DATA TRANSMISSION BETWEEN PCS

PROJECT REPORT

SUBMITTED BY

R. RAJARAJAN G. VENKATESAN B. R. VINOD

UNDER THE GUIDANCE OF

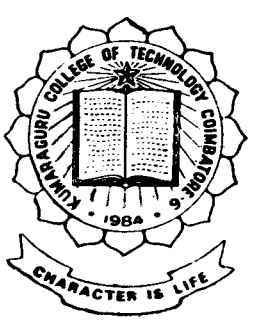
Miss. H. MANGALAM, M. E.

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

IN ELECTRONICS AND COMMUNICATION ENGINEERNiG OF THE

BHARATHIAR UNIVERSITY, COIMBATORE



DEPARTMENT OF

ELECTRONICS AND COMMUNICATION ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641 006

1992-93

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE-641 006

CERTIFICATE

This is to Certify that the Report Entitled

DATA TRANSMISSION BETWEEN PC'S

HAS BEEN SUBMITTED BY

Mr. .....

in partial fulfilment for the award of the Degree of Bachelor of Engineering in Electronics and Communication Engineering Branch of the Bharathiar University, Coimbatore-641 046 during the academic year 1992-93

GUIDE 3/11/93

HEAD OF THE DEPT

Certified that the candidate was Examined by us in the project work viva-voce Examination held on _____ and the University Register Number was _____

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

We thank our Principal **MAJOR T.S. RAMAMURTHI** to carry out our project.

We express our sincere thanks to our professor and the Head of the department **Prof. K. PALANISWAMI ME, MIEEE, MIE, MISTE, MCSI, FIETE**, for his motivation and suggesstion in carrying out our project work.

We are indebted to Miss. **H. MANGALAM M.E.** for her guidance and encouraging words for the success of our project.

We are very grateful to Mr.**M.S. SHANMUGA SHANKAR** B.E for his keen interest and untiring efforts towards the completion of the project.

We also thank the lab technicians for their kind co-operation.

INDEX

CONTENTS

6. **SYSTEM SOFTWARE**

6.1 Flow chart development

6.2 Transmitter & Receiver Software

7. APPLICATION AND SUGGESTIONS FOR FURTHER DEVELOPMENT.

8. CONCLUSION

9. APPENDIX

10. BIBLIOGRAPHY.

SYNOPSIS

A direct coupled modem is the most reliable method of sending digital data through telephone lines for a computer user. Fortunately, the modem is available in the form of a single chip, namely AM 7910.

The computer is used as Data terminal equipment (DTE) and the modem is used as Data Communication equipment (DCE). RS232 -C connector is used for data transfer between DTE & DCE. The interface between the parallel microcomputer and the serial channel - the telephone line - is accomplished by means of the UART (Universal Asynohronous/Receiver/Transmitter).

Asynchronous mode of transmission and reception has been used. The modulation technique employed is frequency shift keying, provision for which is given in the modem chip itself. The V23 standard, which is specified by the C.C.I.T.T. has been selected, as half-duplex mode of operation is desired. The auto answer facility has also been provided.

INTRODUCTION

The multifarious applicatio ns of computers in nearly every fact of present day life have rendered them almost indispensable. This widespread use of computers and intelligent terminals has further accelerated the need for efficient communication. So it would be very helpful if to have an efficient equipment for communicating from one point to another.

Much importance is therefore given to the design and performance of data communication system. Break throughs in materials and components have greatly alleviated several major difficulties in such systems.

This project work is aimed to develop a modem and efficient communication system which may be later widened to serve a larger network. The use of several sophisticated ICs highlights the design.

CHAPTER ONE

COMMUNICATION PRINCIPLES

1.1 DATA COMMUNICATION

Since almost all the computers used nowadays are digital and use binary representation of numbers, the data communication must be basically digital in nature. However in practice, digital pulses cannot be directly transmitted over long distances via a channel. Also such direct transmissions result in appreciable amplitude and phase distortions.

A practical solution to the problem is to modulate the digital pulses before trasmission and demodulate then after reception. This process is accomplished by the use of a MODEM (acronym for Modulator/Demodultor). The modulation technique may be any one of these described in section 1.3.

Communication systems that employ already existing channel networks are always preferable. Therefore, data communications via the telephone is becoming more than a convenient option for many small computer users. With the recent availability of large scale integration modems, designers can now incorporate data communication components into the smallest and the most inexpensive hardware.

1.2 OPERATING MODES

HALF-DUPLEX AND FULL DUPLEX MODES

The modems can operate, in two modes, the half-duplex and full-duplex over the telephone lines. Half-duplex data transmission and reception allows data communication in only one direction. The lines can be "turned around" to allow communication in the opposite direction. In some cases, "a backward channel" is included to allow a low rate channel in the opposite direction of the main channel.

On the other hand full-duplex data transmission allows simultaneous data transmission by the two modems, in different (opposite) directions. Only four wire facilities are capable of full-duplex working.

1.3 MODULATION TECHNIQUES

Modems are generally characterised by the speed and modulation technique for which various standards exist (Table 2.2). Low speed modems upto 1200 bps are implemented using frequency shift keying (FSK) modulation. Medium speed modems (1200 to 4800 bps) are implemented using phase shift keying (PSK) or quadrature amplitude modulation. Higher speed modems (9600 bps) are implemented using quadrature amplitude modulation.

These modulation techniques differ in the method of encoding data into an analog carrier, the number of bits encoded per modulation interval, efficiency of transmission, frequency spectrum usage and complexity of the circuitry required for implementation.

FREQUENCY SHIFT KEYING

This modulation technique encodes one bit for the serial data stream per baud. A logic 'one' in the bit stream placed a mark frequency (fM) on the phone line while a logic 'zero' places a space frequency (fs). As the bit stream switches between one and zero , the analog signal on the line modulates between fM and fs (Fig.1.1b.)

Since it encodes only one bit per baud FSK uses approximately 1 Hz of bandwidth for each per second of data rate. For example at a bit rate of 1200 bps, a substantial portion of the phone line bandwidth is used, allowing only a single channel to be transmitted. At 300 bps, two independent channels can be accommadated within the line's bandwidth using frequencydivision multiplexing. The C.C.I.T.T. V.23 standard for the bandwidth utilisation of the phone line, is shown is Fig 1.2. As this modulation technique has been used, this type of modulation was explained in detail.

BINARY DATA

(a)

AMPLITUDE SHIFT KEYING

(b)

FREQUENCY SHIFT KEYING

(c)

PHASE SHIFT KEYING.

FIG. 1.1. MODULATION TECHNIQUES.

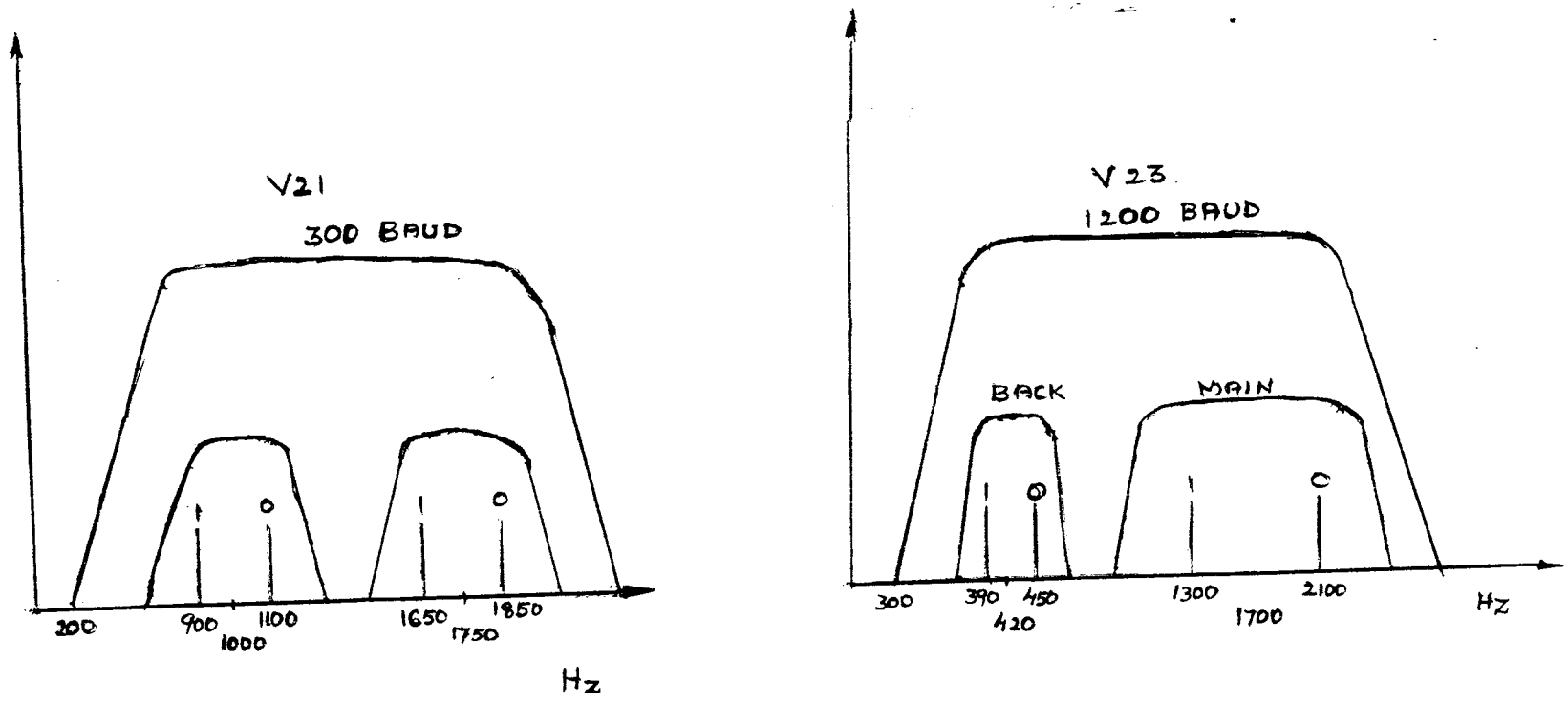


Fig.1.2. V21 & V23 STANDARDS.

CHAPTER TWO

SYSTEM DESCRIPTION

2.1 SYSTEM BLOCK DIAGRAM

The prpocess of handling data for communication is shown in the block diagram (Fig. 2.1). Here the medium of transmision is the telephone lines as the largest network in the world by far is the telephone system. It also turns out to be the most economical link for transmitting digital data.

But the use of telephone line possess two main problems which arise due to the following reasons: (a) The bandwidth of individual telephone lines are barely 3300 Hz and they do not pass DC and so they introduce considerable phase distortion and noise (b) The parallel data from the microcomputer cannot be directly sent through a two-wire transmission line.

As already seen in section 1.1, the first problem can be solved by modulating the digital data before transmission and demodulating after reception. This is performed by the modem chip.

The second problem is solved by using an UART (8250 chip). (UNVERSAL/ASYNCHRONOUS TRANSITTER/RECEIVER) which is capable of converting the parallel data coming from the microcomputer into serial form.

The complexities involved in the above mentioned conversion have been made economically attractive only recently. This is achieved through the integration of those functions onto single IC Chips.

The conversion from parallel to serial form and vice-versa is done by the UART 8250 Chip while the modulation and demodulation are caried out by the MODEM AM 7910.

It is soon from the figure 2.1 that at the transmitting end, the parallel data from the microcomputer is converted into serial data by the UART. Then the modem modulates this serial data to a compatible analog signal which is transmitted through the Telephone lines. At the receiving end, the reverse process takes place.

2.2 THE UART- INTERFACE

The INTEL 8250 is capable of operating with a wide variety of serial communication formats. It has been specially designed for data communications with Intel's microprocesser families, such as MCS-6800,8080,8085. Since many peripheral devices are available with serial interface, the 8250 can be used to interface a microcomputer to a broad spectrum of peripherals.

This device can be programmed by the CPU, to operate using virtually any serial data transimission technique presently in use. The UART accepts data characters from the CPU in parallel

form and then converts them into serial data stream for transmission. Simultaneously it can receive several data stream and then converts them into parallel data characters for the CPU.

The CPU can read the complete status of the UART at any time. The UART will signal the CPU whenever it accept a new character for transmission or whenever it has received a character for the CPU.

COMMUNICATION FORMATS

Serial communications occur in one of two basic formats - ASYNCHRONOUS and SYNCHRONOUS. These formats are similar in that they both require framing information to be added to the data to enable proper detection of the character at the receiving end. The major difference between the two formats is that the asynchronous format requires framing information to be added to each charaacter, while the synchronous format adds framing information to blocks of data or messages.

An example of the asynchronous and synchronous format are shown in Fig.2.2. The asynohronous format starts with the basic data bits to be transmitted and adds a 'START' bit to the front of then and one or more 'STOP' bits behind them an they are transmitted. The START bit is a logical zero and the STOP BIT is a logical one. The START bit tells the receiver to start assembling a character and allows the receiver to synchronise itself with the transmitter. Since this synchronization only has

to last for the duration of the character, this method works quite assuming a properly designed receiver. The STOP bits are added to the end of the character to ensure that the START bit of the next character will cause a transition on the line.

2.3 **FSK MODEM - AM7910**

The AM7910 is a single-chip asynchronous Frequency Shift Keying Voice-band modem. Digital signal processing techniques are employed in the AM7910 to perform all major functions such as modulation, demodulaton and filtering.

The Key Features of AM7910 are:

* Complete FSK Modem in a 28-pin package
* Compatible with Bell 103/113/108, Bell 202,CCITT V21, CCITT V23 specifications
* No external filtering required
* All digital signal processing, digital filters and ADC/DAC included on chip
* AUTO answer capability
* Local copy/test modes
* Pin programmable mode selection.

The internal block diagram of the chip is shown in Fig.2.4. It contains a transmitter and receiver both of which are controlled by interface-control and timing - control sections. The Clock can be generated by attaching a crystal to drive the

internal crystal oscillator or by applying an external clock signal. A data access arrangement must provide the phone line interface externally. The AM7910 is fabricated using N-channel MOS technology in a 28 pin package. All the digital input and output signals except the external clock are TTL compatible. Power supply requirements are ± 5 Volts.

THEORY OF OPERATION

The AM7910 consists of three main sections, as shown in the block diagram. Transmitter, Receiver and Interface Control.

TRANSMITTER (MODULATOR)

The transmitter, shown in Figure 2.5a, receives binary digital data from a source such as a UART and converts the data to an analog signal using frequency shift keying modulation. The analog signal is applied to the phone line. The serial data, which to be transmitted is fed to the TD input causes a sine wave to appear at the analog TRANSMITTED CARRIER (TC) output. The frequencies are different for the two logic levels. In the AM7910, this switching between frequencies is phase continuous. The frequencies themselves are digitally synthosized sine functions.

The process of switching between the two

stage of the demodulator is a simple on-chip analog low-pass filter. The output of this is converted into digital form and filtered by digital band pass filters to improve the signal to noise ratio and reject other independent channel frequencies associated with the phone line in the case of full duplex configuration. The bandpass filtered output is digitally demodulated to recover the binary data. A carrier detect signal is also extracted from the received line carrier to indicate valid data.

INTERFACE CONTROL

This section monitors the handshaking between the modem and the local terminal. It consists primarily of delay generation counters, two state machines for controlling transmission and reception and mode control decode logic for selecting proper transmit frequencies and transmit and receive filters regarding to the selected modem type.

Inputs and outputs from this section are

(i) REQUEST TO SEND

(ii) CLEAR TO SEND

(iii) CARRIER DETECT

(iv) RING

(v) MCO-MC4

(vi) DATA TERMINAL READY

Details of these signals are discussed in Appendix A. Handshake signals must initially have levels listed in Table 2.1. The pin details of the Modem Chip also have been given in Appendix.A.

TABLE 2.1 : INITIAL CONDITIONS.

DATA TERMINAL READY ($\overline{DTR}$)	OFF
REQUEST TO SEND ($\overline{RTS}$)	OFF
CLEAR TO SEND ($\overline{CTS}$)	OFF
TRANSMITTED DATA (TD)	IGNORED
BACK CHANNEL RTS ($\overline{BRTS}$)	OFF
BACK CHANNEL CTS ($\overline{BCTS}$)	OFF
BACK CHANNEL TD (BTD)	IGNORED
RING ($\overline{RING}$)	OFF
CARRIER DETECT ($\overline{CD}$)	OFF
RECEIVED DATA (RD)	MARK.
BACK CHANNEL CD ($\overline{BCD}$)	OFF
BACK CHANNEL RD ($\overline{BRD}$)	MARK.

TABLE 2.2.

