

Don't believe blindly on these most questions.

It is just prediction based!

B.Tech. (IV Sem.) C.S. Solved Papers

- an access
- an access
- (10) It provides 5 hardware interrupts : TRAP, RST5.5, RST6.5, RST7.5, INTR.
 - (11) By providing external hardware (8259 chip) one can increase the interrupt capability of it.
 - (12) It provides status and control signals.

YEARS QUESTIONS

To reduce complications on microprocessor IC, we need multiplexing.

Prob.4 Define general purpose registers and their use.

[R.T.U. 2019]

Sol. General Purpose Registers : 8085 has six 8-bit general-purpose registers to store 8-bit data named as B, C, D, E, H and L. They can be combined to form register pairs – BC, DE and HL to store 16-bits of data. These registers are programmable by user to store/copy any data and used to perform different operations and hence also known as **scratched registers**. The HL pair is used to address members.

Prob.5 What is microprocessor?

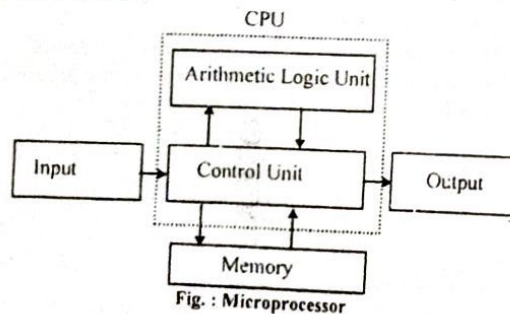
[R.T.U. 2019]

OR

Explain microprocessor.

[R.T.U. 2017]

Sol. Microprocessor : It is a multipurpose, programmable, clock driven, register based electronic device that reads binary instruction from memory, accepts binary data as input and processes them.



Prob.6 Write short note on Multiplexer.

[R.T.U. 2017]

Sol. Multiplexer : Multiplexer means many into one. A multiplexer is a circuit used to select and route any one of the

Microprocessor and Interfaces

several input signals to a signal output. A simple example of a non electronic circuit of a multiplexer is a single pole multiposition switch.

Multiposition switches are widely used in many electronics circuits. However circuits that operate at high speed require the multiplexer to be automatically selected. A mechanical switch cannot perform this task satisfactorily. Therefore, multiplexer used to perform high speed switching are constructed of electronic components.

Multiplexer handle two type of data that is analog and digital. For analog application, multiplexer are built of relays and transistor switches. For digital application, they are built from standard logic gates.

The multiplexer used for digital applications, also called digital multiplexer, is a circuit with many input but only one output. By applying control signals, we can steer any input to the output. Few types of multiplexer are 2-to-1, 4-to-1, 8-to-1 and 16-to-1 multiplexer.

Following figure shows the general idea of a multiplexer with n input signal, m control signals and one output signal.

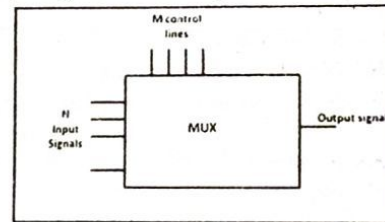


Fig. : Multiplexer Pin Diagram

Prob.7 Write short note on De-multiplexer.

[R.T.U. 2017]

Sol. De-multiplexer : De-multiplexer means one to many. A de-multiplexer is a circuit with one input and many outputs. By applying control signal, we can steer any input to the output. Few types of de-multiplexer are 1-to-2, 1-to-4, 1-to-8 and 1-to-16 De-multiplexer.

Following figure illustrate the general idea of a de-multiplexer with 1 input signal, m control signals, and n output signals.

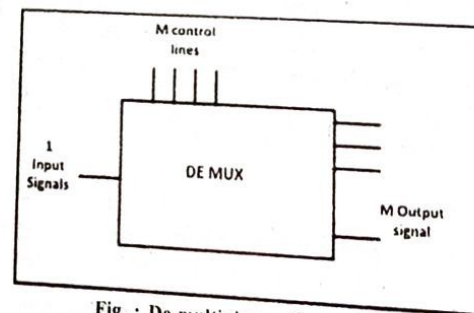


Fig. : De-multiplexer Pin Diagram

ML3

Prob.8 Write short note on Encoder.

[R.T.U. 2017]

Sol. Encoder : "An encoder is a combinational logic circuit that essentially performs a 'reverse' decoder function. It accepts an active level on one of its inputs representing a digit, such as a decimal or octal digit, and converts it to a coded output such as binary or BCD."

Encoders can also be devised to encode various symbols and alphabetic characters. This process of converting from familiar symbols or numbers to a coded format is called Encoding. Encoder has n input lines, only one of which is active at any time and m output lines. In an encoder the number of output is less than the number of input

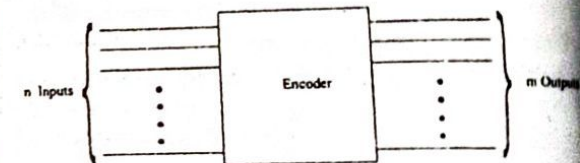


Fig. : Block Diagram of an Encoder

An 8 to 3 encoder has eight active low inputs and three output lines. When the input line 0 goes low, the output is 000. When the input line 5 goes low, the output is 101. For simultaneous inputs, priority encoders are very successful. 74LS148 is an example of 8 to 3 priority encoder.

Encoders are commonly used with keyboards. For each key pressed, the corresponding binary code is placed on the data bus.

Prob.9 Write short note on Decoder.

[R.T.U. 2017]

Sol. Decoder : "Decoder is used to detect the presence of a specified combination of bits (code) on its inputs and indicate that presence by a specified output level."

When generalized, a decoder has n input lines to handle n bits and from one to 2^n output line to indicate the presence of one or more n-bit combinations. Decoding is necessary in applications such as data multiplexing, digital display, digital to analog converters and memory addressing.

For example, if the input to a decoder has 2 bits, the decoder will have 4 output lines. The 2 lines can assume four combinations of input signals 00, 01, 10, 11. Each combination identified by the output lines 0 to 3. If the input is (10), the output line 2 will be at logic 1, and other lines will remain at logic 0. This is called Decoding.

MI.4

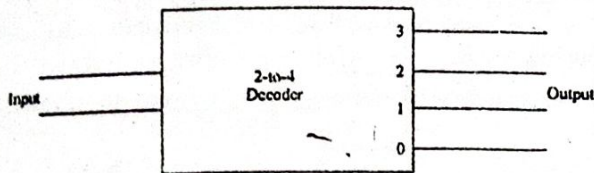


Fig. 1 : 2 to 4 Decoder Logic Symbol

Example of the decoder can be 3 to 8, 4 to 16, 4 to 10 decoders. Decoders have active low enable lines, which decide when to turn it on.

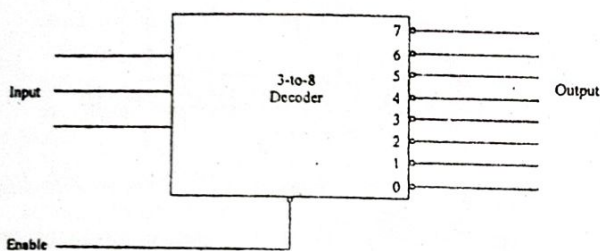


Fig. 2 : 3 to 8 Decoder Logic Symbol with Enable

A decoder is commonly used device in interfacing I/O peripherals and memory. Decoders are also built internal to a memory chip to identify individual memory register.

Basic Binary Decoder : Suppose we wish to determine when a binary 1001 occurs on the inputs of a digital circuit. An AND gate can be used as the basic decoding element because it produces a HIGH output only when all of its inputs are HIGH. Therefore, we must make sure that all of the inputs to the AND gate are HIGH when the binary number 1001 occurs, this can be done by inverting the 2 middle bits (the 0s).

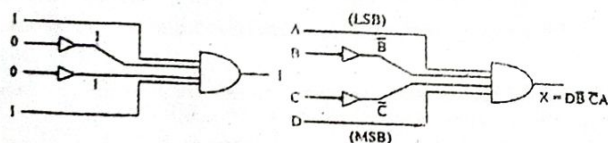


Fig. 3 : Decoding logic for 1001 with an Active-HIGH Output

A is the LSB and D is the MSB. In the representation of a binary number or other weighted code, LSB is always the right most bit in a horizontal arrangement and the top most bit in a vertical arrangement, unless specified otherwise.

Prob.10 Explain microcontroller.

[R.T.U. 2017]

Sol. Microcontroller : It is a device that includes microprocessor memory and I/O signal lines on a single chip, fabricated using VLSI technology.

- Reprogrammable system.
- Embedded system.

B.Tech. (IV Sem.) C.S. Solved Papers

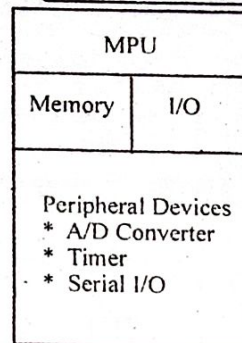


Fig. : Microcontroller

Prob.11 Explain address bus.

[R.T.U. 2017]

Sol. Address Bus : The address bus is a group of 16 lines generally identified as A_0 to A_{15} . The address bus is unidirectional : bits flow in one direction - from the MPU to peripheral devices. The MPU uses the address bus to perform the first function : identifying a peripheral or a memory location.

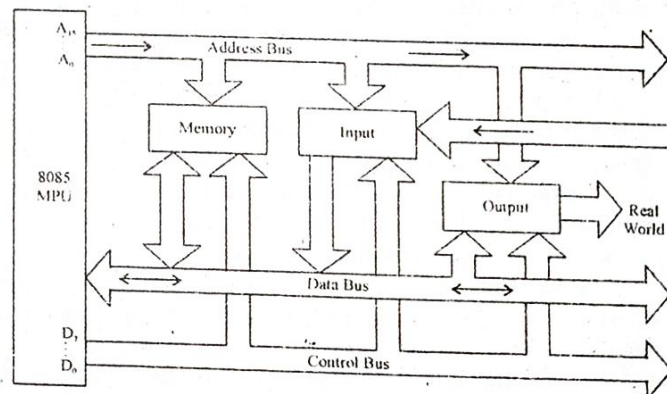


Fig. : 8085 Bus Structure

In a computer system, each peripheral or memory location is identified by a binary number, called an address and the address bus is used to carry a 16-bit address. This is similar to the postal address of a house. A house can be identified by various number schemes. For example, the forty-fifth house in a lane can be identified by the two digit number 45 or by the four digit number 0045. The two digit number scheme can identify only a hundred houses, from 00 to 99. Similarly, the number of address lines of the MPU determines its capacity to identify different memory locations (or peripherals). The 8085 MPU with its 16 address lines is capable of addressing $2^{16} = 65,536$ (generally known as 64K) memory locations. As 1K memory is determined by rounding off 1024 to the nearest thousand, similarly, 65,536 is rounded off to 64,000 as a multiple of 1K.

Microprocesso

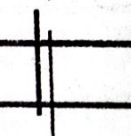
The num
by the desig
considerations
of the process
20 and pentium

Prob.12 Exp

Sol. Data Bu

for data flow
both direction
devices. The
function: tran

The eigh
data ranging
number that
the 8085
Microproces
Motorola 68
16-bit micro
lines; thus th

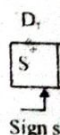


Prob.13

Lang
flags

Sol. Flag R

register. Th
all the cond
flags of the
on the carry
the program
the next in
five bits se



Sign s

Zero s

It is predicted Questions only

Microprocessor and Interfaces

The number of address lines is arbitrary, it is determined by the designer of a microprocessor based on such considerations as availability of pins and intended application of the processor. For example, the Intel 8088 processor has 20 and pentium processor has 32 address lines.

Prob.12 Explain data bus.

[R.T.U. 2017]

Sol. Data Bus : The data bus is a group of eight lines used for data flow. These lines are bi-directional, data flows in both directions between the MPU and memory and peripheral devices. The MPU uses the data bus to perform the second function: **transferring binary information.**

The eight data lines enable the MPU to manipulate 8-bit data ranging from 99 to FF ($2^8 = 256$ number). The largest number that can appear on the data bus is 11111111 (255_{10}) the 8085 is known as an 8-bit microprocessor. Microprocessors such as the Intel 8086, Zilog Z8000 and Motorola 68000 have 16 data lines; thus they are known as 16-bit microprocessors. The Intel 80386/486 have 32 data lines; thus they are classified as 32-bit microprocessors.

PART-B

Prob.13 Explain role of flag register in Assembly Language Programming also describe various flags available in 8085.

[R.T.U. 2019, 09]

Sol. Flag Register : Flag register is a 8-bit special purpose register. The role of the flag register in programming is that all the conditional branching operations are based on the four flags of the flag register as an example INC "addr" is based on the carry flag. If the carry flag value is CY = 0 then only the program control will transfer to the given address otherwise the next instruction will be executed. The flag register and five bits set and reset condition is shown below :

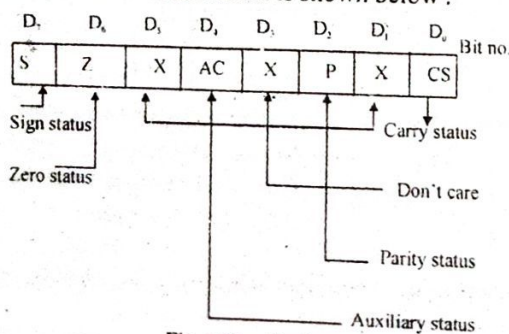


Fig. : Flag Register

MI.5

1. Carry Flag (CY): If an arithmetic operation results in carry, a carry flag is set (1). Otherwise it is reset (0). The carry flag also serves as a borrow flag for the subtraction. The carry flag solves the dual purpose of showing carry borrow in case of addition and subtraction respectively. It is represented by D_0 bit of flag register.

2. Parity Flag (P): This flag bit (D_2) shows the result of the arithmetic operation is having even number of ones (or) add. Since 8085 microprocessor follows even parity system so if the number of ones in accumulation after arithmetic operation is even. This flag is set (1) other wise it is cleared (0).

3. Auxiliary Carry Flag (AC): In an arithmetic operation when a carry is generated by lower nibble and passes to upper nibble, the auxiliary carry flag is set. This flag is used internally for BCD (Binary Coded Decimal) operations D_4 bit of flag represents the AC flag.

4. Zero Flag (Z): The zero flag shows that the result of any arithmetic or logical operation is zero or non zero. The zero flag is set (1) when the result is zero otherwise it is reset (0). Zero flag is represented by D_6 bit of flag register.

5. Sign Flag (S): The sign flag is replica of the D_7 bit of the result. The sign flag is set (1) if bit 7 of the result is (1), otherwise it is reset (0). Sign flag is represented by D_7 bit of flag register.

Prob.14 Explain the role of following in 8085-

- Program Counter
- Stack Pointer

[R.T.U. 2019]

Sol.(a) Program Counter : Program is a sequence of instructions, microprocessor fetches these instructions from the memory and executes them sequentially. The program counter is a special purpose register which, at a given time, stores the address of the next instruction to be fetched. Program Counter acts as a pointer to the next instruction. How processor increments program counter depends on the nature of the instruction; for one byte instruction it increments program counter by one, for two byte instruction it increments program counter by two and for three byte instruction it increments program counter by three such that program counter always points to the address of the next instruction.

In case of JUMP and CALL instruction, address followed by JUMP and CALL instructions is placed in the program counter. The processor then fetches the next instruction from the new address specified by JUMP or CALL instruction. In conditional JUMP and conditional CALL instructions, if the condition is not satisfied, the processor increments program counter by three so that it points the

Prob.19 Explain the function of various control and status signals available on 8085 microprocessor. [R.T.U. 2013]

Sol. Control & Status Signals

There are 2 control signals (\overline{RD} and \overline{WR}) and three status signals (IO/\overline{M} , S_1 and S_0) to identify the nature of operation and one special signal, ALE for distinguishing the higher order and lower order addresses.

