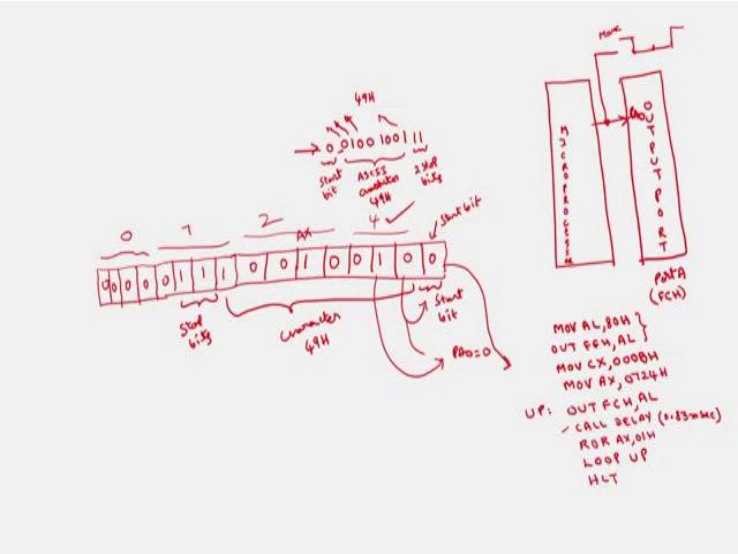
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture – 34
Architecture of 8251

So, we are discussing about the software controlled serial data transmission. So, in that we have discussed about the serial data transmission. So, we can write the program for serial data transmission.

(Refer Slide Time: 00:50)



So, because we have to send ASCII 49 H means you have to send 49 is 0 1 0 0 1 0 0 1 and one start bit will be here 0, 2 stop bit will be 1 1, we have 2 stop bits this is one start bit, this is the flow that we have to give, first we have to send the 0. Then we have to send 1 then we have to send 0. So, on up to this one total this one is ASCII character 49.

So, how to write the program for this? So, we have connected this microprocessor to port A say, this is serial data pin. So, inside this 8255, we have this serial data pin I will connect to the port A output port A of 8255 I will program this as output port say PA naught this is microprocessor. So, I have to send in this manner first you have to send the stop bit then 0. So, I will take this into some 16 bit register. So, I will take this 16 bit register say AX.

So, first bit is start bit, this is 0, then character bits are the first bit will be 0 in this direction then 1 then 0 0, then 1 0 0 then we have 1 this is character this is start bit and this is character then we have 2 stop bits 1 1. So, I will not taken AX is 16 bit register total we have 11 bits only remaining five, I will take as do not cares I will assume that as 0 0 0s. I will take the remaining bits as 0 0 0 0 0 up to here, this is character sorry up to here this is character; these two are start bits, this is character 49 H, this is start bit and these two are stop bits. So, what is the hexadecimal equivalent of this one this is 0 0 0 0 is 0 and then 0 1 1 1 is 7 and this one is 2 and this one is 4, 0 7 2 4 H.

So, first I have to I mean program this port output port A as output port. So, let us do the address of this one is F C H, then control word register will be F F H; I have to first program the port A as output port. So, to program all the ports as output ports we know that 80 H is the control word register we will out onto FFH which is the address of the control word register comma A l. With these two instruction this will program all the ports of 8255 in mode 0 as output ports. Now I have to send one by one total 11 bits.

So, I will take this counter CX with 11 bits 0 0 0 B the equivalent the hexa decimal equivalent of 11 will be B, then because I have to send on PA naught in A X, I will take the total the character start stop bits and the remaining bits 0s. So, this will be 0 7 2 4 H . So, in this we have to send the least significant 11 bits. The most significant by 5 bits will not send their do not cares will stop after the eleventh bit is transmitted for that so, I am going to send only AL contents only.