MC_4	MC_3	MC_2	MC_1	MC_0	
0	0	0	0	0	Bell 103 Originate 300 bps full duplex
0	0	0	0	1	Bell 103 Answer 300 bps full duplex
0	0	0	1	0	Bell 202 1200 bps half duplex
0	0	0	1	1	Bell 202 with equalizer 1200 bps half duplex
0	0	1	0	0	CCITT V.21 Orig 300 bps full duplex
0	0	1	0	1	CCITT V.21 Ans 300 bps full duplex
0	0	1	1	0	CCITT V.23 Mode 2 1200 bps half duplex
0	0	1	1	1	CCITT V.23 Mode 2 with equalizer "
0	1	0	0	0	CCITT V.23 Mode 1600 bps half duplex
0	1	0	0	1	RESERVED
0	1	0	1	0	
0	1	0	1	1	
0	1	1	0	0	
0	1	1	0	1	
0	1	1	1	0	
0	1	1	1	1	
1	0	0	0	0	Bell 103 Orig. loopback.
1	0	0	0	1	Bell 103 Ans. loopback.
1	0	0	1	0	Bell 202 Main loopback.
1	0	0	1	1	Bell 202 with equalizer loopback
1	0	1	0	0	CCITT V.21 Orig loopback.
1	0	1	0	1	CCITT V.21 Ans. loopback
1	0	1	1	0	CCITT V.23 Mode 2 main loopback
1	0	1	1	1	CCITT V.23 Mode 2 with equalizer "
1	1	0	0	0	CCITT V.23 back loop back
1	1	0	0	1	RESERVED
1	1	0	1	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	0	1	
1	1	1	1	0	
1	1	1	1	1	

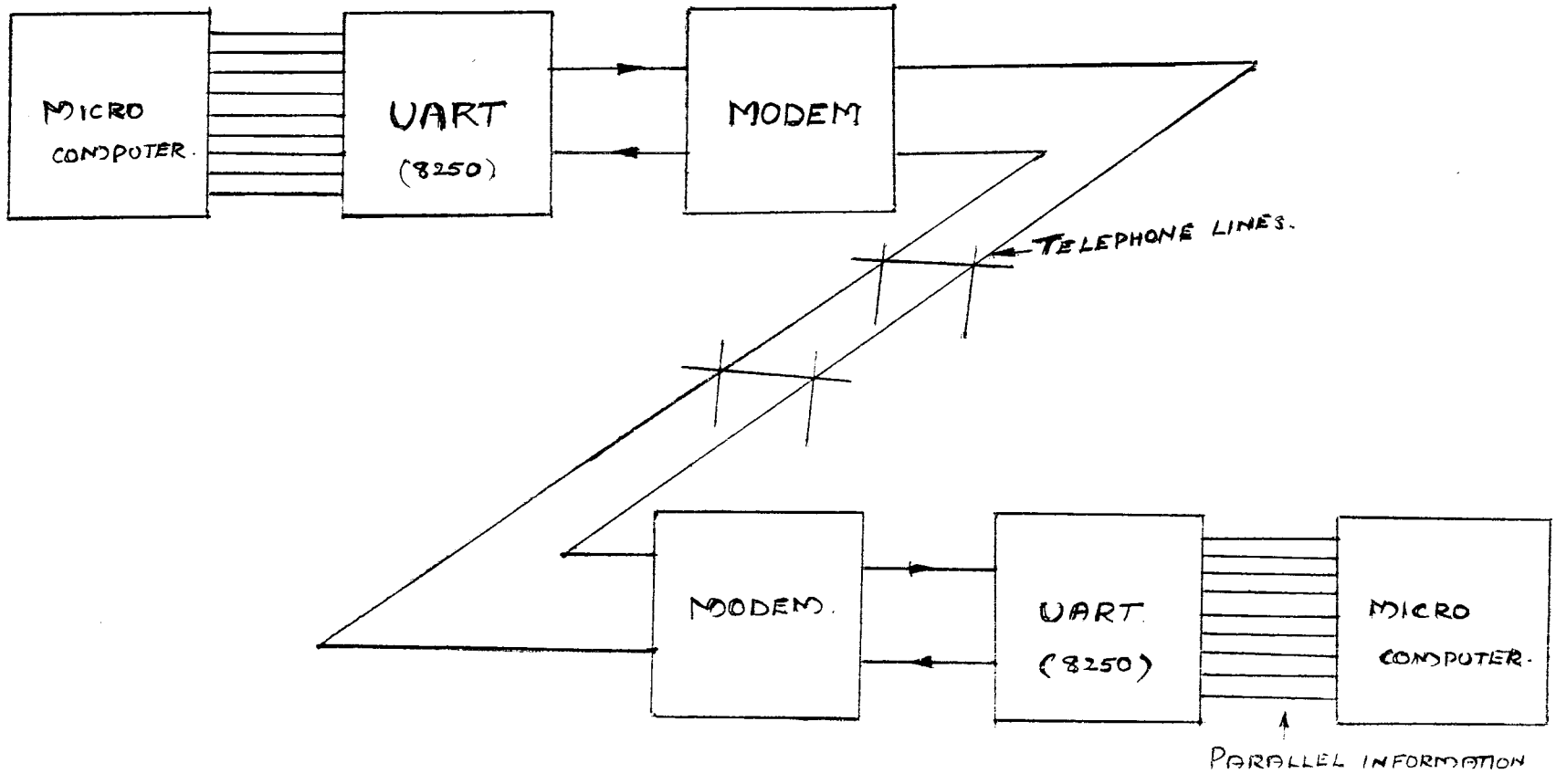


Fig. 2.1 : DATA COMMUNICATION SYSTEM

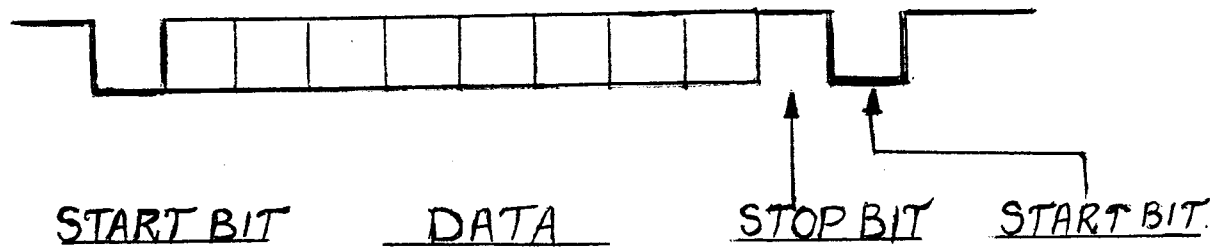


Fig. 2.2 : ASYNCHRONOUS DATA TRANSMISSION

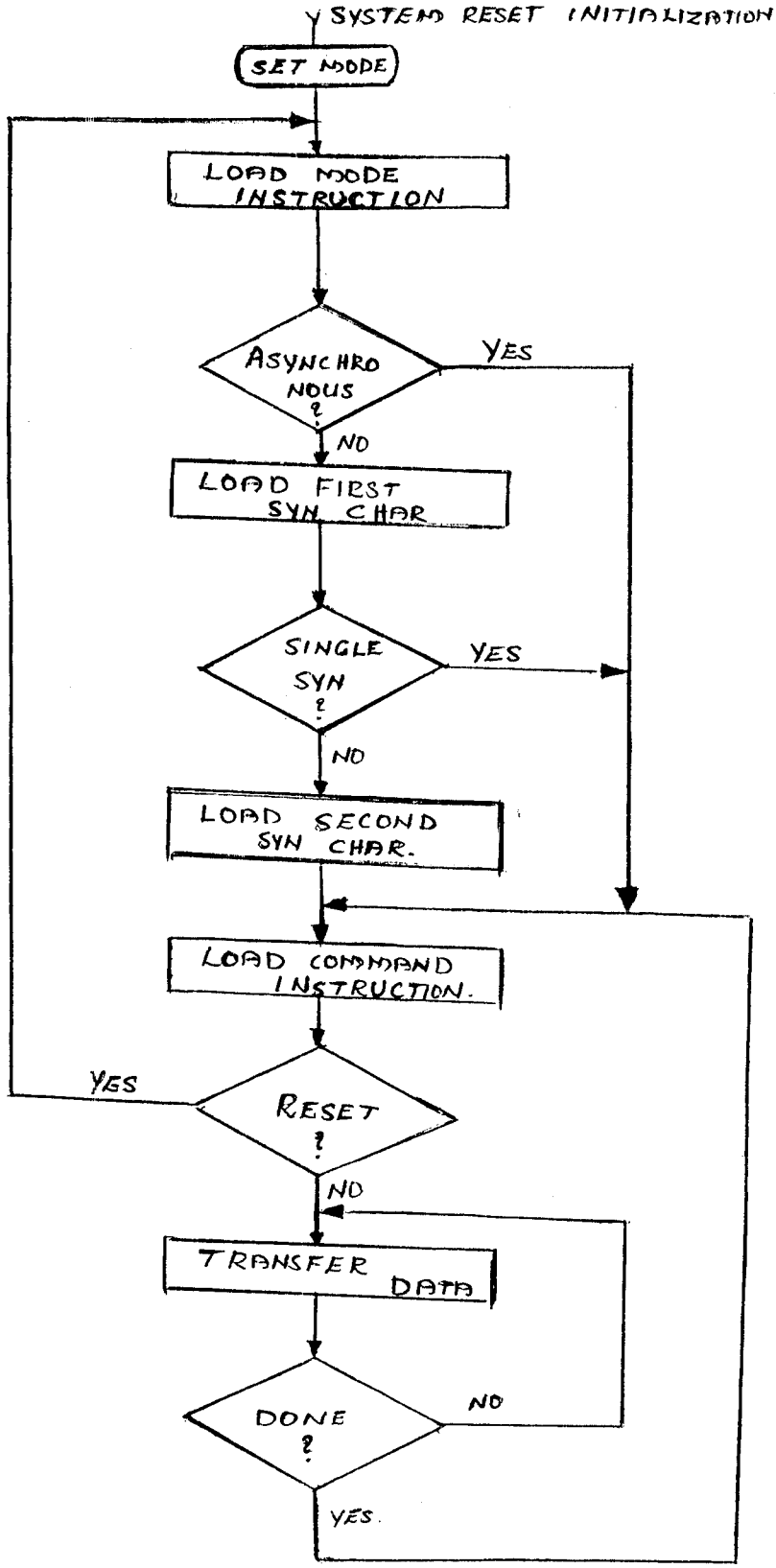


FIG. 2.3: INITIALIZATION FLOWCHART

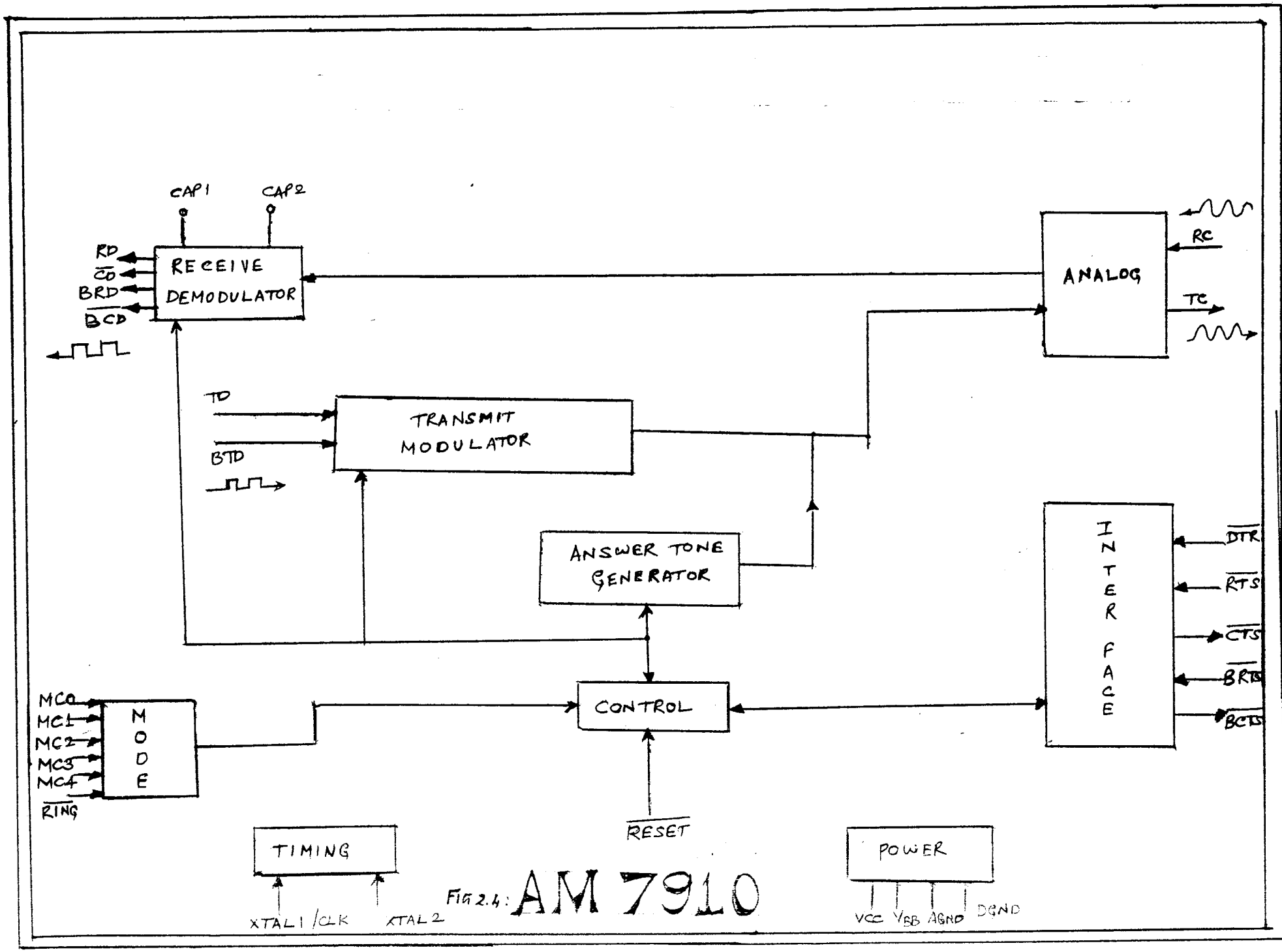


Fig 2.4: AM 7910

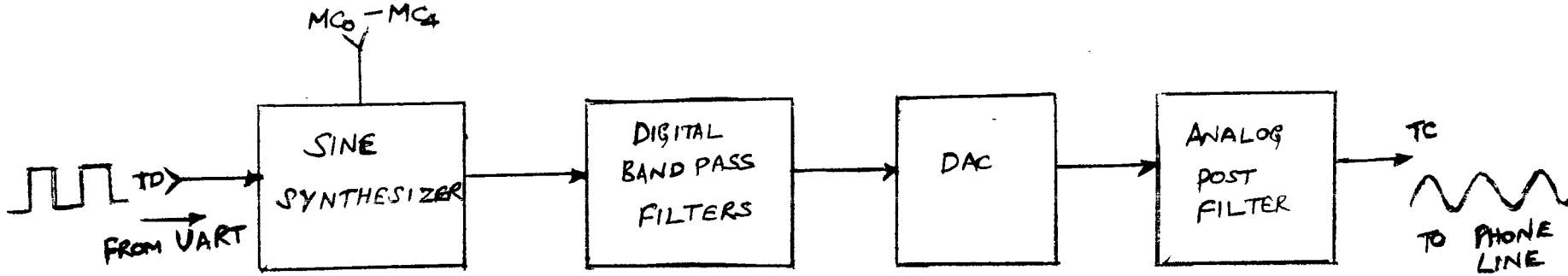


Fig. 2.5(a): MODULATOR

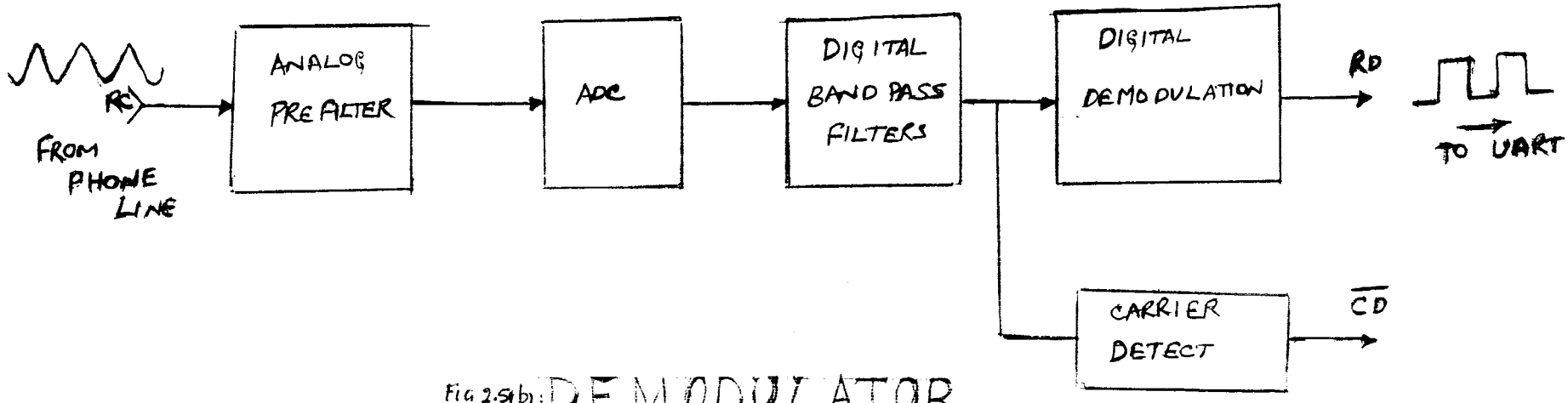


Fig 2.5(b): DEMODULATOR

CHAPTER THREE

MODEM HARDWARE

3.1 HARDWARE DESCRIPTION

The hardware of the system includes the following.