- (a) **ALE (Address Latch Enable)**: This signal is generated in starting of the 8085 operation and is used to latch the lower order address from the multiplexed bus and generate a separate set of eight address lines $A_7 - A_0$. If this is one, address will be passed otherwise data will be passed through $AD_7 - AD_0$.
- (b) **\overline{RD} (Read)**: This is an active low signal and is used to read memory or I/O device and data will be available on data bus. This signal should be low to perform the read operation.
- (c) **\overline{WR} (Write)**: A low on \overline{WR} indicates that data on the data bus which has been placed on it by the processor is to be written into the selected memory or I/O operation.
- (d) **IO/\overline{M}** : This is a status signal used to differentiate between I/O and memory operation. If it is high, I/O operation and if low, memory operation. It is combined with \overline{RD} and \overline{WR} to generate I/O and memory control signals.
- (e) **S_1 and S_0** : These are also status signals and used to identify various operations, according to the following table:

Table : 8085 Machine Cycle, Status & Control Signal

Machine Cycle	Status			Control Signals
	IO/\overline{M}	S_1	S_0	
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	$\overline{RD}, \overline{WR} = Z$ and $\overline{INTA} = 1$
Hold	Z	X	X	
Reset	Z	X	X	

Here Z=Tri-state/high Impedance;
X=Unspecified

Prob.20 Explain the difference between the following
(i) Microprocessor and Microcontroller
(ii) Assembler and Compiler
(iii) High level and low level language
(iv) RAM and ROM [R.T.U. 2012]

Sol. (i) Difference between Microprocessor and Microcontroller :

Table : Comparison Between Microprocessor and Microcontroller

S. No.	Microprocessor	Microcontroller
1.	Microprocessor contains ALU, general purpose register, stack pointer, program counter, clock timing circuit and interrupt circuit.	Microcontroller contains the circuitry of micro-processor and in addition it has built-in ROM, RAM, I/O devices, timers and counters.
2.	It has many instructions to move data between memory and CPU.	It has one or two instructions to move data between memory and CPU.
3.	It has one or two bit handling instructions.	It has many bit handling instructions.
4.	Access times for memory and I/O devices are more.	Less access times for built-in memory and I/O devices.
5.	Microprocessor based system requires more hardware.	Microcontroller based system requires less hardware reducing PCB size and increasing the reliability.
6.	Microprocessor based system is more flexible in design point of view.	Less flexible in design point of view.
7.	It has single memory map for data and code.	It has separate memory map for data and code.
8.	Less number of pins are multi-functioned.	More number pins are multifunctioned.
9.	Few bit hiding information.	Many bit hiding information.
10.	High performance pipeline parallel CPU architecture.	Low performance simple CPU Architecture.

(ii) Assembler and Compiler

S. No.	Assembler	Compiler
1.	An assembler can be considered as a special type of compiler which translates only assembly language to machine code.	Compiler is a computer program that reads a program written in a language and translates it in to another language.

2.	It produces an object code which might have to be linked using linker program in order to run on machine.	Compiler usually produces the machine executable code, directly from a higher level language.
----	---	---

(iii) High Level Language and Low Level Language

S. No.	High level language	Low level language
1.	High level languages are easy to learn.	Low level languages are difficult to learn.
2.	These are near to human language.	These are far from human language.
3.	These are easy to modify.	Program in this language are difficult to modify.
4.	Execution of program is slow.	Execution of program is fast.
5.	These languages are normally used to write application program.	These languages are normally used to write hardware program.

(iv) RAM and ROM

S. No.	Features	RAM	ROM
1.	Elaboration	Random Access Memory	Read Only Memory.
2.	Memory type	Internal Memory	External Memory.
3.	Working type	Both Read and Write operation can be performed over the information stored in RAM.	The ROM only allows the user to read the information. User cannot make changes to the information
4.	Type	1. Static RAM (SRAM) 2. Dynamic RAM (DRAM)	1. PROM (Programmable ROM) 2. EPROM (Erasable Programmable ROM) 3. EEPROM (Electrically Erasable Programmable ROM)

Prob.21 There are two types of read and write in microprocessor interfaces, such as Read and Write from/to input device and output device as well as Read and Write from/to memory device. Explain how they are realized by only three pins in 8085. [R.T.U. 2009]

lines (line 15-8) and lower address lines (line 7-0) combinely holds the 16 bits of the address. When ALE is low (Logic 0): Upper address lines (line 15-8) holds the upper 8 bit address and Lower address lines (line 7-0) holds the "8 bit DATA".

Multiplexing is used to reduce the number of pins of 8085, which otherwise would have been a 48 pin chip. But because of multiplexing, external hardware is required to de-multiplex the lower byte address cum data bus.

The Pin no= 30 of 8085 is the ALE pin which stands for 'Address Latch Enable'. ALE signal is used to de-multiplex the lower order address bus ($AD_0 - AD_7$).

Pins 12 to 19 of 8085 are $AD_0 - AD_7$ which is the multiplexed address-data bus. Multiplexing is done to reduce the number of pins of 8085.

Lower byte of address ($A_0 - A_7$) are available from $AD_0 - AD_7$ (pins 12 to 19) during T_1 of machine cycle. But the lower byte of address ($A_0 - A_7$), along with the upper byte $A_8 - A_{15}$ (pins 21 to 28) must be available during T_2 and rest of the machine cycle to access memory location or I/O ports.

Now ALE signal goes high at the beginning of T_1 of each machine cycle and goes low at the end of T_1 and remains low during the rest of the machine cycle. This high to low transition of ALE signal at the end of T_1 is used to latch the lower order address byte ($A_0 - A_7$) by the latch IC 74LS373, so that the lower byte $A_0 - A_7$ is continued to be available till the end of the machine cycle. The situation is explained in the following figure:

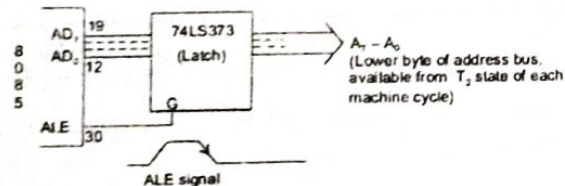


Fig. : Lower byte of address latching achieved by the H to L transition of ALE signal, which occurs at the end of T_1 of each machine cycle

Prob. 23 Draw and explain the functional block diagram of 8085 microprocessor along with the features in detail.

[R.T.U. 2019]

OR

Draw the architecture diagram of 8085 microprocessor and explain functions of various registers.

[R.T.U. 2015]

OR

Draw and explain the functional block diagram of 8085 microprocessor.

[Raj. Univ. 2006, 2004, 2002, 2001]

OR

Draw the diagram of 8085 microprocessor architecture. Explain the components.

[R.T.U. 2014]

Sol. Intel 8085 is an 8 bit, NMOS microprocessor. It is a 40 pin IC package fabricated on a single LSI chip.

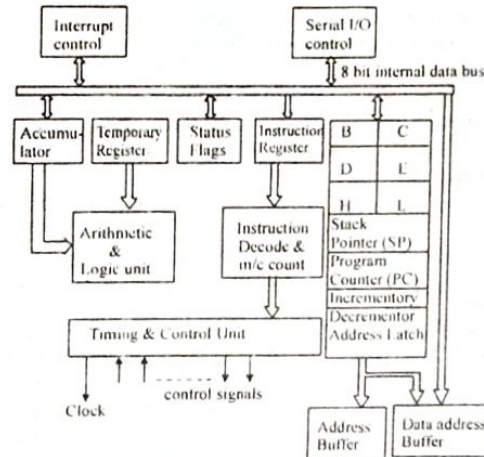


Fig. : Block diagram of 8085 MPU

The intel 8085 uses a single +5V dc supply for its operation. Its clock speed is about 3 MHz. The clock cycle is of 320 ns. The time for the clock cycle of the intel 8085 is 200 ns. It has 80 basic instructions and 246 opcodes.

Fig. shows the block diagram of intel 8085. It consists of three main sections, an arithmetic and logic unit, a timing and control unit and several registers. These important sections are described below:

1. ALU

The arithmetic and logic unit (ALU), performs the following arithmetic and logic operations.

1. Addition
2. Subtraction
3. Logical AND
4. Logical OR
5. Logical Exclusive OR
6. Complement (Logical NOT)
7. Increment (add 1)
8. Decrement (subtract 1)
9. Left shift (add input to itself)
10. Clear

2. Register Array

Register is a group of predefined memory location which is used to store data and result. 8085 has got three types of registers:

(a) General Purpose Registers (GPR) : Refer to Prob.4.

(b) Temporary Registers : W and Z are temporary registers which are not available for user. To perform arithmetic and logical operations, microprocessor assumes one data is available in accumulator and takes another and then performs operation on the 2 data bytes. Temporary registers are used by microprocessor for internal operations to store operand, immediate operand or address of memory.

(c) Special Purpose Registers : 8085 microprocessor contains three special purpose registers which are Program Counter (PC), Stack Pointer (SP) and Increment/Decrement Latch.

(i) Program Counter (PC) : Refer to Prob.14(a).

(ii) Stack Pointer (SP) : Refer to Prob.14(b).

(iii) Increment/Decrement Latch: This is also a 16-bit register used to increment or decrement the contents of SP and PC registers. Address buffer and address/data buffers are used in co-ordination with these registers. Address buffer is an 8 bit unidirectional buffer used for $A_{15} - A_8$ address lines. When they are not used, buffer is used to tri-state these lines. Address/Data buffer is an 8-bit bidirectional buffer for address/data lines $AD_0 - AD_7$ under certain conditions (reset, hold, halt).

3. Interrupt Control

This block is responsible for accepting different interrupt request inputs such as TRAP, RST 7.5, RST 6.5, RST 5.5 and INTR. Interrupts are generated to interrupt the execution of program. When a valid interrupt request is present, it informs control logic to take action in response to each signal.

4. Serial I/O Control Groups

The data transferred via D_0 to D_7 lines is parallel data, but under certain condition it is advantageous to use serial data transfer (bit by bit transmission). Microprocessor 8085 uses SID and SOD signals to transmit data serially. The data on these lines is accepted or transferred under software control by serial I/O control blocks.

5. Instruction Register, Decoder and Control Group

Instruction Register : When an instruction is fetched from memory, it is loaded in instruction register and that instruction is provided to decoder for decoding. This register is only activated when an instruction code or OPCODE is available on internal data bus. It is a non-programmable register.

Instruction Decoder : This accepts a bit pattern from instruction register, decodes it and gives the decoded information to control logic. The information includes what operation is to be performed, who is going to perform it, how many operand bytes the instruction contains etc.

Timing and Control Unit : It contains an oscillator and a controller sequencer. Oscillator is responsible for generating clock signals to synchronize all registers and peripherals for communication to the microprocessor.

ASSEMBLY LANGUAGE PROGRAMMING

2

CHAPTER IN A NUTSHELL

□ Assembly Language

A medium of communication with a computer in which programs are written in mnemonics. An assembly language is specific to a given computer.

□ **Instruction Set** : All the instructions are described fully in terms of its operation and the operand, including details such as number of bytes, machine cycle, T-states hex code and affected flags.

The various abbreviation used- Reg. 8080A/ 8085 Register.

Mem = Memory Location
R = Register
R_s = Register Source
R_d = Register Destination
M = Memory
() = Contents of
XX = Random Information

□ **Flags** : All flags are modified to reflect the result of addition.

S = Sign

Z = Zero

AC = Auxiliary Carry

P = Parity

CY = Carry

Example :

Opcode	Operand	Byte	M-cycle	T-States	Hex code
ACI	8-bit data	2	2	7	CE

Description : The 8-bit data (operand) and the carry flag are added to the contents of the accumulator and the result is stored in accumulator.

□ **Program Structure** : These are of 3 types.

- Sequential** : A program with top location approach, without much complexity.
- Conditional** : Conditions are applied in the form of conditional loops and counters.
- Iterative/ Recursive** : Some execution of a program counter for a finite number of times.

PREVIOUS YEARS QUESTIONS

PART-A

Prob.1 Define instruction & instruction set. [R.T.U. 2019]

Sol. Instruction : An instruction is a command to the microprocessor to perform a given task on a specified data. Each instruction has two parts: one is task to be performed, called the operation code (opcode), and the second is the data to be operated on, called the operand. The operand (or

data) can be specified in various ways. It may include 8-bit (or 16-bit) data, an internal register, a memory location, or 8-bit (or 16-bit) address. In some instructions, the operand is implicit.

Instruction Set : All the instructions are described fully in terms of its operation and the operand, including details such as number of bytes, machine cycle, T-states hex code and affected flags.

The various abbreviation used- Reg. 8080A/ 8085 Register.

Mem = Memory Location
R = Register
R_s = Register Source

7. JNC is used to jump to the given step if their is no carry (3 Byte instruction)
8. JNZ is used to jump to the given step if their is not zero (3 Byte instruction)
9. DCR is used to decrease given register by 1 (1 Byte instruction)
10. HLT is used to halt the program

Prob. 16 Define direct and indirect addressing modes with appropriate examples. [R.T.U. 2016]

OR

Describe various addressing modes available in 8085 microprocessor with two example of each.

[R.T.U. 2013]

OR

Explain various addressing modes with example in 8085 assembly language programming.

[R.T.U. 2013, 10, Raj. Univ. 2001, 2000]

OR

Explain direct and indirect addressing with suitable examples. [R.T.U. 2014]

Sol. Addressing Modes : Each instruction requires certain data on which it has to operate. There are various techniques to specify data for instructions these techniques are called addressing mode. Intel 8085 uses the following addressing modes :

(a) **Direct Addressing :** In this type of addressing, the address of the operand (data) is given in the instruction itself.

Examples :

(1) STA2400 H : Store the content of the accumulator in the memory location 2400 H.

32, 00 24 : The above instruction in the code form.

In this instruction 2400H is the memory address where data is to be stored. It is given in the instruction itself D. The 2nd and 3rd bytes of the instruction specify the address of the memory location. Here, it is understood that the source of data is accumulator.

(2) IN 02 : Read data from the port C

D B, 02 : Instruction in the code form.

In this instruction 02 is the address of the port C of I/O port from where the data is to be read. Here, it is implied that the destination is the accumulator. The 2nd byte of the instruction specifies the address of the port.

(b) **Register Addressing :** In register addressing mode, the operands are in the general purpose register. The opcode specifies the address of the registers in addition to the operation to be performed.

Examples :

(1) MOV A, B : Move the content of register B to register A.

78 H : The instruction in the code form.

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

80 H : The instruction in the code form.

In example (1) the opcode for MOV A, B is 78H. Besides the operation to be performed, the opcode also specifies the registers, which contain data. The opcode 78H can be written in binary form as 01111000. The first two bytes, i.e. 01 are for MOV operation, the next three bits 111 are the binary code for register A and the last three bits 000 are the binary code for register B.

In example (2) the opcode for ADD B is 80H. In this instruction one of the operands is register B (its content is one of the data) which is indicated in the instruction itself. In this type of instruction (arithmetic group), it is understood that the other operand is in the accumulator. The opcode 80H in the binary or is 10000000. The first five bits i.e. 10000 specify the operation to be performed, i.e. ADD. The last three bits 000 are the binary code for register B for 8085 microprocessor.

(c) **Register Indirect Addressing :** In this mode of addressing the address of the operand is specified by a register pair.

Examples :

(1) LXI H, 2500 H : Load HL pair with 2500 H.

MOV A, M : Move the content of the memory location, whose address is in HL pair (i.e., 2500H) to the accumulator.

HLT : Halt.

In the above program the instruction MOV A, M is an example of register indirect addressing. For this instruction, the operand is in the memory. The address of the memory is not directly given in the instruction. The address of the memory resides in HL pair and this has already been specified by an earlier instruction in the program, i.e., as LXI H, 2500 H.

