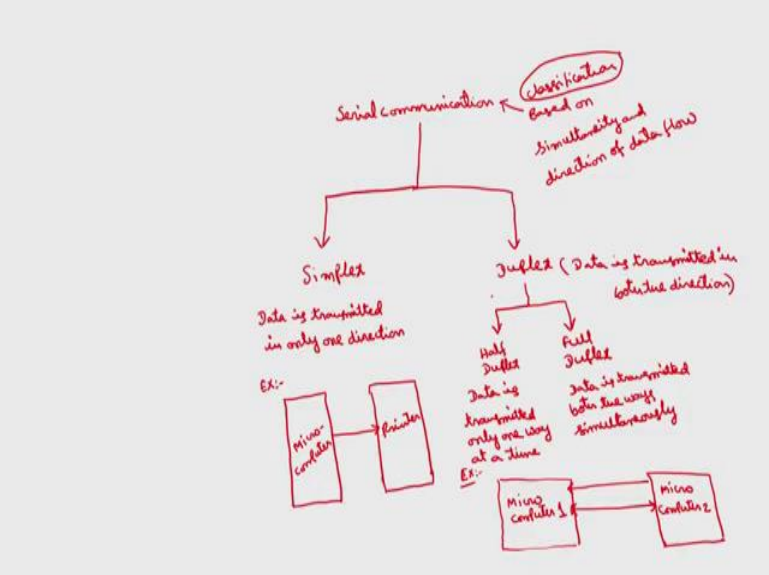
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture – 33
Basic Concepts of serial I/O contd…

(Refer Slide Time: 00:43)



The second way to classify serial communication is based on the direction and simultaneity of data storage. This is a synchronous serial communication. This is based on simultaneity and direction of data flow. The classification based on simultaneity and direction of data flow. So, we have two types of serial communications, one is called simplex and another is called duplex. In case of a simplex, the data transmission will be takes place in only one way.

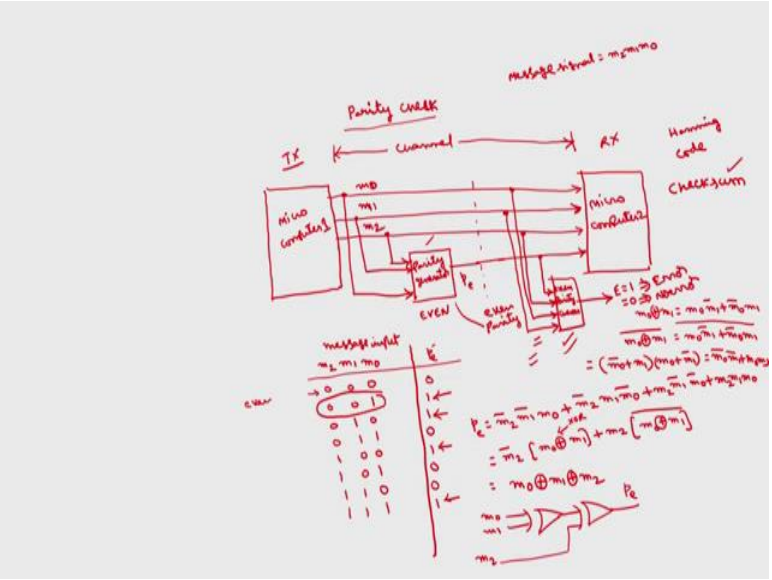
Data is transmitted in only one direction. So, what is the example of the simplex type of a data transfer? So, example is from microcomputer to solid time I am calling as microcomputer means, the CPU of the micro computer is microprocessor. So, this is basically microprocessor based computer. So, if I want to transmit the printer, this will takes place through simplex because only one direction. Printer cannot transfer the data to the microcomputer, but microcomputer only can transmit the data to printer. The second one is duplex. Duplex means data will be transmitted in both the directions and again in both the direction if simultaneous data transfer takes place or at any time only one way is

a enable depends upon that again we have two types of the duplex. One is called half duplex and full duplex. In case of half duplex the data will be transmitted in only one way at a time. Whereas, here data is transmitted both the ways simultaneously.

