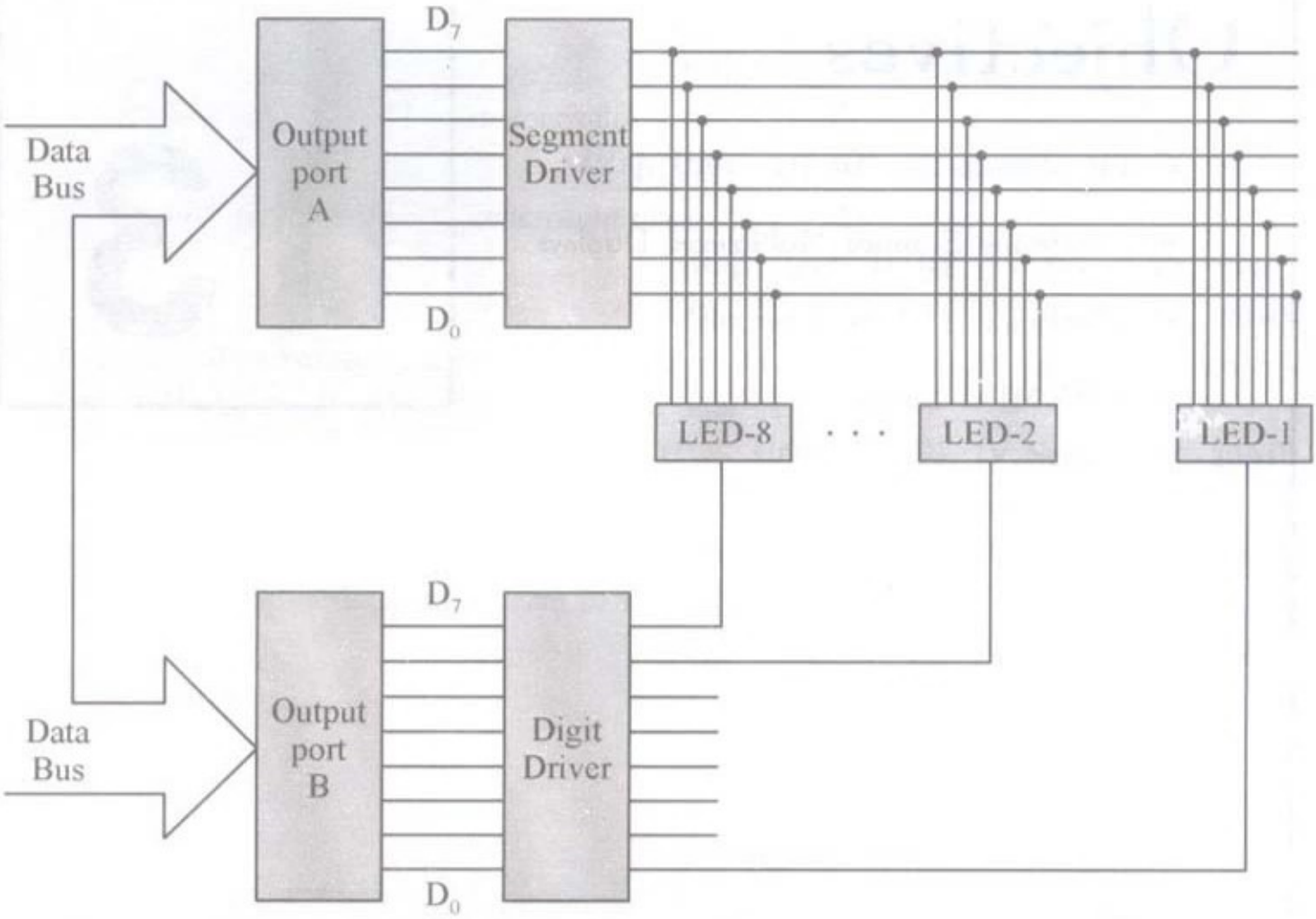
Interfacing Scanned Multiplexed Displays

The basic block diagram for multiplexed display is illustrated in Fig. One output port (Port A) is connected to segment driver and another output port (Port B) is connected to digit driver.