(2) LXI H, 2500 H : load the HL pair with 2500 H.

ADD M : Add the content of the memory location, whose address is in HL pair (i.e. 2500 H), to the content of the accumulator.

HLT : Halt

In this program, the instruction ADD M is an example of register indirect addressing.

(d) **Immediate Addressing :** In immediate addressing mode, the operand is specified with the instruction itself.

Examples :

(1) MVI A, 05 : Move 05 in register A.

3 E, 05 : The instruction in the code form.

(2) ADI 06 : Add 06 to the content of the accumulator.

C6, 06 : The instruction in the code form.

In these instructions the 2nd byte specifies data.

LXI H, 2500 H is an example of immediate addressing 16-bit data.

Which is given in the instruction itself. It is to be loaded into HL pair.

(e) **Implicit Addressing** : There are certain instructions, which operate on the content of the accumulator. Such instructions do not require the address of the operand.

Examples : CMA, RAL, RAR etc.

Prob.17 Write a short note on rotate instructions of 8085 Microprocessor. [R.T.U. 2016]

OR

Explain the all type of Rotate instructions with the help of suitable examples. [R.T.U. 2013]

Sol. 8085 Microprocessor has four types of rotational instructions.

- RLC (Rotate Accumulator Left)
- RAL (Rotate Accumulator Left Through Carry)
- RRC (Rotate Accumulator Right)
- RAR (Rotate Accumulator Right Through Carry)

(i) **RLC (Rotate Accumulator Left)** : Rotate or shift the content of accumulator left by one bit without carry. Seventh bit (D_7) of the accumulator is moved to carry bit (cy) as well as to zero bit (D_0).

$$D_{n+1} \leftarrow D_n, D_0 \leftarrow D_7, CY \leftarrow D_7$$

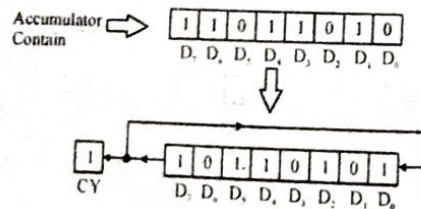
(a) It is 1 byte opcode.

(b) Only carry flag is affected.

(c) Implicit addressing mode (no operand is there, works on accumulator.)

(d) It has 1 machine cycle and 4-T states.

Example : If accumulator contain 8-bit data 11011010. Then applying RLC instruction on it.



Accumulator after applying RLC Instruction.

Use of RLC

1. To multiply the number by two.

2. To check whether bit has carry or no carry.

(ii) **RAL (Rotate Accumulator Left Through Carry)**: This instruction rotate the contents of accumulator left by one bit with carry.

The seventh bit of the accumulator shift to carry and carry bit is shift to zero bit of the accumulator.

Syntax :

$$D_{n+1} \leftarrow D_n, CS \leftarrow D_7, D_0 \leftarrow CY$$

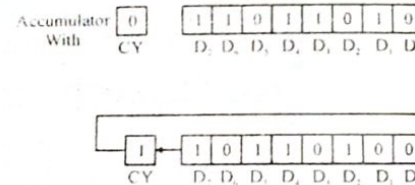
(a) It is 1 byte opcode.

(b) Only carry flag is modified.

(c) Implicit addressing mode.

(d) It has same 1 machine cycle and 4-T states.

Example : If accumulator contain 8-bit data 11011010 and carry bit contain 0 then after instruction execution accumulator contain 10110100 as follows.



Accumulator and carry bit after RAL execution.

(iii) **RRC (Rotate Accumulator Right)** : Rotate shift the contents of accumulator right by one bit with carry. Zero bit of accumulator shift to seventh bit and carry bit.

Syntax :

$$D_7 \leftarrow D_0, CS \leftarrow D_0, D_n \leftarrow D_{n+1}$$

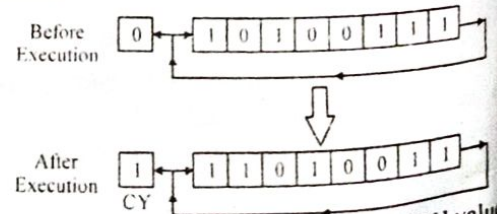
(a) It is 1 byte opcode instruction.

(b) Only carry flag is affected.

(c) Implicit address mode.

(d) It has 1 machine cycle and 4-T state

Example : Accumulator contain A7H and carry is reset to 0.



After execution accumulator contain DEH value.

Use

(1) To divide the numbers by two.

(2) To check whether the bit has carry or no carry.

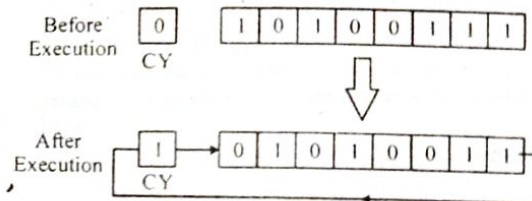
(iv) **RAR (Rotate Accumulator Right Through Carry)** : Rotate of shift the contents of accumulator right by one bit with carry. Zero bit of Accumulator is shift to carry bit and carry bit is shift to seventh bit.

Syntax :

$$D_n \leftarrow D_{n+1}, CY \leftarrow D_0, D_7 \leftarrow CY$$

It has same features like RRC instructions.

Example : If accumulator contain A7H and carry flag contain 0 bit.



Prob.18 Explain instruction cycle of an instruction MVIA, 05 H using timing diagram. [R.T.U. 2016]

Sol. Let the actual physical memory location of register A be 1000H and the machine code for the same be 06H. The data is given to be in 05H. Let the actual physical memory location of this data be 1001 H.

Mnemonics	Machine Code	Memory Locations
MVIA, 05H	06H	1000H
	05H	1001H

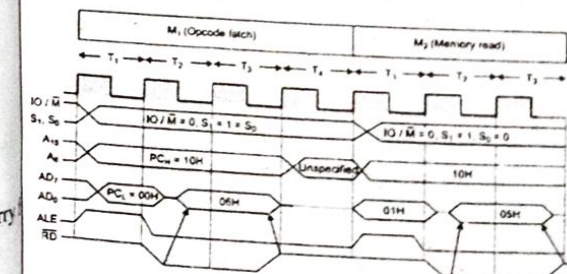


Fig. : Timing diagram for MVIA, 05H

The MVIA, 05H instruction requires 2-machine cycles (M_1 and M_2). M_1 requires 4-states and M_2 requires 3-states, total of 7-states as shown in figure. Status signals IO/M , S_1 and S_0 specifies the 1st machine cycle as the opcode fetch.

In T_1 -st bus A_{15} \leftrightarrow AD_0 the data the data The ins the con status s cycle is and-the and M_2 opcode identifi $A, 05H$ is

Muen

MVI A 05H

C 1 1 1

Prob.1 states a

Sol. LD used to indicat accumu

Examp location the con complet

Ope

L2

N diagram 'LDAX

and $ALE = 1$. In T_2 -state, the RD line goes low, and the data 06H from memory location 1000H are placed on the data bus. The fetch cycle becomes complete in T_3 -state. The instruction is decoded in the T_4 -state. During T_4 -state, the contents of the bus are unknown. With the change in the status signal, $IO/M = 0$, $S_1 = 1$ and $S_0 = 0$, the 2nd machine cycle is identified as the memory read. The address is 1001H and the data byte [7AH] is fetched via the data bus. Both M_1 and M_2 perform memory read operation, but the M_1 is called op-code fetch i.e., the 1st machine cycle of each instruction is identified as the opcode fetch cycle. Execution time for MVIM, 7AH i.e., memory read machine cycle and instruction cycle is:

Clock frequency of 8085 = 3.125 MHz

Time (T) for one clock = $1/3.125 \text{ MHz} = 0.32 \mu\text{s}$

Time for Memory Read = $3T = 3 \times 0.320 \mu\text{s} = 0.96 \mu\text{s}$

Total Execution time for Instruction = $7T = 7 \times 0.320 \mu\text{s} = 2.24 \mu\text{s}$

The Timing Diagram is

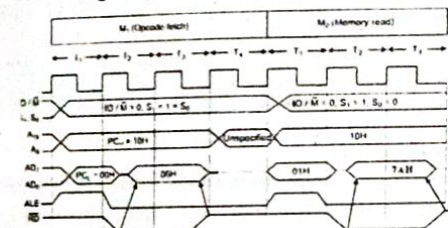


Fig. : Timing Diagram

Prob.27(a) What are peripheral mapped I/O memory and memory mapped I/O? Differentiate b/w them.

[R.T.U. 2015]

OR

Explain memory mapped and I/O Mapped Techniques.

[R.T.U. 2010]

(b) Write the uses of buffers, decoders, encoders and D-flip/flop in Microprocessor? [R.T.U. 2015]

Sol.(a) I/O devices (peripherals) like keyboard, switches, converters, CRT, printers, LEDs etc. are communicated with microprocessor using interfacing methods. In the 8085 microprocessor based system, I/O devices can be interfaced using both techniques : **memory-mapped I/O** and **peripheral-mapped I/O**. The process of data transfer is identical in both.

Peripheral-Mapped I/O

In this method, mostly lower address lines ($A_0 - A_3$) are used to get an address of I/O ports and this method is identified by 8-bit address. Following steps are used in this method:

1. Address lines ($A_0 - A_3$ or $A_8 - A_{15}$) are decoded to generate a unique pulse corresponding to the I/O address on the bus and this is called device or I/O address pulse.
2. Device address pulse with the control signal (\overline{IO} or \overline{IOW}) generates a device select (I/O Select) pulse that is generated only when both signals are asserted.
3. Use the I/O pulse to activate the interfacing device or I/O port.

Memory-Mapped I/O

I/O devices are treated as memory locations. Each I/O device will have 16-bit address, ranges from 0000H to FFFFH. MEMR and MEMW are the control signals used in this technique. The microprocessor communicates with I/O device as if it were one of the memory locations. Steps involved are:

1. Decode the address bus to generate the device address pulse.
2. AND the control signal with the device address pulse to generate the device select pulse.
3. Use the device select pulse to enable the I/O port. 16-bit instructions like LDA, STA are used here.

Comparison of Memory-Mapped I/O and Peripheral-Mapped I/O

Characteristics	Memory-Mapped I/O	Peripheral-Mapped I/O
1. Device address	16-bit	8-bit
2. Control signals for I/O	MEMR/MEMW	$\overline{IO}/\overline{IOW}$
3. Instructions available	STA, LDA, LEAX, STAX, MOV M, R, ADD M, SUB M, ANA M etc.	IN and OUT
4. Data transfer	Between any register and I/O	Only between I/O and the accumulator
5. Maximum number of I/Os possible	The memory map (64K) is shared between I/Os and system memory.	The I/O map is independent of the memory map. 256 input and output devices are connected.
6. Execution speed	13 T-states (STA, LDA) 7 T-states (MOV, M, R)	10 T-states
7. Hardware requirements	More hardware is needed to decode 16-bit address	Less hardware is needed to decode 8-bit address
8. Other Features	Arithmetic or logical operations can be directly performed with I/O devices	Not available

Sol.(b) Buffer : A data buffer (or just buffer) is a region of physical memory storage used to temporarily store data while it is being moved from one place to another.

Buffers can increase application performance allowing synchronous operations such as file reads or writes to complete quickly instead of blocking while waiting for hardware interrupts to access a physical disk subsystem. Instead, an operating system can immediately return successful result from an API call, allowing an application to continue processing while the kernel completes the disk operation in the background. Further benefits can be achieved if the application is reading or writing small blocks of data that do not correspond to the block size of the disk subsystem.

allowing a buffer to be used to aggregate many smaller read or write operations into block sizes that are more efficient for the disk subsystem, or in the case of a read, sometimes to completely avoid having to physically access a disk.

Encoders and Decoders : In digital domain, for ease of transmission of data, the data is often encrypted or placed within codes and then this secured code is transmitted. At the receiver, the coded data is decrypted or gathered from the code and is processed to be displayed or given to the load accordingly.

This task of encrypting the data and decrypting the data is done by Encoders and Decoders.

Encoders : Encoders are digital ICs used for encoding. By encoding, we mean generating a digital binary code for every input. An Encoder IC generally consists of an Enable pin which is usually set high to indicate the working. It consists of 2^n input lines and n output lines with each input line being represented by a code of zeros and ones which is reflected at the output lines.

Decoders : Decoders are digital ICs which are used for decoding. In other words the decoders decrypt or obtain the actual data from the received code, i.e. convert the binary input at its input to a form, which is reflected at its output. It consists of n input lines and 2^n output lines. A decoder can be used to obtain the required data from the code or can also be used for obtaining the parallel data from the serial data received.

D Flip-Flops : They provide a separation of state (before and after the clock signal) and therefore help control things that might otherwise run away if left to propagate on their own.

A D flip-flop allows you to know in your design exactly when you accepted data and froze it. Because of that property, you can then design orderly transitions from state to state, which is very useful in microprocessors which are state machines. State machines proceed in an orderly way from condition to condition, unlike analog systems which constantly change their state. The property also allows you to make sure operations are synchronized across many devices.

Prob.28(a) Explain the difference between 1 byte, 2 byte and 3 byte instructions. Quote suitable examples.

(b) Explain STA with an example. Write the no. of T states and machine cycles involved in it.

[R.T.U. 2014]

Sol. (a)
1, 2 and
respect
instruc
accumu
register
eg. : AL

long. e
content
is the 2
as it co
has to j

1 byte

2 bytes

3 bytes

Note :
instruc
memo

Sol. (b)
used to
locati
instruc
gives
gives
accum
memo
STA i

Op

S

diagram
STA (

PREVIOUS YEARS QUESTIONS

PART-A

Prob.1 What are interrupts?

[R.T.U. 2019]

Sol. Interrupts : Interrupt is a process by which some external device or peripheral informs microprocessor to become ready for data communication with the help of accepting the request made. Interrupt is a process in which the control of the program transfers from the main program to the starting location defined by the interrupt.

Prob.2 Differentiate between Synchronous and Asynchronous data transfer.

[R.T.U. 2019]

OR

What do you mean by synchronous and asynchronous data transfer?

[R.T.U. 2010]

Sol. In synchronous transmission, a receiver and a transmitter are synchronized. In this method, a block of characters is transmitted along with synchronization information. See fig. (a) such type of transmission is used for high speed data transfer (more than 20 k bits / second).

In asynchronous transmission, each character is transmitted with start and stop bits. Transmission begins with one start bit (low), followed by 8 bit code of the character and one or two stops bits (high) at the end. See fig. (b). The asynchronous transmission is used in low-speed data transfer (less than 20k bits / second).

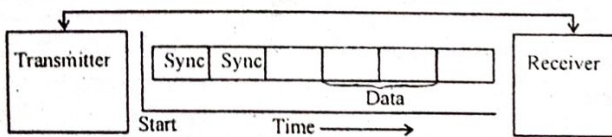


Fig. (a): Synchronous Transmission

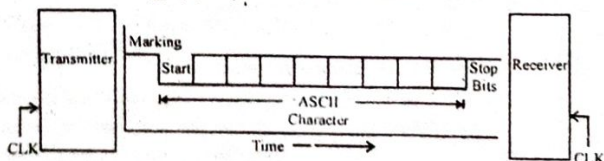


Fig. (b): Asynchronous Transmission

Prob.3 Differentiate vectored and non vectored interrupts of 8085.