So, the example of both half duplex as well as full duplex is computer to computer communication. This is micro computer 1 and this is another microcomputer 2. So, we are going to transmit through serial line, this is serial line in case of half duplex only one single line is enough. So, this can have the microcomputer 1 can transmit the data to microcomputer 2 or 2 can transmit to 1, but at anytime only one way will be enabled. So, in case of full duplex there will be two communication one is from computer 1 to computer 2 second one is from computer 2 to computer 1. So, there will be two vibes this is example of full duplex.

So, you know these different classifications of a serial communications. So, one classification is based on synchronous or asynchronous where the clock is common or clocks are different. And second classification is based on the direction and simultaneity of data flow. So, the next topic that we have to discuss before going for the programmable communication interface is parity.

(Refer Slide Time: 06:32)



So, normally in data communication, what happens is if you want to transmit the data from one computer to another computer. This is microcomputer 1 to microcomputer 2. Suppose,

if I want to transmit 3 bits of the data in half duplex. So, there will be 3 lines this will transmit one bit this will transmit another bit this will transmit another bit.

So, we can have total 8 combinations of the bits that can be transmitted from computer 1 to computer 2. Now during the transmission, the errors maybe encountered ok. So, in such cases how to detect and how to correct the errors at the receiver. This will acts as now transmitter this will acts as receiver and this will be the channel. This is channel here its a telephone lines, I will discuss how the data can be transmitted over telephone lines from one computer to another computer the channel is wired line ok.

So, we know that the in any communication system the common phenomena is in the channel there will be some due to noise there will be some distortion or there will be some errors ok, but in digital communication the error will change bit 0 to bit 1 or bit 1 to bit 0 ok. Now the question is how to detect and correct the errors at the receiver. For that, we have to add an extra bit at the transmitter which is called parity bit, such that the overall information can have either even parity or odd parity. Suppose, if we have a 3 message signals say m 1, m 2 or m 0 m 1 m 2. So, what I will do is at the transmitter instead of transmitting this message signal this m 2, m 1, m 0 is message signal message signal is this is actual the information signal means m 2, m 1, m 0 m 2 is msb m 0 is lsb.

So, in addition to these 3 bits, I will add one more bit which is called as parity bit. So, this is parity generator here I have to go design a parity generator whose inputs are these 3 bits message bits and there will be one output which is called as parity bit P. So, we can design this parity generator for either even parity or odd parity. Suppose, I want to design for even parity even parity generator I will call this bit as P e. If you want to design for odd parity I will called as P o. So, how to design this basically this parity generator, you might have studied in your digital electronics course. So, I will just briefly explain the design of this parity generator.

So, if we have m 2, m 1, m 0 is message input to parity generator, then output of these parity generator will be P e if I want to generate even parity. So, the logic here is so, whatever the combination of this m 2, m 1, m 0 I have to generate P e such that these 4 bits this 4 bits along with P e should have even parity. Say a number is said to be having even parity, if it contains even number of 1s. So, if I want to transmit 0 0 0 information what is the actual parity in the message bits. So, how many number of 1s are there 0 number

of ones. So, 0 is even number this 0 0 0 is already having even parity. So, I will choose this parity bit as 0 such that this 4 bits will be having now even parity.

The second combination is 0 0 1 if you want to transmit without this P e, this is having odd parity if I want to make this as even parity I have to make this as 1. Similarly, 0 1 0 odd parity is there you have to make 1 0 1 1 even parity you have to make 0 1 0 0 odd parity means you have to add 1, 1 0 1 even parity 0, 1 1 0 is also even parity 0, 1 1 1 is odd parity you have to add 1

The expression for P e will be now the Boolean expression, wherever this 1 is there you did the corresponding mean term. This is m 2 is 0 means m 2 bar, m 1 bar, m 0 plus this one m 2 bar, m 1 m 0 bar plus this one m 2, m 1 bar, m 0 bar plus this one is.

