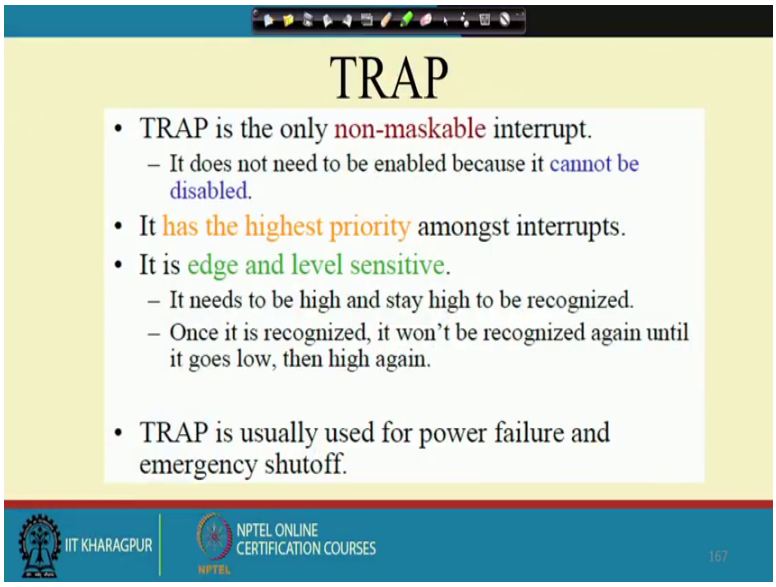
Microprocessors and Microcontrollers
Prof. Santanu Chattopadhyay
Department of E and EC Engineering
Indian Institute of Technology, Kharagpur

Lecture – 19
8085 Microprocessors (Contd.)

So, the other interrupt that we have in 8085 is the trap interrupt, so this is equivalent to RST 4.5 and this is the only non Maskable interrupt so you cannot it is not affected by that E I, D I instructions so it is always enable. So, it does not need to be enabled because it cannot be disabled so there is no question of enabling it.

(Refer Slide Time: 00:39)



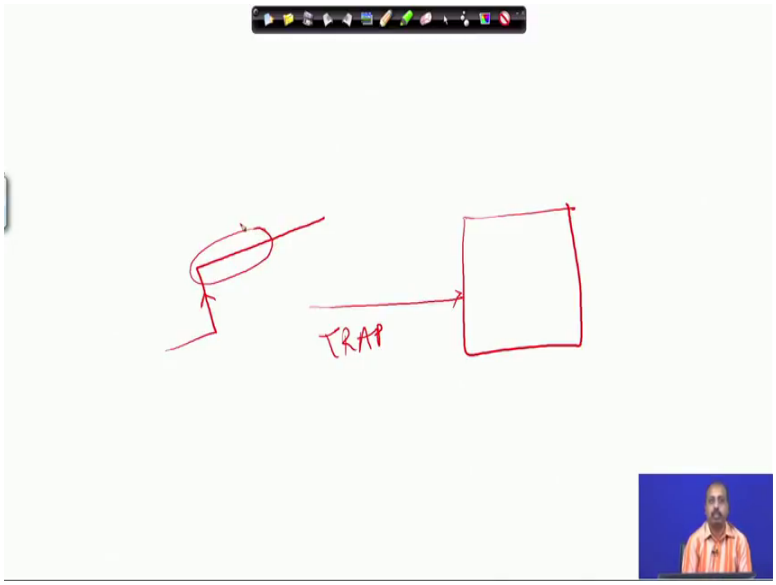
And the priority is the highest among all the interrupt, so this is the high this is the highest priority interrupt so even if some other interrupt is occurred and we are in the ISR even if that E I instruction is not yet executed if there is an interrupt of the trap lines so it will be sensed by the micro processor and it will go into the processing of it, so that makes it a very useful for very useful interrupt for handling critical situation.

Second important point to note is that it is usually used for power failure emergency shutoff so this type of cases so this type of applications this trap interrupt is used. So, power failure is a situation where if the power goes off so there is definitely batter backup, but the emergency operations that are going on there we need to take care log of that. So, that after sometime the battery will also go off so at that time I we will be able

to restart the operation from the from that point onwards, so or maybe the system has got some modules which are power hungry particular peripherals their power hungry and those peripherals had to be shutoff they take along the battery.

