UNIT-I

INTRODUCTION TO 8-BIT MICROPROCESSOR

- **Microcomputer and Microprocessors :-**

Microprocessor & microcomputer are not the same. The main ~~features~~ difference is that the microprocessor is a computer processor is an integrated circuit chip and the microcomputer is a small, relatively inexpensive computer. The microcomputer holds a ~~microcomputer~~ microprocessor inside and not the other way around.

MICROPROCESSOR :-

This is the central processing unit (CPU) of a computer system that conducts arithmetic and logic tasks such as adding, subtracting, moving numbers from one region to another and comparing two numbers. It is frequently referred to as a processor, a central processing unit or a logic chip. When the computer is turned on, it is essentially the engine or the computer's brain that gets things moving. It is a programmable, multifunction device that combines the operations of a CPU on a single integrated circuit (IC).

Origin of the Phrase :-

The Term "micro~~com~~ processor" was used by Viatron Computer systems to describe the unique integrated circuit used in their System 21 small computer system, which was released in 1968.

History :-

A microprocessor is a computer processor that contains the operations of a central processing unit on a single integrated circuit (IC) or a few integrated circuits.

The microprocessor is a multifunctional, clock driven, register-based digital Integrated circuit that receives binary data as input, according to guidelines stored in its memory and processes it and outputs the results.

Microprocessors feature both Combinational logic and Sequential digital logic. Microprocessor work with numbers and symbols represented by the binary number system.

Integrating an entire CPU onto a single or a few integrated circuits decreased the cost of processing power significantly. Integrated circuit processors are mass-produced using highly automated techniques, resulting in a low unit price.

Small computers were developed before microprocessors utilizing racks of circuit boards with many medium and small-scale integrated circuits.

A microprocessor is a computer control unit capable of managing all arithmetic logical unit (ALU) activities. Other actions that a microprocessor may do include computational processes such as addition / subtraction, internal processing, device terminal connectivity and I/O management

MICROCOMPUTER :-

It is an electronic device with a central processing unit that is a microprocessor. It was once a frequent word for personal computers, specifically any family of compact digital computers with a CPU housed on a single integrated semiconductor chip.

This is used for exchanging data of

this sort with peripherals device (ex→ keyboard, video display and printer) and auxiliary storage units, The earliest micro-computers marketed in the mid-1970s contained a single chip on which all CPU, memory and interface circuits were integrated.

As large-scale integration and then very-large-scale integration progressively increased the number of transistors placed on one semiconductor chip, microcomputers processing capacity using such single chips grew commensurately.

- Origin of the Phrase :-

In the early 2000s everyday use of the word "microcomputer" declined significantly from its peak in the mid-1980s. The phrase is most commonly connected with the most popular all in one 8-bit home computers and small-business CP/M-based microcomputers.

In everyday usage "microcomputer" has mainly been substituted by the phrase "Personal computer" or "PC" which specifies a computer that has been created for one

Page No.:
Date :

user at a time, a term first made up in 1959.

Page No.:
Date:

MICROPROCESSOR:- The microprocessor communicates and operates in the binary numbers 0 & 1. called bits Each microprocessor has a fixed set of instructions in the form of binary patterns called a machine language. However ~~Here~~ it is difficult for humans to communicate in the language of 0s & 1s. Therefore, the binary instructions are given abbreviated names, called mnemonics, which from the assembly language for a given microprocessor.

The advantages of assembly language are compared with high-level languages (such as BASIC, C, C++ and Java)

★ A microprocessor is a multipurpose, programmable, clock driven, register-based electronic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions and provides results as output.

A typical programmable machine can be represented with four

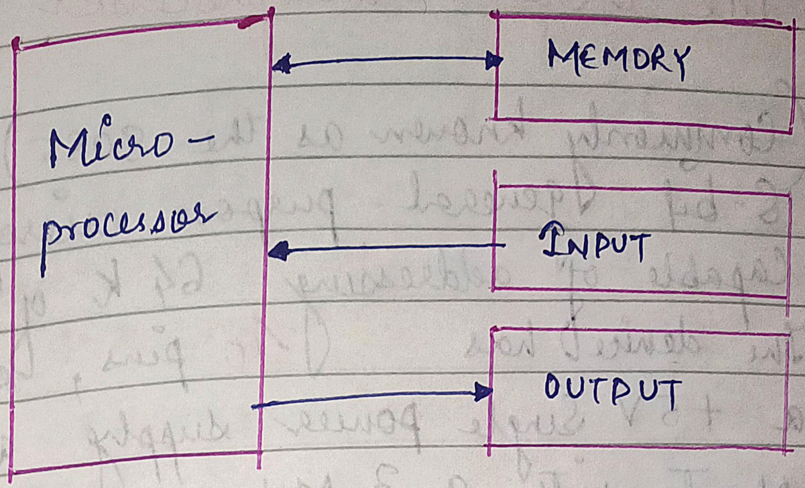


figure → A programmable Machine

Components, microprocessor, memory, i/p & o/p as shown in above figure. These four components work together or interact with each other to perform a given task: Thus they comprise a system. The Physical components of this system are called Hardware. A set of ~~instructio~~ instructions written for the microprocessor to perform a task is called a program. and a group of program is called Software.

Each microprocessor recognizes and processes a group of bits called word and microprocessor are classified according to their word length. for ex. a processor with an 8-bit word is known as an 8-bit microprocessor.