So, if I simplify this we can put in the form of this m 2 bar is common in the first two terms. So, the remaining term is m 0 exclusive or with m 1 and between these 2 if I take m 2 as common this is exclusive naught which is compliment of exclusive or m 0 m 1 exclusive or this is exclusive or symbol the xor symbol.

So, this will be now whole bar because we know that m 0exclusive or m 1 is m 0 m 1 bar plus m 0 bar m 1. If you take the compliment of this one m 0 compliment of m 1 bar this is equal to m 0 m 1 bar plus m 0 bar m 1 whole compliment.

According to Demorgan's law this is equal to m 0 bar plus m 1 into m 0 plus m 1 bar. So, m 0 bar become 0. So, we will get this is equal to m naught m 1 bar plus m naught m 1. So, this is again exclusive or between this and exclusive or operation will be having both associative property and the distributive property. So, we can write this one as m naught exclusive or m 1 exclusive or m 2. No bracket is required because of the associative and distributive property of exclusive or gate. So, the circuitry inside this even parity generator is this basically 3 exclusive or gates.

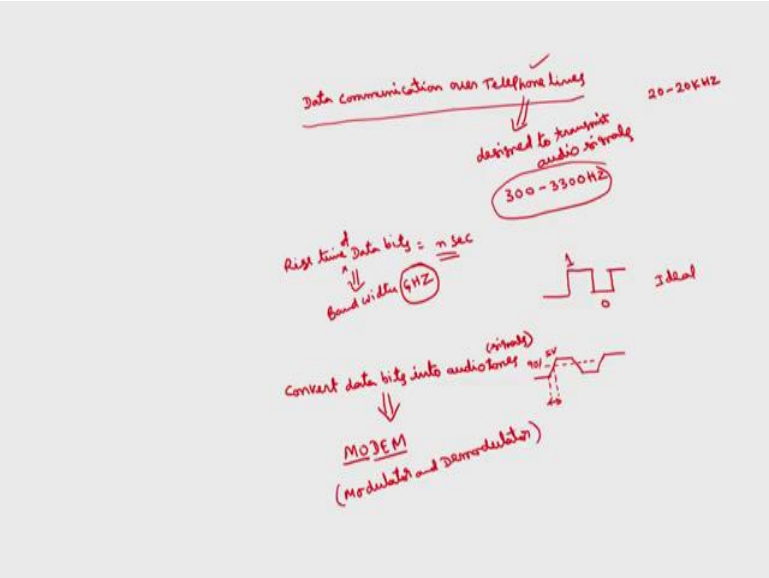
So, this is P e this is m naught m 1 m 2. So, this circuit will be available inside this parity generator. Now at the receiver you have to check for the errors. So, I will receive at the receiver, you have to design a parity checker because at the transmitter you have generate an even parity you have to design here even parity checker . The inputs will be because we are going to transmit 4 bits. So, all these 4 bits will be inputs. Now output of this one will be if I call E is equal to 1 means error E is equal to 0 means no error.

So, here I will assume that at the most one error will occur, if more than one error occurs we have their different techniques. So, here for this a simple technique so, I will assume that at the most one error occurs. So, if one error occurs if I receive here odd parity. So, means there is a error. So, E will be 1. So, accordingly you can design the in a similar way, the even parity checker. If I receive here even parity, no error E should be 0. So, you can design this even parity checker. This is a simple I mean, parity generator and checker circuit which will be used in data communication systems to detect the errors. Even you can correct also if I go for the hamming code using the hamming code, we can detect as well as you can correct the errors ok. This itself is a error control codes is a course.