[R.T.U. 2016]

Sol. Difference between vectored and non vectored interrupts :

S. No.	Vectored Interrupt	Non-vectored Interrupt
1.	Interrupts which are automatically transformed (or vectored) to specific locations on memory page 00H without any external hardware are called vectored interrupts.	Interrupts which are not automatically vectored to specific locations on memory page 00H are called non-vectored interrupts.
2.	Vectored interrupts do not require the \overline{INTA} signal or an input port.	Non-vectored interrupts require external hardware to transfer it to a particular location.
3.	These interrupts and their call location are as follows : <div style="display: flex; justify-content: space-between;"> <div> Interrupts (i) TRAP (ii) RST 7.5 (iii) RST 6.5 (iv) RST 5.5 </div> <div> Call Location 0024 H 003C H 0034 H 002C H </div> </div>	Only INTR interrupt is non-vectored interrupt. INTR requires RSTN instruction for its working.

Prob.4 Write an assembly language program to add a memory block of 10 bytes starting from 2000 H and store the sum in the memory at 200 F H location. If carry generated, store the carry at 2010 H location. [R.T.U. 2012]

Sol. Label Mnemonics

```

LXI H, 2000 H
MVI C, 0A H
XRAA
Loop:  ADD M
      JC END
      INX H
      DCR C
      JNZ Loop
      STA 200 F H
      JMP SKIP
      MVI A, 01 H
      STA 2010 H
SKIP:  HLT
    
```


Subroutines are a powerful programming tool, and the syntax of many programming languages includes support for writing and using them. Judicious use of subroutines (for example, through the structured programming approach) will often substantially reduce the cost of developing and maintaining a large program, while increasing its quality and reliability. Subroutines, often collected into libraries, are an important mechanism for sharing and trading software. The discipline of object-oriented programming is based on objects and methods (which are subroutines attached to these objects or object classes).

The content of a subroutine is its body, which is a piece of program code that is executed when the subroutine is called or invoked.

Subroutine Execution : A subroutine may be written that it expects to obtain one or more data values from the calling program (to replace its parameters or form parameters). The calling program provides actual values, these parameters, called arguments. Different programming languages may use different conventions for passing arguments :

Convention	Description	Common use
Call by value	Argument is evaluated and copy of value is passed to subroutine	Default in most Algol-like languages after Algol 60, such as Pascal, Delphi, Simula, CPL, PL/M, Modula, Oberon, Ada, and many others. C, C++, Java (References to objects and arrays are also passed by value)
Call by reference	Reference to argument, typically its address is passed	Selectable in most Algol-like languages after Algol 60, such as Algol 68, Pascal, Delphi, Simula, CPL, PL/M, Modula, Oberon, Ada, and many others. C++, Fortran, PL/I

Prob.9 What is the use of "Stack"? Illustrate the PUSH and POP operations with help of suitable examples. [R.T.U. 2004]

OR

Explain use of stack during interrupt processing. [R.T.U. 2004]

OR

Explain the Stack with examples. [R.T.U. 2004]

Sol. Stack : "The stack is a group of memory location in R/W memory that is used for temporary storage of binary information during the execution of program".

The stack can be defined as a set of memory locations in the R/W memory, specified by a programmer in the main program. This set of memory location can be used to store the data temporarily during the execution of a program. Sometimes, it becomes necessary during the execution of program to store the contents of certain registers because these registers are required to store data for some other operations in the subsequent steps of a program. The contents of these registers are moved to certain memory location by PUSH instruction. Then these registers can be used for some other operations. After completing these operations, those contents of registers, which were saved in the memory, are transferred back to these registers by POP instructions. Memory locations for this purpose are specified by a programmer in the beginning of a program. These memory locations are called the stack top. A special 16-bit register called the stack pointer register (SP) holds the address of the stack top.

The stack operation can be described in the following way :

(i) In the beginning, the stack pointer (SP) is loaded by 16-bit memory address using the instruction LXI SP, 16-bit address. This instruction will load the 16-bit memory address in the stack pointer register of the microprocessor. Once the stack location is defined, the storing of data begins at the memory location that is one less than the address in the stack pointer register. For example, the instruction LXI SP 3099H will load memory address 3099 in the stack pointer register. Now the storing of data begins at memory location 3098 and continues in the decreasing memory address like 3097, 3096 and so on.

(ii) Now the data bytes in the register pairs can be stored on the stack. The instruction for example PUSH B will copy the contents of register pair BC on the stack, the contents of B register are copied in the memory location 3098 and contents of C register are copied in the memory location 3097. After this Push B instruction, the stack pointer register is also decremented by 2 i.e. it contains the address 3097.

(iii) The register pair BC now can be used to store any other data in subsequent step of a program. When this operation is finished, the contents of register pair BC which were stored on the stack can be transferred back to this register pair by using the instruction POP B. This instruction

will copy the contents of top of the stack memory located 3097 in the register C and 3098 in the register B and stack pointer register again contains the address 30 (incremented by two).

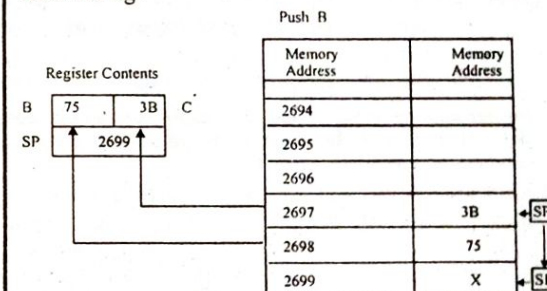
Example :

The following example also explains the stack operation. Consider the contents of register B are 75 and contents of register C are 3B. The stack is initialized with instruction LXI SP, 2699.

Now the instruction PUSH B will copy the contents of register B in the memory location 2698 and the contents of register C in the memory location 2697. After this PUSH B instruction, the stack pointer register is decremented by 2 and now, it contains the memory address 2697 as shown in

After Push B instruction, the stack pointer register contains the memory address 2697. The contents of B register and C register are not destroyed, as these registers can be used to store any other data.

Now the instruction POP B will transfer the contents of 2697 to register 3B and contents of 2698 to register E as shown in fig.



After the POP B instruction, the stack pointer register contains the memory address 2699.

Prob.10 Explain the addressing modes for 8085 microprocessor by giving examples of each. [R.T.U. 2004]

OR

Explain addressing modes of 8085. [R.T.U. 2004]

Sol. There are several different addressing modes for 8085 microprocessor. Some of them are given below :

- Inherent/Implied Addressing Mode :** belongs to the 0-Address class. It is essentially not an addressing mode as no addressing mode is used to refer any operand. Only the opcode for the instruction is provided. There is at most one operand required in some instructions.

Sem.) C.S. Solved Paper

the stack memory location in the register B and contains the address 30

explains the stack operation. B are 75 and contents initialized with instruction

1 B will copy the contents of 2698 and the contents of 2697. After this PUSH instruction, register B is decremented by 1 and address 2697 as shown in the stack pointer register. The contents of B register as these registers can

3 will transfer the contents of 2698 to register E

Memory Address	
3B	← SP
75	
X	← SP

on, the stack pointer register is 9.

Addressing modes for 8085

[R.T.U. 20

085. [R.T.U. 20

Addressing modes for 8085 are given below :

Register Addressing Mode :

It is essentially not a memory addressing mode. It is used to refer to the register. The instruction is provided in some instructions

Microprocessor and Interfaces

and that operand is implicitly used from the accumulator register. Like in case of NOT instruction, the operand is fetched from the accumulator and the result (the complement) is also stored in the same register. HALT is an instruction which doesn't require any operand and hence it uses Inherent Addressing Mode.

- **Immediate Addressing Mode:** In this addressing mode, the operand itself is provided along with the opcode of the instruction. There is no memory address required for the operand to be fetched. It is commonly used to store constant in a register. Like storing the value of π (pi) in the register B. It can also be used in a loop. We can initialize the register with the value for which we want to use the loop. For example - MVI B, 40H (40H is copied into the register B).

- **Direct/Absolute Addressing Mode:** We provide the addresses of the operands along with the opcode for the instruction. The size of the address can be different depending on what type of address we are referring, like I/O address is of one byte whereas memory address is 2 bytes or 16 bits. This address will point to the memory location where the operand is stored. It is the most natural and simple way for providing operands for the instruction and takes three machine cycles to read the address (which can be 2 bytes) and fetch the operand. For example - LDA 3000H (The content at the location 3000H is copied to the register A).

- **Register Addressing Mode:** The operands are stored in the registers. Therefore, we only need to provide the address of the register along with the opcode of the instruction. This results in a shorter instruction which takes less time to read. The operand fetch is also fast as we just have to look into the corresponding register and not fetch it from slow main memory. For example - MOV A, C (the content of C is copied into the register A).

- **Register Indirect Addressing Mode:** In this addressing mode, we provide along with the instruction opcode, an address is stored in a register pair (memory address is 16 bits) which points to memory location that stores the address of the operand. In this case, the effective address is essentially the content of the memory location in the register provided with the opcode. This is slower in comparison to direct mode of addressing as it involves reading from a register along with a memory fetch. But it is a very flexible way of providing the address as we don't have to change the program or the instruction to get a different operand as we can just change the address stored in the register provided as part of the instruction. For example - MOV A, M (data is transferred from the memory location pointed by the register to the accumulator).

Prob.11 (a) C

route

Loop :

Assi
is 3
(b) Writ
bit c
(c) Diff
inte

Sol.(a) Given :

Value of T =
T-States of
In Loop
Total T-State
T-States of F
Therefore
Time Delay =
Value of T = 1
Time Delay =

Sol.(b) 16 bit a
also note that y
NUL-terminat
[org 0x100]

[bits 16]
start:
mov ax, 0
printloop:
push ax
Call myconver

mov ah, 0x09
int 0x21
mov dx, newli
mov ah, 0x09
int 0x21
pop ax
inc ax
cmp ax, 50000

jbe printloop

mov ax, 0x4c

int 0x21
newline: db 1

MI.55

Prob.15 Explain the RIM and SIM instruction briefly.

[R.T.U. 2014]

OR

Explain the format of SIM.

[R.T.U. 2014]

OR

Explain the instructions SIM and RIM and illustrate how to use them for 8085 interrupts.

[Raj. Univ. 2014]

Sol. SIM : Set Interrupt Mask

Opcode	operand	bytes	m cycles	T states
SIM	None	1	1	4

Description : This instruction is used to implement 8085 interrupts (RST 7.5, 6.5, 5.5) and serial data output. The instruction interprets accumulator contents as follows:

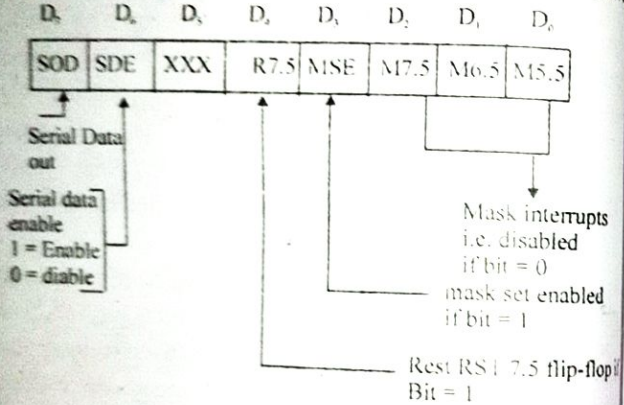


Fig.

SOD - Serial output data : Bit D_7 of accumulator is latched into SOD output time and made available to a serial peripheral if bit $D_6 = 1$.

SDE - Serial data enable : If this bit is 1 it enables serial output to implement serial output bit need to be enabled.

XXX - Don't care (None)

MSE - Mask set enable : If this bit is high it enables function of bit D_2 , D_1 and D_0 . This is master control over interrupt masking bits.

If this bit is low, bit D_2 , D_1 and D_0 do not have any effect on masks.

M7.5	$D_2 = 0$	RST 7.5	is enhanced
	1	RST 7.5	is disabled/masked
M6.5	$D_1 = 0$	RST 6.5	is enabled
	1	RST 6.5	is masked
M5.5	$D_0 = 0$	RST 5.5	is enabled
	1	RST 5.5	is masked

MI.56

RIM : Read Interrupt Mask

This instruction is used to read/check the status of interrupt 7.5, 6.5 and 5.5 and read to serial input bit. The instruction loads eight bits in accumulator with following interpretation.

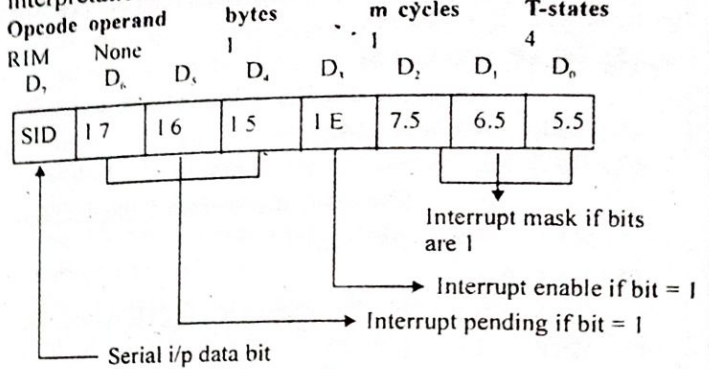


Fig.

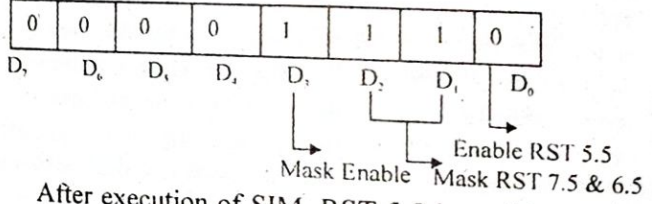
Use of RIM & SIM for 8085 Interrupts

SIM : We want to enable interrupt RST 5.5 and mask other interrupts we can use SIM instructions as follows :

MVI A, 0E : This will set D_3 i.e. RST 5.5 at $D_0 = 0$

SIM : Enable RST 5.5.

In this instruction set, accumulator is loaded with 0EH.



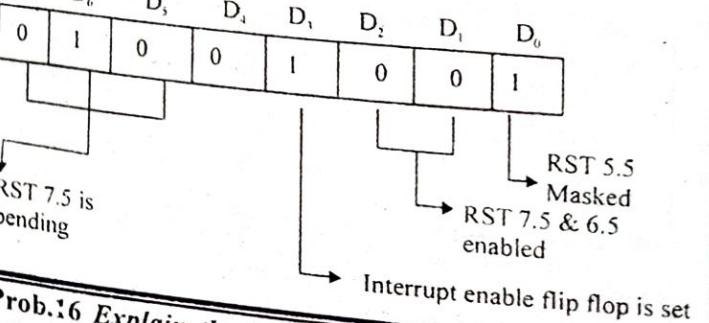
After execution of SIM, RST 5.5 is enable and other interrupts are disabled/ masked.

RIM : If we want to check the status of interrupt either they are enable/disable or pending. This instruction will be best way to do this.

Following instruction interpretation of accumulator as :

Opcode	operand	bytes	m cycles
RIM	None	1	1

After execution of above instruction, accumulator contained 49H that means.



Prob.16 Explain the implementations of stack in 8085 microprocessor programming.

[R.T.U. 2014]

8085 MICROPROCESSOR INTERFACING

4

CHAPTER IN A NUTSHELL

❑ 8255A Programmable Peripheral Interface

8255A, also known as PPI is widely used, programmable, parallel I/O device. It is an important general purpose I/O device that can be used with almost any microprocessor. It can be programmed to transfer data under various conditions, from simple I/O to interrupt I/O. It is economical and flexible and can be used when multiple I/O ports are required.

❑ 8254 Programmable Interval Timer

8254 is popular programmable interval timer chip. Intel 8254 contains three identical 16-bit down counters that can operate independently in any one of the six modes. A 16-bit count is loaded in register and control word is loaded in the control word register. Count is decremented until it reaches zero. When count is terminated, a pulse is generated that can

be used to interrupt the microprocessor. Counter can operate in either binary or BCD. 8254 includes status read command which helps in reading count by microprocessor while the counter is decrementing.