The 8085 Microprocessor :-

The 8085A (Commonly known as the 8085) is an 8-bit general-purpose microprocessor capable of addressing 64 K of memory. The device has 40 pins, requires a +5V single power supply and can operate with a 3-MHZ single-phase clock. The 8085-2 version can operate at the maximum frequency of 5 MHZ. The 8085 is an enhanced version of its predecessor, the 8080A, its instruction set is upward-compatible with that of the 8080A, meaning that the 8085 instruction set includes all the 8080A instructions plus some additional ones.

figure shows the logic pins pinout of the 8085 microprocessor. All the signals can be classified into six groups.

(1) Address bus
(2) Data Bus
(3) Control & status signals
(4) Power supply and frequency signals
(5) Externally initiated signals
(6) Serial I/O ports

Date :

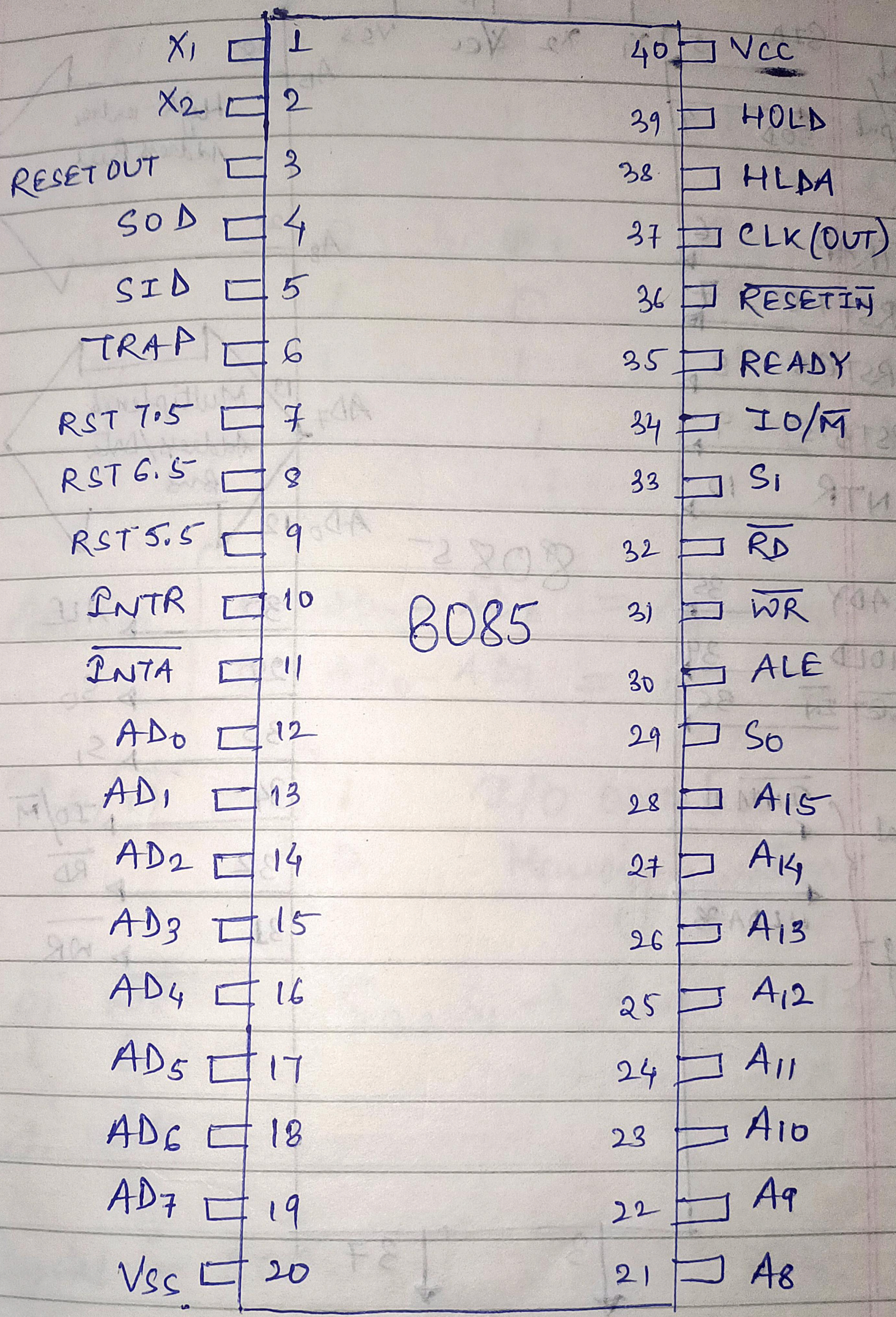
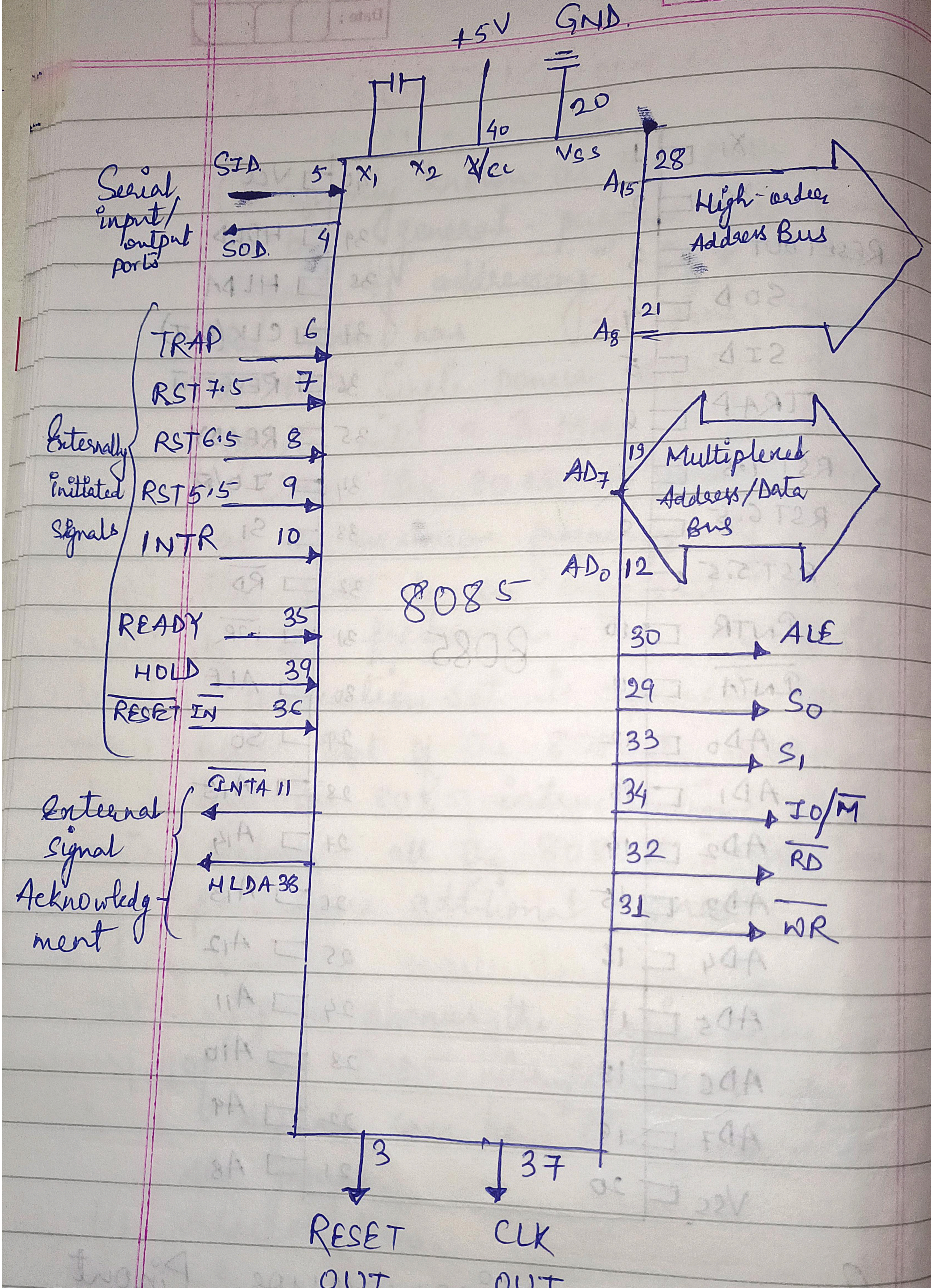