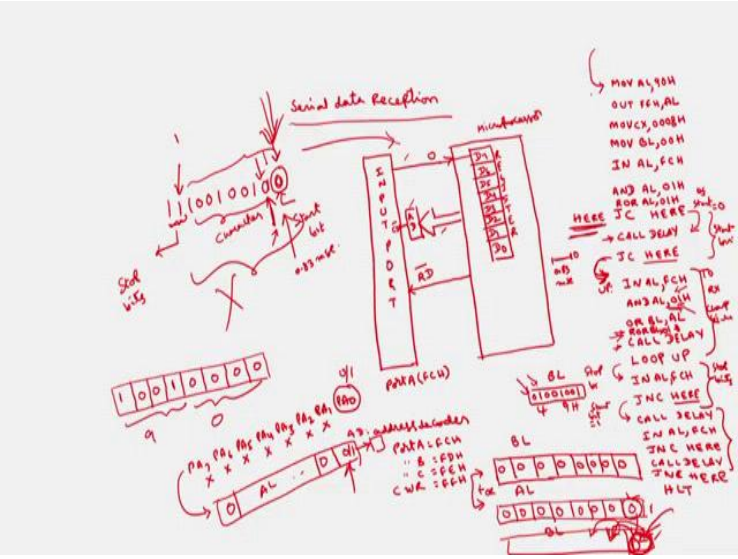
So, each time I will shift this. So,OUT F C H, A L so, what happens this 0 will outputted on P A naught. So, this will be outputted on P A naught. So, this is 0, this P A naught if I take this P A naught waveform, this will be initially without any transmission this will be your mark I have started here so, this 0 this is the point. Then what you have to do is we call a delay equivalent to 0.83 milliseconds in case of 1200 baud rate. So, this 0 will be there for 0.83 milliseconds. Then what I will do is rotate A X to the right by 1 bit position. Now what happens here? This 0 will come here after the rotation that is the first bit of the character, then I will decrement CX because I have already transmitted 1 bit.

So, for that loop up so, in this loop CX will be decremented until CX is equal to 0, this will branch to the address with the label of the up. So, where should be this up now? Up should be

out F C H comma A L. Now what happens second bit 0 will be transmitted this is 0 for 0.83 milliseconds because here delay is there again another shift. So, this 0 will come to the LSB position. So, then sorry this one will comes to the LSB position this one will be transmitted for a 0.83 seconds. So, like that this sequence will be outputted.

Once the all the eleven bits has been outputted, this will come out of the loop simply you have to HALT. This is how using the software we can transmit any character with the specified number of start bit and stop bits. Similarly at the receiver, we will see about this serial data transmission. So, what about the serial data reception?

(Refer Slide Time: 08:49)



So, what will be the block diagram? So, you have to program the port A as the input port; in case of transmission, you are programming the port as output port. Now you have to program say same port A only input port. So, whatever the data that is coming from this input port, you have take into any register. We have take on to D 7 first D 6, D 5, D 4, D 3, D 2, D 1, D naught. This is the register inside the microprocessor. I will write the program for this also, this is microprocessor then the microprocessor will generate the address correspond to this input port.

So, like in previous program we assumed that the port A address is FCH that FCH or whatever the address will be generated by using this address decoder. This will connect to chip select of this signal; this is address decoder A D stands for address decoder. So, this is going to decide the

address of the port A and port B, port C, whatever the logic inside of this. So, I am assuming that the address of port A is FCH, then port B will be the next address port C will be next address and control word register will be next address so the corresponding circuit will be present in this block A D block that will connect the output of that A D block will be connected to chip select signal. Then because you are reading so, the microprocessor sends read bar signal.

In case of transmit write bar signal and this logic is same regardless of whether this is a transmission or a reception and here this has to program as input port in reception output port in case of data transmission, now the characters will be received here. So, what will be the waveform here received will be the first bit will be start bit 0 then 1 0 1 0 0 1 0 0 1 then 1 1. So, this is start bit. First start bit will be received, then the character will be received character is this is 4 and 9 character then 2 stop bits will be received.

So, whatever the data that is received on input line of port A I will take into some register say, let us take I want take this value into register A L or B L. So, what will be the program software for this? So, the first two instructions are same MOV A L, 80 H, sorry here we have to program this port A as input port A in order to program the port A as input port. What is the Control word register this is port C lower we are not using.

So, we can take as output port this is port B we are not using I will take as output port, this is mode selection for port B I will take mode 0, this is port C upper we are not using I will take as output port this is port A, port A you have to program as input port 1. Next mode selection for port A I will want to operate in mode 0 only 0 0 and this is for IO mode 1. So, this will be 9 0 you have to take A L with 90 H. So, that port A will be programmed as input port out control word register is FFH, AL. So, these two instructions we will program port A as input port B, port C as output ports all in mode 0 ok.