So, that way this system designer can decide like what are the components that needs to be shut down when the normal power goes off and it is on the battery backup. So, this trap interrupt line can be connected to the power supply from the power supply point and if there is a power failure then the system will get an interrupt and the essential activities can take place. The important thing to notice that it is the both edge and level sensitive so for sensing of this interrupt so the edge should be there and the level should also be level is also taken level is also to be considered ok.

(Refer Slide Time: 02:35)

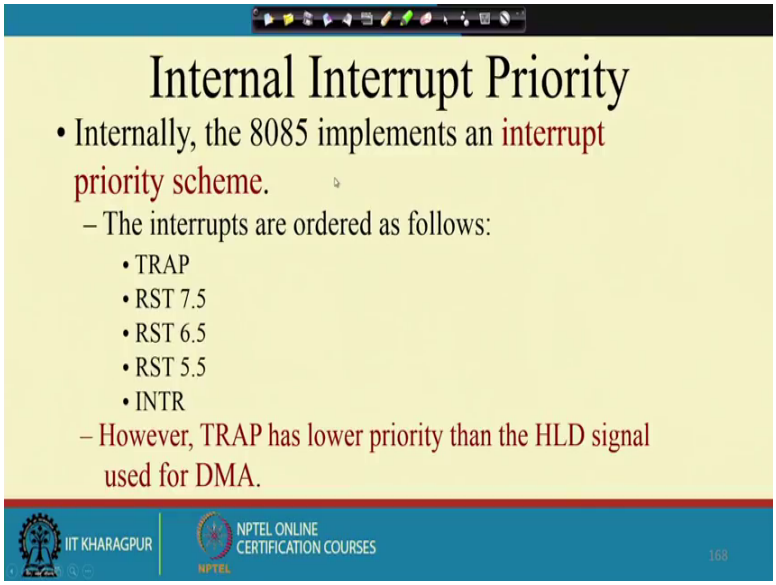


So, it is like this; so if this is microprocessor and this is the trap line so this is the trap line. So, what is required is that? There will be an edge followed by this level so then there is the minimum duration for which this the level should be the signal should remain high, so this edge should be there plus this level should be there so it is connected to it is it will be sense only if both of them are there, so this makes it slightly different compared to the simple edge sensitive one and the level sensitive interrupts. So, edge so level sensitive interrupt means they need to be turned on for quite some time and edge means that some it will be any activity occurring so will get sensed.

So, both of them have got problems because when the power supply line you are connecting to the interrupt pin, so if there is a flicker in the power supply lines it will be a taken as an interrupt if it is only edge triggered, on the other maybe that the flicker comes and goes it is not affecting the system too much so you do not need to shutdown the other shutdown other I O devices. So, in that case that level since it is the minimum duration of level so that will come into picture to interrupt will not be sensed that way so this is this is important, so that is why it is it has been made both edge and level sensitive.

So, it needs to be high and stay high to be recognized so that is the first thing and once it is recognized it would not be recognized until it goes low and then high again, so there is low to high transition is important and it has to remain high for some times so that is also important.

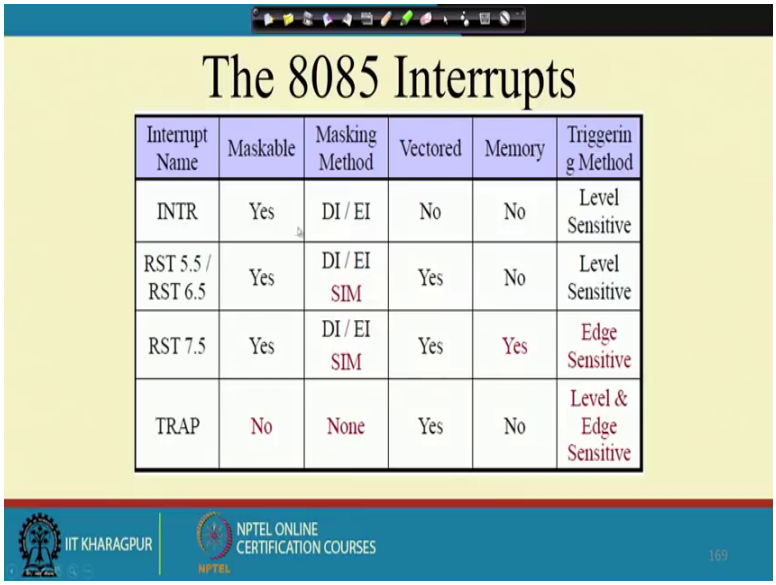
(Refer Slide Time: 04:23)