Block diagram for Multiplexed output display

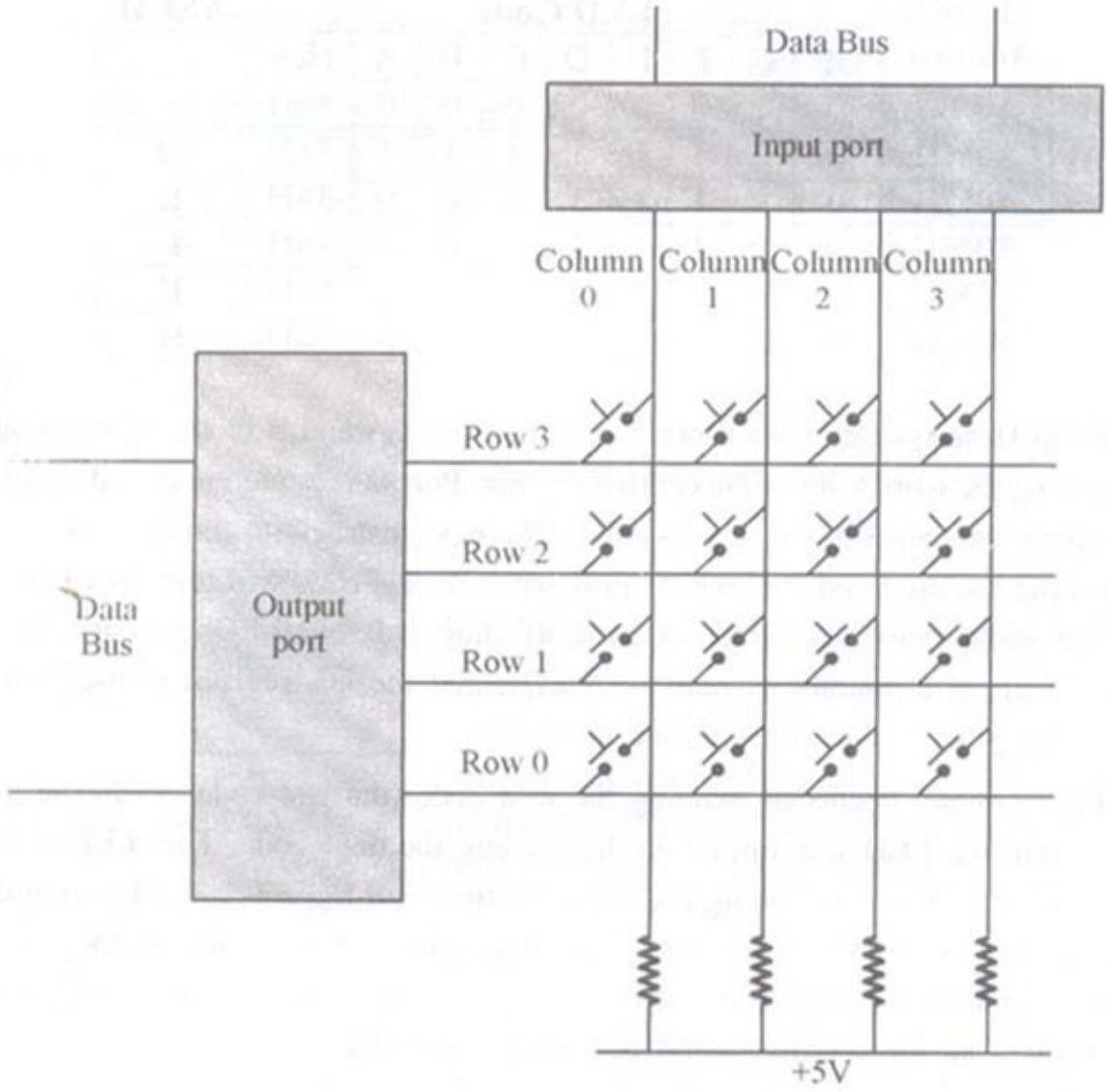
The output data lines of Port A are connected to seven segments of each LED, and the output lines of Port B are connected to the cathodes of each LED. The code of the first digit to be displayed at LED-1 is sent to Port A. A bit is sent to the line corresponding to LED-1 of Port B to turn it on. Similarly, LED-2 is turned on and LED-1 is turned off. In this sequence other LEDs are also turned on and previous LED is turned off. This cycle is repeated at very fast rate and hence all LEDs seem to be on always. This is also known as scanned display.

Additional ICs, Segment Driver (SN 75491) and Digit driver (SN 75492) are used to increase the current output of the I/O port of the programmable devices (e.g., 8255) upto the required level of 10-15 mA.

Interfacing A Matrix Keyboard

A keyboard with 16 keys, arranged in a 4 × 4 (four rows and four columns) matrix as shown in Fig. required eight lines from the microprocessor (4 through output port and 4 through input port) to make all the connections.

When a key is pressed, it shorts one row and column, otherwise, the row and the column do no have any connections. The interfacing of a matrix keyboard requires two ports: one output port and the other input port. Rows are connected to the output port, and the columns are connected to the input port.



Block diagram for Matrix Keyboard

In a matrix keyboard, the major task is to identify a key that is pressed and decode the key in terms of its binary value. This task can be accomplished through either software or hardware (Keyboard encoder MM74C923).

The MPU design can be divided into following parts:

1. Address bus
2. Data bus
3. Control signals
4. Frequency and power supply
5. Externally initiated signals
6. System Buses and their driving capacity
7. Keyboard and Displays
8. Memory

Address Bus

As we have already discussed the 8085 has a multiplexed address/data bus; it must be demultiplexed outside of the microprocessor to separate address and data. In addition, bus drivers are required to provide sufficient driving capacity.

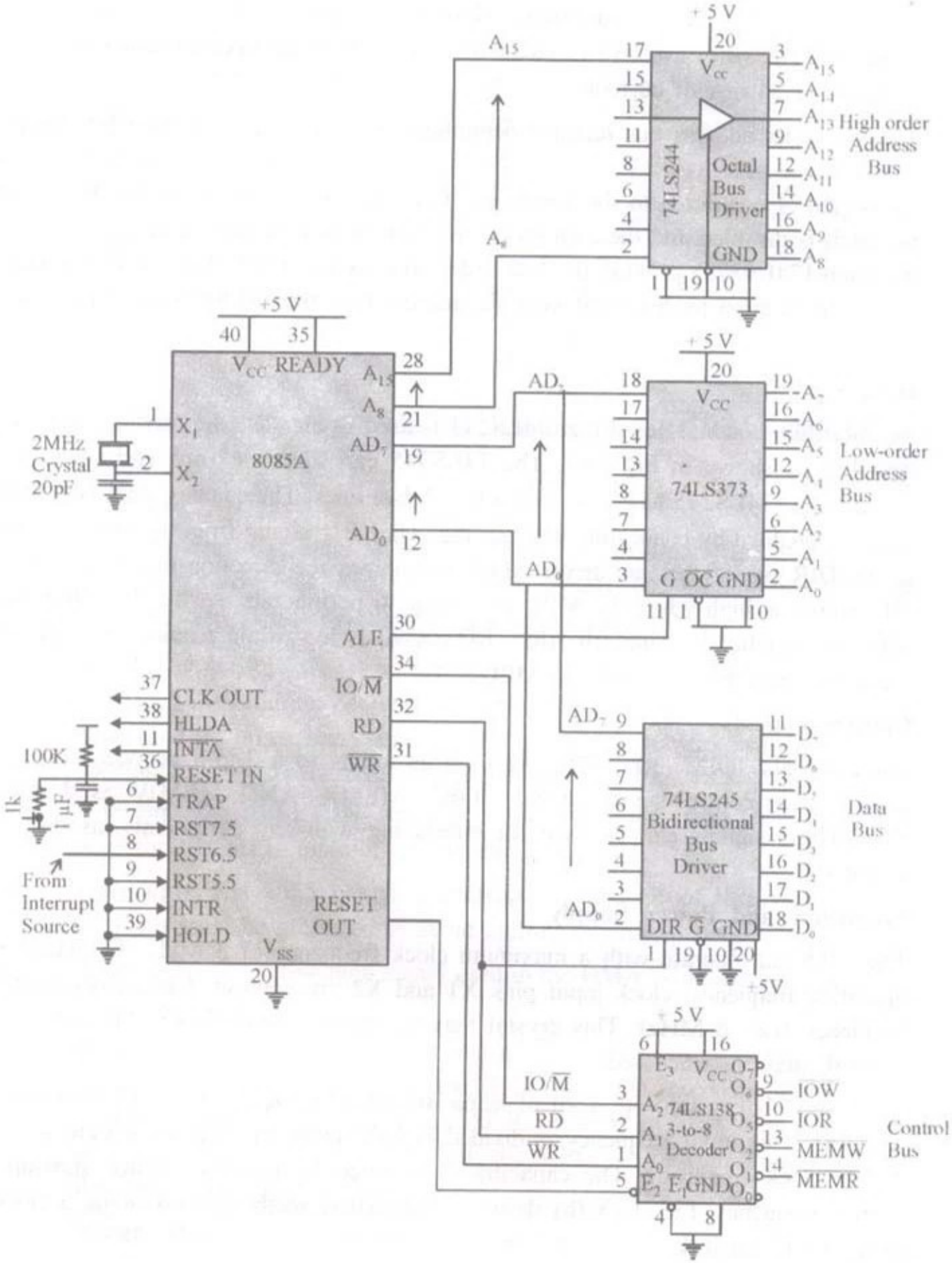
High-order address bus does not require demultiplexing. Hence, an octal bus driver 74LS244, as shown in Fig. is used with the high-order address bus only to increase its driving capacity.

