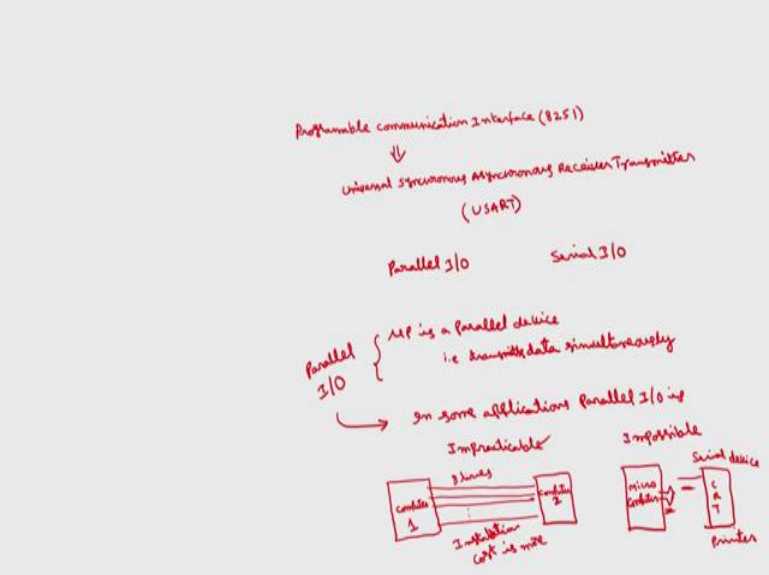
**Microprocessors and Interfacing**
**Prof. Shaik Rafi Ahmed**
**Department of Electronics and Electrical Engineering**
**Indian Institute of Technology, Guwahati**

**Lecture – 32**
**Basic Concepts of serial I/O**

In the last classes, we were discussing about the various peripheral devices. So, initially we started with 8255, the program with peripheral interface. Then we discussed about programmable interval timer 8254 then Programmable Interrupt Controller 8259, then programmable DMA controller 8237. The next programmable device is programmable communication interface.

(Refer Slide Time: 00:59)




8251, this also can be called as USART. The famous name for this one is universal synchronous asynchronous receiver transmitter; Asynchronous Receiver Transmitter USART. So, before discussing about this 8251 or USART, so, we will discuss some fundamental things of this data communication. So, I will discuss the difference between parallel I/O and serial I/O. So, if you take a microprocessor; microprocessor is a parallel device. What does it mean? It transmits the data simultaneously. That is transmits it can be 8-bit or 16-bit of data simultaneously over the data bus.

So, we can have 8-bit data transfer or 16- bit data transfer. So, this type of I/O is called Parallel I/O. In Parallel I/O, we will transmit the data parallely; 8-bit or 16-bit depends

upon the application. But in many situations, the Parallel I/O is almost impracticable and impossible. In some applications, Parallel I/O is impracticable. I will give the examples or sometimes even impossible. The applications where this Parallel I/O is impracticable is if I want to transmit the data over a long distance.

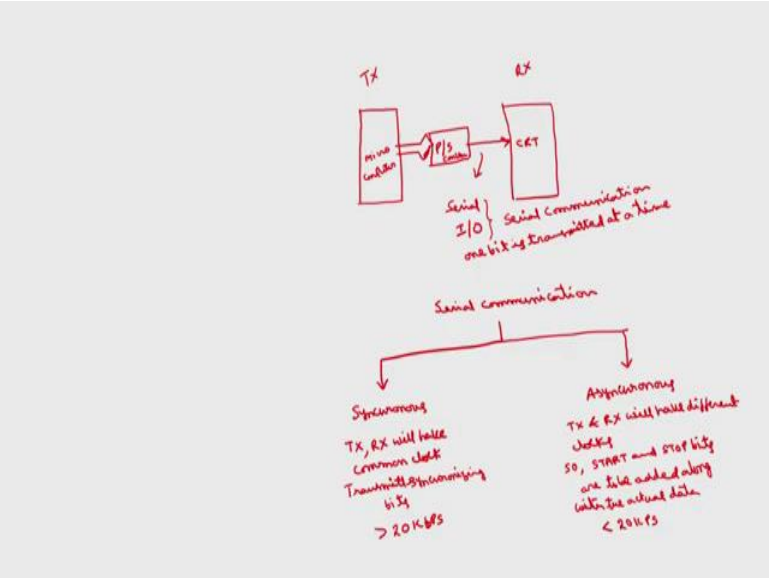
So, I have 1 computer here, this is computer 1 and computer 2 is placed at a remote place. So I want to transmit 8-bit of the data parallely. So I need 8 lines each, I mean a line will carries 1 bit of information. So, I need to connect total 8 lines.