figure → The 8085 Microprocessor Pinout and Signals.


+5V
GND.
20
40
X1
X2
Vcc
Vss
28
A15
High-order Address Bus
21
A8
SID
5
Serial input/output ports
4
SOD
TRAP 6
RST 7.5 7
RST 6.5 8
RST 5.5 9
INTR 10
READY 35
HOLD 39
RESET IN 36
Externally initiated signals
19
AD7
Multiplexed Address/Data Bus
AD0 12
8085
30 ALE
29 S0
33 S1
34 IO/M
32 RD
31 WR
INTA 11
HLDA 38
External signal Acknowledgment
3
37
RESET OUT
CLK OUT

Page No.:
Date :

$IO/\overline{M}$	S_0	S_1	STATUS
0	0	0	HALT
0	0	1	M - write
0	1	0	M - read
1	0	1	IO - write
1	1	0	IO - read
0	1	1	Opcode fetch
1	1	1	INT - ACK

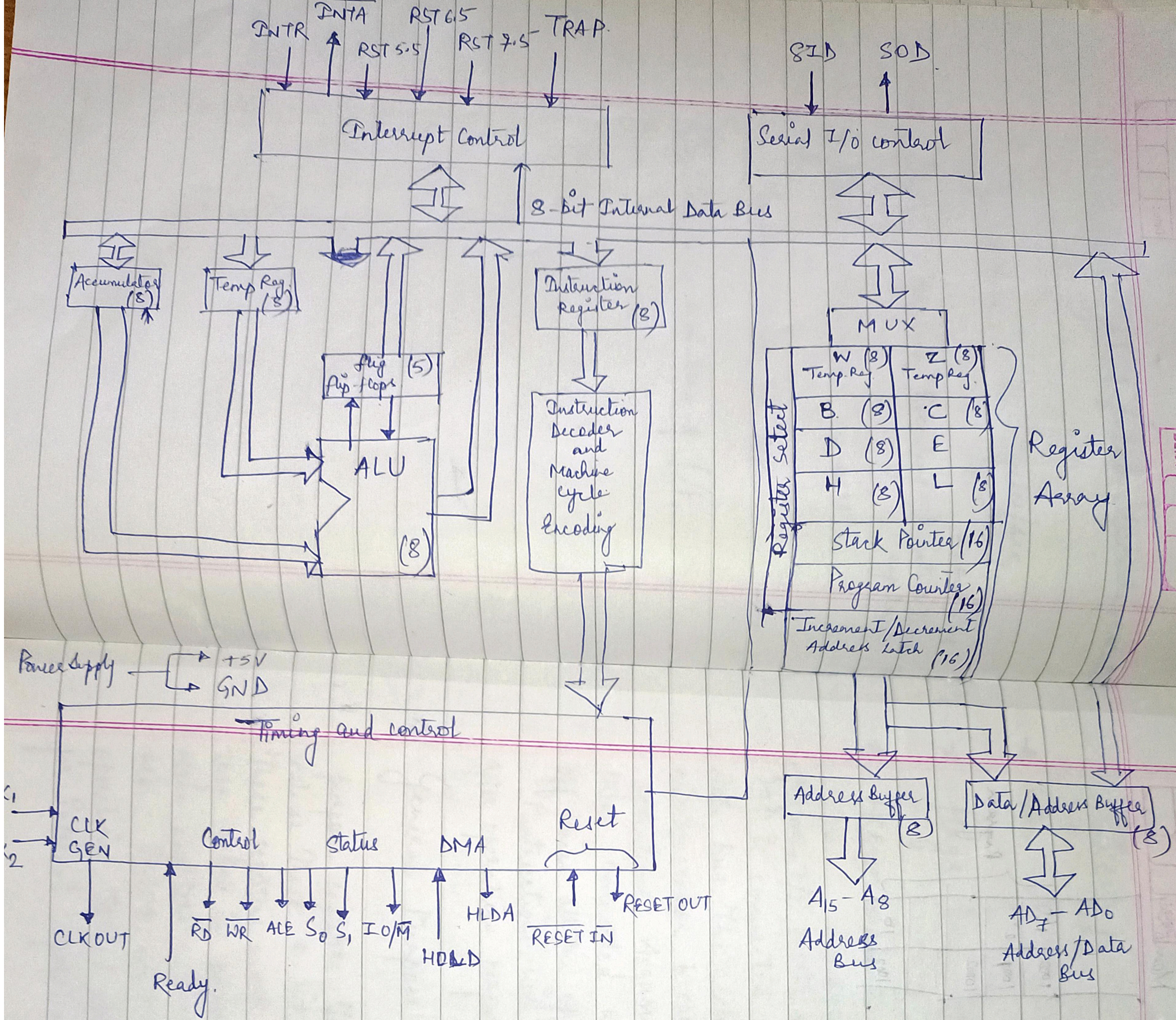


fig:- functional Block diagram & Internal Architecture of The 8085 microprocessor.

Description of 8085 Microprocessor:

* **The ALU :-** By book

* **flags :-** By Book.

* **Accumulator:-**

Accumulator is used to perform I/O, arithmetic and logical operations. It is connected to ALU and the internal data bus. The accumulator is the heart of the microprocessor because for all arithmetic operations Accumulator's 8 bit pin will always there connected with ALU and in most off times all the operations carried by different instructions will be stored in the accumulator after operation performance.

* **General Purpose Registers :-**

There are six general purpose registers. These registers can hold 8-bit values. These 8-bit registers are B, C, D, E, H, L. These registers work as 16 bit registers when they work in pairs like B-C, D-E and H-L. Here registers W and Z are reserved registers. we can't use these registers in arithmetic operations. It is reserved for microprocessor for internal operation like swapping two 16 bit

Page No.:
Date :

numbers. We know that to swap two numbers we need a third variable hence here W-Z registers pair works as temporary registers and we can swap two 16-bit numbers using this pair.

☆ Program Counter :- Program Counter holds the address value of the memory to the next instruction that is to be executed. It is a 16-bit register.

for example :- Suppose current value of program counter [PC] = 4000H

(It means that next executing instruction is at location 4000H. After fetching program counter (PC) always increments by +1 for fetching of next instruction.

☆ Stack Pointer :- In stack, the content of the register is stored that is later used in the program. It is a 16-bit special register. The

Page No.:
Date :

stack pointer is part of memory but it is part of stack operations, unlike random memory access. Stack pointer works in a continuous and contiguous part of the memory. Whereas program counter (PC) works in random memory locations. This pointer is very useful in stack-related operations like PUSH, POP and nested CALL requests initiated by microprocessor. It reserves the address of the most recent stack entry.

☆ Temporary Register :- It is an 8-bit register that holds data values during arithmetic and logical operations.

☆ Instruction register and decoder :- It is an 8-bit register that holds the instruction code that is being decoded. The instruction is fetched from the memory.

☆ Timing & control unit :- The timing & 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 five interrupt signals in 8085 microprocessor: INTR, TRAP, RST 7.5, RST 6.5 & RST 5.5.

Priorities of Interrupts: TRAP > RST 7.5 > RST 6.5 > RST 5.5 > INTR

★ Address Bus & Data Bus: The data bus is bidirectional and carries the data which is to be stored. The address bus is unidirectional and carries the location where data is to be stored.

Date :

☆ Serial Input / Output Control :-

It controls the serial data communication by using serial input data and serial output data.

☆ The flow of an instruction Cycle in 8085 Architecture :

(1) Execution starts with program counter. It starts with program execution with the next address field. It fetches an instruction from the memory location pointed by program Counter.

(2) for address fetching from the memory, multiplexed address / data bus acts as an address bus and after fetching instruction this address bus will now acts as a data bus and extract data from the specified memory location and send this data on an 8-bit internal bus. for multiplexed address / data bus Address Latch Enable (ALE) pin is used.