And then we have check sum technique. So, in the check sum technique what we will do is, we will add all the bytes of the data that is to be transferred. Then we will take the twos compliment of that data and we will transmit that twos compliment of this sum of all the bytes as the last byte. Then we will transmit, set the receiver again we will sum all the bytes including the twos compliment of the earlier transmitter part. Then if you get that I mean sum as 0 means no error sum as one means error that is sum check sum technique and we have a cyclic preferic codes also there are some convolution codes. So, there are different techniques to detect and correct the errors. This is also another important concept in data communications ok.

The next concept is how to transmit this digital data over telephone lines ok; we can transmit the digital data over the telephone lines.

(Refer Slide Time: 17:57)



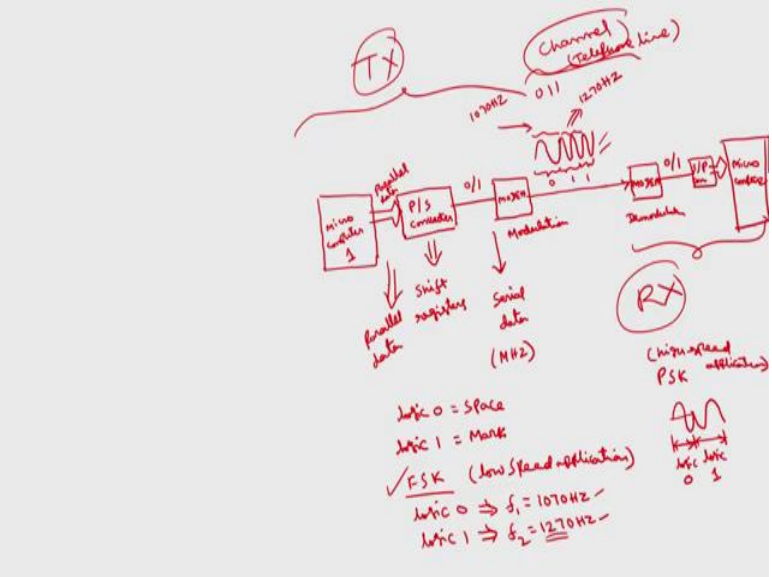
So, what is required to transmit data communication over telephone lines. So, what is required to transmit data over telephone lines? So, we know that telephone lines are basically designed for transmitting the audio signals, telephone lines are basically designed to transmit audio signals ok. What is the bandwidth of audio signals? Normally, we will call as 20 to 20 kilo hertz, but most of the practical signals will lies in range of 300 to 3300 hertz this is called articulation range.

So, most of the audio frequencies lie in this range but if i take this data bits so, what is the rise time of the data bits? We will be of the order of nanoseconds. What is mean by rise time? So, this is I mean transmission this is logic 1 and this is logic 0 and, but this is ideal waveform, but practically this will take some time to change from 0 to 1 as well as 1 to 0. So, this is a practical waveform. So, there are time taken to change from 0 to 90 percent of this final value which is 5 volts, the time taken is this is called rise time.

So, this is going to decide the bandwidth of the digital data transmission system ok. So, this will be of the order of nanoseconds. So, the bandwidth will be of the range of gigahertz. Then the question is how to transmit this large bandwidth signals over the small bandwidth telephone lines. So, for that, so we need to convert the data bits into audio signals. So, we need to convert data bits into audio signals, audio tones or audio signals. So, the revise which is going to convert this data bits into audio tones is called modem.

MODEM is a short form of modulator and demodulator. So, this is our now the block diagram of the data communication system, where the channel will acts as telephone lines will be like this.

(Refer Slide Time: 21:38)



So, we have a microcomputer I want to transmit the data from one microcomputer to another computer, but I want to connect these two through telephone lines. So, we know that the microcomputer is a parallel device this will give parallel data 8 bit or 16 bit depends upon the application. So, what is required is as I have told we require a parallel to serial converter. Now the bit will be here will be having 0s 1s in the sequence sequentially.