Now, I will make this counter to 8, now because I have to store only 8 useful bits and take B L as 0 0 later I am going to use this B L contents 0 0 H, these are some initialisations. Then first I will take the input bit in A L comma F C H.

So, what happens this is F C H port address. So, the first bit that is received, if it is correct I mean received character then first bit should be 0 that will be available here. If I take in AL in

LSB, it is available. So, I want to check whether this LSB is 0 or 1 for that I will and A L with 0 1 H.

So, that the last bit becomes whatever this, the received bit whereas the remaining bit becomes 0 ok. So, whatever you received on input port, this entire input port because we are not using P A 7 to P A 1, P A 6,P A 5, P A 4, P A 3, P A 2 P A 1, P A naught here some bits will be received, but the we are not using these bits.

So, if I write in P L in A L comma F C H, this entire data will comes into AL register. So, what is interested is because this bits will be received on P A naught. So, that is why I am ANDing this one. So, that this bit if this is 0, it remains 0 only; if this is 1, it remains 1; only remaining all bits becomes 0 so, this is the bit that I want to read.

So, that is why I have ANDed this A L contents with 0 1 H. Now I will OR this because in B L I have taken 0 0 H already. So, I want to store finally, this received character onto the B L register so, but the first I mean bit will be the start bit. So, until this start bit is 0 I have to wait because if our start bit is not 0 that is not the valid character ok. So, that is why what I have to do is after ANDing with this one jump on carry here it itself.

So first bit if the first bit is so, I have shifted this bit to the carry flag in R O R sorry, this is after this we have to write R O R A L, 0 1 H. So, that the last bit which is the bit that is received on P A naught will transfer to the carry flag. If this is 0, then this will be false means that is a 0 bit otherwise if this is 1 means this is not the valid character, I will wait until this the last bit which is the first received bit will be 0 this is a infinite loop. So, this will come out of the loop whenever the last bit is 0 that is start bit is 0. If start bit is 0 then it will come out of the loop, then it is not enough actually this start bit has to be 0 for 0.83 seconds.

So, what I will do is I will CALL DELAY of 0.83 seconds after that again I will check jump on carry here. So, if it is not 0 jumps on carry, if this is 2 means not 0 1 means it is not a valid character. So, I will wait here itself so, until a 0 has to be available for 0.83 milliseconds till that time, I will wait once this is coming out of this loop means.

So, the first bit is 0 and this is of duration logic 0, it is of duration 0.83 milliseconds then it will come out of the loop, then I will check for the first I mean bit of the character otherwise this will

wait here until a valid start bit is received, this will wait in the loop. One this is coming out of the loop means a valid start bit has been received. Now I have to check for character bits.

So, first character bit again I will write IN A L, FCH because I have already call the delay. So, the second bit that should be received will be this I have kept this for 0.83 milliseconds. So, after that the bit received will be the next bit which is the first bit of the character so, that I will end with 0 1 H. So, that last bit remains whatever the received bit if received bit is 0 the last bit becomes 0 because we are ending if received bit is 1 the last is 1, because the last bit of this 0 1 will be 1 ok, this I am going to OR with B L and I will store the result in B L itself.

So, in B L initially 0 was there all I mean or 0 bits 0 0 0 0 0 0 0 0 initially. In A L, we have the remaining all bits are 0 except for that these 3 bits are 0 because we have make this ending operation with 0 1 H. So, this 7 bit the first 7 bit becomes 0. This bit will be whatever the received bit, received bit can be either 0 or 1. If this is 0 if I OR these two OR operation, then this B L will be having after this the contents of BL becomes the first 7 bits will be 0 0 0, I will not consider that this bit will be whatever the bit of this one will be transferred. So, 0 if you OR with 0 0 only this is 1 1. If you OR with 1, this will be 1.

So, simply the last bit of BL becomes the first received bit. See in our present case, this is 0. So, the first bit received will be here 0 ok, then I will CALL DELAY before that I will rotate; I will CALL DELAY because this has to be present for delay of 0.3 milliseconds then I will decrement C by 1 because I have already stored the first bit. So, LOOP UP CX will be decremented by 1. So, C X becomes 7 so, this is not 0.

