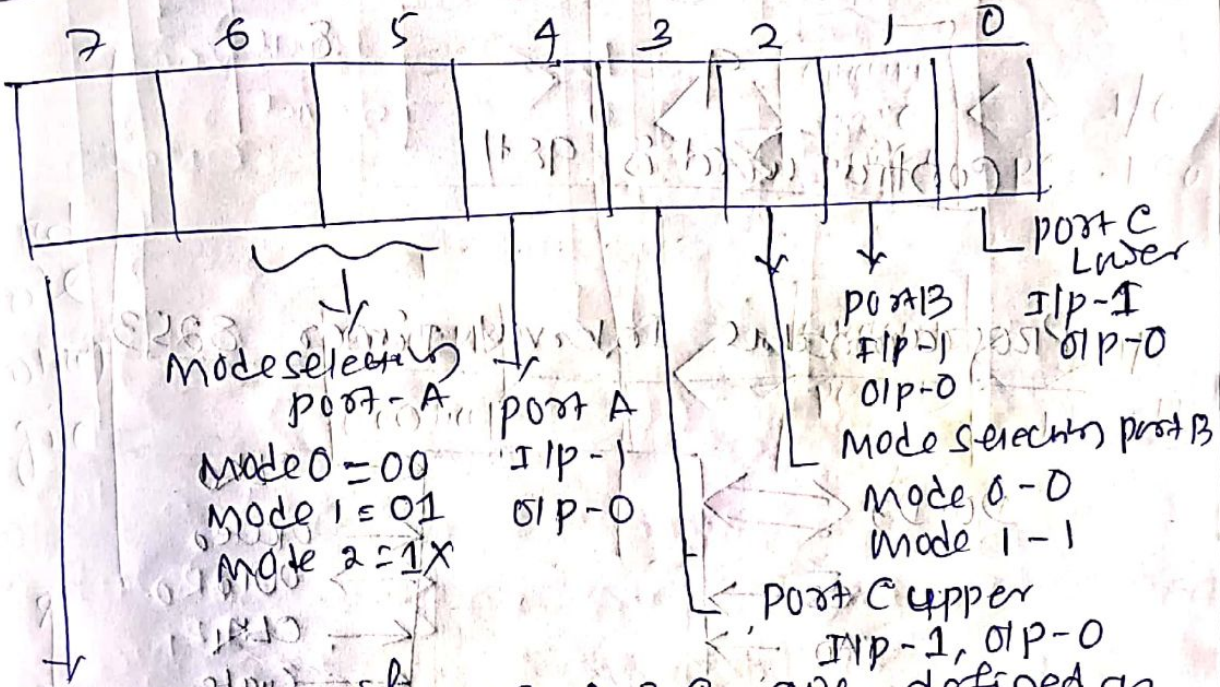
operating modes of 8255

The Intel 8255 has the following three modes of operation

Mode 0 - simple input/output
mode 1 - strobed input/output
mode 2 - bidirectional port

→ The 8255 has two 8 bit ports (port A & B) and two 4 bit port (port C upper & port C Lower)

control word



It is set to 1 if port A, B, C are defined as Input/output port.

It is set to 0 if the individual pins of the port C are to be set or reset

Make control word when the ports of the intel 8255 are defined as follows.

port A - I/P port
Mode of port A = mode 0
port B as O/P port
Mode of port B - mode 0
port C upper - I/P port
port C lower - O/P port