If the distance is more if the distance is of hundreds of or thousands of the kilometers, then the installation cost of these wires will be very large ok. So we cannot afford that much cost. So the installation cost will be more. So this is somewhat impracticable. Still we can establish if we compromise with cost. But it is not a good solution, there are some cases where even if I spend some cost also, it is also impossible. In some applications, it is impossible to perform parallel I/O operation.

So one of this application is, this is microcomputer. In previous example also, computer means microcomputer. If I want to connect to a CRT terminal CRT so this by design itself CRT terminal is a serial device. So it cannot accept the parallel data. So it will accept only serial data only whereas, the microprocessor will give the parallel data. So I cannot connect these 2 serial data. Unless otherwise if you incorporate some additional hardware.

So, CRT device is one of the serial device as well as we can have printer also serial device. So in such applications, it is not possible to perform Parallel I/O operation. So in such applications what you have to do is we have to use Serial-to-Parallel and Parallel-to-Serial Converters.

(Refer Slide Time: 06:29)




So, we take this parallel device microcomputer or microprocessor.

So you take the parallel data, you apply to the parallel-to-Serial Converter. This is serial data, then you connect to the serial device such as CRT display or printer, ok. This is how you can perform this, it is directly not possible. But if I use some additional hardware, we can interface a serial device to the parallel device such as microprocessor. So this type of this serial communication, this is serial communication we can also call as Serial I/O. In Serial I/O, we are going to send 1-bit at a time, Serial I/O means 1-bit is transmitted at a time.

Now we will see some of the different types of Serial I/O or serial communication. So the communication where the serial type of operation will takes place, we can also call it as serial communication. So, there are different types of the Serial I/O. So basically, we need Serial I/O in many of the applications. So, we have to convert this microprocessor parallel data into serial and then you have to perform the serial communication. So serial communication can be broadly classified Serial I/O or serial communication into 2 categories; one is called synchronous serial communication, asynchronous serial communication.

As the name implies in synchronous, the transmitter and receiver; so, we will be having one transmitter and receiver here. This will acts as transmitter this will as a receiver; so both can be computers also. In the earlier example so, transmitter as well as receiver both

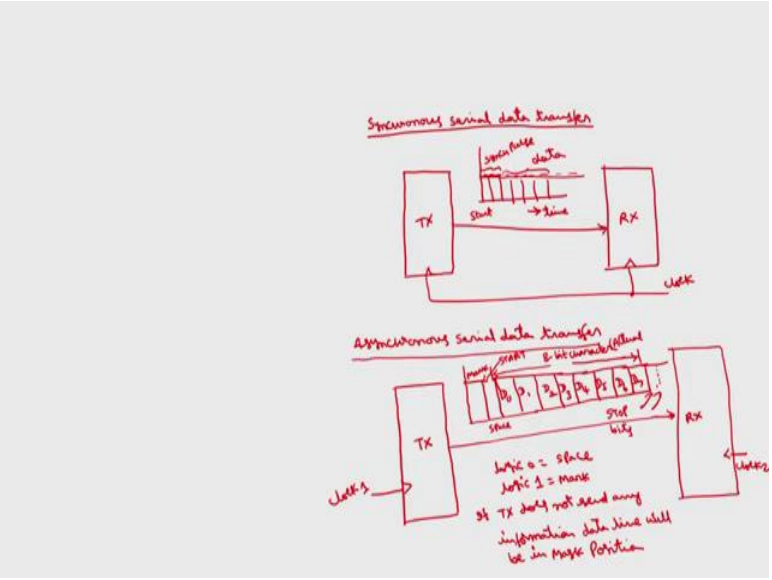
are computers. So in the case of synchronous, the transmitter and receiver will have common clock. So, because of that all the operations will be synchronized.