So, it will go to up where should be this up now? The second bit received has to be checked now. So, LOOP UP means this should be here. So, again I will input the data that is received here into A L in that also we are interested only last bit only. So, we will AND with this last bit and then we will OR with B L. But before this what we have do is we have to rotate B L by 0 1 H, because this first bit that is received in BL, I have to transfer here so that I can store here the next bit that is received.

So, we have to put this rotate operation in loop. So, what happens in the next second bit received will be comes here, again you have to perform rotate operation so that the first bit will moves to

the third position, second bit will moves to this second position, now the last bit position will be free which is used to receive the next bit. So, like that this loop will repeats at the end of this loop B L will contain the ASCII character that is received. So, at the end of this loop BL will contain that 0 1 0 0 which is 4, 1 0 0 1 because the first bit received will be 0, this will be rotated 8 times because this rotate operation is in loop.

So, this will come here, second bit received will be here so like that this is 4 9 H will be placed in B L. So, this is valid if received bit is valid if it follows two stop bits. So, I have to check for stop bits, this program is for checking the start bits and from here to here to receive the character bits.

Now you have to write for checking the stop bits, then only it is a valid character. So, again I will input IN A L comma because every times you are calling this 0.83 second delay. So, after the 8 character also 0.83 DELAY, then I have to check in A L comma port C, F C H. Then jump on not carry now because stop bit should be 1, if it is not 1, 0 this not carry will be true.

So, this will be we have to jump not carry we have to go to here. So, the thing is here. First bit is start bit, we have received as 0 and the remaining 7 bits, we have received as it is after that if this bit is not 1, means this entire character is false. So again you have to start from this point onwards that is why this here should be this here. Once if this bit is 1, it will come out of the loop means first start bit is 1, start bit is 1 then I have to call a delay CALL DELAY, then I have to check for second start bit also. Again in A L comma F C H same logic jump not carry means second start bit is second stop bit is not 1. So, in that case also we will go to again starting of this logic here itself. So, again I will start from start bit onwards because that is not the valid character.

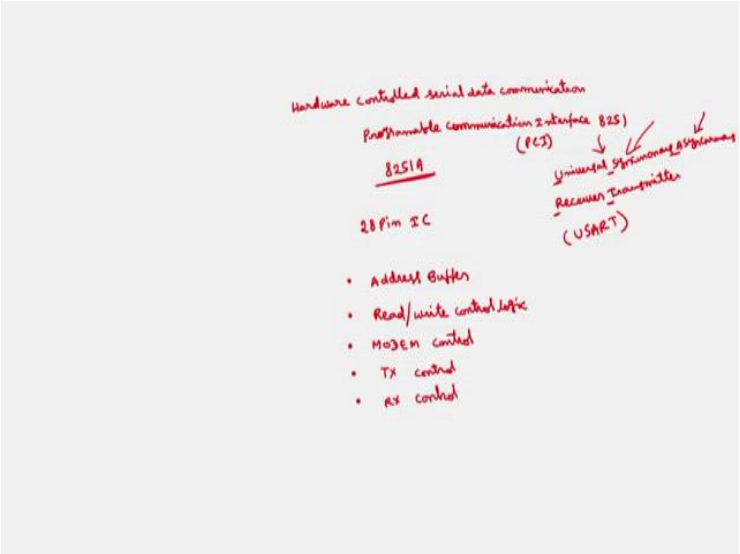
If second start bit is also 1 after the delay also, Jump Not Character, Not Carry; Jump Not Carry HERE, if both the bits are 1s and they are available for 0.83 second, 0.83 millisecond duration then. So, we will confirm that the received character is valid one, then we will simply HALT. So, this is about this serial data reception program. So, you have considered serial data transmission as well as serial data reception and we can write the software programs to send the data as well as to receive the data ok.

So this is about the software control serial data communication. Now the drawback of serial data, I mean software based serial data communication is sometimes it may become slower ok.

As I have discussed in case of 8 2 3 7 DMA controller also. So, why the DMA controller is required? Normally this data transfer will takes place through program instructions, but in some applications if the peripherals are fast like hard disk. So, that much rate, the fast rate the microprocessor cannot write the programs to I mean receive the data at that first rate. So, we have to go for the dedicated hardware.