* MODEM AM7910 is the heart of the system which contains the complete modulator and demodulator section with appropriate filters.

* RS232 which connect the modem to the terminal and the modem is connected to the telephone lines.

* An amplifier Circuit which is used for amplifying the output data (Pin RD) from the modem.

The details of these aspects are dealt in the subsequent sections. The input to this series of blocks is the serial data and the control signals obtained from the UART. The output is the modulated data which is given to the line.

The modem has been discussed in detail in section 2.3. The rest of the hardware features are outlined here.

3.2 **INTERFACES**

An interface can be defined as the line of demarcation between two pieces of equipment. For two pieces of equipment to operate harmoniously, they must obey a complementary problems arise, as there are hundreds of different terminals and a wide range of modems. To avoid this, some standard interfaces have been introduced. Here the TTL/RS232C standard interfaces are used between the Data Terminal Equipment (DTE), the microcomputer and the Data Communication Equipment (DCE), the MODEM.

In long distance serial transmission of data, high frequency signals at TTL levels suffer attenuation and distortion. This can be overcome by conversion of the TTL level signals to the levels conforming to Electronics Industries Association (EIA Standards). For this an interface driver/receiver are required.

So an EIA RS232 interface driver at the transmitting end converts the TTL level to 12v bipolar signals and an EIA interface receiver at the other end reconverts the bipolar signals to TTL levels. Fig. 3.1 illustrates one such interface, between a DTE and DCE using TTL/RS232 interfaces.

DATA COMMUNICATION BETWEEN THE MICROCOMPUTER AND THE MODEM VIA RS232-C INTERFACE.

The control signal and the data communication between the modem and terminal are set along the TTL/RS 232-C connectors. They are discussed below:

* When a data connection is required, the first step is to dial the required number on the telephone in the usual manner. When the call is answered and a communication path exists between the two ends, the telephone line needs to be switched at each and from the telephone to the data terminal. This is done by the modes.

* When the DTE is connected to the line, the modem informs the terminal accordingly by making DSR line high.

* If the distant modem is already connected to the line and is transmitting data, the local modem will make the Carrier

Detect line high.

* Any data that is received will be passed to the terminal over the Received Data (RD) line
* If the terminal wishes to transmit, it will enable RTS line.
* When the modem is ready to accept data for transmission, it replies to the computer by making its CTS line low.
* Data can then be transmitted by the terminal through the Transmit Data (TD) line
* The signal Ground pin provides an essential common return lead for all the circuits.
* The back channel pins namely the BCD, BCTS, BTD,BRD have the same function as that of CD, CTS, TD and RD.

All the above functions are conveyed between the terminal and the modem via the TTL/RS232 connectors only.

3.3 OPERATING A MODEM IN TEST MODE

Loop back:-

In the test mode operation a loopback technique is used. It is shown in Fg.3.2 Ten modes exist to allow both analog and digital loop back for each modem specification met by the AM 7910. When a loop back mode in selected, the signal processing for both the transmitter and receiver is set to process the same channel or frequency band. This allows the analog output (TC) and the analog input(RC) to be connected for local analog back. Alternatively the digital data signals (TD and RD (or) BTD and BRD) can be connected externally, allowing a remote modem to test the local modem with its digital data signals looped back.

When a loop back mode is selected the state machine sequences are altered slightly. First auto answer is disabled. Second, it a half-duplex loop back mode is selected the local CD|BCD is not forced OFF when RTS|BRTS is asserted.

After checking the modem in test loop back mode, disconnect TC and RC loops provided the chip is working properly in the test mode. Connect a telephone wire in the TC output pin of modem and at the other end, connect this telephone wire to RC input of the far end modem. The output data is available at

Receive Data (RD) output pin.

But the modem chip output at the pin RD is very weak and inverted. That is the amplitude of the signal is very low so that it is not possible to display it. In order to avoid this effect we use an inverting Amplifier, making use of operational amplifier (ICNO 358). The circuit diagram is shown in the fig.3.3. The RD output from the modem is given as input to the amplifier and the amplified output is given to the Receive Data input of the 8250 chip present inside the computer terminal. Thus the transmision of data will take place. RS232-C cable is used for transmitting signals from modem to the input part of 8250 chip (UART) in the computer. Grounding should be done perfectly. The ± 15v power circuitry needed for the amplifier is constructed with the help of 7815 & 7915 voltage regulators filters & rectifier circuits.

3.4 **POWER SUPPLY CIRCUITS**

This unit requires two values of power supplies as given below:

1. +5v - 300mAmp power supply
2. -5v - 300 mAmp Power supply

+5v Power Supply (Vcc)

This supplied the required voltage for the TTL IC's. This has been built with a bridge rectifier and a regulator 7805. The

circuit is shown in fig.3.4 Vcc should be decoupled to DGND by a 0.01 microfarad ceramic capacitor.

-5 Power Supply (VBB)

This supplies the required voltage for the MODEM AM7910 and operational amplifiers. This also has been built with a bridged rectifier and a regulator 7905. The circuit is shown is Fig.3.4 VBB should be decoupled to AGND by a 0.01 microfarad ceramic capacitor.

TABLE 3.1 : RS 232 PIN DETAILS.

PIN DETAILS	RS232 PIN
TRANSMITTED DATA.	2.
RECEIVED DATA	3.
REQUEST TO SEND	4.
CLEAR TO SEND	5
DATA SET READY	6.
SIGNAL GROUND	7.
PROTECTIVE GROUND	1.
DATA CARRIER DETECT	8.
BACK CHANNEL DATA CD	12.
BACK CHANNEL CTS	13.
BACK CHANNEL TD	14.
BACK CHANNEL RD	16.
DATA TERMINAL READY	20.