Typically, the 8085 buses can source 400 μA and sink 2 mA of current to drive one Transistor-Transistor Logic (TTL) load. The 74LS244 driver is capable of sourcing 15 mA and sinking 24 mA of current.

The low-order address bus requires demultiplexing. ALE signal is used for this purpose this signal is connected to the enable pin G of the latch (74LS373). As ALE goes low, the latch is disabled and the address on but AD_7-AD_0 is latched, and the output lines of the latch (74LS373) provide the low-order address bus (AD_7-AD_0) until the next ALE signal. In addition to demultiplexing the address bus, the 74LS373 also serves as a bus driver.

Data Bus

An 8-bit bidirectional bus driver 74LS245 is used to increase the driving capacity of the data bus as shown in Fig. The 74LS245 can source 15 mA and sink 24 mA of current. The 74LS245 has eight bidirectional data lines. The enable signal ($\overline{G}$) of the bus driver is enabled by connecting it to ground. The $\overline{RD}$ signal from the MPU is connected to the DIR pin of the bus driver which determines the direction of the data flow. The $\overline{RD}$ signal is high when the MPU is writing to peripherals, so the data flow from the MPU to peripherals. Similarly, the $\overline{RD}$ signal is low while reading from peripherals, hence the data flow toward the MPU.



Schematic of the 8085 MPU with Demultiplex Address Bus and Control Signals

Control Signals

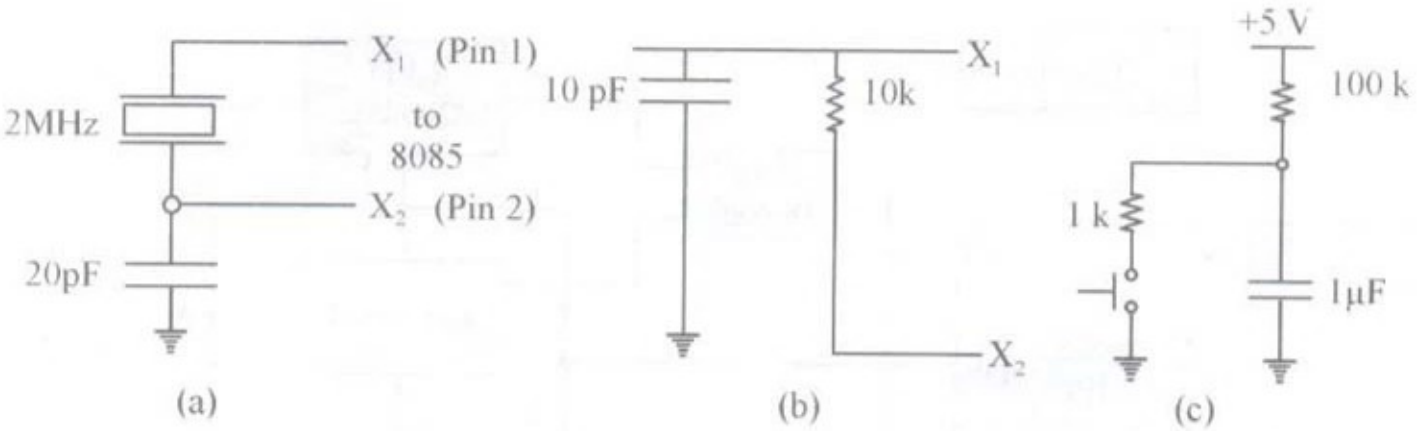
Three signals IO/$\overline{M}$ $\overline{RD}$ $\overline{WR}$ are used as inputs to a 3-to-8 decoder (74LS138) to generate four control signals $\overline{IOR}$, $\overline{IOW}$, $\overline{MEMR}$, AND $\overline{MEMW}$ as shown in Fig These signals can be used for interfacing with any peripherals.

Frequency and Power Supply

The 8085 can operate with a maximum clock frequency of 3 MHz. To obtain 3 MHz operating frequency, clock input pins X1 and X2 are connected to a crystal of double frequency (i.e., 6 MHz). This crystal may be an LC tuned circuit. Alternatively a RC network may also be used.

A 2 MHz crystal with a 20 pF capacitor to drive the clock inputs is shown in Fig.(a). This input frequency is divided in half internally, and the system will run on 1MHz clock frequency. The capacitor is required to assure oscillator start-up at the correct frequency. Fig.(b) shows an alternative method of providing a clock input using a RC network.

The 8085 and other components used in this system require one power supply with + 5V. The MPU and memory components of the system require less than 400 mA.



(a) Clock Circuit with Crystal, (b) RC Clock Circuit and (c) Reset Circuit

Externally Triggered Signals

As discussed, the 8085 has four externally initiated input signals: Reset, Interrupt, Ready, and Hold. Out of these signals, RESET and one interrupt signal (RST 6.5) are used in this system, and the others are disabled.