Similarly, here also in order to I mean accept the data at faster rates. So, instead of using the software programs because here the software program, we have to fetch the software program, we have to execute this will take time. So, instead of that if you have a dedicated processor, dedicated interface device then that can perform these operation transmission and reception at the faster rates.

(Refer Slide Time: 28:14)



So, the second approach is instead of software controlled serial data communication, we have hardware controlled serial data communication. So, the hardware dedicated device which is used to I mean the control serial data communication is called Programmable Communication Interface PCI, the IC number is 8 2 5 1, this also can be called as another name for this one is U S A R T, Universal Synchronous Asynchronous Receiver Transmitter.

So U S A R T. Here as the name implies we are going to discuss both the types of synchronized data transmission as well as asynchronous data transmission. We know what is the difference between synchronous transmission and asynchronous data transmission?

So, this peripheral device 8 2 5 1 is capable of performing both types of the data transfer synchronous data transfer as well as asynchronous data transfer. There is one more this just advanced version of this 8 2 5 1 is 8 2 5 1 A, of course this is compatible with 8 2 5 1 ok. So, I will just write 8 2 5 1 only, it means 8 2 5 1 A also ok.

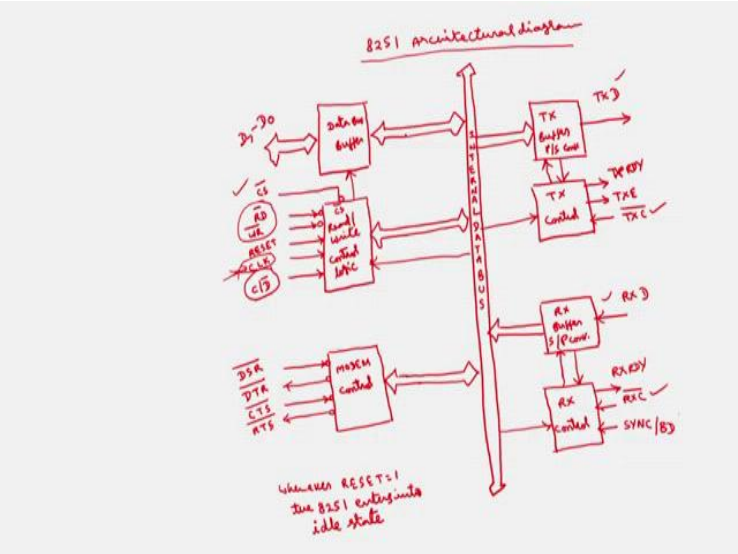
Now coming for this 8 2 5 1 A or 8 2 5 1, this is a 28 pin IC. So, to control these synchronous as well as asynchronous operations totally it has 5 sections of blocks. So, it is simply be having one block is called address buffer block, then this will be having read write control logic. Normally in most of the peripherals, these 2 logics will be common read write control logic, then it has a modem control.

So, we know that so, in case of transmission of the data, digital data over telephone lines we require modem. So, this will have some modern control signals, then we have transmitter control, receiver control these are the some important blocks of this 8 2 5 1 and we have some serial to parallel converter and parallel to serial converter.

Normally so, in case of software controlled you have to use serial to parallel and parallel to serial conversion using the program whereas, here we will have a hardware as I have told this serial to parallel and parallel to serial conversion can be implemented by using shift registers ok.

Now, coming for the block diagram of 8 2 5 1 or 8 2 5 1 A, they will use some internal data bus like all other peripherals.

(Refer Slide Time: 32:04)



This is internal data bus which communicates between the various blocks of this peripheral, as I have told the different blocks are the first block is data bus buffer, this is data bus buffer, this is a bidirectional because data is bidirectional operation data flow, we will be having 8 data lines this 8 2 5 1 is having 8 bit data lines D 7 to D naught, out of 28 pins 8 pins are data lines. Then we have read write control logic. This is also bidirectional its control signals are some control signals are input signal; some are output signals because it can read or write memory as well as IO. Then the, what are the different signals? The very first signal for this any read write control signals will be chip select bar.

So, you have to first of all select this chip using some address decoding logic ok and as the name implies read write, so we have read bar signal, active low signal, write bar active low signal and the various control signals are we have RESET signal, then we have CLOCK, then we have C by D bar signal.