If ALE = 1 (Multiplexed bus is Address Bus otherwise it acts as Data Bus.

Page No.:
Date :

(3) After data fetching data will go into the instruction register it will store data fetched from memory and now data is ready for decoding, so for this instruction decoder register is used.

(4) After that timing and control signal circuit comes into the picture. It sends control signals all over the microprocessor to tell the microprocessor whether the given instruction is for READ/WRITE and whether it is for MEMORY/I-O Device activity

(5) Hence according to timing & control signal pins, logical and arithmetic operations are performed and according to that data fetching from the different registers is done by a microprocessor and mathematical operation is carried out by ALU, and ~~ALU~~ according to operations flag register changes dynamically.

(6) With the help of serial I/O data pin (SID or SOD pins) we can send or recei

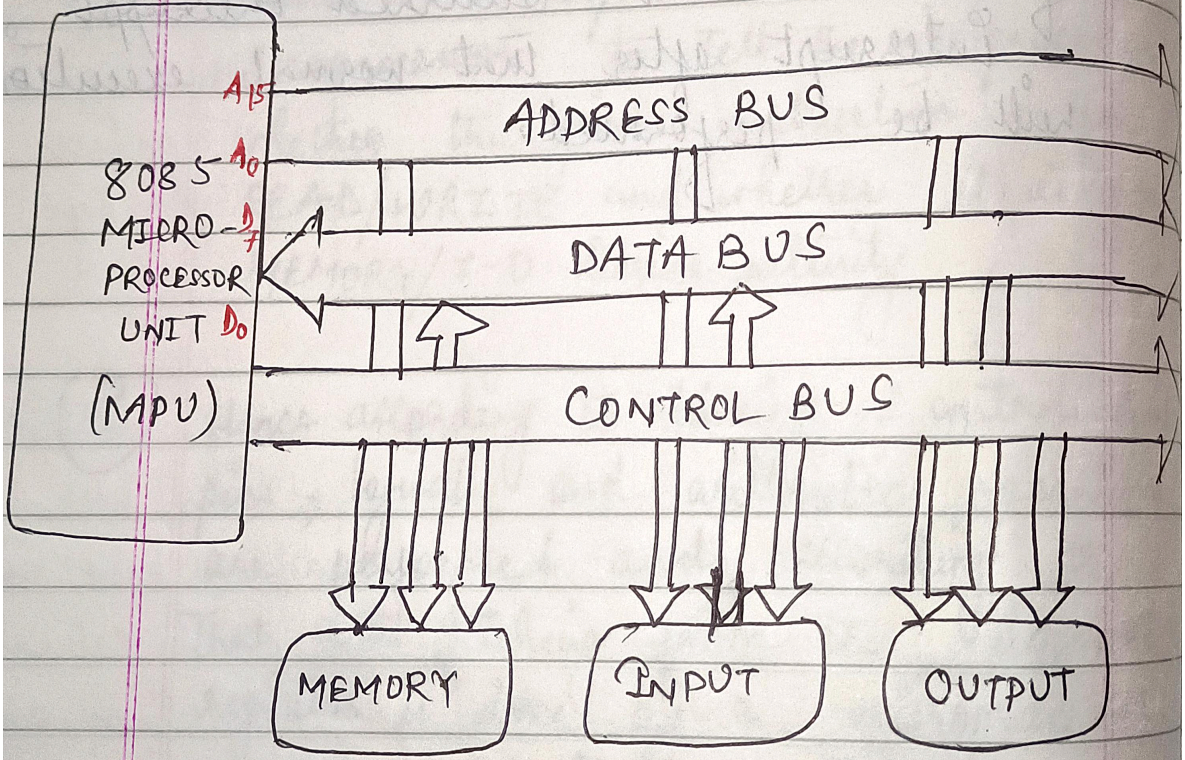
input/output to external devices in this way execution cycle is carried out.

(7.) While execution is going on if there is any interrupt detected then it will stop execution of the current process and invoke interrupt service Routine (ISR) function which will stop the current execution and do execution of the current occurred ~~interrupt~~ interrupt after that normal execution will be performed.

INTERNAL BUS ORGANIZATION :-

Bus is a group of conducting wires which carries information, all the peripherals are connected to microprocessor through Bus.

↳ Diagram to represent bus organization system of 8085 microprocessor.



There are three Types of buses:-

(1) Address Bus:- It is a group of conducting wires which carries address only.
Address bus is unidirectional because

data flow in one direction, from microprocessor to Input/out devices (i.e. out of microprocessor)

Length of address bus of 8085 microprocessor is 16 bit (i.e., four hexadecimal digits), ranging from 0000H to FFFFH. The microprocessor 8085 can transfer maximum 16 bit address which means it can address 65,536 different memory Location.

The length of the address bus determines the amount of memory a system can address. Such as a system with a 32-bit address bus can address 2^{32} memory locations. If each memory location holds one byte, the addressable memory space is 4GB. However, the actual amount of memory that can be accessed is usually much less than this theoretical limit due to chipset and motherboard limitations.

(2) Data Bus:- It is a group of conducting wires which carries Data only. Data bus is bidirectional because data flow in both directions, from microprocessor to memory or Input/output devices and from memory or Input/output devices to microprocessor. Length of Data bus of 8085 microprocessor is 8-bit (i.e. two hexadecimal digits), ranging

from 00H to FFH.
When it is write operation, the processor will put the data (to be written) on the data bus, when it is read operation, the memory controller will get the data from specific memory block and put it into the data bus.