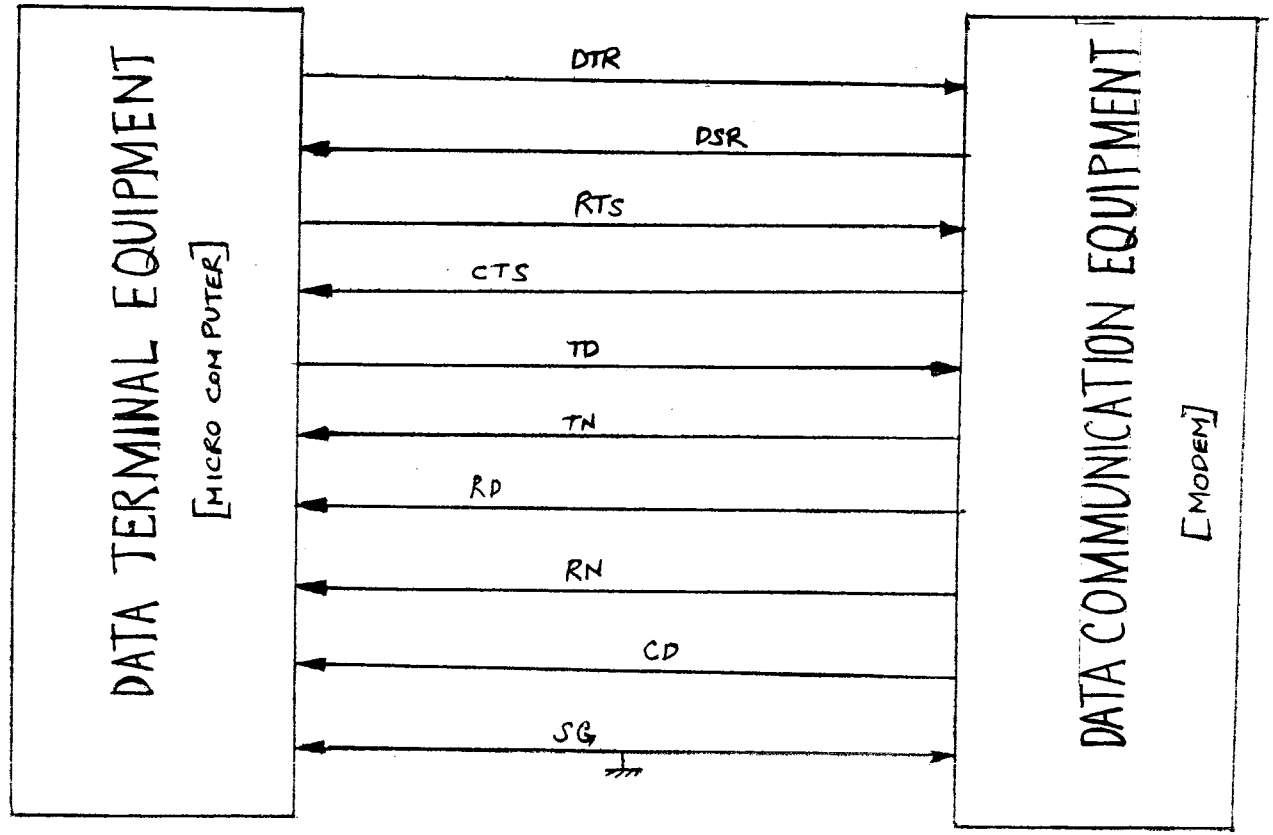


Fig. 3.1 : CONNECTION BETWEEN DTE & DCE USING RS-232

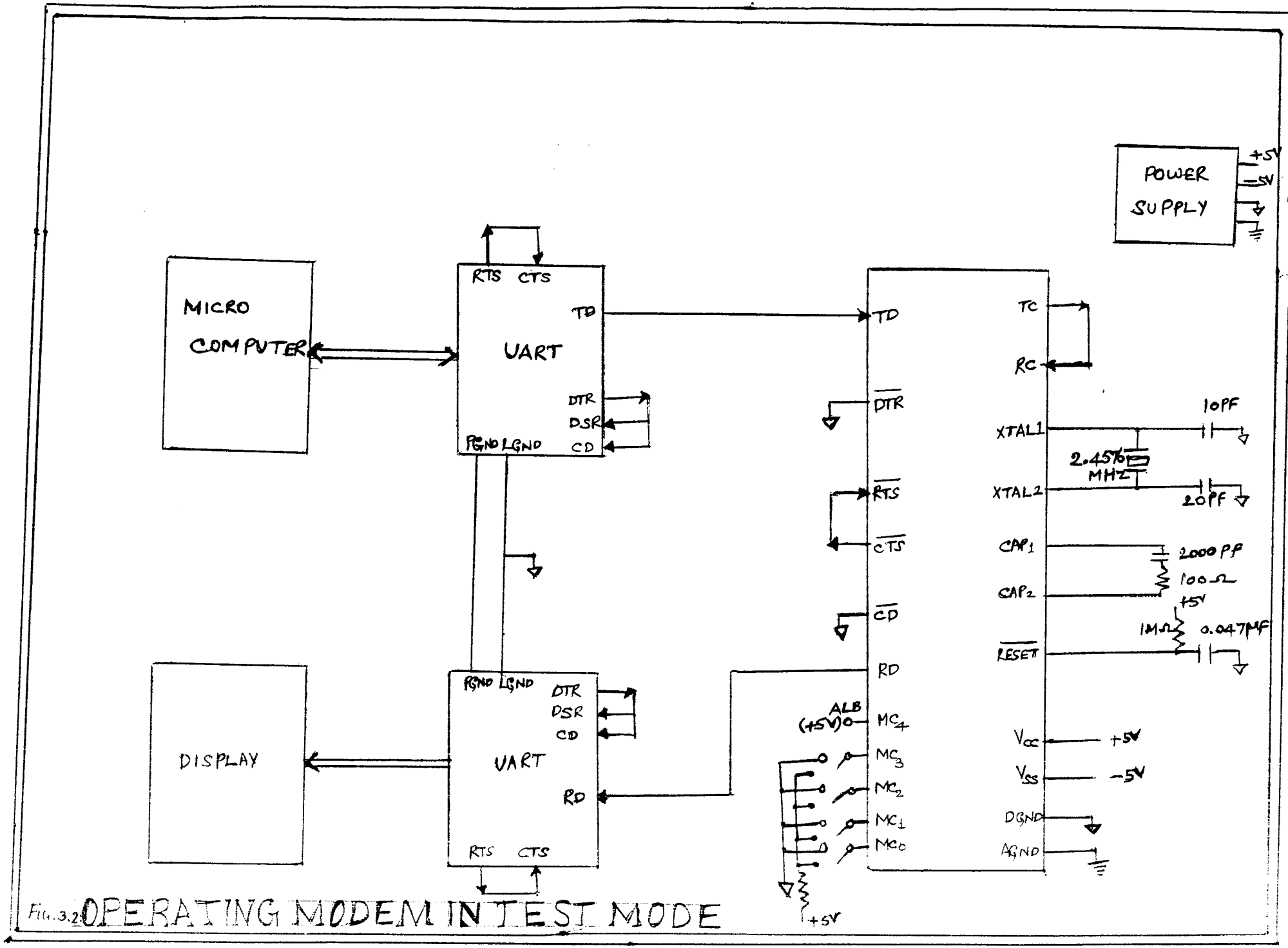


Fig. 3.2 OPERATING MODEM IN TEST MODE

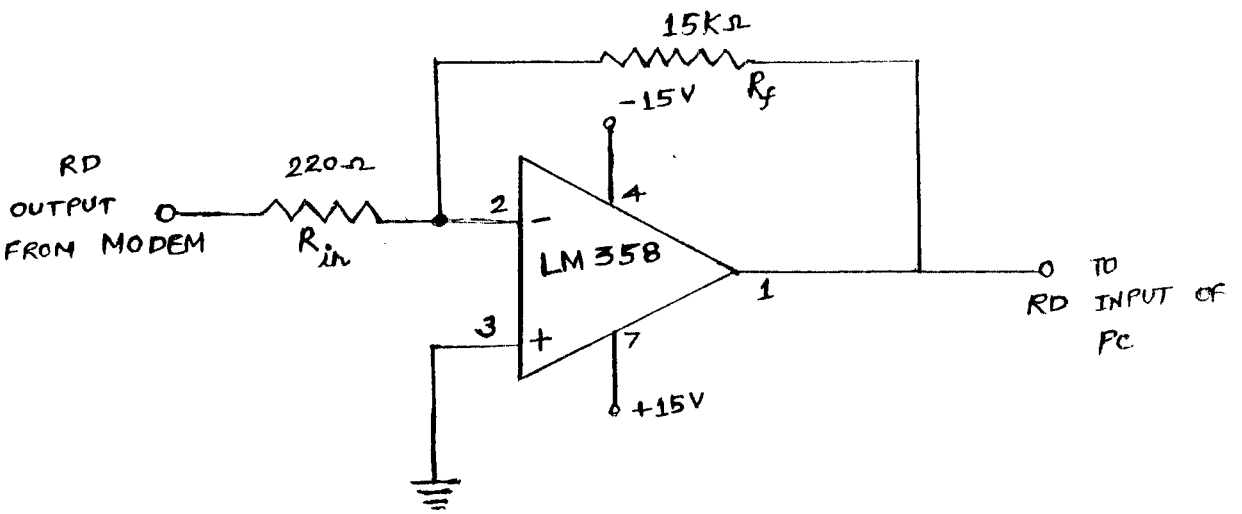
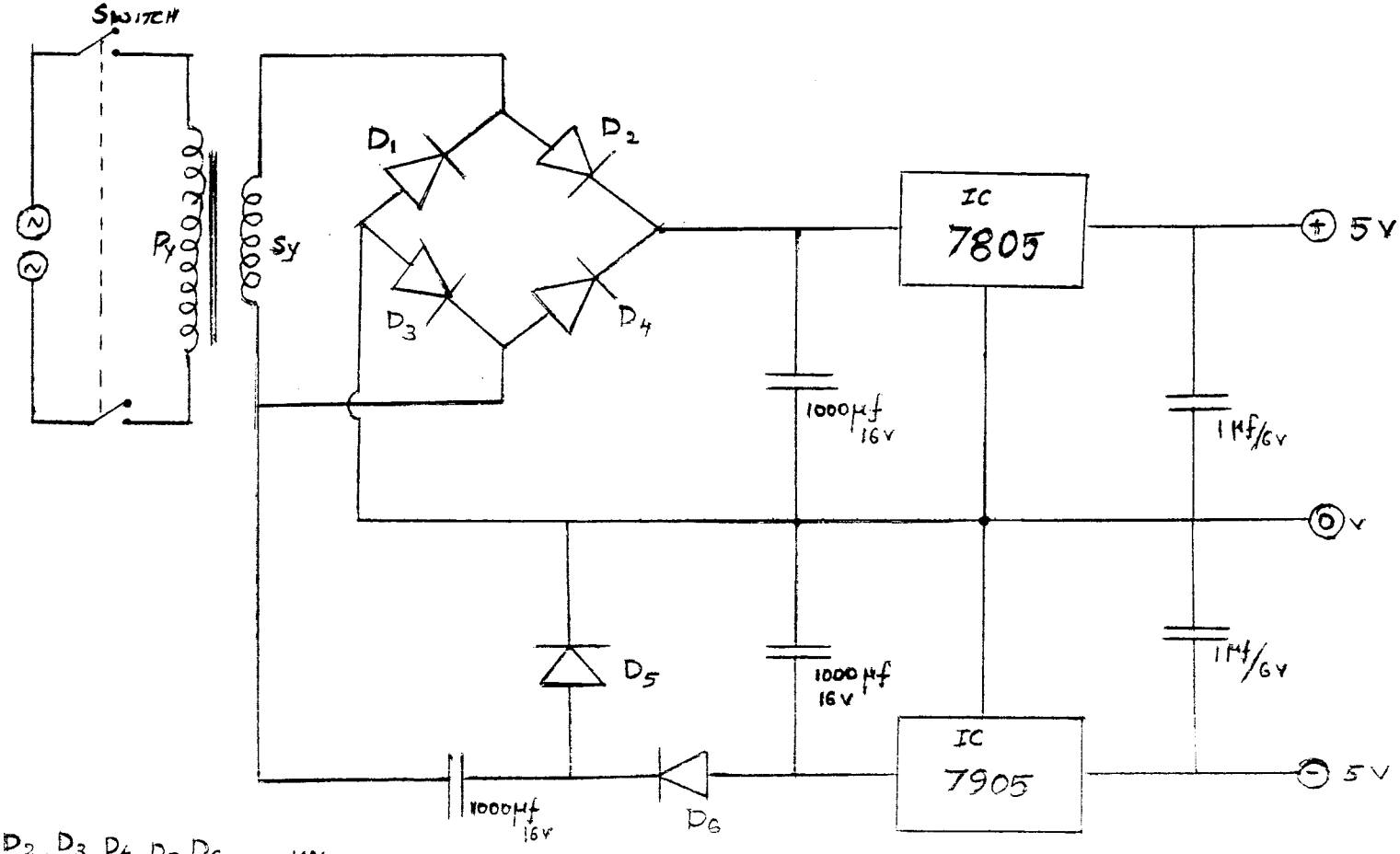


Fig 3.3: AMPLIFIER



$D_1, D_2, D_3, D_4, D_5, D_6$ → IN4001

FIG. 3.4 : POWER SUPPLY

CHAPTER FOUR

COMMUNICATION PROCESS

This part includes Call Establishment between two Microcomputers, Data transmission and Reception in the Half Duplex mode using both main and back channels.

CALL ESTABLISHMENT

Before two modems can exchange data, an electrical connection through the phone system must be estalished. A call may be originated manually or automatically and it may be answered in either fashion.

Manual Calling

Manual Calling is performed by person who deals the number, waits for an answer, then places the calling modem into data transmission mode.

Automatic Calling

Automatic Calling is typically performed by an automatically calling unit (ACU) which generates the appropriate dialing pulse required to call the remote modem. This unit also has the ability to detect an answer tone from the called modem and place the calling modem into data transmission mode.

Manual Answering

Manual Answering is performed by a person who hears the phone ring, lifts the receiver, causes the called modem to send an answer tone to the calling modem, and places the called modem into data transmission mode.

Automatic Answering

This is performed by a called modem with a data access arrgement (DAA). The DAA detects a ringing signal, takes the phone circuit off-hook and instructs the called modem to commence the auto-answer sequence.

DATA TRANSMISSION

Half Duplex Mode of Operation

When the half duplex mode is selected, data transmission can either be on the main channel at 1200/600 baud or on the back channel at 5/75 baud. During normal half duplex operation, a single modem is either transmitting on the main and receiving on the back channel or vice versa. In the AM7910, control of the transmitter and receiver filters to the proper channel is performed by RTS. When RTS is asserted, the transmitter filters and synthesizers are set to transmit on the main channel; the receiver filters are set to the main channel.

If RTS and BRTS are asserted simultaneously, RTS will take precedence.

Handshake Timing Signals

For proper communication to take place between any two Microcomputers, certain control signals are necessary and these control signals are discussed in this section.

Fig.4.1a and 4.1b show the timing diagram of these signals. Depending upon the assertion of the RTS or BRTS signals, transmission will take place either through the main or back channel. The RTS advices the modem, that the terminal wants to transmit and the modem gets ready for transmission by sending the carrier wave down the telephone line. Now the transmitting modem returns a signal called CLEAR TO SEND (CTS) signal to its DTE. This tells the terminal that it can proceed to transmit data. The terminal send its block of data, which is modulated onto the carrier by the transmitting modem. When the sending terminal has finished transmitting the data block, it removes the RTS signal. This causes the transmit modem to drop both the carrier wave and the RTS signal.

DATA RECEPTION

In data reception, **CARRIER DETECT (CD)** may appear at any time after the phone connection has been established and immediately after a ring signal is detected. Reception is independent of Transmission. When the receiver detects a valid carrier for at least a time, the output CD is turned ON the **RECEIVED DATA (RD)** output is released, and valid data can be obtained at **RD**. Data is received until the receiver detects loss of carrier for at least a time of 1/2 sec. At this time **CD** output is turned **OFF** and **RD** is clamped to a **MARK**.

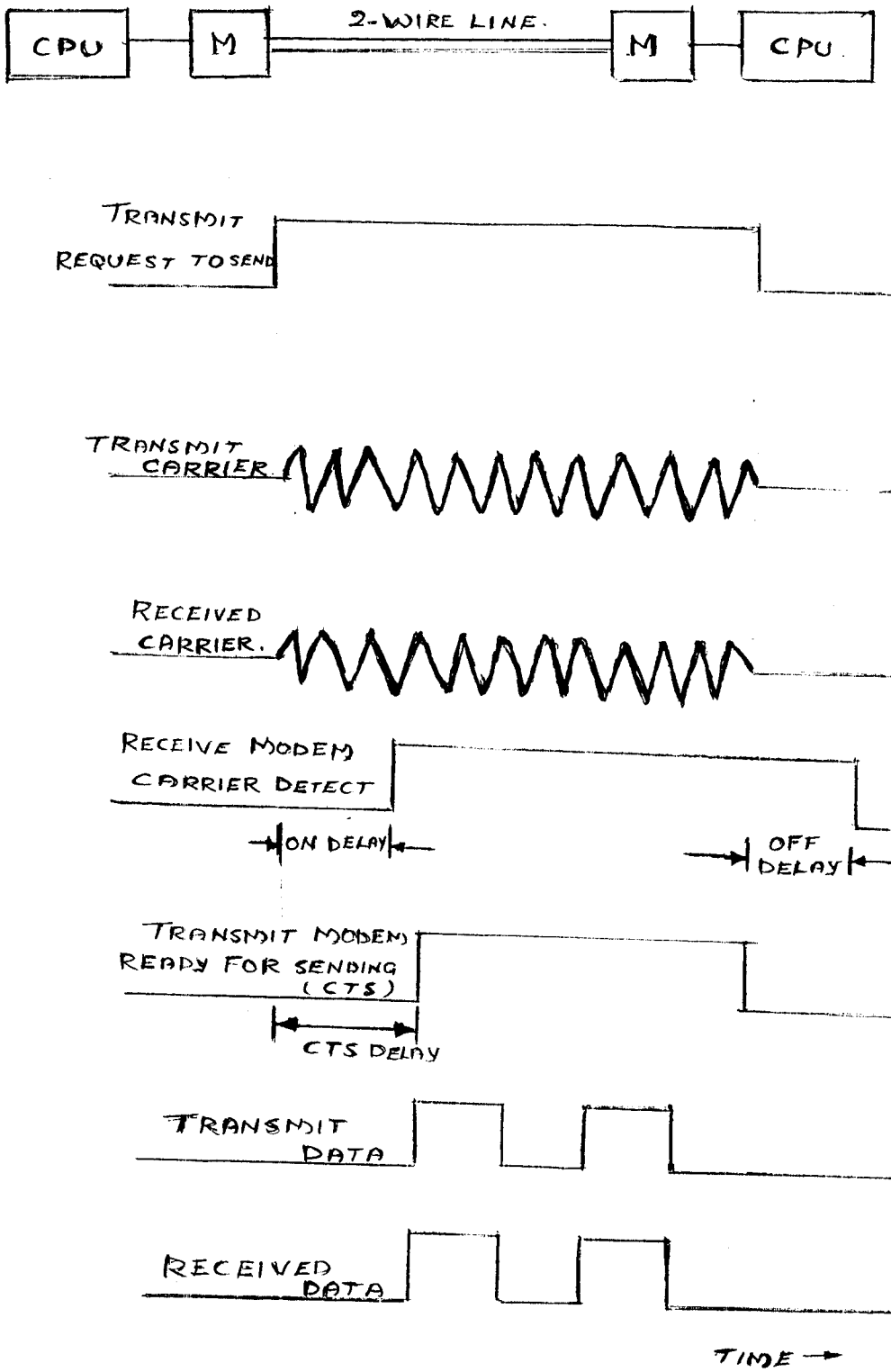


Fig 4.1(a): MODEM TIMING FOR 2-WIRE OPERATION

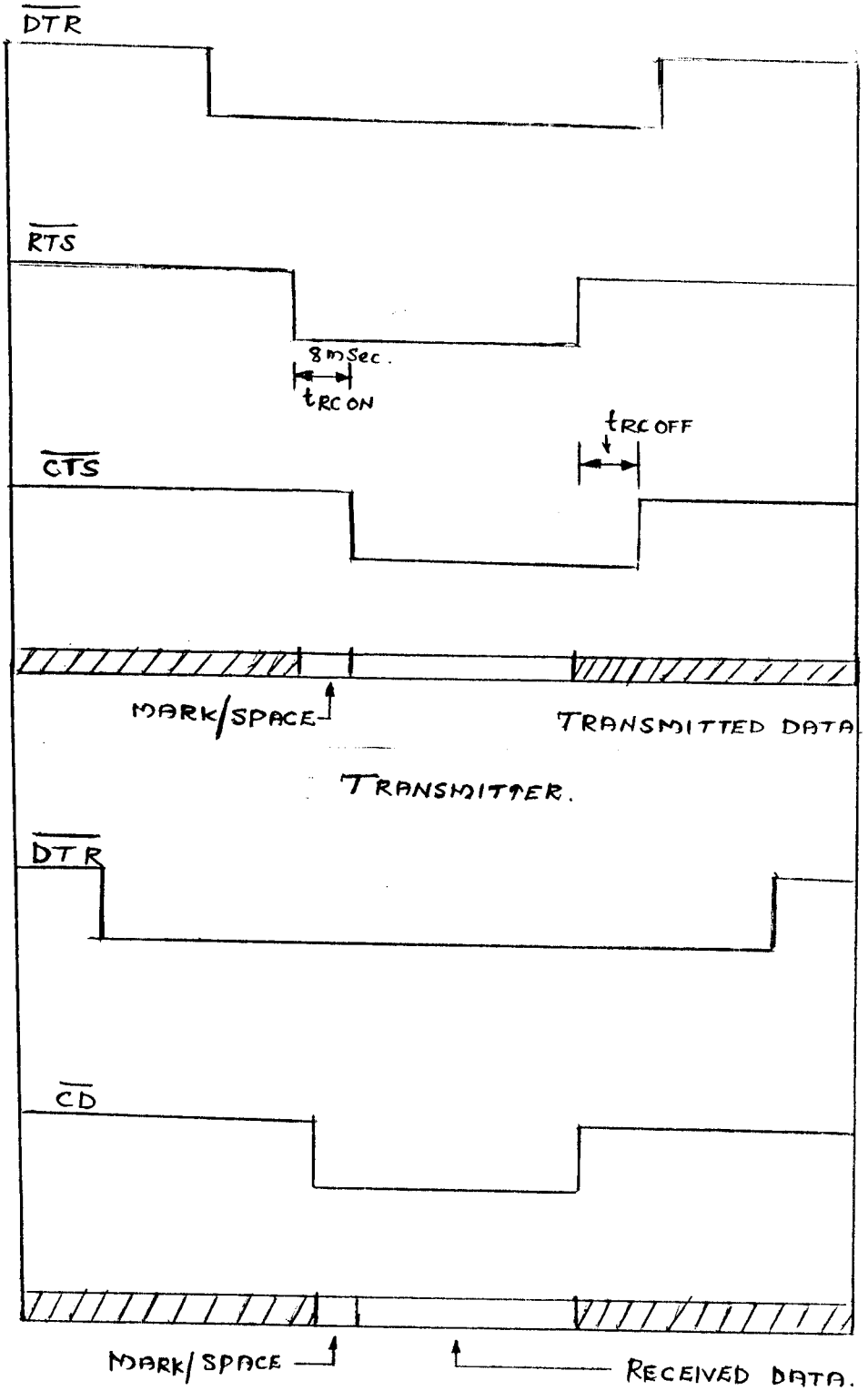


Fig. 4.1(b): HANDSHAKE TIMING.

CHAPTER FIVE

DATA TRANSFER

SERIAL DATA TRANSFER:

UART(8250) chip is used for serial data transfer. The serial transfer of a block of data, from one microcomputer to another, has also been done without modulation.

The Transmitted Data output pin of the UART 8250 of the transmitting microcomputer was directly connected to the Received Data input pin of the UART, in the receiving microcomputer through modem. A common ground has also been given.

The software for the above is implemented in a microcomputer for various data and the transfer is verified.

5.1 Program the 8250 UART chip

Serial communications is so complicated that special chips have been designed to do the work of forming and timing the strings of bits that comprise serial data. Such a chip is called a univeral asynchronous receiver transmitter, or UART(8250).

The 8250 has ten programmable one-byte registers by which to control and monitor the serial port. The ten registers are accessed through seven port addresses, numbers 3F8H - 3FEH(or 2F8H - 2FEH). It is given in table 5.1. Of the ten registers, only six are necessary for simple serial communications. The transmitter holding register holds the byte of data about to be sent and the receiver data register keeps the most recently received byte of data. The line control and line status registers initialize and monitor the serial line, using the baud rate placed in the two baud-rate divisor registers. Of the remaining four registers, the modem control and modem status registers are used only for modem communications and the two interrupt-related registers are used only in interrupt-driven routines.

Interrupts are used in communications for reasons of efficiency. Simple communications routines constantly monitor the line status register, waiting for an incoming character, or waiting unitl the register indicates that it is all right to transmit another byte of data. Because the CPU operates very quickly relative to the 300 or 1200 bit-per-second rate at which serial data typically moves, this method can be wasteful of CPU time that might otherwise be devoted to processing the incoming/outgoing data. For this reason the 8250 may be set up to bring about an interrupt whenever a character arrives, an error occurs, etc.

5.2 Initialize the serial port

When a communications port is initialized all of the parameters by which it operates are set. These parameters include the world length, the number of stop bits, the parity settings and the baud rate. The word length is the number of bits that form the basic data unit. The following paragraph given the initialization of each Registers.

Baud Rate Initialization:

The baud-rate divisor is a number that divides the rate of the system clock-1190000Hz to give a result that equals the desired baud rate. For example, for 1200 bps the baud-rate divisor would be 96, since 119000/96 equals roughly 1200. The larger the divisor, the slower the baud-rate. Baud rates of 300 and under require a two-byte number for the divisor, and for this reason the 8250 chip needs two registers to hold the divisor. The high byte is send to 3F9H (or 2F9H), and the low byte to 3F8H (2F8H). In both cases, bit 7 of the line control register at 3FBH (2FBH) must be set to 1 before sending values; Table 5.2 gives some common baud rates.