So, I will explain what is meant by the C by D bar signal and this is chip select signal. So, totally we have 6 different signals for this particular block and then the other block that I have discussed is modem control. So, for modelling material, you have 4 active low signals, 2 are input signals and 2 are output signals, one is dataset ready. I will explain the functions of the all these signals in detail data terminal ready, clear to send, ready to send.

So, these are the different 3 blocks here, this side the other side we have parallel to serial converter which will acts as transmit buffer, transmitter buffer which is basically a parallel to serial converter. So, it will take the parallel data, it will convert it to serial and there is one serial data signal which is called as transmit data T X D, T X D is transfers transmit data.

Then we have transmitter control signals, we have 3 control signals corresponding to the transmitter, one is transmit ready whenever the transmitter is ready it will inform the device through a signal transmit ready T X R D Y transmitter ready, if transmitter is empty this will indicate through another signal called transmit empty, then this transmitter will be having some clock this is transmitter clock, this is going to decide the rate of transmission, this is active low signal. Then we have receiver buffer basically this is serial to parallel converter.

So, this will receive the data on a signal called RXD and this is reverse to that of TXD. So, the serial data will be received, this will be converted into parallel that will be given to the internal bus of this peripheral. Similarly, we have receiver control. Here also there are 3 signals receiver control. So, we have this signals from here and there will be connection between these two, the signals of this one is receiver ready, similar to transmitter ready and then receive clock C active low signal. Instead of transmitter empty there is a one signal called here sync signal which is synchronous signal.

So, these synchronous pulses are required in case of a synchronous data transmission ok or this can be also called as break character. So, these are the different signals of 8 2 5 9; 8 2 5 1 sorry 8 2 5 1 architectural diagram.

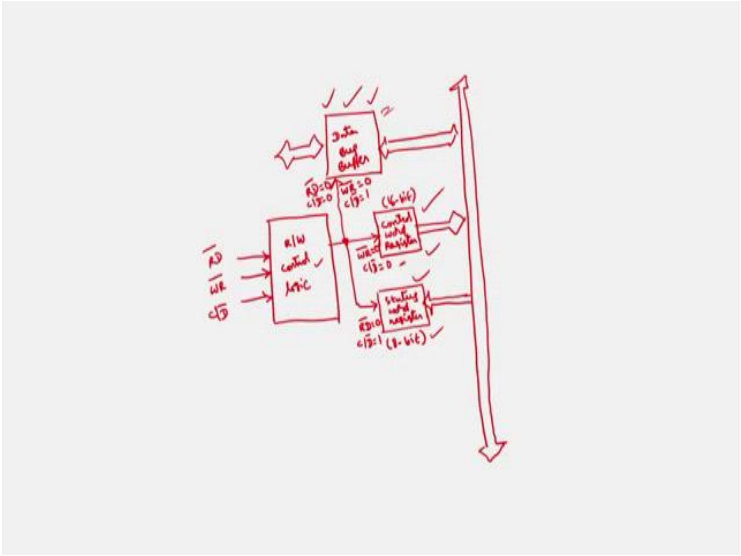
So, total how many signals? Now these are 8 signals 8, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26 and the one ground and another is VCC. So, total 28 signals ok, now coming for the functions of each signal.

So, chip select signal is obvious the operation of the chip select is so, whenever the chip is selected in order to select this chip CS bars should be 0 ok. So, read bar write bar signal I will come to this read bar write bar in connection with C by D bar. So, these 3 signals are related I will come to that later, reset whenever this 8 2 5 1 receives a reset signal 8 2 5 1 will enter into the idle state. Whenever reset is 1 the 8 2 5 1 enters into idle state, one we got this inside this we

have the sequential logic so, it requires some clock to operate this all functions at a some rate, one important point here is this clock is no where related to the transmitter and receiver speeds. This is only this internal to this particular blocks only, internal blocks of this clock is not going to decide the speed of transmission and reception, the speed of transmission reception will be decided by 2 clocks called receiver clock and transmitter clock ok.

Now, coming for this read by write bar and C by D bar; if you go into some more details of this read bar write bar C by D bar, this is internal data bus.