So, if we looking into this internal into priority of this interrupt that internal priority, so that trap has got the highest priority followed by; 7.5 RST 7.5, RST 6.5, RST 5.5 INTR. So, this is the order; however, trap has lower priority than the hold signal used for DMA so will come that, so there are if you remember there are 2 pins hold and hold acknowledge available in 8085 so they are used for a particular type of data transfer which is known as direct memory access or DMA.

So, in those cases this signal will be activated. So, if trap and hold both are activated simultaneously then hold has got the higher priority than the trap.

(Refer Slide Time: 05:16)



So, why this thing happen? So, we will see that later once we go to the DMA controller part so then will explain this thing. So, to summarize so we have got all these interrupt so INTR, RST 5.5, 6.5, 7.5 interrupt out of them all of them are Maskable accepting this trap, so masking method for INTR we do not have this individual flip flops so we have got this internet enabled flip flop.

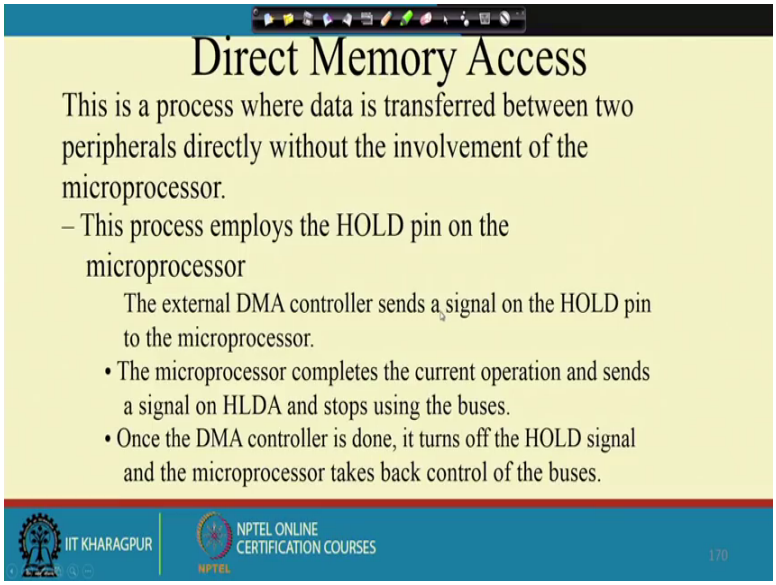
So, this D I, E I instructions can be used for masking this interrupt pin INTR pin whereas for 6.5, 5.5, 6.5 and 7.5 we have got two ways to do this masking one is D I, E I another is by using the SIM instruction, so we know that there is the individual flip flop for enabling disabling of these RST's, so they can be utilized is in the SIM instruction and D I, E I will be affecting the interrupt enable flag so as a result entire thing will be disabled, trapped cannot be masked so this is the only non Maskable interrupt that 80 85 has.

Next is if you try to classify with respect to the vectoring, so INTR is not vectored because so when INTR occurs there is no fixed address to which the processor with jump compared to 5.5, 6.5 and 7.5, so we know that these values multiplied by 8 so whatever be the location so the processor will jump to that location and it will expect that the ISR for the for the interrupt will be starting from that address.

Whereas, for INTR you do not have anything so for INTR what happened? INTR the processor enters into an interrupt acknowledge machine cycle and then it expects that on that cycle the address of the ISR we will put on the database by the device and the processor will take that address and then branch to that particular address. So, that way the INTR goes, but 5.5, 6.5, 7.5 and trap also so all of them are these they are all vectored interrupt so to get the vector address so you just multiplied this number by 8 so trap is 4.5 so multiply 4.5 by 8. Memory so whether they rem whether they can remember that something has happened so when the processor was doing something else maybe another ISR at that time the interrupt were disabled maybe, so at that time something has happened whether the system can remember that they are there is this interrupt has occurred.