Always set the baud rate registers first since they are the only ones that require that bit 7 equal 1 in the line control register. Then set the contents of the line control

register, making bit 7 equal 0 so that all subsequent register accesses are correct. since the line control register is write-only, there is no way to set bit 7 back to 1 without redoing all the bits in the register.

Line Control Register Initialization:

The bit settings for the line control register at 3FBH (or2FBH) are given in table 5.3. The bits 5-7 are set to 0. The others are given the values of the desired communications protocol.

The Interrupt-enable register:

Even when interrupts are not used, you should access the interrupt-enable register to be sure that interrupts are disabled. Simply place 0 in the register. The interrupt identification register may be ignored. Here we are not using any interrupt interface.

Monitor the status of the serial port

The line status register of the 8250 UART sets up the communications protocol. Ordinarily it is constantly monitored during communications activity. During data transmission, the

register tells when the prior character has been sent off, lest the program write the next character on top of it. In data reception, the register informs the program when a character arrives, so that the program can remove it before it is overlaid by the one that follows. The contents are given in Table 5.4.

5.3. Initialize and monitor the modem

There are six lines by which modems communicate with the computer. They are DTR,RTS,DSR,CTS,DCD,RI. Each one is explained in the Appendix.A.

First the computer turns the data terminal ready signal on, and then it instructs the modem to dial the remote station. Once the modem has established a connection, it turns on the data set ready signal. This informs the computer that the modem is ready for communications, and at that point the computer can turn on the request to send signal. When the modem replies with clear to send, transmission can begin.

The two standard lines by which the computer controls the modem may be accessed through the modem control register on the 8250 UART chip. Bit pattern in the register is explained in Table 5.5.

Ordinarily bits 0 and 1 of the modem control register are set to 1, and the others are set to 0, Bit 2 is set to 0 unless a modem's manufacturer has given it a special use. Bit 3 is set to 1 only when interrupts are used. Finally, bit 4 is a special feature that is useful for testing communications programs without actually going on line. The output signal from the UART is looped back so that the UART receives it as serial input. This feature may be used to test whether the chip is functioning properly.

The four lines by which the modem sends information to the computer are monitored through the modem status register. The register is located at the port address that is 6 higher than the base address of the communications adaptor in use the bit patterns are given in Table 5.6.

Programs constantly monitor these bits during communications operations. Note that the four low bits parallel the four high bits. These bits are set to 1 only when a change has occurred in the status of the corresponding high bit since the last time the register was read. All four low bits are automatically restored to 0 after the read operation. Most modems have many more capabilities like autodial and autoanswer are controlled by control strings.

TABLE 5.1 REGISTERS OF 8251 [UART]

PORT ADDRESS	REGISTERS
3F8H (bit 7=0 at 3FBH)	TRANSMITTER HOLDING REGISTER
3F8H (bit 7=0 at 3FBH)	RECEIVER DATA REGISTER
3F8H (bit 7=1 at 3FBH)	BAUD RATE DIVISOR (LOW BYTE)
3F9H (bit 7=1 at 3FBH)	BAUD RATE DIVISOR (HIGH BYTE)
3F9H (bit 7=0 at 3FBH)	INTERRUPT ENABLE REGISTER
3FAH (IN)	INTERRUPT IDENTIFICATION REGISTER
3FBH (OUT)	LINE CONTROL REGISTER
3FCH (OUT)	MODEM CONTROL REGISTER
3FDH (IN)	LINE STATUS REGISTER
3FEH (IN)	MODEM STATUS REGISTER

TABLE 5.2 : BAUD RATE INITIALIZATION.

BAUD RATE.	3F9 H	3F8 H
110	04 H	17 H
300	01 H	80 H
600	00 H	C0 H
1200	00 H	60 H
1800	00 H	40 H
2400	00 H	30 H
3600	00 H	20 H
4800	00 H	18 H
9600	00 H	0C H

TABLE 5.3 : LINE CONTROL REGISTER INITIALIZATION.

BITS.	
1-0	Character length 00 = 5 bits, 01 = 6 bits 10 = 7 bits, 11 = 8 bits
2	Number of stop bits 0 = 1, 1 = 1.5 if the character length is 5, else = 2.
3	Parity: 1 = parity bit is generated, 0 = not
4	Parity type. 0 = odd, 1 = Even
5	Stick Parity
6	Set Break. 0 = disabled, 1 = break
7	Toggless port addresses of other reg. on chip.

TABLE 5.4 : LINE STATUS REGISTER.

BIT	
0	1 = a byte of data has been received.
1	1 = received data has been overrun.
2	1 = Parity error.
3	1 = Framing error
4	1 = break detect.
5	1 = Transmitter holding register empty.
6	1 = Transmitter shift register empty.
7	1 = time out (off-line)

TABLE 5.5 : MODEM CONTROL REGISTER.

BITS.	
7-5	(always 0
4	1 = UART output looped back as i/p
3.	Auxiliary user designated o/p # 2
2	Auxiliary user designated o/p # 1
1	1 = "Request to send" is active.
0	1 = "Data terminal ready" is active.

TABLE 5.6 : MODEM STATUS REGISTER.

BITS	
7	1 = "Data carrier detect"
6	1 = "Ring indicator."
5	1 = "Data set ready"
4	1 = "Clear to send"
3	1 = Change in "data Carrier detect"
2	1 = Change in "ring indicator"
1	1 = Change in "data set Ready"
0	1 = Change in "Clear to send"

CHAPTER SIX

SYSTEM SOFTWARE

6.1 FLOW CHART DEVELOPMENT

(a) **Transmitter Section:**

(i) Initializing UART (8250) control and status registers.

(ii) Setting DTR and checking for DSR.

(iii) After receiving DSR, issue RTS and waiting for CTS.

(iv) When CTS is received, get the data from memory and dump the character in the UART.

(v) Updatting the transmitter holding register in UART with character from memory locations, until all characters are sent.

(vi) Remove RTS, to terminate the transmission.

(b) **Receiver Section:**

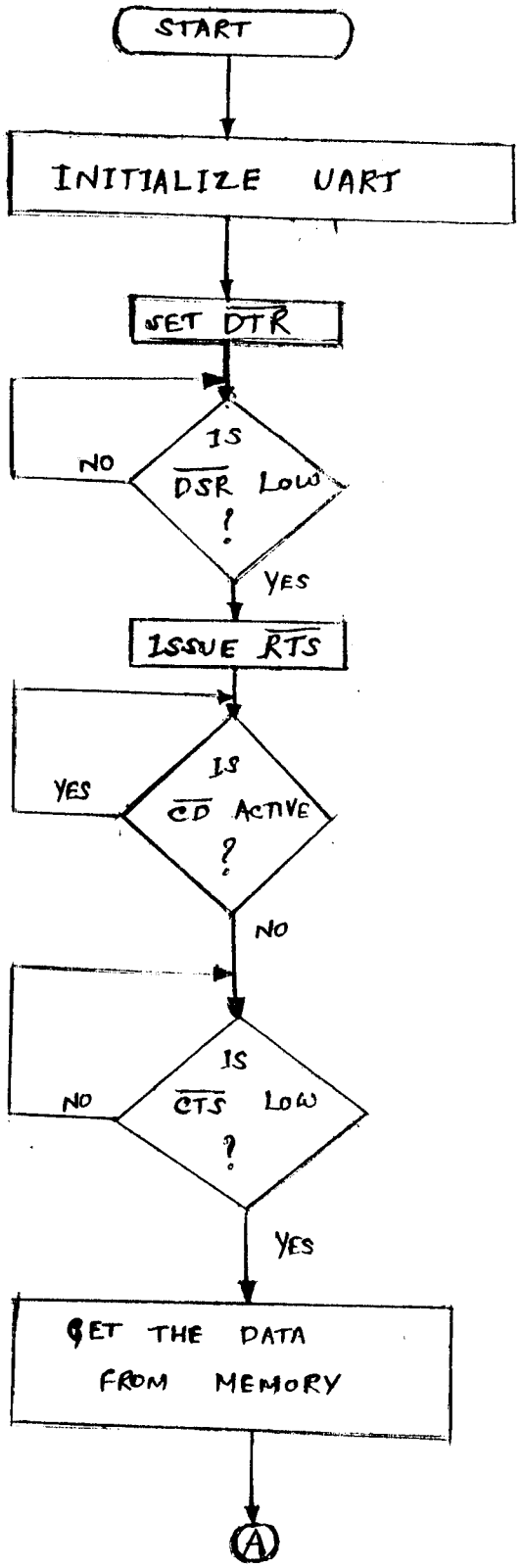
(i) Initializing the ports in UART(8250).

(ii) Setting DTR & checking for CD.

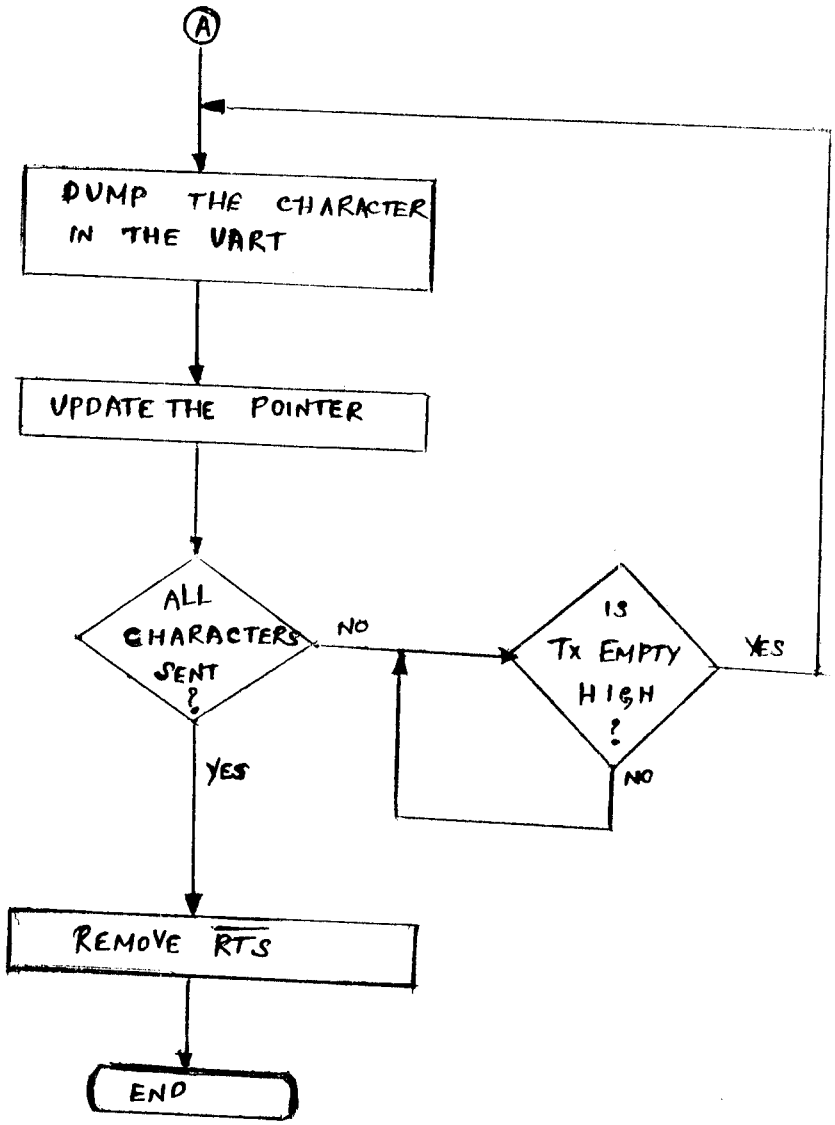
(iii) When CD is received, get the character from UART.

(iv) Dump in memory & update the character of receiver data register in UART, once new character entered the UART. Every instance of character present at port, CD is detected.

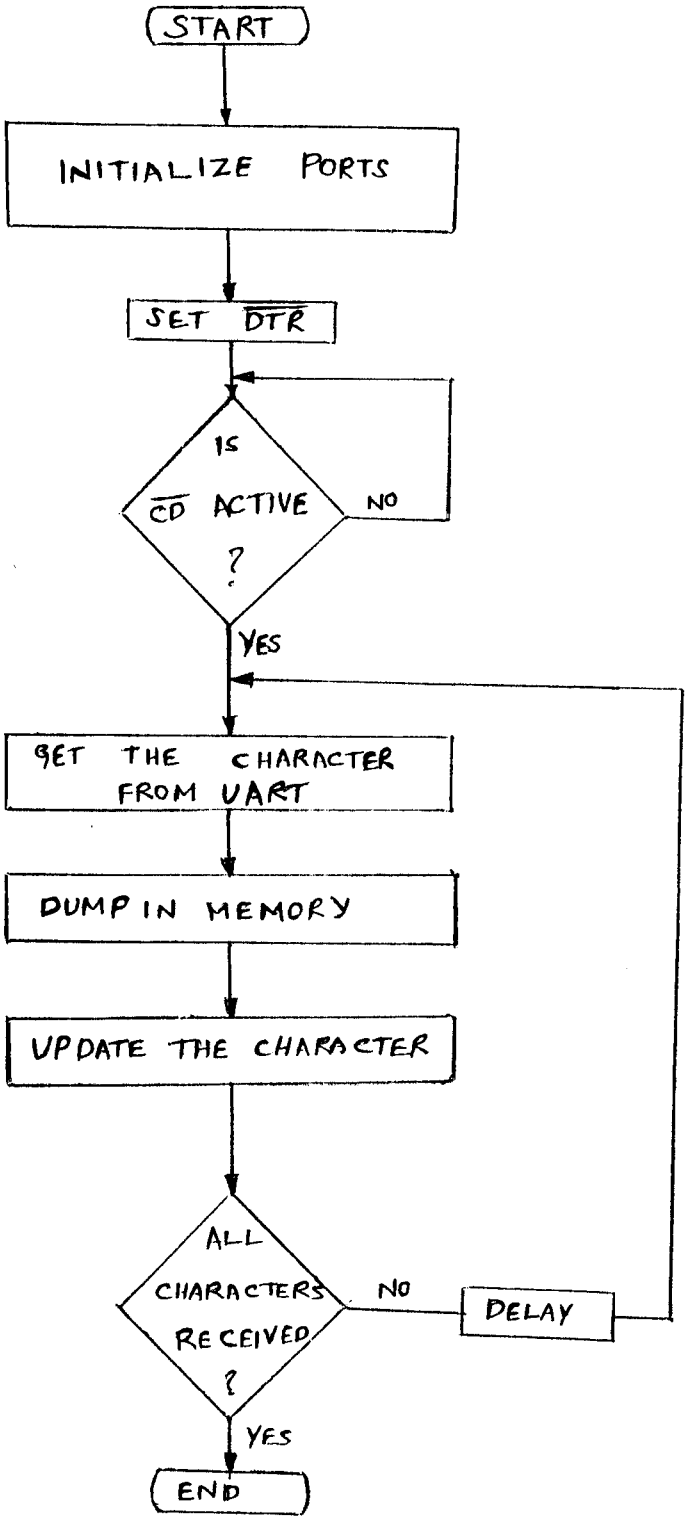
(v) Once all characters are received in similar fashion CD will be disabled.



FLOWCHART 6.1 TRANSMITTER.