So, what is the circuit which I mean converts this parallel to serial conversion or serial to parallel conversion? This can be implemented by using shift registers; you might have studied on digital electronics. So, you can implement this parallel to serial conversion as well as serial to parallel conversion using shift registers. Now, you have got serial data this is parallel data and here we have got serial data, but what is the bandwidth of this serial data as I have told this is of the order of megahertz, but I want to transmit over the telephone line I am going to connect only one transmission line.

So, what is required is here modem. So, the output of the modem will be. So, this will be audio signals. So, we have only two possibilities here either, we have to transmit logic 0 which is called as space or you have to transmit logic 1 which is called as mark. So, we

will use in case of frequency shift keying, there are different types of modulation schemes that can be employed in modem. If I use frequency shift keying F S K. So, logic 0 space will be represented by 1 frequency say f 1 and logic 1 will be transmitted with another frequency f 2.

So, normally this will be 1070 hertz 1270 hertz. Then the output of the modem will be if I want to transmit say. So, this is a high frequency signal this is low frequency signal suppose, if I want to transmit 0 1 1 say 3 bits serially. So, 0 is correspond to one frequency 1070 which is low frequency signal. Frequency is low means; time period will be more this will be something like this is logic 0. Then we have to send 2 ones. So, the frequency of this next two signals will be less. So, this will correspond to 0, this will correspond to 1, this will correspond to 1.

So, this particular portion this is correspond to 1070 hertz signal which is low frequency means time period will be more. And this signal is corresponding to 1270 hertz signal. Frequency is more time period will be less. Now this will be connected to another modem at the receiver this is the transmitter part from here to here is transmitter and this is channel will be telephone line because now the frequencies are in audio range, we can transmit over the telephone lines. Now from here to here is receiver. So, in the receiver also we will be having modem. So, in the transmitter the modulation will takes place.

So, what type of the modulation you have used frequency shift keying. So, at the same signal will be received assuming that there are no errors and here the corresponding demodulation will takes place. So, what will be output of modem 1s and 0s. So, input is this audio tones output will be 0 1. Then this will be connected to serial to parallel converter, this parallel data will be given to the second microcomputer. This is about the complete system, transmitter, receiver in between you have channel. We are going to transmit digital data over telephone lines. So, because of this change I mean different bandwidths, what we are going to do is we are going to use the modem to convert the data bits into audio signals again audio signals into data bits.

So, in normally for low speed applications, we will use FSK we can also use some other techniques called as P S K phase shift keying for normaly high speed applications we will use PSK. In PSK the phase of the signal will be change. So, logic 0 is as it is we will transmit the signal logic 1 means we will make 180 degrees phase shift. This is logic 0

actual signal and this is 180 degrees phase shift is there from here to here this is logic 1. This is phase shift keying ok. So, next one is we have Q P S K that depends upon the system we have different types of this modulation schemes. So, this is about the next topic which is data transmission over the telephone lines. Now, coming for the standards serial IO or serial communication standards.

(Refer Slide Time: 28:29)

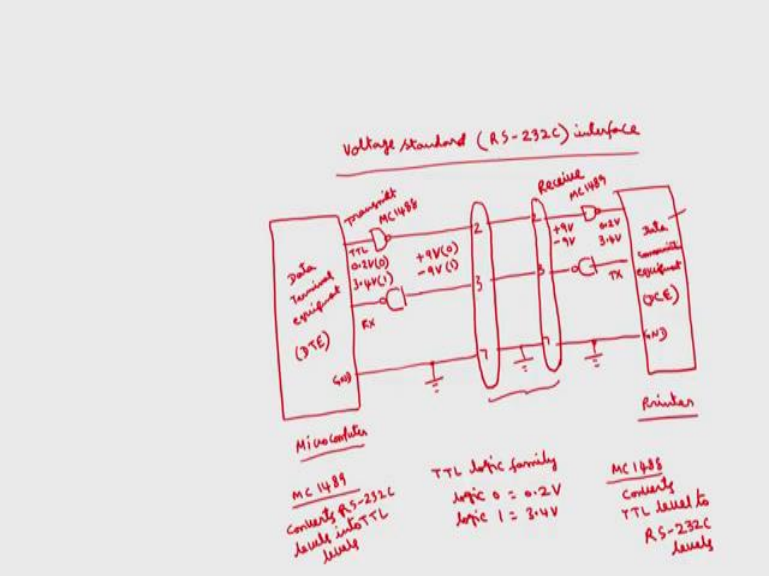
So, in order to transmit the data from one computer to another computer ok. So, different manufacturer will have different computer configurations similarly the peripherals. So, if I take this I mean micro computers and peripherals. Micro computers and peripherals will be manufactured by different industries different manufacturers. Now how to make compatibility suppose I want to I mean connect a peripheral which is designed by some x company and I have to connect to a microcomputer which is designed by some y company ok. So, for that you should have a some common standard, if you have a common standard, this enables to connect one manufacturers device with another manufacturer.