So, 5 INTR 5.5 and 6.5 for them we can we do not have any memory since trap also do not have any memories, only 7.5 has got this feature so remember that there is a memory element associated with 7.5 that stores the occurrence of the interrupt, so that later on the processor may check the occurrence of that interrupt and look into that. Triggering method so this is INTR is level sensitive 5.5, 6.5 they are also level sensitive, 7.5 is 8 sensitive and this trap so this is both the level and edge sensitive, this way we can have different classification of the interrupts of 8085 from different angles if you view so you can come up with different classifications.

(Refer Slide Time: 08:40)



Next we look into another way of data transfer between the processor and the device, so this is known as direct memory access. So, in a normal system what happens is that whenever the IO device wants to transfer some data so the philosophy maybe like this; that the, device sense and interrupt to the 8085 and as a result of 8085 goes into the corresponding interrupt service routine and that interrupt service routine it reads from the device the data that it wants to transmit or if there is some data to be out put it to a device then the when the device is ready can send an interrupt to the processor telling that i am ready and then the processor can start transmitting the data to the device through the database.

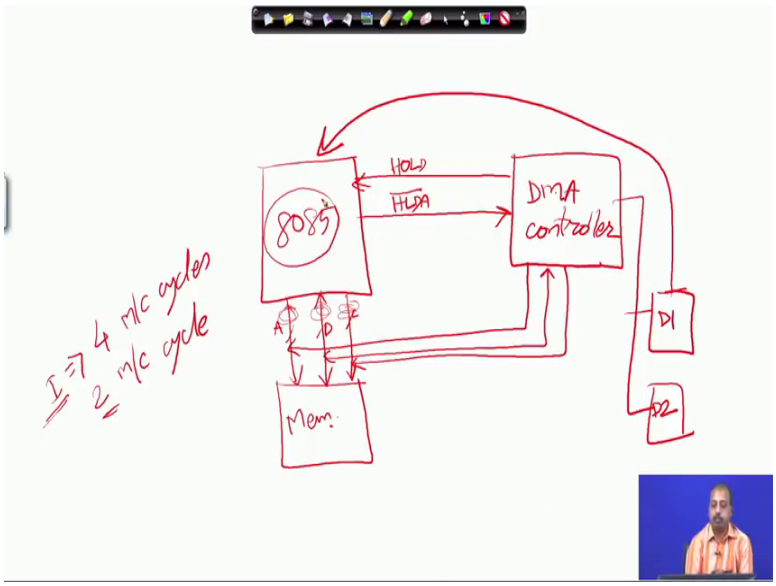
However, what happens is that? These devices that we are considering they are most of the time they are mechanical device or electro mechanical devices and many a times they involve human being like; say, there is a keyboard if i connect keyboard to the microprocessor then when the user is pressing a key may be the interrupt is given in the processer that the key that a key has been pressed.

So, accordingly the processor will go into the ISR for that, now this users are much slower compared to this is the electronic processor that we have considering. So, if the every time user presses a key so the it is interrupted so that way it quite some time or many a time we need to transfer data from the secondary storage which is likes say; disk to the primary memory that you which is the ram maybe in the system, so when you are considering a complete computer system that has got processor memory and secondary storage, so many a times we need to transfer the bulk of data from the secondary storage to the primary storage or wise versa from the primary storage to the secondary storage.

Now while this thing is going on since the process cannot do anything so possessor has to remain idol so it cannot do any operation, so to facilitated this process so this direct memory access is a procedure by which we can transfer data between two peripherals directly without the involving the microprocessor. So, there is a hold pin in the 8085 processor this hold pin is used for this purpose, the DMA controller sensor signal on the hold pin to the microprocessor when this so the hold pin is activated by the external master external DMA controller and the microprocessor will complete the current operation and send a signal on the hold acknowledge and stop the all the buses ok.

So, all the buses and now floating and it is given to the DMA controller. So, DMA controller can now use the bus and once it is done once the DMA controller has finished the operation it can turn off the whole time and the microprocessor can take back the control of the bus, so will explain it with respect to one DMA controller.

(Refer Slide Time: 12:07)



So, suppose you have got as a system consisting of say consisting of say the processor this is the 8085 processor and we have got so this 8085 processor has got the memory connected to it and there is address bus data bus and control bus, so this is the address bus this the date bus and this is the control the read write control, now if there is another d m a controller so this is another chip which is sorry so this is the d m a controller, so there are some standard chips available so for that several such chips are available.