Reset The $\overline{RESET\ IN}$ is an active low signal used to reset the system. When this pin goes low, the program counter is set to 0, the Interrupt Enable and HLDA flip-flops are reset, and all buses are placed in tri-state. A RC network with a sufficiently long time constant is used as the reset circuit as shown in Fig. (c) When the Reset key is pushed, the $\overline{RESET\ IN}$ signal goes low and slowly rises to + 5V, providing sufficient time for the MPU to reset the system.

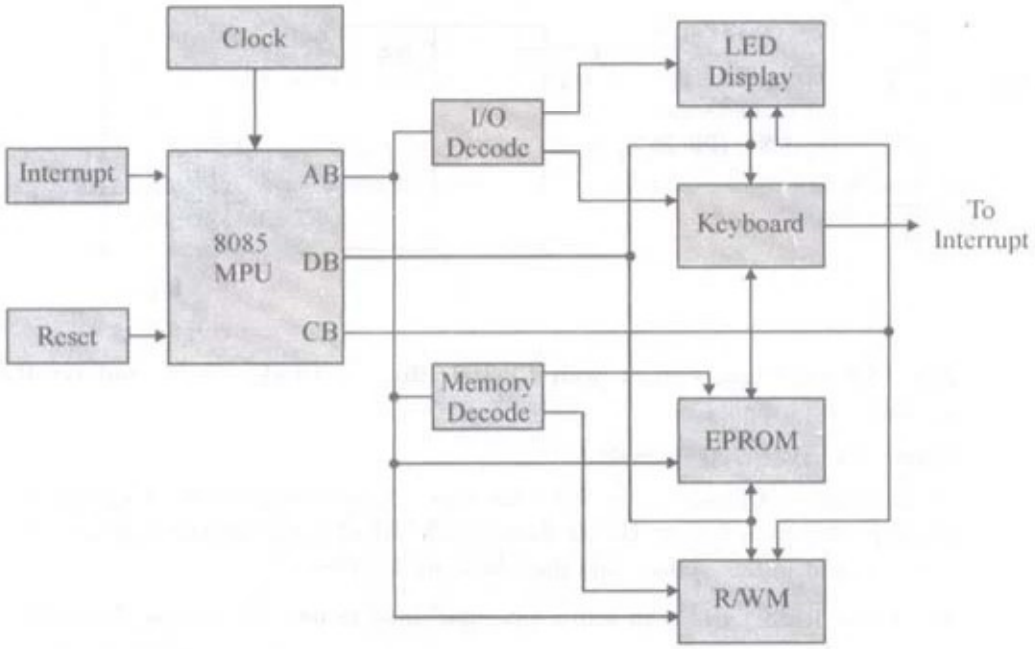
Interrupts The 8085 has five interrupt signals, all of then active high. The unused interrupt must be grounded to avoid malfunctioning.

Hold This is an active high signal used in the DMA. This signal should also be grounded if not required.

Ready When this signal is high, it indicates that the memory or peripheral is ready to send or receive data. When READY goes low, the MPU enters the Wait state until READY goes high; then the MPU completes the Read or Write cycle. This signal is used primarily to synchronize slow peripherals with the MPU. To prevent the MPU from entering the Wait state, this pin is tied high.

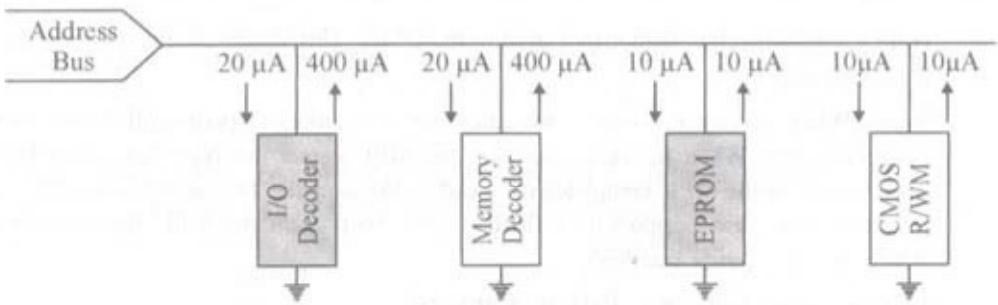
System Buses and their Driving Capacity

To design an MPU the driving capacity of the buses must be decided on the basis of the load current. To make a rough estimation of load current let us consider the Fig.



Block Diagram of a Single-Board Microcomputer

The figure shows that the address bus has to drive two memory chips (CMOS 6264 and EPROM 2764) and two decode circuit (I/O and memory decoders). Also see the next fig to calculate the bus loading as follows:



Loading on the Address Bus

	Low-level input current I_{IL}	High-level output current I_{IH}
Two decoder = 400 μA × 2	= 800 μA	20 μA × 2 = 40 μA
R/W memory	= 10 μA	= 10 μA
2764 EPROM	= 10 μA	= 10 μA
	820 μA	= 60 μA

But, the driving current capacity of the address bus is of 400 μA and sinking capacity of 2mA, which is enough for the calculated load currents. Hence, there is no need of bus drivers. However, to make this single-board microcomputer suitable for general-purpose interfacing we will use the 74LS244 as a bus driver to increase the driving capacity, as a precaution. The 74LS244 is an octal buffer/driver, capable of sourcing 15 mA and sinking 24 mA of current. Thus, the 8085 address bus can drive additional devices with sufficient driving capacity.

The multiplexed bus also has the driving capacity similar to that of the address bus. Since, the data bus is bidirectional; the loading on the bus varies considerable. Therefore, similar to address bus, we will precautionary use a bidirectional buffer (74LS245) as a data driver.

Keyboard and Displays

Interfacing with scanned display and keyboard has been discussed previously may be incorporated in this MPU design.

Another approach introduces a keyboard encoder (MM74C923), which replaces the keyboard software. For scanned display the software is used as before.