8254 is an upgraded version of 8253, and both of them are pin-compatible.

❑ I/O Interfacing

I/O devices (peripherals) like keyboard, switches, converters, CRT, printers, LEDs etc. are communicated with microprocessor using interfacing methods. In the 8085 microprocessor based system, I/O devices can be interfaced using both techniques : **memory-mapped I/O** and **peripheral-mapped I/O**. The process of data transfer is identical in both.

PREVIOUS YEARS QUESTIONS

PART-A

Prob.1 What is the need of DMA in microprocessor?

[R.T.U. 2019, 10]

Sol. DMA : The bulk data transfer from fast I/O devices to the memory or from the memory to I/O devices through the accumulator, is a time consuming process. For such a situation, the direct memory access (DMA) technique is preferred. In DMA data transfer scheme, data are directly transferred from I/O devices to RAM or from RAM to I/O devices. For DMA data transfer, the data and address buses come under the control of the peripheral device which wants DMA transfer. The microprocessor has to retain the control of the address and data buses for DMA data operation on

the request of the I/O device. For DMA data transfer, I/O device must have its own registers to store by the control and memory address. It must also be able to generate control signal required for DMA data transfer.

Prob.2 Write a control word of 8255 in IO mode 0, if port A and port B is input and port C is in output port.

[R.T.U. 2019]

Sol. The ports A, B and C can be configured as simple input or output ports by writing the appropriate control word in the control word register. In the control word, D_7 is set to 1 (to define a mode set operation) and D_6, D_5 and D_2 are all set to 0 to configure all the ports in mode 0 operation. The status bits D_4, D_3, D_1 , and D_0 then determine whether the corresponding ports are to be configured as Input or Output. In port A and port B are to operate as input ports and port C lower and upper as output, the control word that we have to be loaded into the control register will be as follows

PART-B

Prob.5 Explain various modes supported in 8254 Timer in detail. [R.T.U. 2019]

Sol. Operation Modes of 8254 : The 8254 can operate in six different modes :

Mode 0: Interrupt on Terminal Count

In this mode, initially the OUT is low. Once a count is loaded in the register, the counter is decremented every cycle, and when the count reaches zero, the OUT goes high. This can be used as an interrupt. The OUT remains high until a new count or a command word is loaded. Figure shows that the counting ($m = 5$) is temporarily stopped when the gate is disabled ($G = 0$) and continued again when the gate is at logic 1.

Mode 1: Hardware-Retriggerable One-Shot

In this mode, the OUT is initially high. When the Gate is triggered, the OUT goes low, and at the end of the count, the OUT goes high again, thus generating a one-shot pulse (Figure, Mode 1).

Mode 2: Rate Generator

This mode is used to generate a pulse equal to the clock period at a given interval. When a count is loaded, the OUT stays high until the count reaches 1 and then the OUT goes low for one clock period. The count is reloaded automatically, and the pulse is generated continuously. The count = 1 is illegal in this mode.

Mode 3: Square-Wave Generator

In this mode, when a count is loaded, the OUT is high. The count is decremented by two at every clock cycle, and when it reaches zero, the OUT goes low, and the count is reloaded again. This is repeated continuously, thus a continuous square wave with period equal to the period of the count is generated. In other words, the frequency of the square wave is equal to the frequency of the clock divided by the count. If the count (N) is odd, the pulse stays high for $(N + 1)/2$ clock cycles and stays low for $(N - 1)/2$ clock cycles.

Mode 4: Software-Triggered Strobe

In this mode, the OUT is initially high; it goes low for one clock period at the end of the count. The count must be reloaded for subsequent outputs.

Mode 5: Hardware-Triggered Strobe

This mode is similar to Mode 4, except that it is triggered by the rising pulse at the gate. Initially, the OUT is low and when the gate pulse is triggered from low to high, the count begins. At the end of the count, the OUT goes low for one clock period.

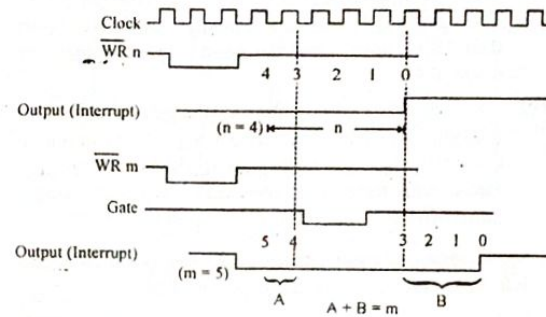
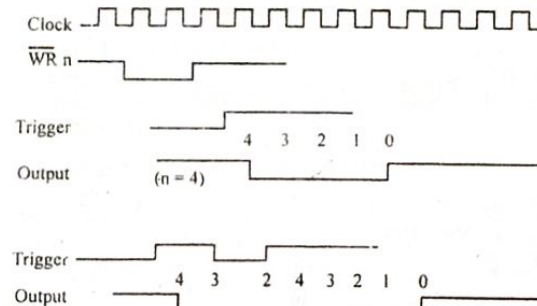
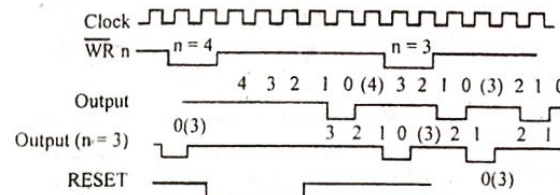
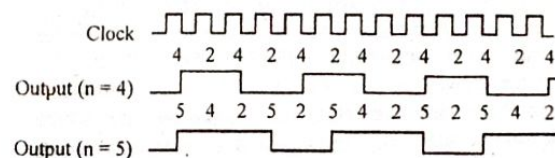
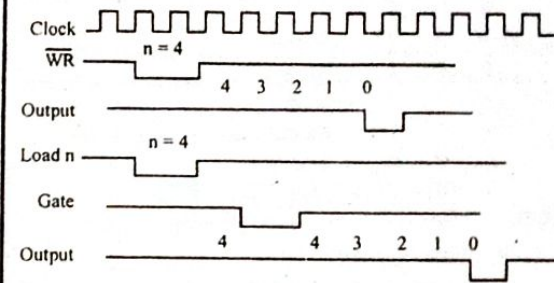
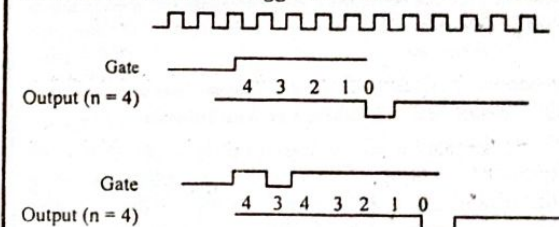
Mode 0: Interrupt on Terminal Count**Mode 1: Retriggerable One-Shot****Mode 2: Rate Generator Clock****Mode 3: Square Wave Generator****Mode 4: Software Triggered Strobe****Mode 5: Hardware Triggered Strobe**

Fig. : Six Modes of 8254

Prob.6 Write and explain control word in 8254 Timer in detail. [R.T.U. 2019]

Sol. Control Word of 8254 PIT : The register selected when line A_0 and A_1 are at logic 1 ($A_0 A_1 = 11$). It is used to write a command word which specifies the counter to be used. Its mode is either a read or a write operation. The control word format and mean of particular bit is shown :

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
-------	-------	-------	-------	-------	-------	-------	-------

SC_1 SC_0 RW_1 RW_0 M_2 M_1 M_0 BCD

Select Counter (SC)

SC_1	SC_0	
0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2
1	1	Read back command

RW (Read Write)

RW_1	RW_0	
0	0	Counter latch command
0	1	Read / Write least significant bit only
1	0	Read / Write most significant bit only
1	1	Read / Write least significant bit first than most significant bit

that is programmed for mode 2 becomes a "divide by n" counter. The out pin of the counter goes to low for one input clock period. The time between the pulse going low is dependent on the present count in the counter's register, i.e., the time of the logical pulse. For example, to get an output frequency of 1000 Hz, the period would be $1/1000 \text{ s} = 1 \text{ ms}$ or $1000 \mu\text{s}$. If an input clock of 1 MHz was applied to the clock input of the counter #0, then the counter #0 would need to be programmed to 1000 μs . This could be done in decimal or in BCD. The formula is

$$N = \frac{F_{\text{clk}}}{F_{\text{out}}}$$

Where,

N = Value to be loaded

F_{clk} = Input clock frequency

F_{out} = Output frequency

Thus, in our case

$$F_{\text{out}} = \frac{1}{6} F_{\text{clk}}$$

$$F_{\text{clk}} = 6 F_{\text{out}}$$

Putting values, we get

$$N = \frac{6F_{\text{clk}}}{F_{\text{out}}}$$

$$N = 6$$

To get an output frequency equal to $\frac{1}{6}$ of input clock frequency, the period would be $1/(1/6) \text{ s}$ or 6 s.

PART-C

Q.17 Explain the organization and architecture of 8255 Programmable Peripheral Interface IC with a function block diagram also draw the interfacing scheme of 8255 and 8085 in memory mapped I/O mode.

[R.T.U. 2019]

Sol. Block Diagram of 8255 : The 8255 is widely used programmable peripheral interface i.e. PPI 8255. It is a general purpose programmable I/O device.

Microprocessor and Interfaces

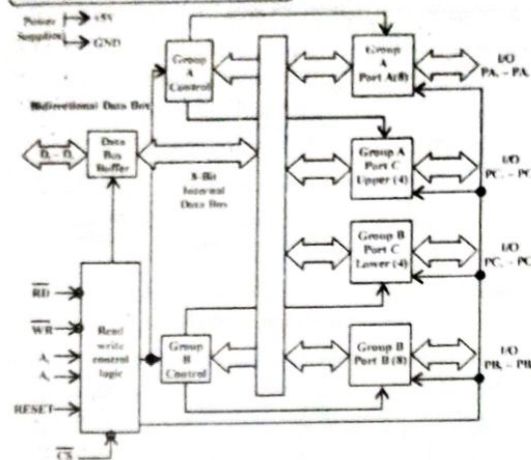


Fig. 1 : Block Diagram of 8255A

The Block diagram of 8255 is shown in above fig.1. It shows:

- Data bus buffer
- Read / Write logic control
- Group A and Group B control
- Port A and B
- Port C

Control Lines and I/O Port Address of 8255

- RD** : The $\overline{\text{RD}}$ (read) signal enable the read operation. When this signal goes low, the microprocessor reads data from the selected I/O port of 8255.
- WR** : The WR (write) signal enables the writes operation. When this signal is low, the microprocessor writes into a selected I/O port of control register.
- RESET** : When the reset signal is high, it clears the control register and sets all ports in the input mode.
- CS, A_0 and A_1** : These are the device select signals. CS is the master chip select and A_0 and A_1 specify one of the I/O port or the control register.

Operation mode of 8255 : Control word is written in the control register of 8255. The control word written in the control register, specify the I/O function for each port. The control register can be accessed to write a control word when address lines A_1 and A_0 are at logic 1.

In the control word, bit D_7 is 1, then bit $D_6 - D_0$ determine the input or output function of various ports and

the mode of operation. If D_7 bit is 0 then port C operates in Bit Set/Reset mode. In the BSR mode, functions of port A and port B are not affected.

The control word of 8255 is divided into two different control word formats which designate two basic modes :

- BSR mode (when $D_7 = 0$)
- I/O mode (when $D_7 = 1$)

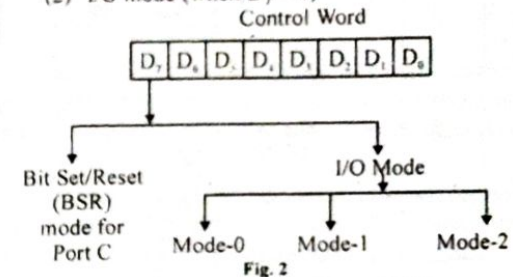


Fig. 2

(1) BSR Mode : The bit Set/Reset BSR is concerned with eight bits of port C only. The individual bit of port C can be set or reset by writing an appropriate control word in the control register. BSR bit does not used after any previously transmitted control word with bit $D_7 = 1$.

In BSR mode, individual bit of port C can be used for applications such as an on/off switch.

Control word format in the BSR mode is given in figure 3.

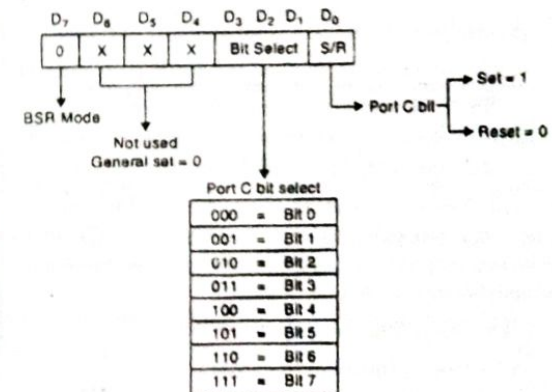


Fig. 3 : Control word format in BSR mode

(2) I/O Mode : A control word with bit $D_7 = 1$ is recognized as a I/O mode, and it is classified into three different I/O modes.

(i) Mode-0 (Simple input or output) : This mode is also called basic I/O mode. Port A, B, C are used as simple I/O ports in this mode.

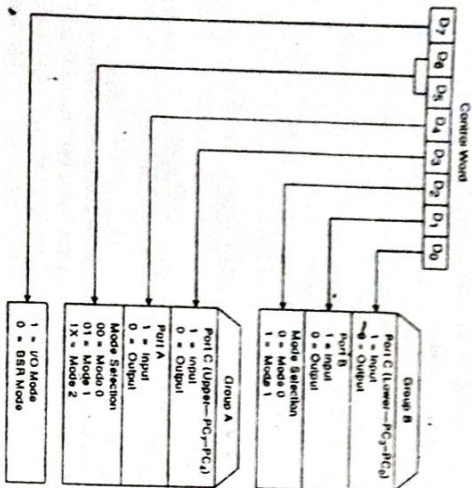


Fig. 4 : 8255 control word format for I/O mode

- It has following features:
- (a) Two 8-bit ports (Port A and port B) and two 4-bit ports (Port C_{upper} and C_{lower}). Port C can be used as 8-bit port.
 - (b) Any port can be used either as I/O or O/P port.
 - (c) Output ports are latched and input ports are not latched.
 - (d) A maximum of four ports available which is to say that overall 16 I/O configurations are possible.

(ii) **Mode-1 (Input or Output with Handshake):**

This mode is also called as strobed I/O mode. In this mode, handshake signals are exchanged between the microprocessor and peripherals prior to data transfer.

It has following two different modes.

- (A) Mode-1 (Input Control Signal)
- (B) Mode-1 (Output Control Signal)

(A) **Mode-1 (Input Control Signal) :** When port A and B are configured as input port, the associated control signals used for handshaking are shown in fig.5.

Port A uses the upper three signals: PC₇, PC₆ and PC₅. Port B uses lower three signals: PC₂, PC₁ and PC₀ for handshaking. PC₆ and PC₇ which can be used as input or output as programmed by bit D₃ of the control word. The functions of these control signals are given below:

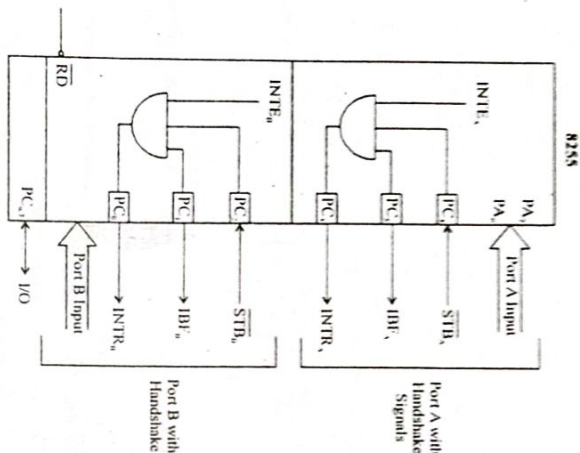


Fig. 5 : 8255 Mode 1: Input Configuration

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	1	1	I/O	1	1	X

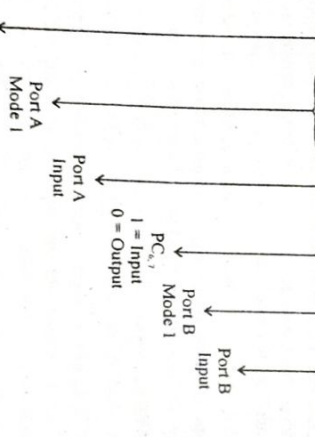


Fig. 6 : Control Word Mode 1 Input

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
I/O	I/O	INTR _A	INTR _B	INTR _A	INTR _B	INTR _A	INTR _B

Fig. 7 : Status Word Mode 1 Input

STB (Strobe Input) : This is an active low signal. This is generated by peripheral device, when transmitting a data byte.