So, DMA controller can it has al also got this address data and control lines so they can be connected to the DMA controller, so this address data and controller lines can also be can also be given by the DMA controller and it has got two extra pins one is the; one is to give the hold single to the microprocessor, so this is the hold line and in turn this microprocessor will respond with the hold acknowledge.

Now in this situation what happens is that whenever this now if i have got some IO device so this IO devices are connected to the DMA controller, so these are the; this is a device 1 this is a device 2 like that.

So, when these devices are already for data transfer what to the system so what will happen is? That the, they will tell the DMA controller and DMA controller in turn will send a hold request to the 8085 processor. So, 8085 processor it was executed some instruction so maybe it was executing say; instruction I, which takes a total of say 4 machine cycles and at present it is in the second machine cycle, when this hold has occurred when this hold signal has been activated the processor is in the second machine cycle of the instruction.

So, what will happen? The processor will immediately suspend it is operation if it is going to accept that hold request so it will suspend it operations it will send the hold acknowledge signal to the DMA controller and it will release the control of these buses. So, this processor will no more control buses so now we can understand that the processor cannot do any memory read write operations now, so there buses are released so after releasing the bus is the hold acknowledge activated so DMA controller you will know that the buses and now free, now the DMA controller will execute and it will transfer the bits between the device and the memory using this address data and control buses, so this is how this hold operation is done and once this DMA controller is done with the all the data transfer that it has to do it will take off this hold signal and the 8085 will understand that the hold signal has gone off, so it can gain back the control of these the 3 buses and continue it is normal operation.

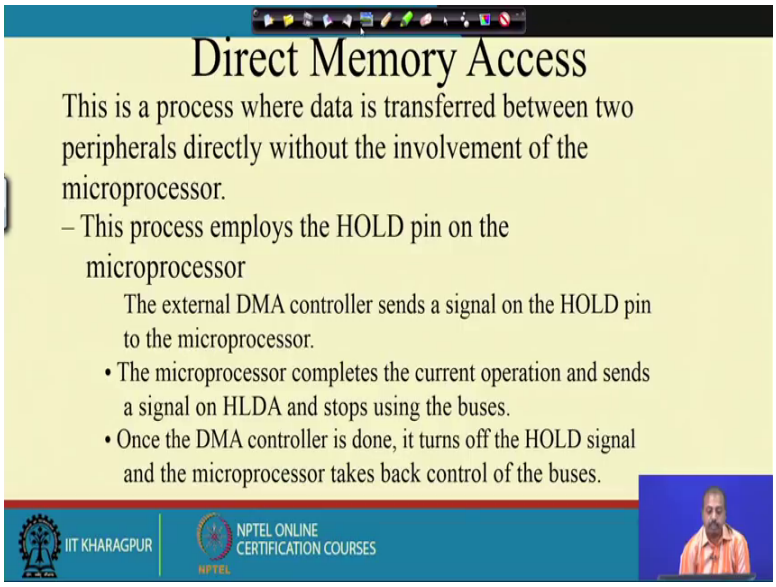
So, you can in some sense you can say that since this the processor is not using this external buses. So, 8085 from the outside world you can say that as if this entire operation has frozen so it is not doing anything so it entire operation is frozen and unlike interrupt where if the interrupt had if an interrupt had occurred when the processor was executing this instruction I which takes 4 machine cycles and interrupt has occurred in the second machine cycle.

So we said that the processor completes the current instruction and then it goes into the interrupt service routine whereas, for this DMA operation so it does not happen like that so it completes the current machine cycle only and it goes to the DMA operation because it is not doing any memory access, so nothing is going to happen inside the processor that is why it does not need to take care of this they are saving of it does not need to take care of completion of current instruction and all that so nothing so all the register contents are frozen you can say.

So, when this hold comes back when hold become deactivated, then the PC still contains the correct value so the instructions continues from the wherever it had stopped when the hold signal has occurred. So, this way this DMA controller can be operated can be interfaced with the 8085 and that way we can do data transfer between device and memory which is very fast otherwise what will happen is that?