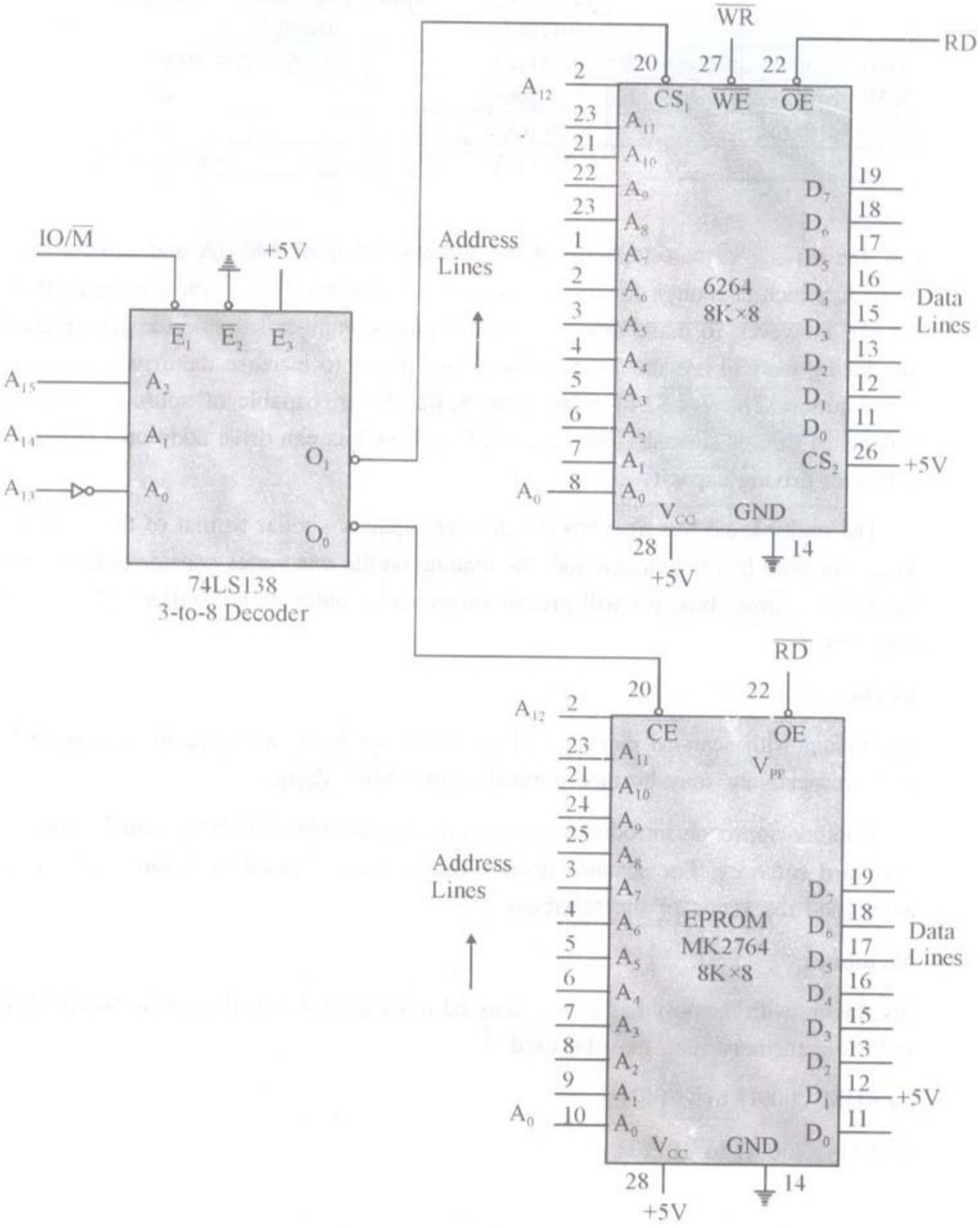
Memory:

Interfacing with memory has been discussed earlier . For this single board MPU the following memory map may be used.

EPROM 0000H to 1FFFH

R/WM 2000H to 3FFFH

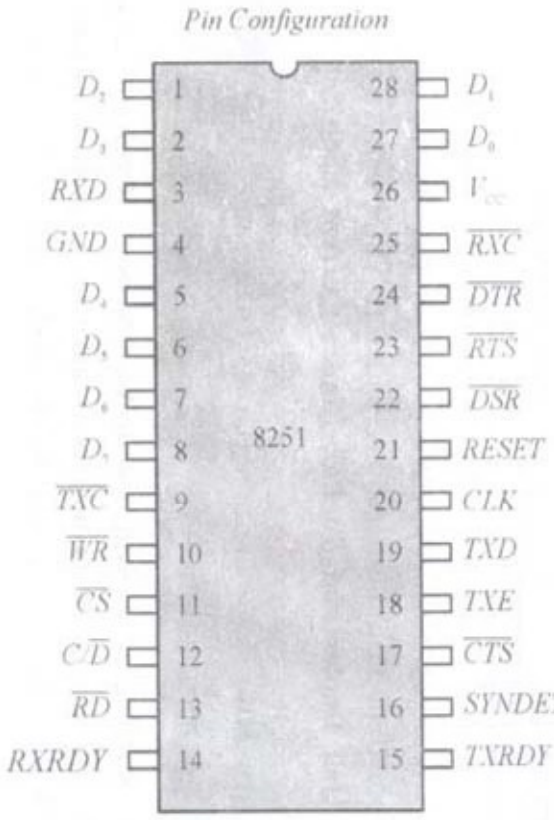
It is shown in the figure below:



Memory Map

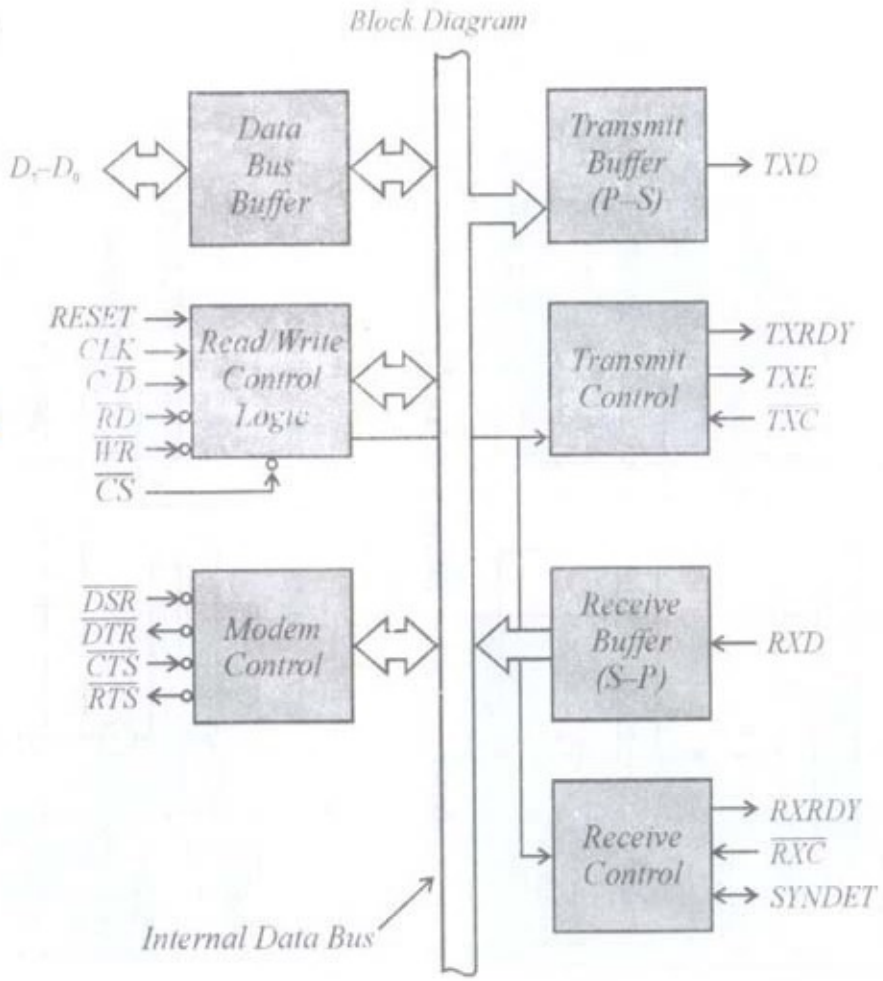
Programmable Communication Interface (USART) - 8251

The USART stands for Universal Synchronous/Asynchronous Receiver/Transmitter. It is a programmable chip designed for synchronous and asynchronous serial data communication. It is of 28 pins. The pin diagram and the block diagram of 8251 are shown in fig. below.



<-- 8251 PIN DIAGRAM

8251 BLOCK DIAGRAM -->



Functions of Various Pins of 8251

The functions of various pins and their signals are explained below:

$\overline{TXC}$ (Transmitter Clock Input) – It controls the rate at which the character is to be transmitted.

TXD (Transmitter Data Output) – It is an output pin, used to send serial stream of transmitted data bits along with other information like start bit, stop bit, and parity bit.

$\overline{RXC}$ (Receiver Clock Input) – It controls the rate at which the character is to be received.

RXD (Receiver Data Input) – This input pin receives the stream of data to received by 8251.

RXRDY (Receiver Data Ready) – It is an output signal, which indicates that the 8251 contain a data to be read by the microprocessor. This signal may used to interrupt the microprocessor.

TXRDY (Transmitter Ready) – It is also an output signal, which inform the microprocessor that transmitter is ready to accept a new data for transmission from microprocessor.

$\overline{DSR}$ (Data Set Ready) – It is an active low input signal, used to check, if the data set is ready, when communicating with a modem. Its status can be checked using status read operation.

$\overline{DTR}$ (Data Terminal Ready) - It is an active low output signal, used to indicate that the device is ready to accept data, when communicating with modem.

$\overline{RTS}$ (Request to Send Data) - It is an active low output signal, used to indicate the modem that the receiver is ready to receive a data byte from the modem.

$\overline{CTS}$ (Clear to Send) - It is an active low input signal, used to enable the 8251 to transmit the data. It can be enabled or disabled using command word.

TXE (Transmitter Empty) – It goes high, when while transmitting, 8251 has no data to transmit.

SYNDET/BD (Synchronous Detect/Bread Detect) –In synchronous mode it is used for detecting Synchronous characters. In this mode, it may be used as input or output. When used as an output, the SYNDET pin goes high, to indicate that 8251 has located a Synchronous character. When used as an input, a signal at this pin causes 8251 to start assembling a data character.

In asynchronous mode, the pin acts as a break detect output. It goes high whenever RXD pin goes low, through two consecutive stop bit sequence.

A_0 – Connected to A_0 line of the address bus. It should be high while writing the command or reading status and low when transferring data.

$\overline{CS}$ (Chip Select) – It is an active low input signal. A low signal at this pin selects the chip.

C / $\overline{D}$ (Control/Data) – The high signal at this pin, used to address control register or the status register. And low signal at this pin selects data buffer.

$\overline{RD}$ - Read signal. It goes low for read operations, like, reading status or input data from data buffer.

$\overline{WR}$ - Write signal. It goes low for write operations, like, writing control word or send output to data buffer.

RESET – A high signal on this pin resets the 8251.

CLK - Clock input, connected to system bus.

Transmitter and Receiver

The block diagram the important sections are: Transmitter, Receiver and Modem Control.

The **transmitter section** converts the parallel word received from the microprocessor, converts it into serial data and transmits them over the TXD line to a peripheral. It has two registers :

(i) **Buffer Register** – to hold 8 bits and

(ii) **Output Register** – to convert 8 bits into a stream of serial bits. (See fig.).