Microprocessor and interfaces

IBF (Input Buffer Full) : This is active high acknowledgement signal generated by 8255 to indicate that input latch has received the data byte.

INTR (Interrupt Request) : This is active high output signal, used to interrupt microprocessor. This signal is generated if STB, IBF and INTE are all at logic 1 and this is reset by the falling edge of the RD signal.

INTE (Interrupt Enable) : This signal is used to generate INTR signal. Two flip flop INTE_A and INTE_B are used for this purpose.

Control and Status Words : The status word is accessed by reading port C i.e. when A₁A₀ = 10 and RD and CS both are low.

(B) **Mode-1 (Output Control Signal) :** When the port A and B configured as output port, Port A uses three signals: PC₇, PC₆ and PC₅. Port B uses another three signals: PC₂, PC₁ and PC₀ for handshaking signals. Leaving PC₄ and PC₃ which can be used as input or output as programmed by bit D₃ of control word.

OBF (Output Buffer Full) : This is an active low output signal. It goes low when microprocessor write data into output latch of 8255.

ACK (Acknowledge) : It is active low signal and indicate that output peripheral has taken the data from output buffer.

INTR (Interrupt Request) : This is active high output signal. INTR is set when OBF, ACK and INTE are all one and reset by falling edge of WR.

INTE (Interrupt Enable) : This is used for enable or disable the INTR signal.

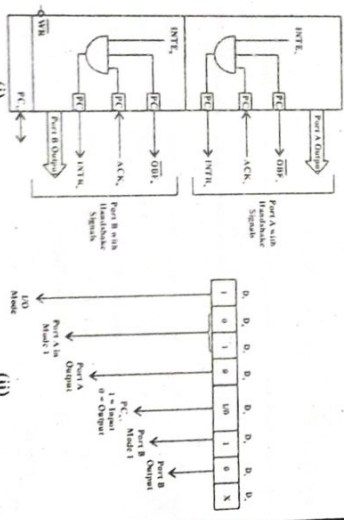


Fig. 8 : 8255 Mode 1: Output Configuration

Control and Status Words : When Port A and B are configured as output ports in the handshake, mode 1, the control word which will be required to be loaded into the control register.

(iii) **Mode-2 (Bidirectional Data Transfer)**

is also called as strobed bidirectional I/O mode. The operation is only for port A. In mode-2, port A uses signals from port C as handshaking signals for data and the remaining three signals as simple I/O or active for port B.

(A) **Mode-2 (Input Control Signal) :**

STB_A (Strobe) : This is an active low input signal.

IBF_A (Input Buffer Full) : When data is available in input buffer, 8255 will enable IBF_A signal for per devices.

(B) **Mode-2 (Output Control Signal) :**

OBF_A : This indicate that data available in output buffer.

ACK_A : This is active low signal, indicate peripheral reads data from port A and acknowledge reading.

INTR (Interrupt request) : This is an output signal given by 8255 to request MPV service.

(C) **Mode-2 (Status Word) :** The status of the operation in Mode-2 can be verified by reading the control word of port C.

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
INTR	INTR	INTR	INTR	X	X	X	X

Fig. 9

Interfacing 8255 with 8085

Description

Step 1:

- Lower order of 8-bit address A0-A7 is selected from AD0-AD7 using address latch/buffer 74373 and ALE signal.
- The separated address lines A₀-A₇ are connected to A₀-A₇ input pins of 8255 and the separate bus D₀-D₇ are connected to D₀-D₇ pins of 8255.
- Reset out of 8085 is connected to reset pin of 8255.

Step 2:

- 8255 does not have internal (separate) control generator, hence the IO/M(bar), RD(bar), WR(bar) control signals are not connected to 8255. These pins are 1st given to decoder using 3:8 decoder (Ex: IC 74138).
- The generated control signals IOA₀, IOA₁, IOA₂ are connected to RD(bar) and WR(bar) input of 8155.

Step 3:

- An active low signal of chip select logic is obtained decoding remaining address lines of lower order addresses A_2-A_7 .
- Chip select logic and I/O port address for this interfacing circuit are as:

Chips select address lines	Address select port	HEX address	Selected I/O								
A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0				
1	0	0	0	0	0	0	0	80H		PORT A	
1	0	0	0	0	0	0	1	81H		PORT B	
1	0	0	0	0	0	1	0	82H		PORT C	
1	0	0	0	0	0	1	1	83H		Chip Select	
										Register	

Interfacing Diagram

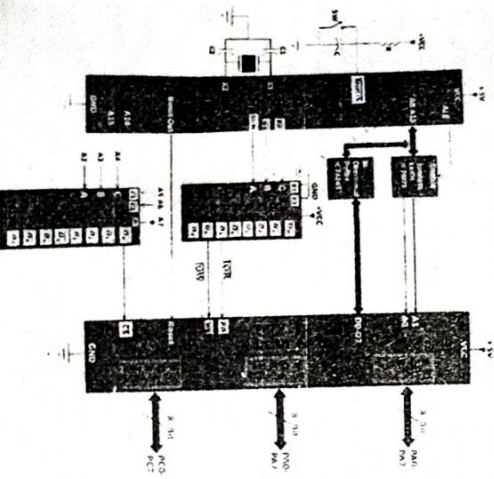


Fig. 10

Memory mapped I/O interfacing with 8085 microprocessor

In memory mapped I/O interfacing with 8085 microprocessor, the I/O devices are not given separate addresses other than treated as a memory location. Whose address range between 0000h to FFFFh (64k). But some part of the space is reserved for I/O devices. The advantage is

any instruction that references memory can also transfer data between an I/O device and the microprocessor, as long as the I/O port is assigned to the memory address space rather than to the I/O address space. The register associated with the I/O port is simply treated as memory location register.

This memory mapped I/O interfacing with 8085 microprocessor with an example in which address bit A_{15} designates whether instructions reference memory or an I/O device. If $A_{15} = 0$, a memory register is addressed. If $A_{15} = 1$, then a memory mapped I/O device is addressed. This assignment elevates the first 32k bytes of memory address space to memory and second 32k to memory mapped I/O devices. External logic generates devices select pulses for memory mapped I/O only when $\overline{IO/\overline{M}} = 0$, the appropriate address is on the address low and a \overline{WR} or \overline{RD} strobe occurs.

Input and output transfer using memory mapped I/O are not limited to the accumulator. For example, same of 8085 A instructions that can be used for input from memory mapped I/O ports.

MOV r, m : Move the contents of input port whose address is available in (H,L) reg pair to any internal register.
LDA addr : Load the acc with the content of the input port whose address is available as a second and third byte of the instruction.

Other instructions include, ANA M, ADD M, M provide input data transfer and computation in a single instruction. Same instruction that out the data from memory mapped ports are :

- MOV M,r
- STA addr
- MVI M, data
- SHLD addr

LHLD and SHLD carry out 16-bit I/O transfers with single instructions which reduce program executive time considerably. The price paid for this added capability is a reduction in directly addressable main memory and the necessity of decoding a 16-bit rather than an 8-bit address.

When a microprocessor puts out an address and generates a control strobe for a memory read, it has no way of determining whether the device that responds with data is a memory device or an I/O device. It only requires that the devices that respond with in the allowable access time or uses the READY line to request a sufficient number of WAIT states. It supplies an address data and a write strobe and continues its operations, external logic determines whether memory, I/O or anything at all receives the data transferred.

Microprocessor and Interfaces

Memory address decoding is nothing but to assign an address for each location in the memory chip. The data stored in the memory is accessed by specifying its address. Memory address can be decoded in two ways:

1. Absolute or fully decoding
2. Linear select or partial decoding

Absolute address decoding has many advantages such as :

1. Each memory location has only one address, there is no duplication in the address.
2. Memory can be placed contiguously in the address space of the microprocessor.
3. Future expansion can be made easily without disturbing the existing circuitry.

There are few disadvantages in this method :

1. Extra decoders are necessary.
2. Some delay will be produced by these extra decoders.

The main advantage of linear select decoding is its simplified decoding circuit. This reduces the hardware design cost. But there are many disadvantages in this decoding

1. Multiple addresses are provided for the same location.
2. Complete memory space of the microprocessor is not efficiently used
3. Adding or interfacing ICs with already existing circuitry is difficult.

Absolute Address Decoding: The 8085 microprocessor has 16 address lines. Therefore it can access 2^{16} locations in the physical memory. If all these lines are connected to a single memory device, it will decode these 16 address lines internally and produces 2^{16} different addresses from 0000H to FFFFH so that each location in the memory will have a unique address.

74LS138 address decoder to generate the chip select signals for each memory block. In this decoder when the address lines A_{15} , A_{14} and A_{13} are 000, the output line Y_0 will be activated as shown in Fig. 11. This in turn selects the first memory block. Similarly when these lines are 001 ($C=0$, $B=0$ and $A=1$) Y_1 will be activated and the second memory block will be selected.

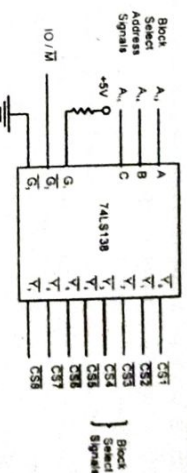


Fig. 11

In this type of memory interfacing, all the address lines (A_0 to A_{15}) have been used. Each location in the memory will have a single address. This type of address decoding is called as absolute or fully decoded addressing.

According to the value of A_0 and A_1 , any one register will be selected and to select one memory chip we need one chip select signal, CS signal as shown in the Figure 12.

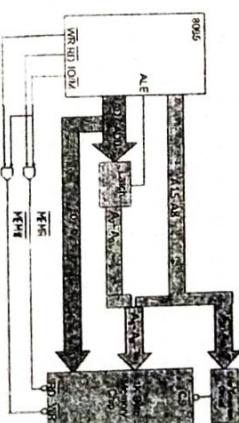


Fig. 12

Pro 13 Explain 8279 Keyboard Interface in detail.

- OR [R.T.U. 2019]
- Explain 8279 with block diagram. [R.T.U. 2016]
- OR
- Draw and explain the block diagram of 8279 keyboard/display interface. [R.T.U. 2015]
- OR
- Explain working and modes of 8279 keyboard/display interface. [R.T.U. 2014]
- OR
- Explain various programming modes of 8279 keyboard and display controller. Also draw a block diagram showing its interfacing with microprocessor 8085. [R.T.U. 2013]
- OR
- List the major components of 8279 keyboard/display interface and explain their functions. [R.T.U. 2012]
- OR
- Explain input modes provided by 8279. [R.T.U. 2008, Raj. Univ. 2001]

Keyboard Modes

(i) **Scanned Keyboard Mode with 2 Key Lockout :** In this mode of operation, when a key is pressed, a debounce logic comes into operation. During the next two scans, other keys are checked for closure and if no other key is pressed the first pressed key is identified. The key code of the identified key is entered into the FIFO with SHIFT and CNTL status, provided the FIFO is not full, i.e. it has at least one byte free. If the FIFO does not have any free byte, naturally the key data will not be entered and the error flag is set. If FIFO has at least one byte free, the above code is entered into it and the 8279 generates an interrupt (on IRQ line) to the CPU to inform about the previous key closures. If another key released before the first key, the key code is entered into FIFO. If the first pressed key is released before the others, the first will be ignored. A key code is entered to FIFO only once for each valid depression. Independent of other keys pressed along with it, or released before it. If two keys are pressed within a debounce cycle (simultaneously), no key is recognized till one of them remains closed, and the other is released. The last key, that remains depressed is considered as single valid key depression.

(ii) **Scanned Keyboard with N-Key Rollover :** In this mode, each key depression is treated independently. When a key is pressed, the debounce circuit waits for 2 keyboard scans and then checks whether the key is still depressed. If it is still depressed, the code is entered in FIFO RAM. Any number of keys can be pressed simultaneously and recognized in the order, the keyboard scan recorded them. All the codes of such keys are entered into FIFO. Note that, in this mode, the first pressed key need not be released before the second is pressed. All the keys are sensed in the order of their depression, rather in the order the keyboard scan senses them, and independent of the order of their release.

(iii) **Scanned Keyboard Special Error Mode :** This mode is valid only under the N-Key rollover mode. This mode is programmed using end interrupt/error mode set command. If during a single debounce period (two keyboard scans) two keys are found pressed, this is considered a simultaneous depression and an error flag is set. This flag, if set, prevents further writing in FIFO but allows generation of further interrupts to the CPU for FIFO read. The error flag can be read by reading the FIFO status word. The error flag is set by sending normal clear command with CF = 1.

(iv) **Sensor Matrix Mode :** In the sensor matrix mode the debounce logic is inhibited. The 8-byte FIFO RAM now acts as 8×8 -bit memory matrix. The status of the sensor switch matrix is fed directly to sensor RAM matrix. Thus the sensor RAM bits contain the row-wise and column-wise status of the sensors in the sensor matrix. The IRQ line goes high, if any change in sensor value is detected at the end of sensor matrix scan or the sensor RAM has a previous entry to be read by the CPU. The IRQ line is reset by the first data read operation, if AI = 0, otherwise, by issuing the end interrupt command. AI is a bit in read sensor RAM word.

Display Modes : There are various options of data display. For example, the command number of character can be 8 or 16, with each character organized as single 8-bit or dual 4-bit codes. Similarly there are two display formats. The first one is known as left entry mode or type writer mode since in a type writer the first character typed appears at the left-most position, while the subsequent characters appear successively to the right of the first one. The other display format is known as right entry mode, or calculator mode since in a calculator the first character entered appears at the rightmost position and this character is shifted one position left when the next character is entered. Thus all the previously entered characters are shifted left by one position when new character is entered.

(i) **Left Entry Mode :** In the left entry mode, the data is entered from the left side of the display unit. Address 0 of the display RAM contains the leftmost display character and address 15 of the RAM contains the rightmost display character. It is just like writing in our note books, i.e. from left to write. If the 8279 is in autoincrement mode, the display RAM address is automatically updated with successive read or writes. The first entry is displayed on the leftmost display and the sixteenth entry on the rightmost display. The seventeenth entry is again displayed at the leftmost display position.

(ii) **Right Entry Mode :** In the right entry mode, the first entry to be displayed is entered on the rightmost display. The next entry is also placed in the right most display to after the previous display is shifted left by one display position. The leftmost character is shifted out of that display at the seventeenth entry and is lost, i.e. it is pushed out of the display RAM.