So, this common standard this will enable the flexibility among various manufacturers. Now, what are the standards in serial communication? So, that we can connect these two at the device manufactured by any company so, there are basically two standards one is current standard another is voltage standard. Basically, you have to transmit 1 or 0 ok. So, in this two standards we have one is current standard in this logic 1 is transmitted by passing a current of the order of 20 milliamps or some 60 milliamps. It has logic 0 passage

of current will be blocked because here we are going to transmit this logic 0 and logic 1 using current.

The advantage of this type of current standard is. So, this is a less immune to the noise, the effect of noise will be less here. So, that this can be used for long distance communications this suitable for long distance communication.

(Refer Slide Time: 32:18)



On the other hand we have the voltage standard also which is the most important and most popular standard it is called RS 232 C interface. Here we will transmit the signal by means of voltage. So, normally the devices will be constructed using TTL logic ok. So, if I take this transmitter; this is called data terminal equipment which is some sort of transmitter type this is called D T E such as micro computer.

So, we can have R S 232 interface, I will just show only the three signals of R S 232 interface, the remaining signals you can refer to any data book. I will show the signal 2 signal 3 and signal 7. This is R S 232 interface this is also corresponding 2 3 7, one is male connector another is female connector. So, this will be transmits this is transmit, this will be received by another terminal which is called as data communication terminal. Data communication terminal called DC equipment sorry data communication equipment called DCE this is receive ok.

Similarly, this can transmit through this is transmit portion of transmit TX and this can receive this is full duplex this is receive and this is ground signal this will be normally connected to the ground. The example of if this is microcomputer example of DTE IS microcomputer, example of DCE can be a printer. In this case only one way communication, if this has also another microcomputer we can have both the way communication. Now one important thing here is. So, this data terminal equipment will sends signals logic 0 and logic 1, 0.2 volts is corresponding to logic 0 3.4 volts is corresponding to logic 1 in case of TTL logic family.