TRANSMITTER SECTION (cont'd)



FLOWCHART 6.2 : RECEIVER SECTION

THE FILE TRANSFER & RECEIVER PROGRAM

```
# define PORT 0
# include "dos.h"
# include "stdio.h"
void send();
char receive();
void init(),ackno();
int stat_com();
int choice,i;
char name[25];
char c,d,k,dtr=0x01,dsr,rts,cts,cd,d1;
int mcr,msr=0,lcr,tr,rr,no;
int p=0;
main()
{
 FILE *fp;

  printf("Enter the communication port no 1/2\n");
   scanf("%d",&no);
   switch(no)
            {
             case 1 : mcr=0x3fc;
                      lcr=0x3fb;
                      tr=0x3f8;
                      rr=0x3f8;
                      msr=0x3fe;
                      p=0;
                      break;
             case 2 : mcr=0x2fc;
                      lcr=0x2fb;
                      tr=0x2f8;
                      rr=0x2f8;
                      msr=0x2fe;
                      p=1;
                      break;
                      }
                      init(p,251);
  printf("MAIN MENU\n");
  printf("1:SEND\n");
  printf("2:RECEIVE\n");
  printf("ENTER YOUR CHOICE");
  scanf("%d",&choice);
  switch(choice)
  {
  case 1 : printf("Enter the file name foe sending\n");
           scanf("%s",name);
           fp=fopen(name,"r");
           if ( fp == 0)
           {
            printf("File not found\n");
            exit(1);
            }
```

```
                fp=fopen(name,"r");
                printf("DTR sending\n");
                outportb(mcr,dtr);
                printf("waiting for DSR\n");
                tt:d1=inportb(dsr);
                d1=d1&0x02; /* change to 20 if not working */
                if ( d1 != 0x02) goto tt;
                printf("DSR Receive \n");
                printf("sending RTS\n");
                outportb(mcr,0x03);
                printf("Waiting for clear to send\n");
tt1:            d1=inportb(msr);
                d1 = d1  & 01 ; /* change to 10 if not working */
                if ( d1 != 0x01) goto tt1;
                printf("CTS received \n");
                printf("DATA Transmission started\n");
                do
                {
                  send(PORT,'&');
                  printf("Checking for acknowledgement\n");
                  } while( !(stat com(PORT)&256));
                 c=receive(PORT);
                 if (c=='.') printf("Recevier ready,ack received\n");
                 while ( (c=getc(fp)) != EOF)
                 {
                   printf("sending RTS\n");
               outportb(mcr,0x03);
               printf("Waiting for clear to send\n");
tt2:           d1=inportb(msr);
               d1 = d1  & 01 ; /* change to 10 if not working */
               if ( d1 != 0x01) goto tt2;
               printf("CTS received \n");
                   send(PORT,c);
                   while( receive(PORT) != '.')
                   printf("Acknowledgement verification for the charcater %
c);
                   }
                   send(PORT,'$');
                   printf("end of file sent\n");
                   fclose(fp);
                   break;
case  2   :
            check : k=receive(PORT);

            if(k!= '&')
           {
            printf("==========================\n");
            printf("Checking for transmitter\n");
           goto check;
           }
           if ( k== '&') printf("PERMISSION GRANTED\n");
                       outportb(mcr,0x03);
              printf("Waiting for clear to send\n");
```

```
tt3:            d1=inportb(msr);
                d1 = d1  & 01 ; /* change to 01 if not working */
                if ( d1 != 0x01) goto tt3;
                printf("CTS received \n");
             send(PORT,'.');
             printf("acknoledgement sent\n");
          beg:   d=receive(PORT);
             printf("%c",d);
             tt4:                  d1=inportb(msr);
                d1 = d1  & 01 ; /* change to 01 if not working */
                if ( d1 != 0x01) goto tt4;
                printf("CTS received \n");

             send(PORT,'.');
             if ( d == '$') exit(1);
             else
             goto beg;
             break;
 case 3    : exit(1);
       }
       }



          void ackno(int port)
          {
                if(receive(port)!='.') {
                  printf(" communication error ");
                  exit(1);
                 }
          }



          void send(port,c)
          int port;
          char c;
          {
                union REGS r;
                r.x.dx = port;
                r.h.al = c;
                r.h.ah =1;
                int86(0x14,&r,&r);
                if(r.h.ah & 128) {
                  printf(" send error detected ");
                  exit(1);
                }
          }
```

```
    /* read a charecter from the port*/
  char   receive(port)
    int port;
    {
     union REGS r;/* wait for a charecter*/
     while (!(stat_com(PORT)&256))
       if (kbhit())
       {
        getch();
        exit(1);
      }
   r.x.dx=port;
   r.h.ah=2;
   int86(0x14,&r,&r);
   if(r.h.ah & 128)
   printf("read error in serial port");
   return r.h.al;
   }

   /* check status of serial port  */
int   stat_com(port)
   int port;
   {
   union REGS r;
   r.x.dx=port;
   r.h.ah=3;
   int86(0x14, &r,&r);
   return r.x.ax;
   }

   /* initialise the port  */
   void init(port,code)
   int port;
   unsigned char code;
   {
   union REGS r;
   r.x.dx=port;
   r.h.ah=0;
   r.h.al=code;
   int86(0x14, &r,&r);
   }
```

CHAPTER SEVEN

APPLICATIONS AND SUGGESTIONS FOR FURTHER DEVELOPMENT

APPLICATIONS

Nowadays the use of computers in different fields is so widespread that communication between them is very necessary and should also be an efficient one.

There are much larger computer communication networks now in service and the coordination required for efficient network use is complex. Such networks may have hundreds of terminals and many small processors located at dozens of dispersed sites. These sites, in turn, are linked by different transmission channels to larger host computers. The task of network design is to select and coordinate the network components so that the necessary data is moved to the right place, at the right time, with a minimum of errors, and at the lowest possible cost.

Following are the types of communication processors used by network designers to achieve their goals. They are

1. (i) **REMOTE CONCENTRATOR:**

This reduces the transmission cost by receiving terminal

input from many low speed lines and then concentrating and transmitting a compressed and smooth stream of data on a higher speed and more efficient channel.

(ii)**MESSAGE SWITCHER**

This receives and analyses data messages from points in the network, determines the destination and the proper routing, and then forwards the messages to other network locations.

(iii) **FRONT-END PROCESSOR**

This is usually located at a central computer site. Its purpose is to relieve a main computer i.e., host computer, of a number of the functions required to control and interact with the communication network.

2. **LOCAL AREA NETWORK:**

In the previous case, communication situations, in which data is transmitted between sites far apart, had been discussed. But in many organisations data is also transmitted between computer terminals, word processing stations and other devices that are all located within a compact area such as an office building or a campus. The communication system used to

link these nearby devices together is referred to asd a local network.

3.SYSTEMS SUPPORTED BY DATA COMMUNICATIONS:

(i) REAL TIME PROCESSING SYSTEMS

A real time processing system is in a parallel time relationship with an ongoing activity and is producing information quickly enough to be useful in controlling this current live and dynamic activity. Thus the words 'realtime' describe a direct access or online processing system with severe time limitation. A real time processing requires immediate transaction input from all input originating terminals. Many stations are directly tied by high speed telecommunications lines into one or more CPU's. Several stations can operate at the same time. A few examples of real time systems supported by telecommunications are in the reservation systems used by airlines, hotels etc. Military systems where the computers are used to accept, store and constantly update masses of data from world wide radar installations, and in Air traffic control systems where millions of aircraft flights are tracked across the nation each year by air traffic controllers which are monitored by computers.

(ii)**TIMESHARING AND REMOTE COMPUTING SERVICE SYSTEMS**

Timesharing is a general term used to describe a processing system with a number of independent, relatively low-speed, on line and simultaneously usable stations. Each station provides direct access to the CPU. The use of special programs allows the CPU to switch from one station to another and to do a part of each job in an allocated 'time slice' until the work is completed.

A number of organisations sell time sharing and remote computing services to their customers. These organisations install terminals in customers offices and then use telecommunications channels to link the terminals to their central processors. These organisations generally offer a library of on line application programs to their clients who need only supply the input data and access the programs to obtain the desired information.

(iii) ELECTRONIC MAIL SYSTEMS:

The ability to use telecommunication line to send electronic messages between distant points is not only limited

to personal computer users, but also to send intracompound or intercompany messages.

(iv) BANKING SERVICE SYSTEMS:

Banks communicate with each other and send funds transfer instructions over telecommunications networks. Bank, at home systems also use telecommunication lines to permit people to interact to their individual banks.

(v) DATA BASE RETRIEVAL SYSTEMS:

These computer networks enable people to retrieve information from dozens of data bases, such as weather and traffic updates, article summaries from magazines and news accounts from radio etc.

SUGGESTIONS FOR FURTHER DEVELOPMENT

This modem can be extended to include auto dialling facilities, by using an unit called automatic calling unit (or) a host processor which has been programmed to perform the tasks. The basic steps involved in automatic calling are

1) Generate the off hook command to DAA

2) Detect dial tone from the central office

3) Generate dial pulses

4) Detect remote modem answer

5) Transfer system to data mode.

This modem can be made to operate in a synchronous mode with minor modifications with software. Direct memory access technique can be used to increase speed of data transfer.

CONCLUSION

The Data Communication system has been designed and fabricated to conform to the target specifications.

The hardware circuitry was neatly constructed on a general purpose Board. The necessary interface arrangements in the microcomputer kit were performed. The software for serial data transfer was developed using C-language. The computer print out for serial data transfer of a file has been taken and attached to this report.

Comprehensive details of the Modem 7910, UART (8250) have been presented. The report also features basic communcation principles and major application of this system.

Design is not just a science but an art as well. No design can be a unique solution and for that reason can not be perfectly ideal. What is achieved can at best be a techno-compromise to give an optimum performance. This system is one such.

APPENDIX - A

PIN DESCRIPTION OF AM7910

MC0 - MC4 (Control Inputs)

These five inputs select one of the thirty-two modem configurations according to the Bellor CCITT specifications listed in Table. Only 19 of these 32 modes are actually available to the user.

The most commonly used standards are the C.C.I.T.T. specifications of V21 and V23. They are as shown in Fig.1.2. The V21 has 300 baud originate and answer. The V23 mode has 1200/75 baud for the originate and answer.

Data Terminal ready: $\overline{DTR}$:

A low on this input indicates the data terminal i.e., the micro computer desires to send and/or receive data via the modem.

REQUEST TO SEND: $\overline{RTS}$

A low on this input instructs the modem to enter the trasmit mode. This input must remain low for the full duration of data transmission. The signal has no effect of DATA TERMINAL READY.

CLEAR TO SEND: CTS

This output goes low at the end of a delay initiated when **RTS** goes low. This indicates the modem is ready to begin transmission.

CARRIER DETECT: CD

A low on this output indicates that a valid carrier is present at the receiver. This pin remains high, when **DTR** is high.

TRANSMITTER DATA: TD

Data bits to be transmitted are presented on this input serially. High corresponds to logic 1 and low corresponds to logic 0. This data determines which frequency appears at any instant at the TRANSMITTED CARRIER output pin. No signal appears at the TRANSMITTED CARRIER output, unless **DTR** is low, and **RTS** is low.

RECEIVED DATA: RD

Data bits demodulated from the RECEIVED CARRIER input are available serially at this output. High indicates logic 1 and low indicates logic 0.

BACK REQUEST TO SEND : BRTS

The back channel input signal is equivalent to **RTS** for the main channel. It is required in the V23 mode (half-duplex) only.

BACK CLEAR TO SEND: BCTS

Back CARRIER DETECT **BCD, BACK RECEIVED DATA BRD,** are similar to those signals of the main channel.

TRANSMITTED CARRIER:

This analog is the modulated carrier to be conditioned and sent over the phone line.

RECEIVED CARRIER:

This input is the analog signal received from the phone line. The modem extracts the information contained in the modulated carrier and converts it into a serial data stream for presentation at the RECEIVED DATA output.

RING

ThThe input signal permits auto-answer capability by responding to a ringing signal from the DAA. If a ringing signal is detected (**RING** low) and **DTR** is low, the modem begins a sequence to generate an answer tone at the TC output.

XTAL1, XTAL2:

Master timing of the modem is provided by either a crystal connected these two inputs or an external clock inserted into XTAL1. The value of the crystal or the external clock frequency must be 2.4576 MHz+01%

Vcc : +5 volts power supply (+5%)

Vbb : -5 volt power supply (+5%)

RESET :

This input signal is for a reset circuit which operates in either of two modes. The AM7910 should be reset upon initial application of power.

The details of the various pins are shown in Fig.A.1

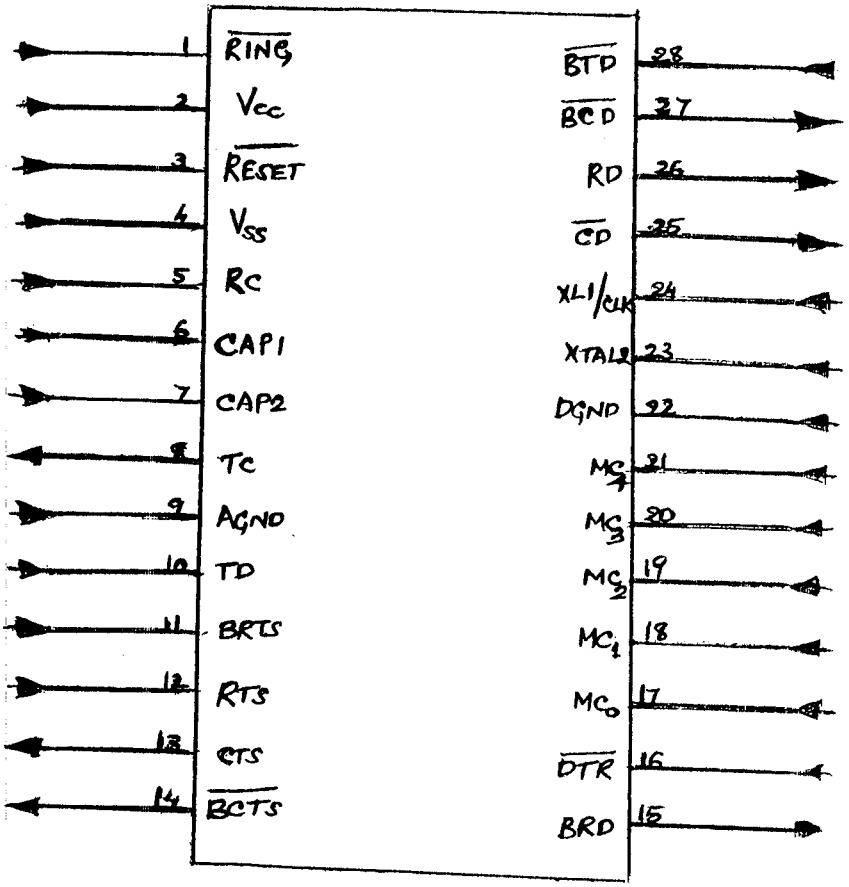


FIG A-1: AM 7910

PIN DETAILS

LM78XX Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

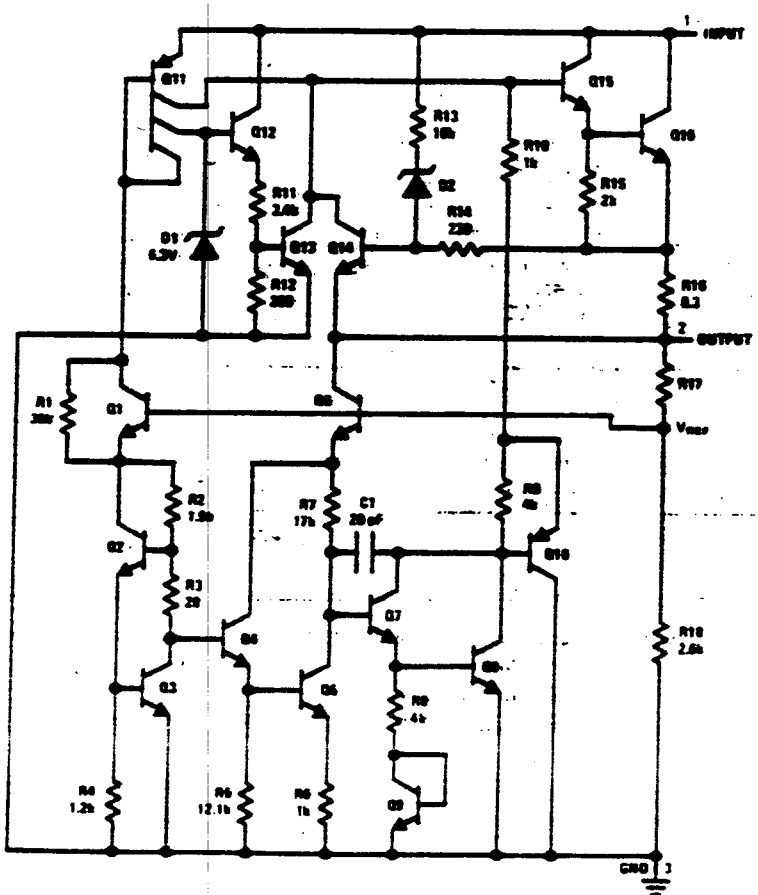
Features

- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

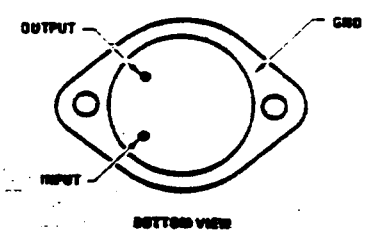
Voltage Range

LM7805C	5V
LM7812C	12V
LM7815C	15V

Schematic and Connection Diagrams



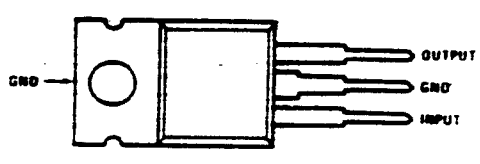
Metal Can Package
TO-3 (K)
Aluminum



BOTTOM VIEW

Order Numbers
LM7805CK
LM7812CK
LM7815CK
See Package KC02A

Plastic Package
TO-220 (T)



TOP VIEW

Order Numbers:
LM7805CT
LM7812CT
LM7815CT
See Package T03B

Input Voltage (V_O = 5V, 12V and 15V)	35V
Internal Power Dissipation (Note 1)	Internally Limited
Operating Temperature Range (T_A)	0°C to +70°C
Maximum Junction Temperature	
(K Package)	150°C
(T Package)	125°C
Storage Temperature Range	−65°C to +150°C
Lead Temperature (Soldering, 10 seconds)	
TO-3 Package K	300°C
TO-220 Package T	230°C

Electrical Characteristics LM78XXC (Note 2) 0°C ≤ Tj ≤ 125°C unless otherwise noted.