Interfacing of 8279 with microprocessor 8085:

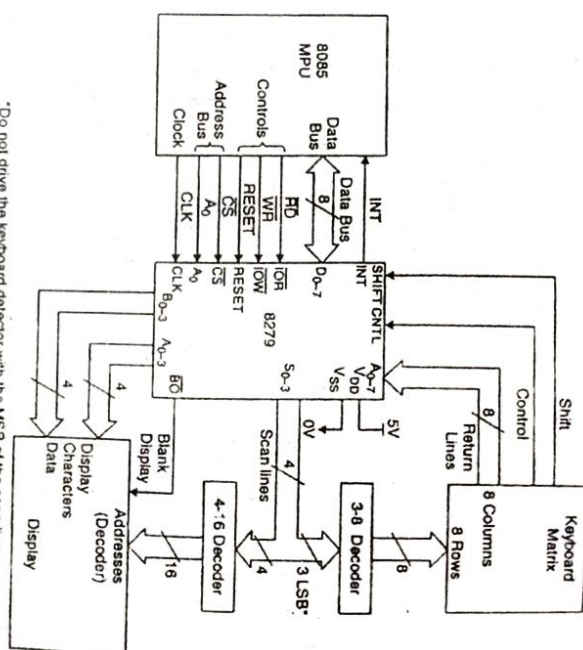


Fig. 1 : Interfacing Connection Block diagram of 8279

8279 Keyboard / Display Interface : 8279 is a general purpose programmable keyboard and display I/O interface. It can be interfaced with any type of CPU. It relieves CPU scanning keyboard refreshing display, debouncing the checking display etc.

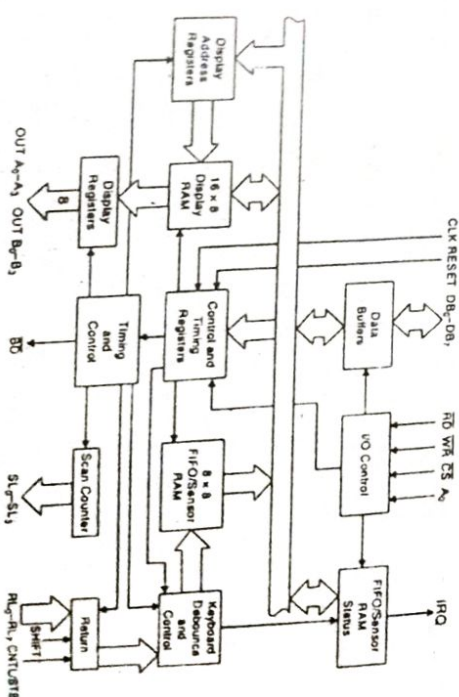


Fig. 2 : Block Diagram of 8279

The component of 8279 keyboard/display interface categorize in different section that are following:

1. Keyboard Section : It consists of :

- (i) Keyboard debounce
- (ii) FIFO / Sensor RAM
- (iii) Return Buffer
- (iv) Status Logic

(i) **Keyboard Debounce :** This section has 8 lines ($RL_0 - RL_7$) that can be connected to 8 columns of a keyboard plus two additional lines SHIFT and CNTL/STB (Control/Strobe). The status of the SHIFT key and the control key can be stored along with a key closure. The keys are automatically debounced and keyboard can operate in two modes (a) Two-key lockout (b) N-key roll over.

(a) **Two-key Lockout Mode :** In two key lockout, if two keys are pressed almost simultaneously then only 1st key is recognized.

(b) **N-key Roll Over Mode :** In the N-key roll over mode, simultaneously keys are recognized, their codes are stored in the internal buffer.

(ii) **FIFO / Sensor RAM :** The keyboard section also includes 8×8 FIFO (First in First Out) RAM. Its functions depends upon operating mode of 8279. This FIFO RAM consists of 8 registers that can store 8 keyboard entries; each is then read in the order of entries.

(iii) **Return Buffer :** The $RL_0 - RL_7$ lines are buffered and latched in this block. It scans $RL_0 - RL_7$ lines during each row scan. In scanned keyboard or sensor matrix mode $RL_0 - RL_7$ lines are not scanned in strobed mode.

(iv) **Status Logic :** The status logic keeps track on the number of entries and provides an IRQ (Interrupt Request) signal when the FIFO is not empty.

2. Scan Section : The scan section has a scan counter and four scan lines ($SL_0 - SL_3$). These scan lines can be decoded using a 4 to 16 decoder to generate 16 lines for scanning. These lines can be connected to the rows of a matrix keyboard and the digit drivers of a multiplexed display.

3. Display Section : Display section has 8 output lines divided into two groups $A_0 - A_3$ and $B_0 - B_3$. These lines can be used, either as a group of 8 lines or as two group of four in conjunction with the scan lines for multiplexed display. The display can be blanked by using the \overline{BD} line. This section includes 16×8 display RAM. The MPU can read from or write into any of these registers.

(i) **Display Address Register :** It contains two type of address registers i.e. microprocessor controlled address register and timing controlled address register.

(ii) **Display Register :** These registers convert the codes of character to be displayed. They are divided into two nibbles i.e. nibble A and nibble B.

Nibble A and nibble B can be blanked and inhibited individually.

4. MPU Interface Section : It consists of data buffer I/O control, control and timing register and timing and control logic.

(i) **Data Buffer :** These are bidirectional tristate buffers. It includes 8 bidirectional data bus ($DB_0 - DB_7$).

(ii) **I/O Control :** It uses \overline{CS} , A_0 , \overline{RD} and \overline{WR} lines to control data flow to and from the various internal registers and buffers.

(iii) **Control and Timing Registers :** This block consists of control and timing register. 8279 provides 8 control and timing registers which are distinguished by higher 3 bits of command word.

Input Modes By 8279 : The 8279 is a 40-pin device which functionally consists of (i) a CPU interface, (ii) a set of scan lines, (iii) input lines for key data, and (iv) output lines for display data.

There are three input modes :

- (i) Scanned Keyboard
- (ii) Scanned Sensor Matrix and
- (iii) Strobed input

(i) **Scanned Keyboard Mode :** This can have either enclosed or decoded scan lines. Keys are automatically debounced. In this scanned keyboard mode, there are two further alternative ways of operation :

- (1) 2-key lockout, and
- (2) N-key-rollover

In this mode, when a key is pressed, a unique 6-bit data is generated, characteristic of the key position data along with the CNTL and SHIFT states and 8-bit word is formed which is stored in a RAM inside the 8279. As soon as the 8-bit word is stored in this FIFO (First in First out) RAM, the interrupt request IRQ pin of the device goes high. This could be connected to one of the hardware interrupt lines of the CPU, which could be followed by, in the interrupt service routine program, an instruction to read the FIFO RAM. As the FIFO RAM is read the IRQ goes low, but becomes high again, if the FIFO RAM contains further data.

The data format for the scanned keyboard mode, is as follows :

The three scan bits (D_3, D_4, D_5) are from the scan counter and indicate the row on which the pressed key was located. The three return bits (D_0, D_1, D_2) are from the column counter and indicate the column on which the pressed key was located.

(1) **The 2-key lockout option :** In this mode when any key is pressed, the debounce logic is set. For the next two scans, it is checked whether other keys are pressed. At this stage several alternatives are possible.

(a) No other keypresses are found. This is then considered a single key depression, the data is taken into FIFO and IRQ is activated. (If the FIFO is full, the key data is not entered, an error flag is set).

(b) If one or more additional key depression is detected, no data entry to FIFO can occur. This status is resolved as follows :

(A) If all the keys are released before the key first pressed, then the first key data is entered into the FIFO.

(B) If the first key is released before others, then the keypress is entirely ignored. Note that the first key is the crucial one (there may be many other key presses in many different orders); it is the first keypress that is either entered or ignored.

If however, two keys are pressed within one debounce cycle, this is considered as simultaneous depression. Neither of them is recognized till one of them is released. The one that still remains pressed, is considered to be equivalent to a single key depression.

(2) **The N-Key rollover option :** In this mode, when a key is pressed, the debounce circuit waits for two scans and then checks again whether the key is still pressed. If it still is, the key data is then taken into the FIFO. There is no limit to the number of keys that can be pressed. For a simultaneous depression, the key data are entered according to the order of the keypresses.

A special error mode can be set in this option by End Interrupt/ Error Mode set command, if during signal debounce cycle, two keys are found pressed, then it sets an Error Flag, which present further data entering into FIFO, and sets an interrupt. The error flag is read from the FIFO STATUS word, and is reset by sending CLEAR command $C_2 = 1$.

(ii) **Scanned Sensor Matrix Mode :** In this mode, the keys are placed in the form of a matrix (scan lines comprising the columns and the return lines comprising the

rows), and the key status which can be address 8×8 (with encoded lines).

The data on each in eight columns of therefore maps to a SHIFT and CNTL size

In this mode, circuits) may be connected that device (or logic scan lines, the output data into the return

In this mode, has the advantage it was closed. The IRQ to have changed at the auto increment

the first data read set to one, then by (iii) Strobed accepted from the

enter at the rising (placed on return another encoded k (now correspondi lead to an 8-bit w

Prob.14 Explain all the n

Sol. 8254 Progr

8254 is popular 8254 contains three operate independent count is loaded i control word register zero. When count be used to interrupt in either binary command while the count 8254 is an are pin-compatible Control Operate

MICROPROCESSOR APPLICATIONS

5

CHAPTER IN A NUTSHELL

Microprocessor Application
Applications include such examples as the scanned LED display, the matrix keyboard and LCD etc.

LCD (Liquid Crystal Display)

An LCD is used in system where low power consumption is necessary. Watches, calculators and consumer electronics displays are its examples.

Interfacing a Matrix Keyboard

A matrix keyboard is commonly used as input device when more than eight keys are necessary instead of a linear format row of keys.

Keyboard Display Controller 8279

It is a hardware approach to interface a matrix keyboard and a multiplexed display. The software approach to interface a matrix keyboard and a multiplexed display of seven segment LEDs is used.

8279 reduces time in checking the keyboard and refreshing the display.

Keyboard entries redelayed and stored in the internal FIFO (first in first out) memory, an interrupt signal is generated with each entry.

The display segment can provide a 16-character scanned display interval with such device as LEDs.

This segment has 16×8 R/W memory (RAM) which can be used to read/write information for displayed purpose.

The display can be set up is either right entry or left entry format.

Level converters MC 1488

MC 1489

Communication buses: Centronics IEEE 488

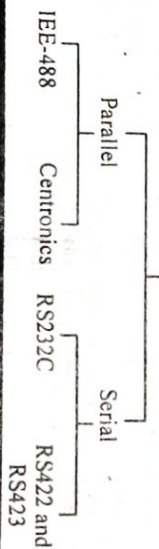
Current loop

RS 232C

RS 422A and RS423A

Communication Buses

Through interfacing these are of two types: Communication Buses



PREVIOUS YEARS QUESTIONS

PART-A

Prob.1 List the features of 8251.

[R.T.U. 2019]

Sol. Features of 8251 Microcontroller : The features of 8251 Microcontroller are namely :

1. The Intel 8251A is an universal synchronous and asynchronous communication controller.

2. It supports standard asynchronous protocol with :

- 5 to 8 bit character format.
- Odd, even or no parity generation and detection.
- Baud rate from DC to 19.2 Kbaud.
- False start bit detection.
- Automatic break detect and handling.
- Break character generation.

3. It has built in baud rate generator.

4. It supports standard synchronous protocol with :

Microprocessor and Interfaces

- 5 to 8 bit character format.
 - Internal or external character synchronization.
 - Automatic sync insertion.
 - Baud rate from DC to 64 Kbaud.
5. It allows full duplex transmission and reception.
 6. It provides double buffering of data both in the transmission section and in the receiver section.
 7. It provides error detection logic, which detects parity, overrun and framing errors.
 8. It has Modern Control Logic, which supports basic data set control signals.
 9. It provides separate clock inputs for receiver and transmitter sections, thus providing an option of fixing different baud rates for the transmitter and receiver section.
 10. It is compatible with an extended range of Intel microprocessors.
 11. It is fabricated in 28 pin DIP package and its all inputs and outputs are TTL compatible.
 12. It is available in standard as well as extended temperature range.

Prob.2 Write difference between serial communication and parallel communication.

[R.T.U. 2017]

Sol. Serial and Parallel Communication : Data can be transmitted between a sender and a receiver in two main ways: serial and parallel.

Serial communication is the method of transferring one bit at a time through a medium.

0	1	0	0	0	0	1	0
---	---	---	---	---	---	---	---

Parallel communication is the method of transferring blocks, e.g: BYTES, of data at the same time.

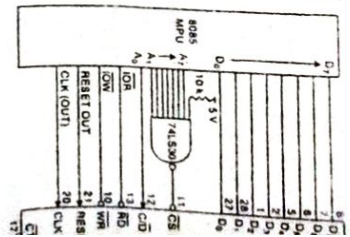
0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

As parallel communication is faster than the serial communication, therefore it is preferred over serial. For this

reason, the internal components are linked together to use the use of parallel communication is unnecessary. Therefore, takes place over serial.

Prob.3 Draw a schematic cable is connected.

Sol. Block Diagram



Sol. RS-232C: Fig. 1(b) shows the RS-232C 25-pin connector

Fig. 1(a): Minimum Configuration of RS-232C Signals and Voltage Levels

- Secondary Transmitted Data
- Transmission Signal Element Timing (DCE Source)
- Secondary Received Data
- Receiver Signal Element Timing (DCE Source)
- Unassigned
- Secondary Request to Send
- DCE → Data Terminal Ready (DTR)
- Signal Quality Detector
- Ring Indicator
- Data Signal Rate Selector (DTE/DCE Source)
- Transmit Signal Element Timing (DTE Source)
- Unassigned

Prns	Signals
14	Protective Circuit
1	Transmitted Data (T.D.) → DCE
5	Received Data (R.D.) → DTE
16	3
3	Request to Send (RTS) → DCE
17	4
4	Clear to Send (CTS) → DTE
18	5
5	1
6	Data Set Ready (DSR) → DTE
20	7
7	Signal Ground
21	8
8	Received Signal Termination
22	9
31	10
24	11
34	12
25	Unassigned
13	See Pin 14 on Sig. Director
2	See Pin 14 on Sig. Director
12	See Pin 14 on Sig. Director

Typically, data transmission with a handshake requires eight lines. Listed in Table 1. Specific functions of handshake lines differ in different peripherals and, therefore should be referred to in the manufactures manuals.

For high/speed transmission, the standards RS-422A and RS-423A are used. These standards use differential amplifiers to reject noise levels and can transmit data at higher speed with longer cable. The RS-422A allows a maximum speed of 10 kbaud for a 40-ft distance and 100 kbaud for 4000 ft. The RS-423A is limited to 100 kbaud for a 30-ft distance and 10 kbaud for 300 ft. Table 2 shows comparison of the three standards RS-232C, RS-422A, and RS-423A.

Pin No.	Signals	Pin Name	Functions
2	Transmitted Data	TxD	Output: transmits data from DTE to DCE
3	Received Data	RxD	Input: DTE receives data from DCE
4	Request to Send	RTS	General purpose output from DTE
5	Clear to Send	CTS	General-purpose input to DTE; can be used as a handshake signal
6	Data Set Ready	DSR	General-purpose input to DTE; can be used to indicate that DCE is ready
7	Signal Ground	GND	Common reference between DTE and DCE
8	Data Carrier Detect	DCD	Generally used by DTE to disable data reception
20	Data Terminal Ready	DTR	Output: generally used to indicate that DTE is ready

Specifications	RS-232C	RS-422A	RS-423A
Speed	20 Kbaud	10 Mbaud at 40 ft 100 Kbaud at 4000 ft	100 Kbaud at 30 ft 1 Kbaud at 4000 ft
Distance	50 ft	4000 ft	4000 ft
Logic 0	+3 to +25 V	B > A	+4 to +6 V
Logic 1	<-3 to -25 V	B < A	-4 to -6 V
Receiver Input			
Voltage	±15 V	±7 V	±12 V

1. The advantage of current loop method is that signals are noise free.
2. Suitable for transmission over a distance.

1. **Demerits**
The same standard is used for communication between computers and peripherals and the roles of a data terminal and a modem have become ambiguous.
2. The lines used for transmission and reception will differ, depending on the manufacturer's role definition of its equipment.