(Refer Slide Time: 41:05)



So, I am trying to going to draw that detailed diagram of this C by D bar controlled logic, this is somewhat data buffer; data bus buffer. Then we have read write control logic. There are 2 registers called one is control word register, another is status register. This is the read write control logic. The remaining signals I am not going to discuss here; I am going to discuss only 3 signals read bar, write bar and C by D bar ok.

So, this will go here here or here, it will decide by these 3 signals. So, this is one is called as control word register and this is status word register. This is 8 bit, whereas this is 16 bit. So, the microprocessor will write some data into control register. So, the direction will be here from control word register to internal data bus whereas, the microprocessor can read the status of 8 2 5

1. So, that the direction will be like this so, as the name implies in case of control word register because it is going to write, write bar is 0 here.

So when write bar is 0, here also we can read write data. So, if write bar is 0 whether this control read write control logic will refer to data bus or control word register that will be decide by C by D bar. So, if this is data C by D bar is equal to 1 and write bar is equal to 1 means data buffer will be selected. If C by D bar is 0 and write bar is equal to 1, control word register will be selected. Is it clear now?

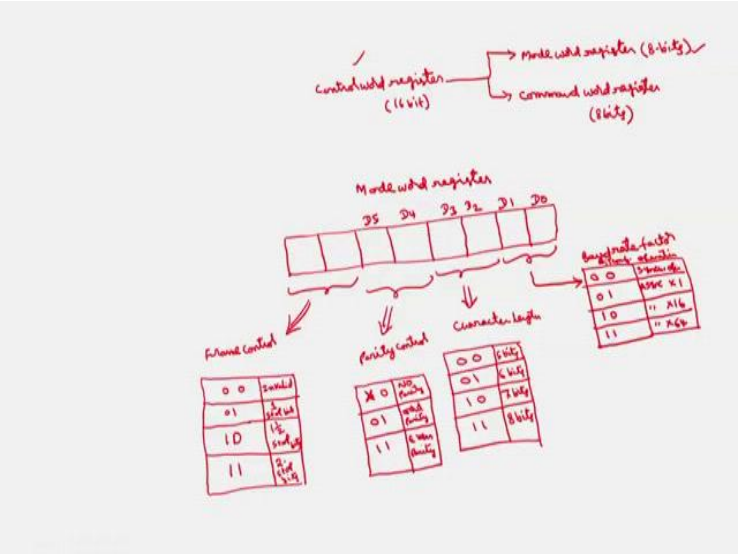
So, C by D bar is going to decide whether the control word register or data bus buffer is selected even if you select the write bar as equal to 0. Similarly read bar is equal to 0. So, we can read something, either you can read from the status register or you can read from the data bus also here also read bar can be 0.

Now, when read bar is equal to 0 whether the microprocessor is going to read from the data bus buffer or status word register that will be again decide by C by D bar. If C by D bar is equal to 1 and R D bar is equal to 0, this refer to status word register if C by D bar is equal to 0, R D bar is equal to 0 it refers to data bus buffer ok. So, this is about the functions of read bar, write bar and C by D bar signals. Now what about this 16 bit control word register and status word register? What are the formats of this control word register, status word register?

So, like in the case of other peripheral devices we have some format which is going to decide the modes of operations and the various I mean details of this transmitter as well as receiver.

Now, what are the formats of this control word register and status word register?

(Refer Slide Time: 45:20)



Control word register is 16 bit, but again this will be independently divided into two 8 bit registers one is called mode word register, another is called command word register. Each is 8 bits. So, what is the format of these 2 registers? First I will take mode word register. So, the format is this is 8 bit register the last two bits; L S B two bits are this is D 0, D 1. These two bits are called as baud rate factor, this is going to decide baud rate factor. We know what is baud rate. If this is 0 0, this is bit configuration 0 0 the operation and bits, bit combinations; 0 0 means synchronous operation as I have told 8 2 5 1 can operate as either synchronous mode or asynchronous mode ok.

So, if this is 0 0, this is synchronous operation. If this is 0 1, then this is asynchronous operation and the baud rate factor is 1. Whatever the baud rates we have set that is to be multiplied by one means that will be taken as it is ok; I will explain this what is meant by this baud rate factor 1, 16 and all.