The width of the data bus is directly related to the largest number that the bus can carry, such as an 8-bit bus can represent 2 to the power of 8 unique values, this equates to the number 0 to 255. A 16 bit bus can carry 0 to 65535.

(3) Control Bus:- It is a group of conducting wires, which is used to generate timing and control signals to control all the associated peripherals, microprocessor uses control bus to process data, that is what to do with selected memory location. Some control signals are:-

- Memory read
- Memory write
- I/O read
- I/O write
- Opcode fetch

Control Signals and Status Signals:-

This group of signals includes two signals ($\overline{RD}$ and $\overline{WR}$), three status signals ($IO/\overline{M}$, S_1 & S_0) to identify the nature of the operation and one special signal (ALE) to indicate the beginning of the operation. These signals are as follows:

ALE:- Address latch Enable:- This is a positive going pulse generated every time the 8085 begins an operation (machine cycle). It indicates that the bits on $AD_7 - AD_0$ are address bits. This signal is used primarily to latch the low-order address from the multiplexed bus and generate a sperate set of eight address lines. $A_7 - A_0$.

$\overline{RD}$:- Read:- This is a Read control signal (Active low)

This signal Indicates that the selected I/O or memory device is to be read and data are available on the data bus.

$\overline{WR}$:- Write:- This is a write control signal (Active low). This signal indicates that the data

Page No.:
Date:

on the data bus are to be written into a selected memory or I/O location.

$IO/\overline{M}$:- This is a status signal used to differentiate between I/O and memory operations. When it is high, it indicates an I/O operation; when it is low, it indicates a memory operation. This signal is combined with $\overline{RD}$ (Read) and $\overline{WR}$ (write) to generate I/O and memory control signals.

S_1 and S_0 :- These output signals, similar to $IO/\overline{M}$, can identify various operations but they are rarely used in small systems. (All the operations and their associated status signals are listed in Table.)

Page No.:
Date :

8085 Machine cycle status and control signals

Machine cycle	Status $IO/\overline{M}$	S_1	S_0	Control Signals
Opcode fetch	0	1	1	$\overline{RD} = 0$
Memory Read	0	1	0	$\overline{RD} = 0$
Memory Write	0	0	1	$\overline{WR} = 0$
I/O Read	1	1	0	$\overline{RD} = 0$
I/O write	1	0	1	$\overline{WR} = 0$
Interrupt Acknowledge	1	1	1	$\overline{INTA} = 0$
Halt	Z	0	0	$\overline{RD}, \overline{WR} = Z$ & $\overline{INTA} = 1$
Hold	Z	X	X	
Reset	Z	X	X	

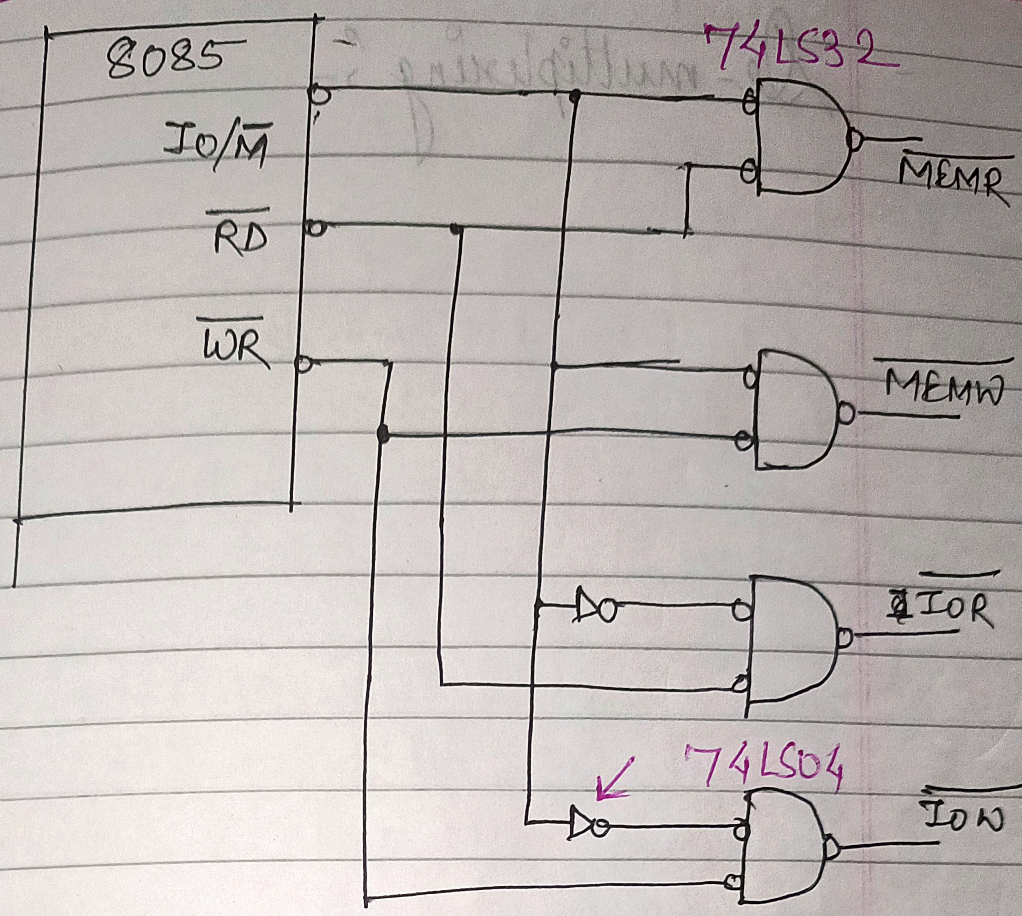
NOTE :- Z = Tri-state (High Impedance)
X = Unspecified.