In addition here, this serial data you have to transmit in case of synchronous we have to transmit some synchronizing bits also. So this type of data transfer is normally preferred because we are going to transmit at a faster rate. So, this type of a data transmission that is synchronous serial communication is normally used for the data rates greater than 20 kilobits per second.

So, I will explain this with examples also whereas, in case of asynchronous the clock is different from transmitter and receiver. Transmitter and receiver will have different clocks. As a result of that, because the speeds are different now, so we need to transmit here start and stop bits. So, start and stop bits are required where the data starts, where the data ends. So you have to transmit in addition to the actual information, you have to transmit some additional bits called as start and stop bits; USART to be added along with the information.

So because of this start and stop bits, so the process becomes slower. So this will be normally used for data rates less than 20 kilo bits per second.

(Refer Slide Time: 11:39)




So if you want to explain in detail of the synchronous and asynchronous data transfers so in case of synchronous data transfer, synchronous serial data transfer. So we have 1

transmitter, 1 receiver. So we are going to connect through the serial line. So the data format will be here, this is time axis; this is start of the frame. So you have to use some synchronous pulses before the data to be transmitted. So, these are some sync pulses. Then this will be transmitted by data, this is data.

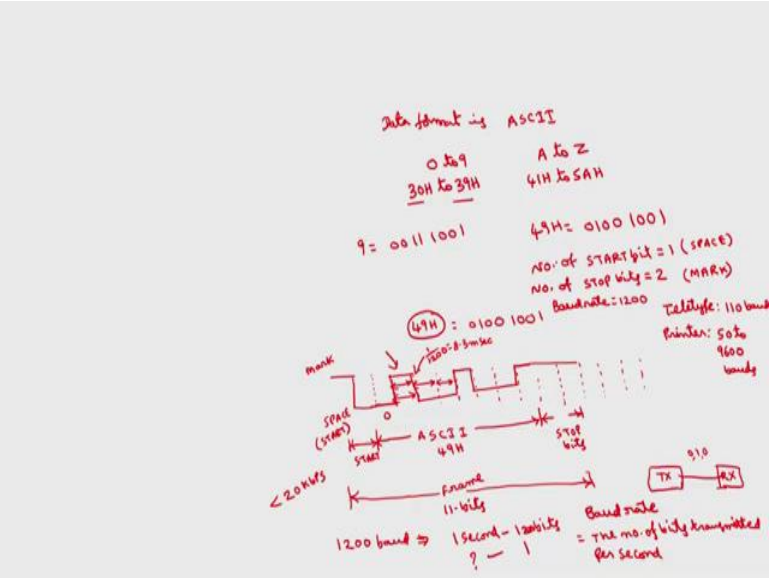
So this type of data transfer is faster because both will be operates at same clock, this is important. Both devices will operate at same clock whereas, in case of asynchronous data transfer, so both transmitter and receiver will be connected by a serial line, because this is serial communication. But these two will be having different clocks. This is clock 1 transmitter clock is still operated, some other clock, clock 2. This is the main difference between synchronous and asynchronous serial data transfer.

So because the clocks are different, we have to transmit start and stop bits along with the actual data. So if I take this frame structure, frame is nothing but transmission of 1 complete byte.

So this is mark and we have one space. Then we have data bits 1, 2. If I assume that 8-bits are the data: 4, 5, 6. So, this will be of same duration 4, 5, 6, 7 and 8. Then this will be high for 2 clock periods. This is called mark, this is called space normally, logic 0 is represented by space, logic 1 by mark. So, this first mark represent that. So if the transmitter does not send any information, if transmitter does not send any information. So the data line will be in mask position.