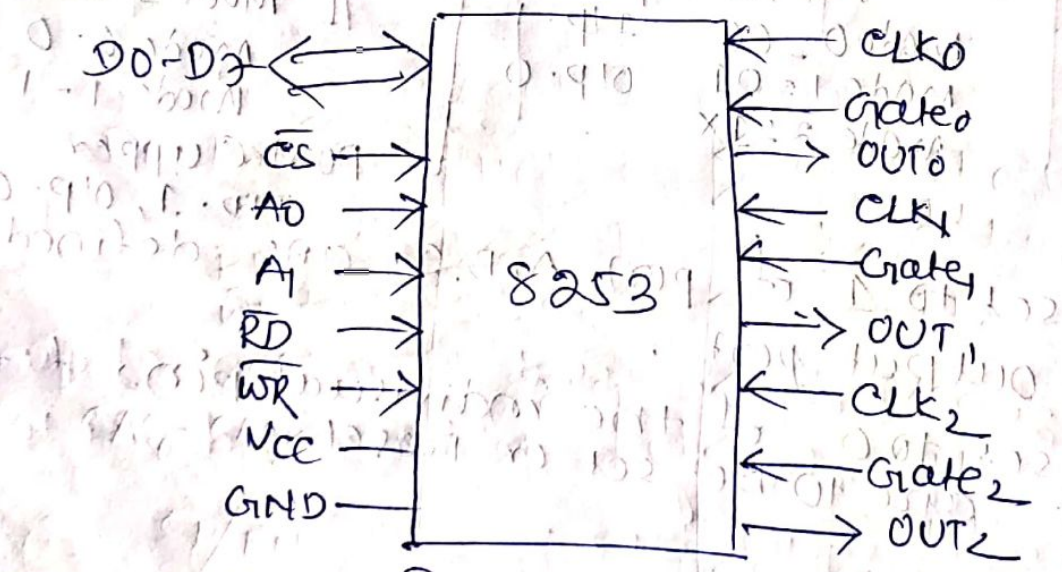
7	6	5	4	3	2	1	0
1	0	0	1	1	0	0	0

control word bit

9 8

Control word is 98H

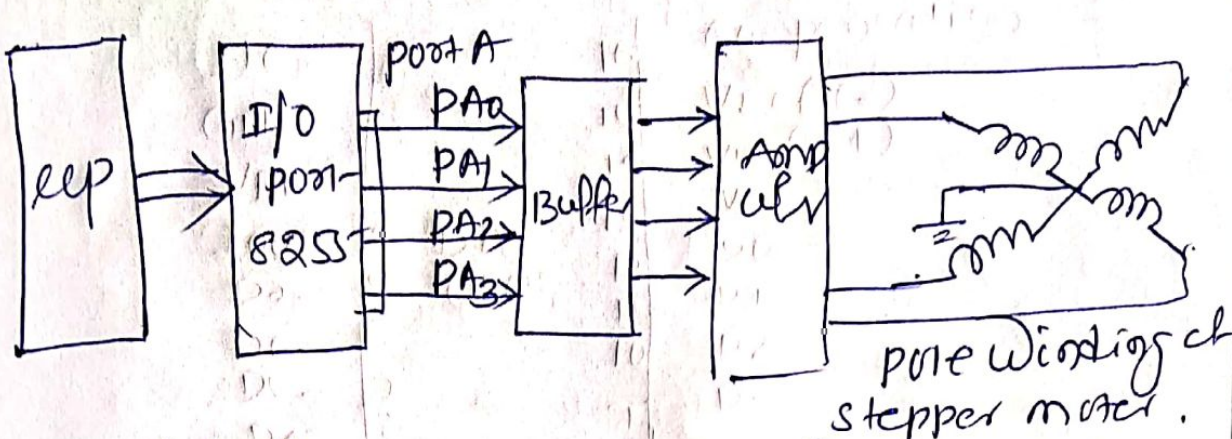
Programmable Interval Timer 8253



→ The 8253 is 24 pin IC and operates at 5Vdc. It contains three independent 16 bit counters.

→

Speed control of DC motor.



pole windings of stepper motor.

general interfacing connections. for a stepper motor. 12V dc supply is used to energise the poles. Pulse sent by the microprocessor switches rated ~~supply~~ voltage to the windings of the desired poles.

→ A delay subroutine is incorporated in the program, after energising one set of pole windings some delay is provided, then the power supply is switched on to the other set of the pole windings.

→ This delay time governs the speed of the motor.

Scanned with CamScanner