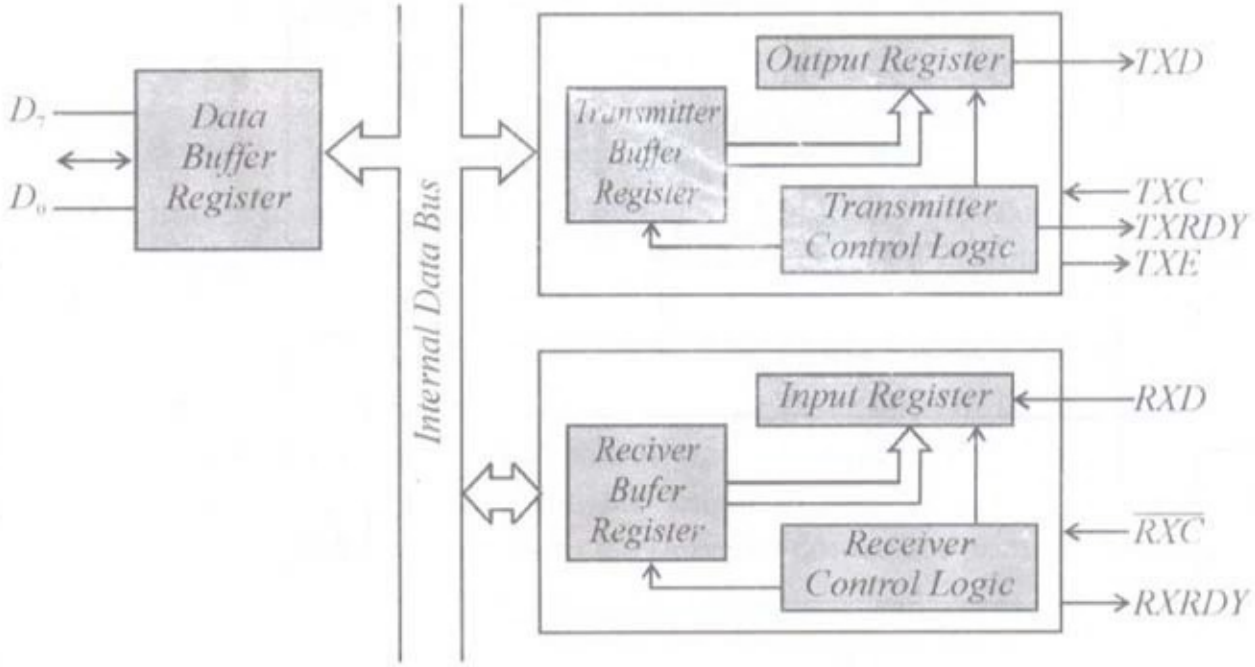
TXD, $\overline{TXC}$, TXRDY and TXE pins are associated with this section.

The **receiver section** receives serial bits from the peripheral, converts them into a parallel word and transfers the word to the microprocessor. The Modem Control section is used to establish the data communication through modems over telephone lines. It has two registers :

(i) **Input Register** – to accept the serial data and

(ii) **Buffer register** – to hold 8 bit until read by the microprocessor. (See fig.).

RXD, $\overline{RXC}$ and RXRDY pins are associated with this section.



Expanded Block Diagram of Transmitter and Receiver Section of 8251

Control, Status and Data Buffer Registers

Control Register : It is a16-bit output register, used to store the control word. The control word consists of two independent bytes: (i) Mode Instruction Word (ii) Command Instructin Word. This register can be accessed as an output port, with $C/\overline{D}$ signal high.

Status Register : It is an input register, used to check the ready status of a peripheral. This register can be accessed with $C/\overline{D}$ signal high. The port address of status register is same as of the control register.

Data Buffer Register : It is a bidirectional register, which is used as input/output port to accept/send data, with $C/\overline{D}$ signal low.

Initialisation of 8251

Before starting the data transfer, microprocessor must inform the 8251 about mode, baud, stop bit, parity etc., through a set of Control Words. The 16-bit control word is divided into two parts: Mode Word and Command Word. The microprocessor must also check the Status Word, by reading the Status Register, for readiness of the peripheral.

Serial I/O Standards

The serial Input/Output technique is used to communicate with various devices such as printers and modems. These devices are manufactured by various companies. Hence, they should follow some common standards. A standard is normally defined by a professional organisation (such as IEEE – Institute of Electrical and Electronics Engineers). A standard may include items, such as, number of pins, pin position for various signals, voltage levels, speed of data transfer, length of cables etc.

In serial I/O, data can be transmitted as either current or voltage.

When data are transmitted as current, current loops are used. e.g., in teletype, current loops 20/60 mA are used. The advantage of current loop method is that signals are relatively noise-free and suitable for transmission over a distance.

When data are transmitted as voltage, RS-232C cable is normally used. The disadvantages of voltage level methods are that (i) the voltage levels are not compatible with TTL logic levels and (ii) the rate of data transmission is limited to a maximum of 20 k baud and a distance of 50 ft. For high speed data transmission, two new standards: RS-422A and RS-423A are developed, but they are also not used much.

These serial I/O standards are compared in the following table:

Characteristics	RS-232C	RS-422A	RS-423A
Input Voltage	±15V	±7V	±12V
Speed	20 k baud	10 M baud at 40 ft 100 k baud at 4000 ft.	100 k baud at 30 ft. 1 k baud at 3000 ft.
Distance	50 ft.	4000 ft.	4000 ft.
Logic 1 (–ve true logic)	B < A*	–4V to –6 V	–3V to –25V
Logic 0	+3V to +25V	B > A*	+4V to +6 V