So this is not transmitted and this particular low bit or space bit this represent start pulse this is of 1 clock period duration. So, this is actual data from here to here actual data 1, 2, 3. So on up to this is D0, D1, D2, D3, D4, D5, D6, D7, total 8 bits. This is 8-bit character. You can call as character or data. And there are 2 high signals; this is mask. So, these 2 are called as STOP bits. I will explain this in detail by transmitting some of the characters.

(Refer Slide Time: 18:01)




So the format, data format is normally the type of the data that we will be send through the serial link will be data format will be one of the commonly used data format is ASCII. This I have already explained American Standard Code for Information Interchange. So in this, this is a 7-bit code, but we will represent it with 8- bits. So 0 to 9 decimal will be represented by 30 H to 39 H. Similarly, capital letters A to Z will be represented by 41 H to 5A H. Total 26 letters. Similarly, some special symbols will be having some specific codes. So normally, we will use 8-bits.

So, 9 means if you want to transmit 9 is 0 0 1 1 1 0 0 1 we will transmit this 8-bits. Suppose if I want to transmit say 49 H, this is corresponding to one of the letters. So this is 41 is A 42 is B. So, these 41 H is corresponding to one of the alphabets. So this is nothing, but 0 1 0 0 1 0 0 1. So what is the format that you have to send? And you have to assume this depends upon your serial interface, the standards. I want to use 1 STOP bit.

Number of start bits is equal to 1. So number of START bits I want to take 1 and number of STOP bits I want to take 2. So start bit will normally space and stop bit will be limited by mark. So what is the format that you have to send now? For 49 H format of 49 H. So first one is initially without transmitting the data mark. So this is the point where I am just starting the transmission of the data. This will be one clock period duration if you

assume that this is the clock duration i will come to this clock duration, what should be the clock duration? So these are the equal duration pulses you have to transmit.

So this is space which represents 1 START bit then 49 H is equivalent to 0 1 0 0 1 0 0 1. So the first bit is also 0. So we can transmit 0, this is first bit 0, from here to here ASCII 49. Then we have one this is 1 then 2 zeros, 2 zeros then single 1 then 2 zeros then 1 then followed by two stop bits. So this is still on, this will be both are on. 1 0 0 1 up to here the ASCII character ends, but these 2 are STOP bits. So total you have to send starting from the start bit, this is start bit; so total from here to here total 11 bits each called 1 frame.

So, 1 frame consisting of 11 bits in this case. So this is what we are going to send if we want to send 49 H. Now the rate at which the bits will be transmitted decided by the duration of this clock signal, these durations are same. So, what is this duration? That will be decided by the rate at which the bits will be transmitted from the transmitter. This is transmitter, this is receiver. So what is the rate at which we are going to transmit the bits? 0, 1, 0 something; so the rate at which you are transmitting the same rate, you have to operate the receiver also otherwise there will be some errors ok.

So, the rate at which we are going to transmit the bits is called baud rate. This is the one of the important parameter baud rate is defined as the number of bits transmitted, number of bits transmitted per second. So this actually depends upon the various devices this varies. See if I take teletype, teletype operation normally we will use a baud rate of 110 110 baud rate will be used. But in most of the commercial applications printer say, this will use this adjustable baud rate will be used between 50 to 9600 bauds bits per second. Suppose if I want to transmit here the specifications I have given in addition to this if I give the specification as baud rate also, baud rate I want to operate at 1200 say. Then what will be this gap? What will be this duration of this signal?

So, what will be this duration? How to compute this? So baud rate is defined as bits per second ok. 1200 baud means implies in 1 second 1200 bits, 1200 bits. So, 1 bit what is the time taken? This will be 1 by 1200. So this duration of this one will be 1 by 1200. This will comes around some 8.3 milliseconds ok. So this is how we can transmit the data in case of asynchronous mode of serial communication.

So we require start and stop bits normally because of the start and stop bits the rate of data transmission is here less. So, we have to use this type of data transfer for the data rates less than 20 kilo bits per second, ok. This is one difference between synchronous and asynchronous type of the data transfer. So we can classify this serial communication based on the data type as well as the direction of the data. So, on the based on that we have simplex and duplex operations that we will discuss in the next lecture,

Thank you.