OUTPUT VOLTAGE				5V			12V			15V			
INPUT VOLTAGE (unless otherwise noted)				10V			19V			23V			
PARAMETER		CONDITIONS		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
V_O	Output Voltage	T_j = 25°C, 5 mA ≤ I_O ≤ 1A		4.8	5	5.2	11.5	12	12.5	14.4	15	15.6	
		P_D ≤ 15W, 5 mA ≤ I_O ≤ 1A, V_{MIN} ≤ V_{IN} ≤ V_{MAX}		4.75 (7 ≤ V_{IN} ≤ 20)		5.25	11.4 (14.5 ≤ V_{IN} ≤ 27)		12.6	14.25 (17.5 ≤ V_{IN} ≤ 30)		15.75	
ΔV_O	Line Regulation	I_O = 500 mA	T_j = 25°C, ΔV_{IN}		3 (7 ≤ V_{IN} ≤ 25)	50		4 (14.5 ≤ V_{IN} ≤ 30)	120		4 (17.5 ≤ V_{IN} ≤ 30)	150	
			0°C ≤ T_j ≤ +125°C, ΔV_{IN}		(8 ≤ V_{IN} ≤ 20)	50		(15 ≤ V_{IN} ≤ 27)	120		(18.5 ≤ V_{IN} ≤ 30)	150	
		I_O ≤ 1A	T_j = 25°C, ΔV_{IN}		(7.3 ≤ V_{IN} ≤ 20)	50		(14.6 ≤ V_{IN} ≤ 27)	120		(17.7 ≤ V_{IN} ≤ 30)	150	
			0° ≤ T_j ≤ +125°C, ΔV_{IN}		(8 ≤ IN ≤ 12)	25		(16 ≤ V_{IN} ≤ 22)	60		(20 ≤ V_{IN} ≤ 26)	75	
ΔV_O	Load Regulation	T_j = 25°C	5 mA ≤ I_O ≤ 1.5A		10	50		12	120		12	150	
			250-mA ≤ I_O ≤ 750-mA			25			60			75	
		5 mA ≤ I_O ≤ 1A, 0°C ≤ T_j ≤ +125°C				50			120			150	
I_Q	Quiescent Current	I_O ≤ 1A	T_j = 25°C			8			8			8	
			0°C ≤ T_j ≤ +125°C			8.5			8.5			8.5	
ΔI_Q	Quiescent Current Change	5 mA ≤ I_O ≤ 1A				0.5			0.5			0.5	
		T_j = 25°C, I_O ≤ 1A, V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(7.5 ≤ V_{IN} ≤ 20)	1.0		(14.8 ≤ V_{IN} ≤ 27)	1.0		(17.9 ≤ V_{IN} ≤ 30)	1.0	
		I_O ≤ 500 mA, 0°C ≤ T_j ≤ +125°C, V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(7 ≤ V_{IN} ≤ 25)	1.0		(14.5 ≤ V_{IN} ≤ 30)	1.0		(17.5 ≤ V_{IN} ≤ 30)	1.0	
V_N	Output Noise Voltage	T_A = 25°C, 10 Hz ≤ f ≤ 100 kHz			40			75			90		
$\frac{\Delta V_{IN}}{\Delta V_{OUT}}$	Ripple Rejection	f = 120 Hz	I_O ≤ 1A, T_j = 25°C or	62	80		55	72		54	70		dB
			I_O ≤ 500 mA, 0°C ≤ T_j ≤ +125°C	62			55			54			dB
		V_{MIN} ≤ V_{IN} ≤ V_{MAX}			(8 ≤ V_{IN} ≤ 18)			(15 ≤ V_{IN} ≤ 25)			(18.5 ≤ V_{IN} ≤ 28.5)		
	Dropout Voltage	T_j = 25°C, I_{OUT} = 1A			2.0			2.0			2.0		
R_O	Output Resistance	f = 1 kHz			8			18			19		mΩ
	Short-Circuit Current	T_j = 25°C			2.1			1.5			1.2		
	Peak Output Current	T_j = 25°C			2.4			2.4			2.4		
	Average TC of V_{OUT}	0°C ≤ T_j ≤ +125°C, I_O = 5 mA			0.6			1.5			1.8		mV/°C
V_{IN}	Input Voltage Required to Maintain Line Regulation	T_j = 25°C, I_O ≤ 1A		7.3			14.6			17.7			

NOTE 1: Thermal resistance of the TO-3 package (K, KC) is typically 4°C/W junction to case and 35°C/W case to ambient. Thermal resistance of the TO-220 package (T) is typically 4°C/W junction to case and 50°C/W case to ambient.

NOTE 2: All characteristics are measured with capacitor across the inut of 0.22 µF, and a capacitor across the output of 0.1 µF. All characteristics except noise voltage and ripple rejection ratio are measured using pulse techniques (t_W ≤ 10 ms, duty cycle ≤ 5%). Output voltage changes due to changes in internal temperature must be taken into account separately.

TABLE 12.1 RS-232 Interface

Sl.No.	Pin No.	Signal	Signal Name	Source	Destination
1.	1	—	Frame Ground	—	—
2.	2	TXD	Transmit Data	DTE	DCE
3.	3	RXD	Receive Data	DCE	DTE
4.	4	RTS	Request to Send	DTE	DCE
5.	5	CTS	Clear to Send	DCE	DTE
6.	6	DSR	Data Set Ready	DCE	DTE
7.	7	SG	Signal Ground	—	—
8.	8	RLSD or CD	Received Line Signal Detect (Carrier Detect)	DCE	DTE
9.	20	DTR	Data Terminal Ready	DTE	DCE
10.	22	RI	Ring Indicator	DCE	DTE

called modem issues the RI Signal to its DTE, and sends an answer tone for 2 s to the calling modem. Then the calling modem sends an 8 ms duration tone on the telephone line. Now the called modem issues CD (Carrier Detect Signal) to its DTE. The CD is an indication to the DTE that it will soon be receiving the data sent by the other end DTE.

12.3 SERIAL PORT IN PC

The RS-232 standard interface is implemented in a wide variety of computer related equipment, such as terminal, printer, mouse, optical scanner, bar code reader, voice synthesiser, OMR (Optical Mark Reader), OCR (Opitical Character Reader), process control systems, etc. These equipment are linked to the computer through modems when they are in a remote area. If the distance is considerably small, the terminal equipment can be directly connected to the computer provided appropriate modifications are done in the cable. Since both the computer and the terminal are DTEs they have identical modem interface. As each end expects a modem (DCE) to be connected to it, the interconnections should cheat both the ends, as if each of them is linked to a modem. Such an interconnecting cable is known as a null modem and is shown in Fig. 12.7.

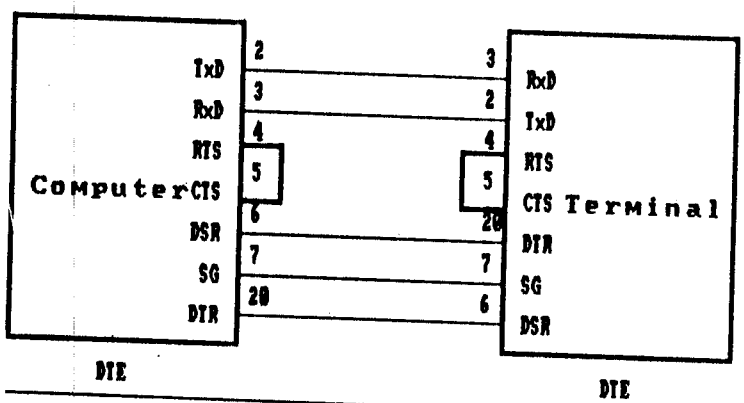


FIG. 12. 7 A Null Modem Linking Two DTEs

In the RS-232C interface, the handshake control signals, such as RTS, CTS, DTR, etc., follow positive logic whereas the data bits follow negative logic.

12.3.1 UART

The Universal Asynchronous Receiver Transmitter (UART) is usually a programmable LSI device having necessary hardware circuits for implementing asynchronous serial communication. Figure 12.8 shows the essential components of the receiver section and transmitter section in an UART. The receiver converts serial data bits received on the line into parallel bytes. The transmitter converts the parallel bytes into serial bits to be sent on the line. The frequency of the receiver clock has to match the baud rate of the receive data (RXD). The SIPO (Serial In Parallel Out) logic deserialises the serial data bits into a parallel byte. The RPE (Receiver Parity Error) signal is generated by the parity checker if there is a wrong parity in the received character. In the transmitter section, the PISO (Parallel In Serial Out) logic converts the parallel byte into a serial bit stream. It also adds start bit, parity bit and stop bit to the data.

The frequencies of the transmit clock and the receive clock need not be equal. But the baud rate of the sending end transmitter should be equal to the baud rate of the receiving end receiver. If the baud rates are not equal the receiver section will generate an RFE (Receiver Framing Error) signal. When the receiver section finds an invalid data format it issues the RFE signal. The received data is said to be invalid when

(a) The start bit is sensed but no stop bit is found after the data bits and the parity bit time.

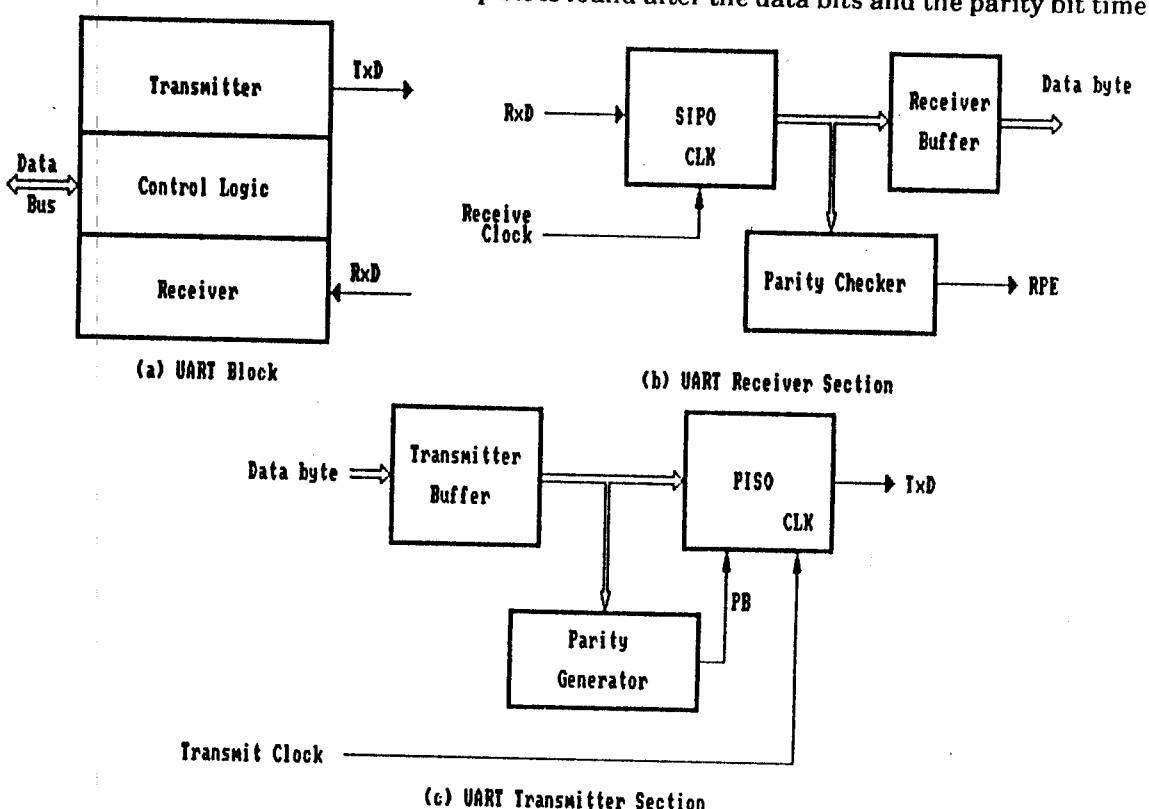


FIG. 12.8 UART Block Diagram

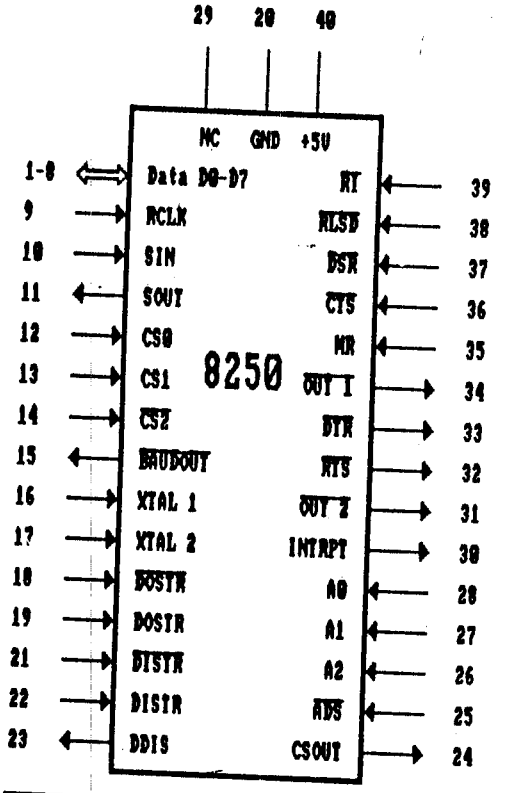


FIG. 12.9 8250 Pinout Diagram

(b) The start bit is sensed but its duration is less than a baud period.

If the receiver clock frequency and the data format are proper and still the RFE is generated, it is obvious that there is either noise on the line, or a fault in the line or receiver circuits.

The UART samples the line condition at a fixed frequency which is 16 times the baud rate. To satisfy this, the clock inputs to the UART should be 16 times the desired baud rate.

12.3.2 National Semiconductor 8250 UART

The 8250 UART is the programmable LSI used in asynchronous serial port in the PC. Figure 12.9 gives the pin details of 8250 UART. Table 12.2 gives the pin description of 8250 UART. The major features of 8250 are briefly outlined below:

1. The 8250 is a programmable asynchronous communication controller. The communication parameters and formats are specified by the software.
2. The 8250 provides modem control and status signals.
3. The Baud rate is programmable.
4. Generates standard baud rates upto 9600 bauds.
5. The number of data bits per character is programmable.
6. The parity type (Odd, Even or No parity) is programmable.
7. The number of stop bits is programmable.
8. The 8250 has seven I/O ports and ten internal registers.
9. The 8250 provides double buffering of data both in the transmitter section and in the receiver section.
10. Data transfer between the 8250 and the system can be performed either in the programmed mode or in the interrrupt mode.
11. The 8250 provides separate clock inputs for receiver and transmitter sections thus providing an option of fixing different baud rates for the transmitter and receiver section.
12. The 8250 can detect a false start bit.
13. The 8250 provides line-break generation and detection.
14. The 8250 provides a loopback feature which is used by diagnostic software.
15. The 8250 has multiple pins for each of the following functions: Chip Select, Read (Data Input Strobe) and Write (Data output strobe)
16. The 8250 has an internal clock oscillator circuit facilitating direct connection of a crystal.
17. The 8250 provides external status signals to indicate when the chip has been selected and when the CPU is reading data from it.
18. The 8250 can be programmed to raise interrupts for four different reasons.