So, we will be discussing about some few examples 1 0 asynchronous, but baud rate factor is 16, 1 1 asynchronous only baud rate factor is 64 ok for only 0 0 the 8 2 5 1 will operate in synchronous serial transmission mode and as for the remaining 3 combination asynchronous. In one case direct the baud rate is taken and in other cases in one case it is a 16 factor is taken in other case 64 factor will be taken. That is about the last 2 bits. The next 2 bits are character

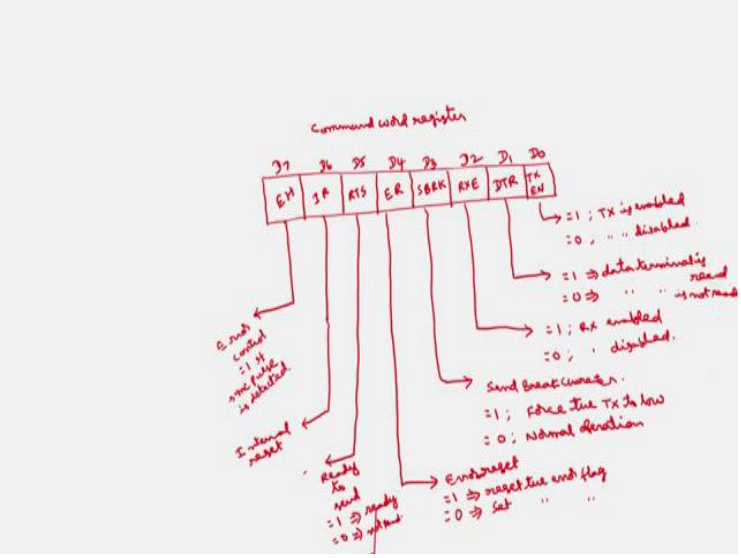
length D 2 D 3 will decide the character length. In case of ASCII character length is 8, even though it is a 7 bit code, but we are going to use 8 bits. This is going to decide here. This is 0 0 character length is 5 bits, 0 1 character length is 6 bits, 1 0 character length is 7 bits, 1 1 character length is 8 bits.

So, we can have the character lengths from 5 to 8 range, we can have character length of 5 6 7 and 8. The next two bits D 4 and D 5 or parity control signal, this will be parity control as I have discussed in the earlier lecture, we have to send some parity signals in order to detect and correct the errors at the receiver ok. We can transmit either even parity or odd parity or we can have no parity also for short distance communications even without parity also we can send.

So, that will be decided with these 2 bits. This is 0 0 no parity. In fact X 0 and this will be 0 1 odd parity, this is 1 0; 1 1, 1 0 is actually included in this, 1 1 is even parity and the last two bits are called frame control signals, these are two are frame control. They will be used for only asynchronous mode only. This is going to decide the number of stop bits and this is 0 0, this is invalid means in case of synchronous where this stop bits are not required probably this can use 0 0, 0 1 means its one stop bit , 1 0 means one and half stop bit.

So, this one and half periods of the clock, 1 1 is 2 stop bits. I n the earlier example you have taken 2 stop bits.

(Refer Slide Time: 52:12)



So, this is the format of mode word register which is the first part of control word register and the second one is command word register. So, the last bit of command word register is transmitter ready this is D naught bit, this is transmitter ready or the transmitter enable in fact, say if this bit is equal to 1 means transmitter will be enabled, 0 means disabled. Then this is data terminal ready; 1 means data terminal ready as I have told data terminal ready will be the transmitter basically, 0 means not ready. Then we have receiver enable, this is transmitter enable, this is receiver enable, this is equal to 1 means receiver is enabled, 0 means receiver is disabled. This is S B R K send break characters; this is send break character.

So, if this bit is 1; it means it will force the transmitter to low. If this is 0 normal operation, then the next bit is error signal E R this is D 3, D 4 error reset. So, this is equal to 1 means it will reset the error flag. So, received character can be, can have error also if this is 0 means it will set the error flag. Then we have RTS I R that enable set E H these are the different signals D 5, D 6, D 7 ok RTS; ready to send, I R this is equal to 1 its ready for sending 1 means 0 means not ready and then we have I R internal reset 1 means reset, 0 means set. This is also error control signal. So, in the last bit is error control this is equal to 1, if error is if sync pulse is deducted. So, we know that in case of synchronous transmission we will use a sync pulses whereas, in case of asynchronous we will use start and stop bits.

Thank you.