☆ Generating Control Signals :-

figure shows the $\overline{RD}$ (Read) as a control signal. Because this signal is used ~~for~~ both for reading memory and for reading an input device, it is necessary to generate two different Read signals :- one for memory and another for input. Similarly, two seperate write signals must be generated.

figure shows that four different control signals are generated by combining the signals $\overline{RD}$, $\overline{WR}$ and $IO/\overline{M}$. The signal $IO/\overline{M}$ goes low for the memory operation. This signal is ANDed with $\overline{RD}$ and $\overline{WR}$ signals by using the 74LS32 quadruple two input OR gates as shown in figure. The OR gates are functionally connected as negative NAND gates. When both input signals go low, the outputs of the gates go low and generate $\overline{MEMR}$ (Memory Read) and $\overline{MEMW}$ (Memory write) control signals. When the $IO/\overline{M}$ signal goes high, it indicates the peripheral I/O operation. figure shows that this signal is complemented using the Hex inverter 74LS04 and ANDed with the $\overline{RD}$ and $\overline{WR}$ signals to generate $\overline{IOR}$ (I/O Read) and $\overline{IOW}$ (I/O write) control signals.

fig→ Schematic to generate Read/Write Control signals for memory and I/O fig



External Address / Data bus multiplexing :-

The signal lines $AD_7 - AD_0$ are bidirectional :- they serve a dual purpose. They are used as the low order address bus as well as data bus. In executing an instruction, during the earlier part of the cycle, these lines are used as the low-order address bus. During the later part of the cycle, these lines are used as the data bus. (This is also known as multiplexing the bus). However, the low-order address bus can be seperated from these signals by using a latch.

Serial Communication :-

The 8085 has two signals to implement the serial transmission →

SID (Serial input data)

SOD (Serial output data)

In serial transmission data bits are sent over a single line, one bit at a time, such as the transmission over Telephone lines.

→ Data Transfer is controlled through Two instructions :- SIM and RIM.

Instructions SIM and RIM are used for two different processes: Interrupt and serial I/O. ~~These~~

Serial output Data (SOD) :- The instruction SIM is necessary to output data serially from the SOD line. It can be interpreted for serial output as shown in figure

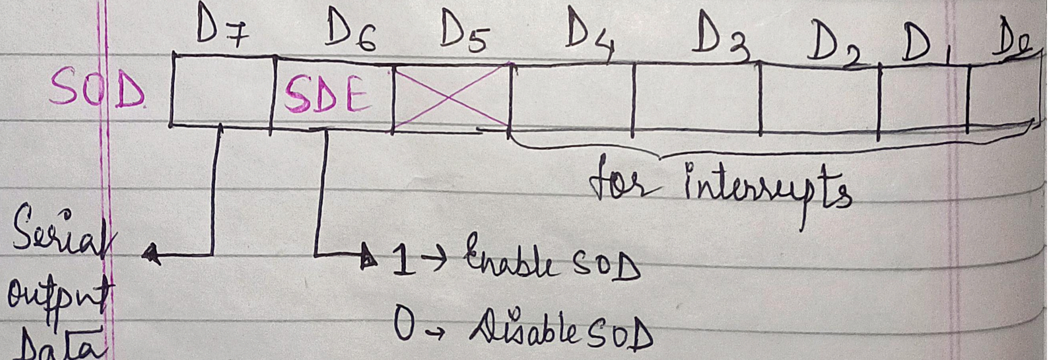


fig → Interpretation of Accumulator content by the SIM instruction.

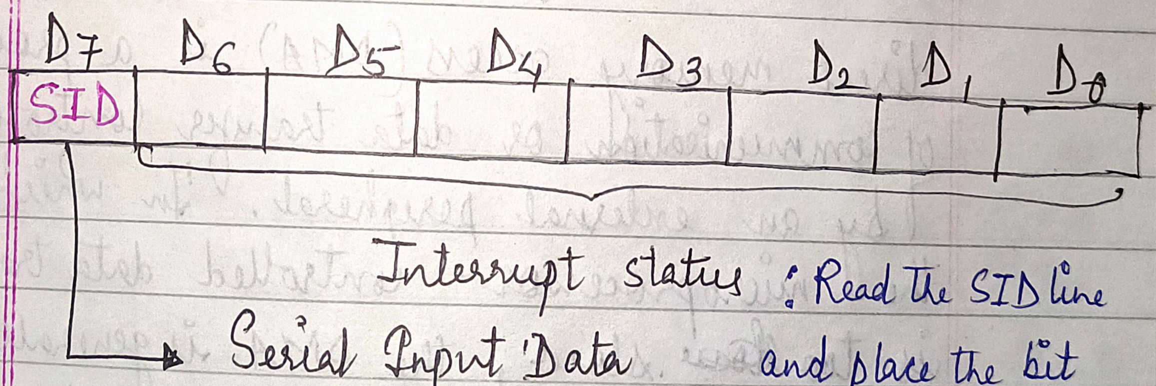
INSTRUCTIONS :-

MVI A, 80H ; Set D_7 in the accumulator = 1
RAR ; Set $D_6 = 1$ and bring carry into D_7
SIM ; Output D_7

In this set of instructions, the serial output line is enabled by rotating 1 into bit position D_6; the instruction SIM outputs the carry bit through bit position D_7.