TABLE 12.2 8250 Pinout

Sl.No.	Pin No.	Pin Name	Type	Remarks
1.	1/8	D0/D7	I/O	Bidirectional data bus to CPU; tri-state pins
2.	9	RCLK (Receiver Clock)	I	Clock input to the receiver section; 16 × baud rate clock
3.	10	SIN (Serial Input)	I	Serial data input to the receiver section
4.	11	SOUT (Serial Output)	O	Serial data output from the transmitter section
5.	12	CS0	I	Three chip select inputs; the chip is selected only when all these three signals are active
6.	13	CS1	I	
7.	14	$\overline{CS2}$	I	
8.	15	BAUDOUT	O	16 × clock signal to the transmitter section; may be connected to the RCLK input pin (9)
9.	16	XTAL 1	I	Crystal is connected to these pins or an external clock can be connected to XTAL 1
10.	17	XTAL 2	I	
11.	18	$\overline{DOSTR}$	I	Write control inputs; when either of these pins is active and the chip is selected, the data sent by the CPU, D0/D7 is entered into 8250
12.	19	DOSTR (Data Output Strobe)	I	
13.	20	GND	—	Ground Reference
14.	21	$\overline{DISTR}$	I	Read control inputs. When either of these pins is active and the chip is selected, the 8250 outputs data on the data pins D0/D7.
15.	22	DISTR (Data Input Strobe)	I	
16.	23	DDIS (Drive Disable)	O	Data bus buffer control; goes low whenever the CPU is reading the 8250
17.	24	CSOUT (Chip Select Out)	O	Data bus buffer control; goes high when the chip has been selected
18.	25	$\overline{ADS}$ (Address Strobe)	I	May be used as the latch control input to latch the chip select (CS0, CS1, $\overline{CS2}$) and register select (A0, A1, A2) signals
19.	26	A2	I	These three inputs indicate the port (register number) addressed by the CPU during read/write operation. The DLAB (Divisor Latch Access Bit) in the line control register is used with the A2, A1 and A0 for selecting the required port register by the CPU
20.	27	A1	I	
21.	28	A0	I	
22.	29	NC	—	No connection

TABLE 12.2 8250 Pinout *(Contd.)*

Sl.No.	Pin No.	Pin Name	Type	Remarks
23.	30	INTRPT	O	Interrupt Signal; becomes active for four different conditions if interrupt is enabled by IER (Interrupt Enable Register). The four causes are (a) Receiver Error (b) Receiver Data Available (c) Transmitter Holding Register Empty (d) Modem Status
24.	31	$\overline{OUT2}$	O	Programmable output by bit 3 in the modem Control Register
25.	34	$\overline{OUT1}$	O	Programmable output by bit 2 in the modem Control Register
26.	35	MR	I	Master Reset; when active, clears the internal registers except the receiver buffer, transmitter holding register and the divisor latch register
27.	36	$\overline{CTS}$ (Clear to Send)	I	Control input from modem; enters bit 4 (CTS) of the modem Status Register
28.	37	$\overline{DSR}$ (Data Set Ready)	I	Modem Status input; enters bit D5 (DSR) of the modem Status Register
29.	38	$\overline{RLSD}$	I	Received Line Signal Detect; Indicates that the modem has detected the data carrier. Available in bit 7 (RLSD) of the modem Status Register.
30.	39	$\overline{RI}$	I	Ring Indicator; indicates that the modem has received a ringing signal; available as bit 6 (RI) in modem Status Register
31.	40	Vcc	-	+5V dc supply
32.	32	$\overline{RTS}$	O	Request to send; informs the modem that the 8250 wants to transmit data; programmable by bit 1 in modem Control Register
33.	33	$\overline{DTR}$	O	Data Terminal Ready; informs the modem that the 8250 is ready for communication; programmable by bit 0 (DTR) in the modem Control Register

Internal Registers

The 8250 has 10 registers which are accessed by the CPU as I/O ports. The three address bits (A2, A1 and A0) along with DLAB bit are used to select one of the registers. Table 12.3 lists the registers and the addressing scheme.

Line Control Register (LCR)

The program loads the desired character format in this register. The contents of the LCR can be read back by the program at any time. Figure 12.10 illustrates the format of the LCR. The different bits in LCR are defined in Table 12.4.

Line Status Register (LSR)

This register is used by the receiver and the transmitter sections in the 8250 to indicate various status and error conditions related to data transfer. Figure 12.11 illustrates the format of the LSR. The different bits in the LSR are defined in Table 12.5.

TABLE 12.3 Internal Registers

Sl. No.	DLAB	A2	A1	A0	Register	Remark
1.	0	0	0	0	Receiver Buffer (RB) or Transmitter Holding Register (THR)	During the input operation by CPU, the receiver data buffer (register) is accessed. During the output operation, the transmitter holding register is accessed
2.	0	0	0	1	Interrupt Enable Register (IER)	Output register, four bits mask for four different interrupts.
3.	X	0	1	0	Interrupt Identification	Input register; indicates two types of information: (a) Interrupt Pending Status (b) Interrupt level which has been given priority
4.	X	0	1	1	Line Control	Input/Output register; format of an asynchronous character is stored by the program
5.	X	1	0	0	Modem Control	Controls the output signals to the modem; also provides loopback testing
6.	X	1	0	1	Line Status	Indicates different status conditions in the receiver and transmitter sections of the 8250
7.	X	1	1	0	Modem Status	Indicates the status of signals from the modem
8.	X	1	1	1	—	—
9.	1	0	0	0	Baud rate divisor (L Byte)	The program issues two bytes to these registers
10.	1	0	0	1	Baud rate divisor (H Byte)	The 8250 divides the clock input by this 16 bit binary number to obtain 16 x baud rate clock

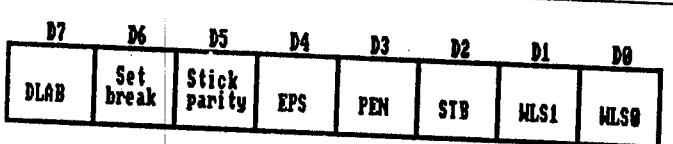


FIG. 12.10 LCR Format

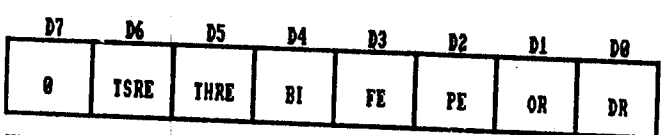


FIG. 12.11 LSR Format

TABLE 12.4 Line Control Register

Bit	Name	Function
D7	DLAB	Divisor latch access bit; set to 1 by program before accessing baud rate divisor registers; set to 0 by program to access the RB, THR and IER
D6	Set Break	When this bit is 1, the 8250 forces the serial output (data) line to the spacing level (logical 0)
D5	Stick Parity	This bit decides the polarity of the parity bit. It is dependent on the D4 (EPS) bit also. If D5 is 1 and D4 is 1, then the parity bit follows negative logic. If D5 is 1 and D4 is 0, then positive logic is followed for the parity bit.
D4	EPS	Even Parity Select. If D4 is 1, even parity is followed. If D4 is 0, odd parity is followed.
D3	PEN	Parity Enable bit. When this bit is 1, the 8250 enables parity bit; when this bit is 0, the parity bit is suppressed.
D2	STB	Number of stop bits. If D2 is 0, one stop bit; if D2 is 1, 1.5 stop bits for 5 data bits and 2 stop bits for more than 5 data bits.
D1 D0	WLS1 WLS0	Word length; these bits define the number of bits in the character. The combinations are as follows:

WLS1	WLS0	Word Length
0	0	5 bits
0	1	6 bits
1	0	7 bits
1	1	8 bits

TABLE 12.5 Line Status Register

Bit	Name	Function
D7	-	Not used; always 0
D6	TSRE	Transmitter Shift Register Empty; when 1 it indicates that the TSR has completed previous character and is idle. When a character is transferred from the THR to TSR, the TSRE bit becomes 0.
D5	THRE	Transmitter Holding Register Empty; when 1 it indicates that the THR is empty and the 8250 is ready to accept next character for transmission. If THRE interrupt is enabled, an interrupt is raised by the 8250 under this condition. When a character is loaded in the THR by the CPU, the THRE bit becomes 0. When the THR is transferred to TSR, the THRE bit becomes 1.
D4	BI	Break Interrupt; when the received data line is in space condition (logical 0) for more than one character duration, the BI bit is made 1 by the 8250. This signifies that a break character has been received on the line as an indirect message asking for attention from the other end.
D3	FE	This bit is set when there is a framing error.
D2	PE	Parity Error; this bit is set when there is parity error in the character received.
D1	OR	Overrun error; this indicates that data (previous byte) in the RBR has not been read by the CPU but the next byte has entered the RBR; previous byte is destroyed.
D0	DR	Data Ready; the DR bit is set when a character is received and stored in the RBR.

When the CPU reads the LSR, the OR and PE bits are reset. The DR bit is reset when the CPU reads the RBR or when the CPU writes 0 into this bit. The BI, FE, PE and OR bits also raise appropriate interrupts.

Interrupt Enable Register (IER)

The IER bits enable different interrupt types. It is used as the interrupt mask register. The format of IER is illustrated in Fig. 12.12 and defined in Table 12.6.

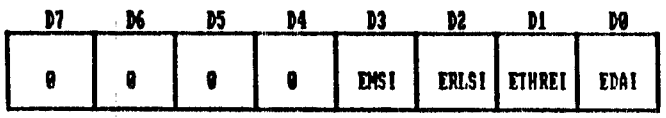


FIG. 12.12 IER Format

TABLE 12.6 Interrupt Enable Register (IER)

Bit	Name	Function
D3	EMSI	Enable Modem Status Interrupt
D2	ERLSI	Enable Receive Line Status Interrupt
D1	ETHREI	Enable THR Empty Interrupt
D0	EDAI	Enable Data Available Interrupt

Modem Control Register (MCR)

The MCR controls the modem or data set. Figure 12.13 illustrates the format of the MCR. The different bits in the MCR are defined in Table 12.7.

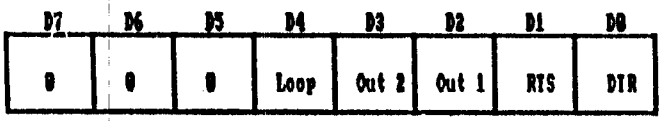


FIG. 12.13 MCR Format

TABLE 12.7 Modem Control Register

Bit	Name	Function
D4	LOOP	This bit provides for an internal loopback of transmitter section output into the receiver section input by logically disconnecting the line interface signals. This loopback feature is used to verify the transmit and receive logics in the 8250.
D3	OUT2	When this bit is 1, the $\overline{OUT2}$ pin of 8250 becomes low and when this bit is 0, the $\overline{OUT2}$ pin becomes high.
D2	OUT1	The $\overline{OUT1}$ pin of 8250 is made low or high depending on whether this bit is 1 or 0 respectively.
D1	RTS	Request To Send; the $\overline{RTS}$ pin of 8250 is made low or high depending on whether this bit is 1 or 0 respectively.
D0	DTR	Data Terminal Ready; the $\overline{DTR}$ pin of 8250 is made low or high depending on whether this bit is 1 or 0 respectively.

Modem Status Register (MSR)

The MSR stores the current state of the modem status signals. Figure 12.14 illustrates the format of the MSR. The different bits in the MSR are defined in Table 12.8.

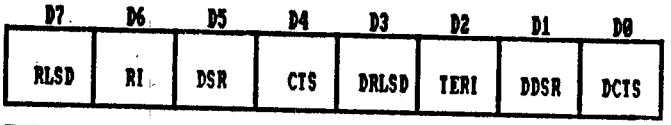


FIG. 12.14 MSR Format

TABLE 12.8 Modem Status Register

Bit	Name	Function
D7	RLSD	Received Line Signal Detect; this bit is the complement of the $\overline{RLSD}$ pin of 8250 except during Loopback.
D6	RI	Ring Indicator; this bit is the complement of the $\overline{RI}$ pin of 8250 except during Loopback.
D5	DSR	Data Set Ready; this bit is the complement of the $\overline{DSR}$ pin of 8250 except during Loopback.
D4	CTS	Clear to Send; this bit is the complement of the $\overline{CTS}$ pin of 8250 except during Loopback.
D3	DRLSD	Delta Received Line Signal Detector; this bit indicates that the $\overline{RLSD}$ pin of 8250 has changed state.
D2	TERI	Trailing Edge of Ring Indicator; this bit indicates that the $\overline{RI}$ pin of 8250 has changed from 1 to 0.
D1	DDSR	Delta Data Set Ready; this bit indicates that the $\overline{DSR}$ pin of 8250 has changed state after it was read by the CPU the last time.
D0	DCTS	Delta Clear To Send; this bit indicates that the $\overline{CTS}$ pin of 8250 has changed state after it was read by the CPU the last time.

Interrupt Identification Register (IIR)

The IIR stores the status of the internal priority interrupt controller in the 8250. This register identifies the type of interrupt given priority at the moment and the status of further pending interrupts. Figure 12.15 illustrates the format of the IIR. The different bits in the IIR are defined in Table 12.9.

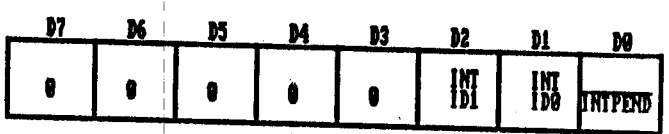


FIG. 12.15 IIR Format

12.3.3 Programming Considerations

The BIOS contains routines for performing data transfer with the 8250 in interrupt mode. These can be called by a software interrupt, i.e., INT14H. The calling program supplies the line parameters to the BIOS. These parameters are: baud rate, word length (number of data bits), number of stop bits, and parity bit desired. On the other hand, the program can also directly control the serial port by bypassing the BIOS. The sequence of actions to be performed by the BIOS are outlined below.

TABLE 12.9 Interrupt Identification Register

Bit	Name	Function
D2	INT ID1	These two bits contain encoded
D1	INT ID0	information about the interrupt which has been given priority. The four different combinations are given below:

D2	D1	Interrupt Type
1	1	Receiver Line Status
1	0	Receiver Data Available
0	1	THR Empty
0	0	Modem Status

D0	$\overline{\text{INT PEND}}$	When an interrupt is pending, the bit is 0. Otherwise, this bit is 1 and the D1 and D2 bits are irrelevant.

1. Load the desired pattern in the line control register (Hexa 3FB) with OUT instruction to fix the character format.

2. Load a 16 bit binary number in the two baud rate divisor registers (Hexa 3F8 and 3F9) by two OUT instructions. This is related to the desired baud rate as follows:

16 × Baud Rate = Clock frequency input/baud rate divisor

3. By OUT instruction to the MCR (Hexa 3FC), DTR bit is set.

4. If the program wants to perform data transfer in interrupt mode, the IER should be loaded by an OUT instruction on port Hexa 3F9 with appropriate pattern in bits 0/3.

5. By an IN instruction on LSR (Hexa 3FD), the status of the receiver and transmitter sections can be read and analysed as a preparation for the actual data transfer.

6. If the modem or data set is connected to the serial port, the program should observe the modem protocol by reading the MSR (Hexa 3FE) and analysing the signals from the modem. The control signals to the modem are issued by the program by an OUT instruction to the MCR (Hexa 3FC).

7. During transmit operation, the program waits for the THRE bit in the LSR. Once this bit becomes high, the program can issue one character of data to the Transmit Data Register (TDR) (Hexa 3F8). Before sending the very first character, the program should also check for the TSRE bit being high. For interrupt mode of data transfer, the program can enable THRE interrupt in the IER. If the program outputs a character to the TDR when the THRE bit is 0, this character is not accepted by the 8250 and is lost. On the other hand, there is no urgency in outputting data to the TDR when the THRE is 1 since gaps are allowed between two characters.

8. During receive operation the program should read data from the Receive Data Register (RDR) once the DR bit in the LSR is high. If the OR bit is also high it implies that one byte received from the line has already been lost due to sluggishness on the part of the program in matching the higher rate of incoming data.

9. The program should also verify that the PE and FE bits in the LSR are zero. Otherwise there is no use of reading the RDR.

10. If the BI bit in the LSR is high, it is an indication that the other (receiving) end wants to interrupt this end. Hence the program should stop its transmission and go to receive mode. Similarly when the program senses errors in the received data, it can send a break character to the other end by making set break bit in the LCR.

BIBLIOGRAPHY

1. Douglas V.Hall, "MICROPROCESSOR INTERFACING programming and Hardware" McGraw-Hill Book Company, 1986.

2. Robert Jourdain, "Programmer's Problem Solver for the IBM PC, XT & AT" - BPB Publications , 1990.

3. James Martin, "Telecommunications And Computer" Prentice Hall, Inc., Eagle-Wood Cliffs, New Jersey.

4. Donald H.Sanders, "Computers Today", McGraw Hill International Book Co., Asian Students Edition, Singapore.

5. William Sinnema, "Digital, Analog and Data Communication" Restron Publishing Company, Prentice-Hall Company, Restron, Virginia.

6. Elector Electronics India - Journal October-'84.

7. Electronics for you - Journal April-'91

8. John G.Stephenson and Bob Cahill, "How to Maintain And Service Your Small Computer" - BPB Publications - 1987.

9. Product Specification - Advanced Micro Devices.

10. National Semi Conductor Manual.