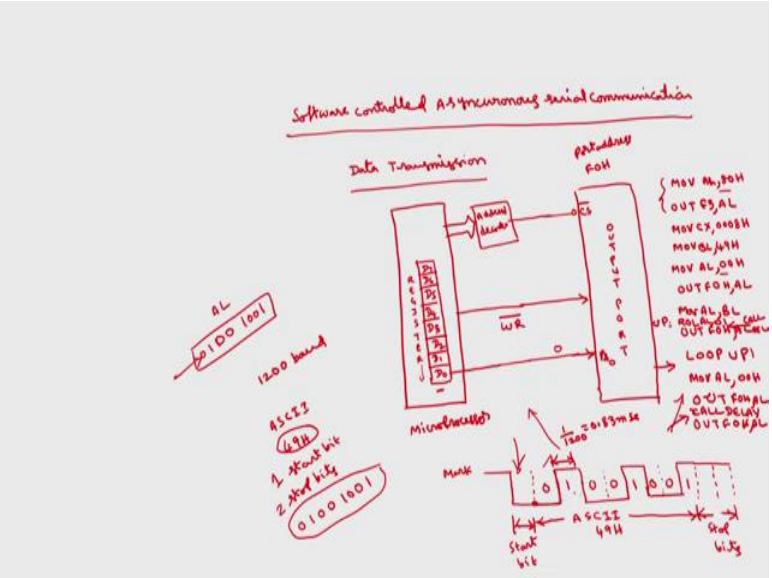
So, in TTL logic family transistor transistor logic 0 will be represented by 0.2 volts and logic 1 will be represented by 3.4 volts. This is correspond to logic 0 this is correspond to logic 1 these are TTL levels, but R S 232 C will not support TTL signals. So, it have I mean its own logic 0 and logic 1. So, logic 0 is nothing, but plus 9 volts and minus 9 volt is logic 1. This is logic 1 logic 0, these are the voltages supported by R S 232 C. So, in order to convert this TTL into R S 232 C voltage levels, we can use a circuit called MC 1488, this MC 1488 converts TTL level to R S 232 C level.

Similarly, the levels will be here now plus 9 volts minus 9 volts, but the again this DCE you will operate at TTL levels. So, here I want plus nine volts is correspond to 0.2 volts and minus 9 volts should be corresponding to 3.4 volts. For that we will use another I C which is called as M C 1489 this will convert the reverse operation this will do the reverse operation. Say MC 1489 converts R S 232 C levels into TTL levels.

This is how we can make the communication between and this R S 232 C have total the nearly some 24 signals, I have not shown all the signals 2 signal is transmit signal, 3 is receive signal, 7 is ground signal.

So, remaining signals I have not shown here, you can refer to any data book. So, this is how we can make this common standard. So, that, we can have the comparability or flexibility among the various manufacturers. So, these are all basic concepts of data communications. So, with this data communication concepts, we will go for the implementation of these data communications. We can do this implementation either using the hardware or software ok.

(Refer Slide Time: 38:59)



So, software controlled asynchronous serial communication. First I will discuss the software base after that we will go to the hardware base serial communication how to enable the serial communication using hardware?

So, in case of software so we will use the microcomputer here if I take data transmission. So, we have the microcomputer or the microprocessor and we are going to use accumulator or any other register this is some register. I will take 8 bit register or 16 bit register this is up to you. If I take 8 bit register this will you have D naught, D 1, D 2, D 3, D 4, D 5, D 6 and D 7. So, the data that is to be transmitted on the serial line, I will first take into some register because this register will contain parallel data. So, I will transmit bit by bit first I will transmit D 0 onto serial communication.

So, I will have here output port because we are going to transmit you have to use output port you can use 8 2 5 5 output port. You can program one of the ports of 8 2 5 5 as output port this is output port. So, one of the pins of these output port, we have to connect say if you want to connect to P A naught. Only one bit because serial communication this will be transmitted from here through PA naught. Suppose if I want to transmit the 49 H ASCII code for that we have generated the waveform 1 start bit 2 stop bits. So, what did the waveform that is to be generated here and here the same is here also.

So, what type of waveform yesterday also we have discussed this one. So, 1 is 1 start bit this is mark without transmission , the transmitter will remain at mark position. Then 1 bit

position of low which represents start bit then 49 will be 0 1s 0 0 1 0 0 1 this is start bit . Then another 0 the first bit of 49 H is also 0 0 then 1 1 these are all of same duration, then 2 0s then again 1 1 for 1 duration then 2 0s 1 1 then we have 2 stop bits. So, this is ASCII 49 up to here and these two are stop bits. Total 11 bits we are going to transmit ok.