Serial input data (SID) :-

Instruction RIM is used to input serial data through the SID line. Instruction RIM can be interpreted for serial I/O as shown in figure.



; Read the SID line and place the bit in the accumulator at D_7

fig → Interpretation of Accumulator Contents after the RIM Instruction.

In the content of serial I/O, instruction RIM is

Similar to instruction IN, except RIM reads only one bit and places it in the accumulator at D_7.

The SID and SOD lines in the 8085 eliminate the need for an input port and an output port in the software-controlled serial I/O. Essentially, SID is a 1-bit input port and SOD is a 1-bit output port. Similarly, instruction RIM is equivalent to a 1-bit IN instruction and instruction SIM is equivalent to a conditional 1-bit OUT instruction. ~~The software necessary to implement serial I/O using SID and SOD lines is conceptually~~

Direct Memory Access (DMA):

The direct memory access (DMA) is a process of communication or data transfer controlled by an external peripheral. In which the microprocessor-controlled data transfer is too ~~slow~~ slow, the DMA is generally used eg. data transfer between a floppy disk and R/W memory of the system.

The 8085 microprocessor has two pins available for this type of I/O communication: HOLD (hold) and HLDA (Hold Acknowledge).

Conceptually, this is an important I/O technique, it introduces two new signals available on the 8085 - HOLD & HLDA.

(1) **HOLD** — Hold. This is an active high input signal to the 8085 from another master requesting the use of the address and data buses. After receiving the HOLD request, the MPU released the buses in the following machine cycle. All buses are tri-stated and a Hold Acknowledge (HLDA) signal is sent out. The MPU regains the control of buses after HOLD goes low.

(2) HLDA – Hold Acknowledge → This is an active high output signal indicating that the MPU is released the control of the buses.

Typically, an external peripheral such as DMA controller sends a request - a high signal to the HOLD pin. The processor completes the execution of the current machine cycle; floats (high impedance state) the address, the data, and the control lines and sends the Hold acknowledge (HLDA) signal. The DMA controller takes control of the buses and transfers data directly between source and destination thus by-passing the microprocessor. At the end of

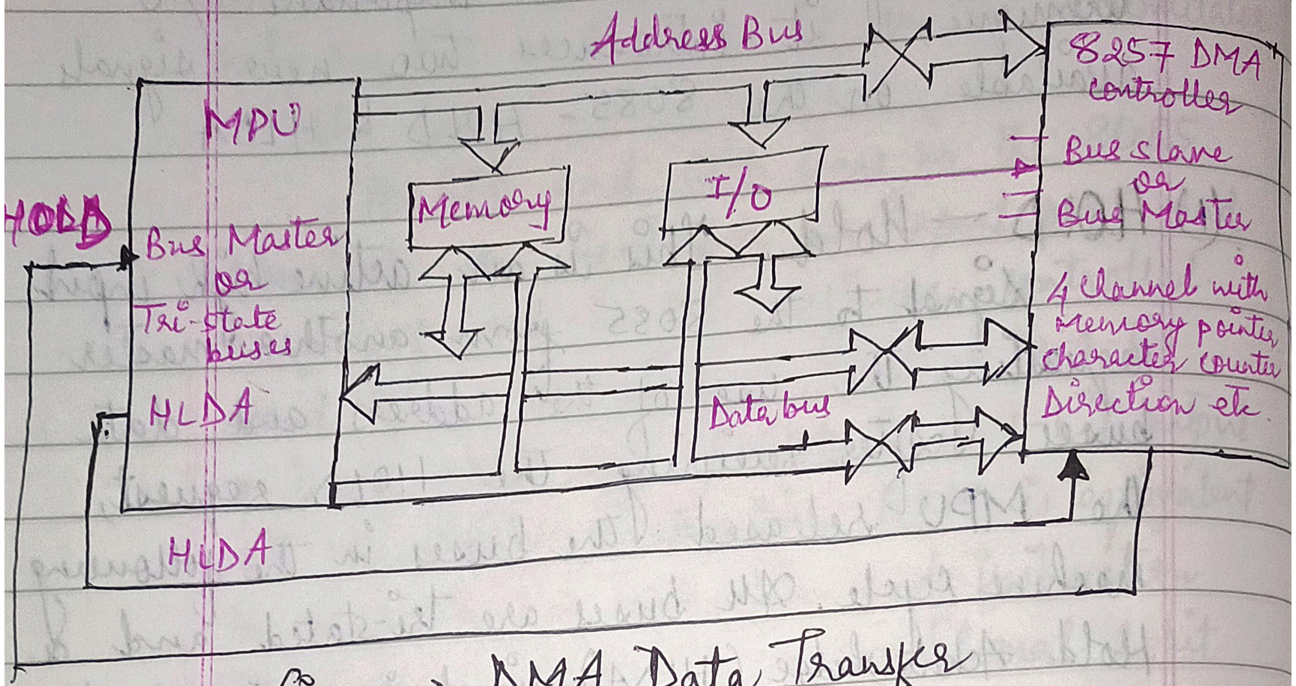


figure → DMA Data Transfer

data transfer, the controller terminates the request by sending a low signal to the HOLD pin and the microprocessor regains control of the buses. Typically, DMA controllers are programmable LSI Chips. ~~One such~~ ~~chip~~