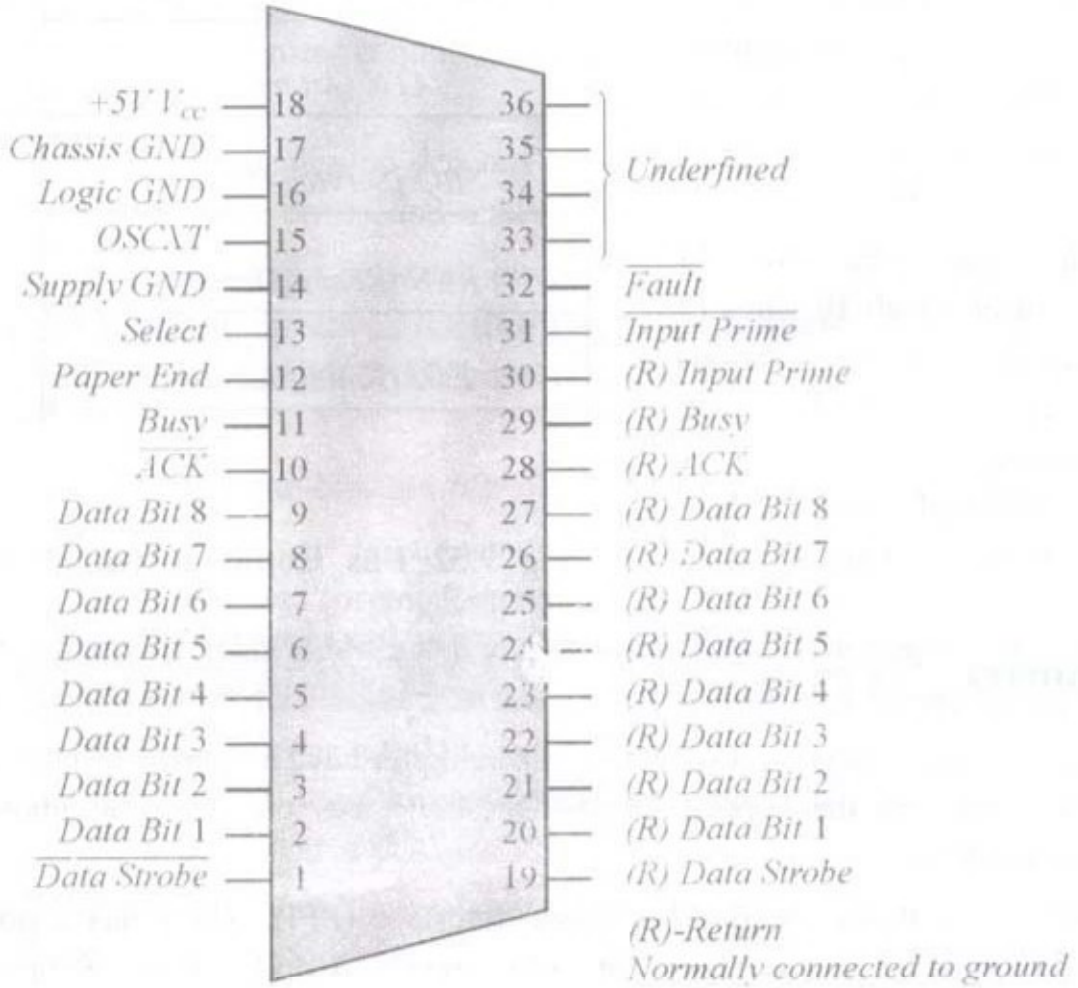
* A and B are different input to the operational amplifier.

Parallel Interface

Parallel I/O ports may be connected to a multiconductor cable with eight lines for incoming data, eight lines for outgoing data and a few handshake signal lines. For simple TTL drivers, the length of such cable may be 1 to 2 m. When high power drivers are used, the cable lengths may be about 50 m.

Centronics Parallel Interface

The Centronics parallel interface is a standard I/O interface for transmitting parallel data, used for connecting printers with computers. This interface includes a cable with 36-pin male/female connector. The cable plugs into a 25-pin parallel port of the computer system. The 8-bit data lines of the interface are unidirectional, hence data flows only in one direction, i.e., from computer to printer. The other lines of the interface are used to read status information and send control signals. Usually, the Centronics data port is driven by TTL, which limits the cable length to 10 ft or less. It was designed by Centronics Corporation, originally to interface the dot-matrix printers. The signals in the Centronics interface are show in fig. below.



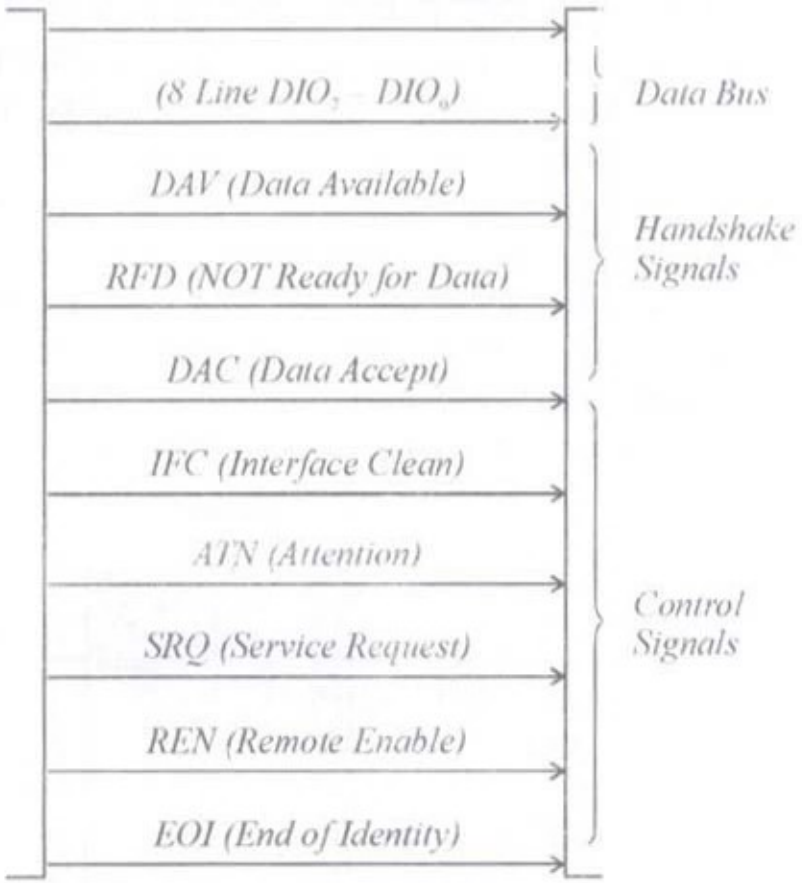
The 36-Pin Connector for Centronics Parallel Interface

IEEE 488 Parallel Interface

It is a standard interface for transmitting parallel data between electronic test instruments and microprocessor-based system. It is also known as GPIB (General Purpose Interface Bus). The concept of such bus was first developed by Hewlett-Packard as the name HP-IB (Hewlett-Packard Interface Bus). Then it was standardised by IEEE as the IEEE-488.

It can transfer data at the rate of 1 Mbyte/Sec or more at a distance of 50 m. The bus speed is limited by the speed of the slowest device connected to the bus. It is a bidirectional bus. GPIB permits maximum upto 15 devices to be connected together. Each device on the bus can be configured to have any one of the three functions: talker, listener or controller. Each device on the bus must have a unique address, ranges from 0-30.

The bus cable has 24 wires, out of which 16 wires (8 for data lines and 8 for various handshake and control signals) are shown in the fig. Eight additional wires are provided for shielding and grounds.



Bus Configuration of IEEE-488