This is logic you start a bit this is 0 1 0 0 1 0 0 1 is the total 49 H. So, this will be transmitted on D naught, but what we will take is first I will send the start bit I can write the program also later. So, then this bit 0 first I have to transmit, you take into some register this data and you transmit the LSB bit position onto the port a P A naught then you shift this data towards right and then again you transmit onto P A naught and you set a counter the number of bits to be transmitted ok. So, after this I mean count is over you have to stop this. So, this is about this counting operation ok. Then we have write bar signal because this is writing operation, we need write bar signal then this output port will be having, we know that this will be having chip select signal chip select bar 8 2 5 5 if we take as 8 2 5 5 or anything.

So, here this microprocessor we will be having address bus using this address bus using a decoder address decoder you can select the chip select. This is address decoder this is about the data transmission block diagram. So, we can write the software program to transmit the data for example, if I assume that the port A address is some 80 H 81 will be port B 82 will be port C 83 will be control word register ok. Suppose if I want to transmit the same thing. So, what is the program? First I will take MOV A L comma 80 H. So, because this 80 H will be, I mean control word register for making all the ports as port output ports in mode 0 because this is also 80 H to avoid the confusion I will take this as F 0 H. F 0 H is the port address of port a.

Then F 3 H will be control word register. So, out F 3 H comma A L. So, this will program all the ports as output ports, but of course, we are using only one port only ok. Now, I have to set the counter. Total how many bits I want to transmit? So, I want to transmit total 8 bits I will take MOV C X comma 0 0 0 8 H then I will take this data that is to be transmitted 49 H into some register say A L MOV or rather we will take it into B L B L comma 49 H. First I want to transmit a start bit which is logic 0. So, for that I can take MOV A L comma 0 0 H out port A address is F 0 H comma A L. So, what happens to P A naught.

So, this all 0s will be transmitted PA naught to PA 7 becomes 0. So, thereby here 0 will be transmitted that is this 0 bit will be transmitted that is start bit. Then you have to transmit the first bit of BL onto the output port ok. So, for that what we will do is? I will take that BL contents onto AL contents MOV AL comma BL. So, that AL will be having 49 H rotate right AL by 0 1 bit. So, that the contents of R O L becomes, first I will transmit as it is lower order bit onto the this port A PA naught. So, I will just transmit this later I will rotate. So, if I write this out F 0 H comma A L. So, what happens? Now, what happens here? I have taken here this 4 9 H in AL in AL I have 4 9 H 0 1 1 0 1 0 0 1 0 0 1.

So, I want to transmit first this 0 bit ok. So, this 0 bit means, we have to take otherwise you can take this data in the reverse order, if I take this data instead of 49 H if I take 9 4 H. So, I want to I mean transmit this in this direction. If I take 9 4 H 0 1 0 0 the first bit I will be transmitted will be this. You know I will just shift these bits. So, I will rotate this bits R O L AL comma 0 1 then out on to this one. So, only one bit will be outputted. Now D C decrement DEC CX or you can write directly loop because C X we can use the loop instruction loop up. So, this up will be so, you have to rotate 1 bit position.

So, you have to rotate this by 1 bit position, you can write here that rotating instruction R O R OR R O L AL comma 0 1 so, this up. So, that all the 8 bits will be after a transmitting this 8 bit ASCII character it will come out of the loop, then you transmit to stop bits MOV AL comma 0 0 H out F 0 H comma AL then again out F 0 H comma AL in between I have to give delay call delay. So, what should we this delay if I want to use 1200 baud rate yesterday we have computed this duration of each bit will be this will be 1 by 1200 which is equal to 0.83 milliseconds. So, you have to write this delay program which is equivalent to 0.83 milliseconds. Even between two consecutive this bit positions also.

So, you transmit here after this rotate here we have to call delay also same call same delay of 0.83 millisecond. So, that if I transmit first a 0 here you call the delay for 0.83 milliseconds. So, that it will end here again we will send one you call delay again. So, in every I mean bit transmission you have to call the delay. So, likewise we can transmit all the 8 bits. This is about the data transmission using software, we can write the program. Similarly, you can receive the data also data reception also we can do it by using software that we will discuss in the next class.

Thank you.