[R.T.U., 2016]

Sol. Liquid Crystal Display: A liquid crystal display (LCD) is an output device. It is a handy method of providing a low-power consumption user interface. LCDs are everywhere. You probably have seen them as displays in watches, calculators, cellular phones, and even as a color television type monitor on a video camera. LCDs are available in monochrome as well as color displays. They are also available in two general categories:

- (1) Character display

In general, a LCD display consists of two panes of polarized glass with a liquid suspended between the two panes. The liquid between the two panes of glass may be of

either the twisted nematic (TN) (STN) type. In general, the TN has but it has poorer contrast and a na the STN technology. Whichever characters and graphics are for arranging the liquid crystals into seg liquid crystal segments are prop properties may be adjusted to tr that characters or graphics are f

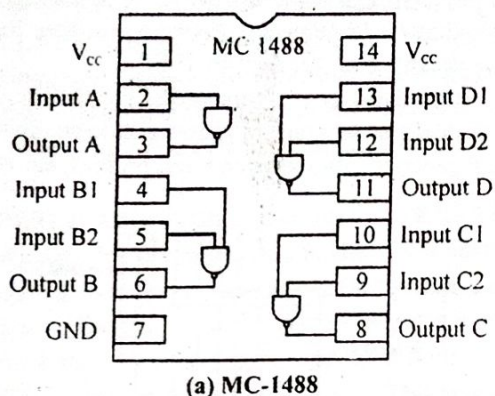
Features available on LC panels, backlit displays for easi configurations of characters per number of dot matrix formats controlled by a microprocessor affixed to the backside of microprocessor converts the i module into the proper excitati microprocessor is equipped with library and a RAM that hold displayed. So when you establ 68HC12 and the LCD display, an interface between two pro

Prob.7 What is the difference between macrocontrolle and microcontrolle

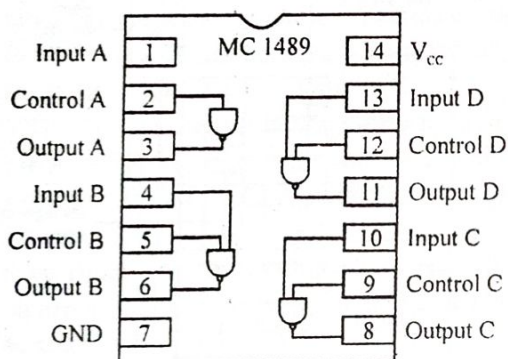
Sol. A microprocessor is an has CPU inside it. It only has Intel Pentium processors. It RAM and ROM and other peripheral devices are connected to be provided externally to function in certain way. Simultaneously, instructions are all external, each instruction hence it is relatively slower. Therefore, more operations are needed and more operations are needed in Laptops and Desktop PCs to perform numerous tasks like photo and video editing. Therefore, input and output depends on more resources like high speed hence can't be embedded or their cost is very high. The high clock rate at around 100 MHz is very complex tasks and peripheral devices.

A micro-controller contains an amount of RAM, ROM and CPU on a single chip. It Since components are in

The MC-1488 contains four drivers and each driver converts a TTL level signal to RS-232C level signal.



(a) MC-1488



(b) MC-1489

Fig. : Pin Diagram of (a) MC-1488 (b) MC-1489

MC-1489: The MC-1489 is a quad line receiver to interface DTE and DCE through RS-232C bus standard. It converts RS-232 output into TTL-compatible logic. i.e., -9V into TTL-logic 1 (approximately +3.4V) and +9V into TTL-logic 0 (approximately 0.2V). The pin diagram of MC-1489 is shown in fig. (b).

The MC-1489 contains four receivers and each receiver converts a RS-232C level signal to TTL level signal.

PART-C

Prob.12 Explain RS422A Standard in detail. [R.T.U. 2019]

Sol. RS 422A : High speed data transmission between computer system components and peripherals over very long distance, under high noise condition is very difficult with single ended driver and receivers. As an improvement over single

ended drivers and receivers EIA standard introduced RS 422A which uses low impedance differential signals. This standard achieves transmission rates from 100 Kbps to 100 Mbps. RS 422A is used for balanced transmission. It support transmission distance which is greater than 1000 m. RS 422A use a low impedance signal which is a differential signal.

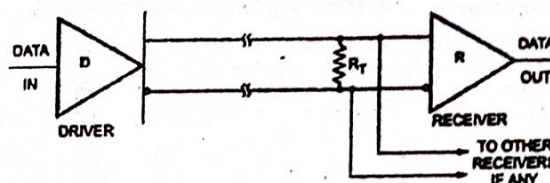


Fig. 1 : RS 422A interface circuit

A basic RS 422A circuit has a data line driver, differential transmission line and loads, where a load may consists of one or more receivers (R) and the line termination resistor (R_T), as shown in Fig. 1.

Following tables-1 and 2 shows the driver and receiver requirements for RS 422A

Table 1 : RS 422A driver requirements

Parameter	Specification
Driver output impedance	100 Ω or less
Output differential voltage	2.0 V to 6.0 V
Difference between opposite polarity differential output voltages	less than 0.4 V
The driver output offset voltage (V_{OS}), measured from the junction of the two 50 Ω terminator and driver ground	less than 3.0 V
Driver output current	less than 150 mA
Output off-state leakage current	less than 100 μ A
Output voltage transition time (t_r)	less than 20 ns
Overshoot and undershoot magnitudes	less than 10% of V_{SS} where V_{SS} is the difference between the two steady state values of the output

Table 2 : RS 422 A receiver requirements

Parameter	Specification
Differential data input threshold sensitivity	± 200 mV
Input impedance	Greater than or equal to 4 k Ω

Drivers and receivers : The SN75172 B and SN75175 are driver and receiver for EIA RS 422A standard, respectively.

Prob 13 Explain Parallel Interface-Centronics & IEEE 488 in detail.
[R.T.U. 2019]

Sol. Centronics : A standard 36pin parallel interface for connecting printers and other devices to a computer. It defines the plug, socket and signals used and transfers data asynchronously up to 200 Kbytes/sec. The plug has 18 contacts each on the top and bottom. The socket contains one opening with matching contacts.

This de facto standard was developed by Centronics Corporation, maker of the first successful dot matrix printers. Centronics Data Computer Corporation was a pioneering American manufacturer of computer printers, now remembered only for the parallel interface that bears their name.

The Centronics Interface provides a handshake protocol between a computer and a printer and supports a maximum data transfer speed of about 100 kb/s. The printer side of the interface is a 36 pin connector and the PC side is a 25 pin D type connector. The PC uses 36 pin flat cable in which every alternative wire is for the ground. Most of the signals should have twisted pair wiring in the cable. The signals are TTL level signals and the twisted pair return ground wire for each signal is connected to the signal ground level. To prevent noise effects the twisted pair wires are shielded and the shield is connected to the chassis ground in the system box.

Figure shows the signals in the centronics interface. Since a PC uses 25 pin connector, there is a shortage of pins for four twisted pair return signals. Usually, four data signals don't have twisted pair wires. The pin configurations for both printer side and PC side are listed.

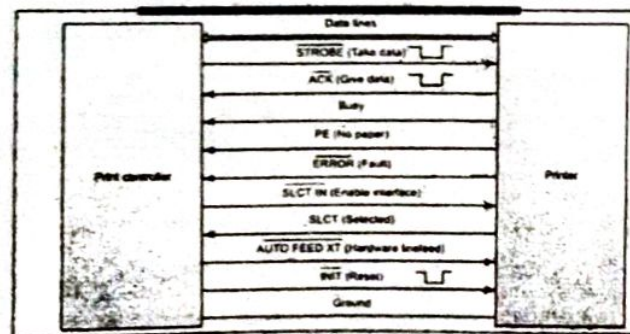


Fig. : Signals in Centronics Interface

IEEE 488 : It is known as General Purpose Interface Bus (GPIB). GPIB is designed by Hewlette Packard so it is also called as *HPIB* (Hewlette Packard Interface Bus). The GPIB is accepted as standard by IEEE.

So it has been give a number IEEE 488. The GPIB is basically designed to interface instruments to computers such as signal generator, digital voltmeters, printers, etc. as shown in Fig.

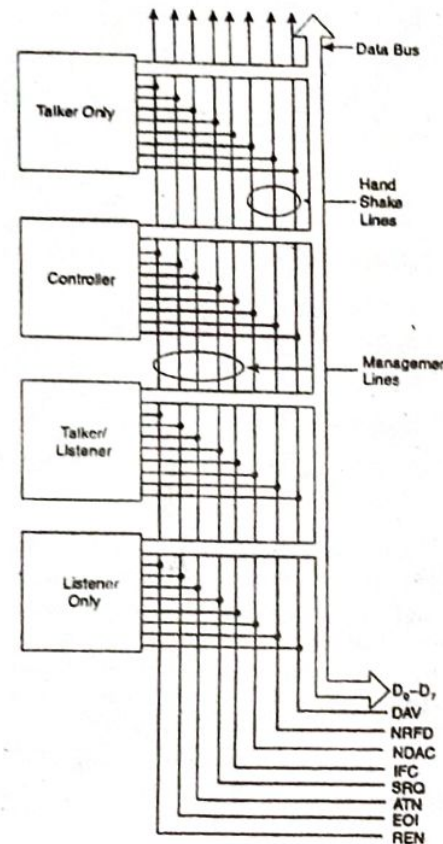


Fig. : IEEE 488 (GPIB Structure)

The interfacing device for IEEE 488 can be divided into 3 categories :

1. Listener
2. Talker
3. Controller

1. Listener : These are instruments which only accept data from other instruments. For example, printers, display, programmable power supply, programmable signal generator etc. All the examples of listener will be an output devices.

2. Talker : These are instruments which only send data to other instruments. For example, Digital voltmeter, tap

reader, frequency counter etc. All the examples of talker will be an input devices.

3. Controller : These are instruments which initialises and controls the other instruments in the system. It also gives service to all devices and decides the sequence for the devices.

The IEEE 488 has three group of buses as shown in Fig.

- (i) Bidirectional data lines
- (ii) Management lines
- (iii) Handshake signals

(i) Bidirectional Data Lines : These are eight lines used to transfer 8-bit of information.

(ii) Management Lines : The GPIB has five management lines which are used to manage data transfer on $D_0 - D_7$, the different signals are :

- (a) IFC (b) ATN (c) SRQ (d) REN (e) EOI.

(a) IFC (Interface Clear) : This is an output signal generated by controller. It is used to force the instruments in system to a known or predetermined state. It is similar to system reset signal issued by controller to reset the system.

(b) ATN (Attention) : This is an output signal generated by controller. It is used to differentiate between data or status from address commands.

(c) SRQ (Service Request) : This is an input signal to the controller generated by peripherals. It is an interrupt signal requesting service of controller.

(d) REN (Remote Enable) : This is an output signal by controller used to enable the instrument to be controlled by the system. If REN is not generated, the instrument will be controlled by its front panel controls.

(e) EOI (End of Identify) : This is an input signal to controller generally generated by talker to indicate end of data transfer.

(iii) Handshake Signals : The IEEE 488 has three handshake signals which helps in data transfer, these are :

- (a) DAV (Data Valid)
- (b) NRFD (Not Ready for Data)
- (c) NDAC (Not Data Accepted)

The signals DAV, NRFD and NDAC are open collector active low signals. Since it has open collector structure any listener can pull NRFD low to indicate it is not ready for data. The NRFD line will not go high to indicate its readiness unless all the addressed listeners has released it.

Similarly, any listener can pull NDAC low to indicate it has not accepted data. The NDAC line will not go high to indicate data accepted unless all listener accepts data.

Prob.14 Explain the organization and architecture of 8251 (USART) with a functional block diagram.

[R.T.U. 2019]

OR

How can you interface USART 8251 with microprocessor 8085? Explain.

Sol. The 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART) package in a 28 pin dual-in-line package made by Intel. It is typically used for serial communication. The 8251 in I/O mapped mode is shown in fig.

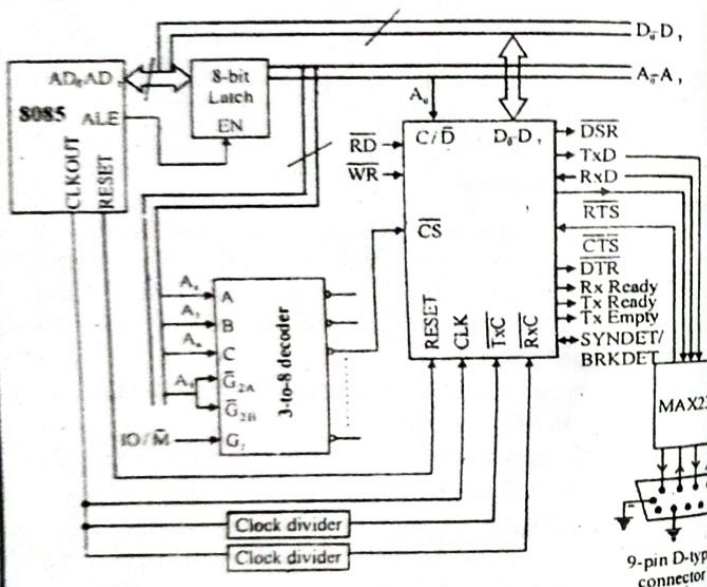


Fig.

1. Address lines A_6-A_4 are used to generate the chip select signal via a 3-to-8 decoder.
2. A_7 and IO/\bar{M} are used as enable for the decoder.
3. A_0 of 8085 is connected to C/\bar{D} of 8251 to provide internal addresses.
4. The data lines D_0-D_7 are connected to D_0-D_7 of the processor to achieve parallel data transfer between 8085 and 8251.
5. The output clock of 8085 is divided by suitable clock dividers to get a suitable clock for serial transmission/reception.
6. The TTL logic levels of the serial data lines and the control signals necessary for serial transmission and reception, are converted to RS232 logic levels using MAX232 and then terminated on a standard 9 pin D-type connector.

7. The device which requires serial communication with processor can be connected to this 9-pin D-type connector using 9-core cable.
 8. The signals TxEMPTY, TxRDY, and RxRDY can be used as interrupt signals to initiate interrupt driver data transfer between 8085 and 8251.
- I/O addresses of 8251 interfaced to 8085 is

Internal device of 8251	Binary address								Hex address
	Decoder input & enable				Input to address pin of 8251				
	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	
Data Buffer	0	0	1	0	X	X	X	0	20
Control register	0	0	1	0	X	X	X	1	21

Prob.15 Explain serial communication controller USART 8251. Explain its all modes with example.

[R.T.U. 2017]

Sol. Serial Communication: Using 8251 : 8251 is a Universal Synchronous and Asynchronous Receiver and Transmitter compatible with Intel's processors. This chip converts the parallel data into a serial stream of bits suitable for serial transmission. It is also able to receive a serial stream of bits and convert it into parallel data bytes to be read by a microprocessor.

1. Basic Modes of Data Transmission

- (a) Simplex
- (b) Duplex
- (c) Half Duplex

(a) Simplex Mode : Data is transmitted only in one direction over a single communication channel. For example, the processor may transmit data for a CRT display unit in this mode.

(b) Duplex Mode : In duplex mode, data may be transferred between two transceivers in both directions simultaneously.

(c) Half Duplex Mode : In this mode, data transmission may take place in either direction, but at a time data may be transmitted only in one direction. A computer may communicate with a terminal in this mode. It is not possible to transmit data from the computer to the terminal and terminal to computer simultaneously.

The data buffer interfaces the internal bus of the circuit with the system bus. The read / write control logic controls the operation of the peripheral depending upon the operations initiated by the CPU. The CPU decides whether the address on internal data bus is control address / data address. The modem control