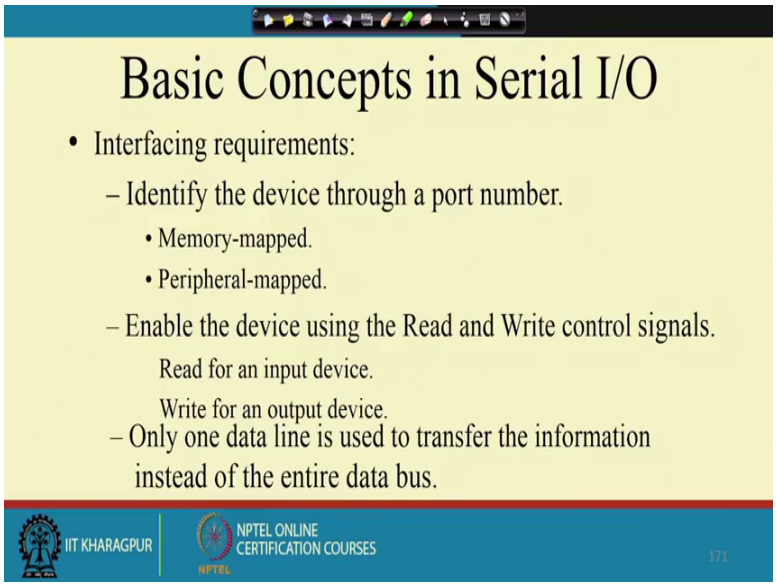
This 8085 it has to go in to if this interrupt defenses service like if this D1 is allowed to interrupt the processor directly, then in that interrupt service routine so it has to read the data from device and put into memory so byte by byte it has to do that transfer and possibly that is an overhead, so in later processor so we can see that when this DMA is activating so 8085 this processor can do the internal operation. So, that way we can have some parallelism as well.

(Refer Slide Time: 17:44)



So, this is the DMA direct memory access philosophy. So, we can have this thing we can have this microprocessor will be put into a hold condition and the DMA controller will be taking care of this transfer.

(Refer Slide Time: 18:00)

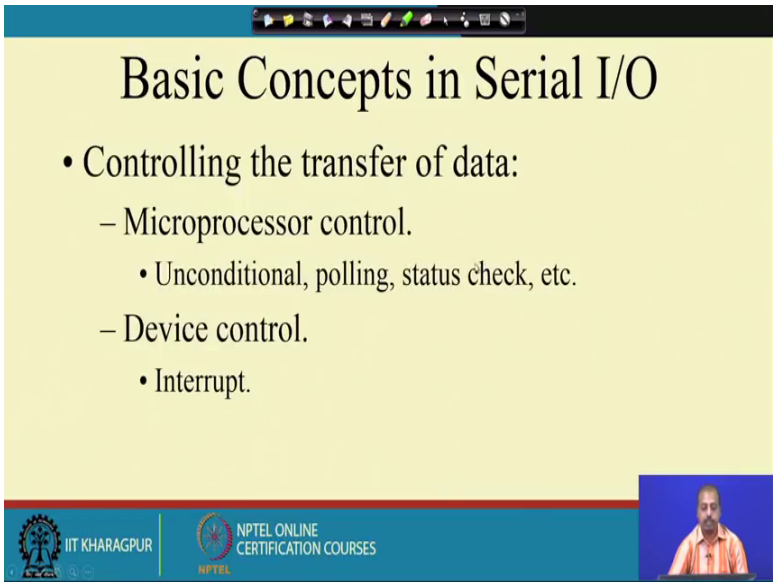


Next will look into another very important concept which is for the serial input output operation, the serial input output operation so we have to we want to transfer data serially like; while looking into that SIM and RIM instruction so we have seen that there are serial data input and output and also those bits were there in the RIM and SIM instructions and also we have seen that there are serial in SID and SOD pins are available in 8085 through which this serial input can be activated, so a very simple way of data transfer from processor to the to the device may be serially through this SID SOD pins.

So, to do this operation so we have to we have to transfer we have to look into this serial IO part so with the basic requirements are like this; so need to identify the device through a port number, so this one maybe mapped on to the memory or it may be mapped on to the peripheral so which is identified by that IOM bar line. So, we will come to this the later, then we enable the device using read write control we read for an input device and write for an output device and only one data line is used to transfer information instead of the entire data bus.

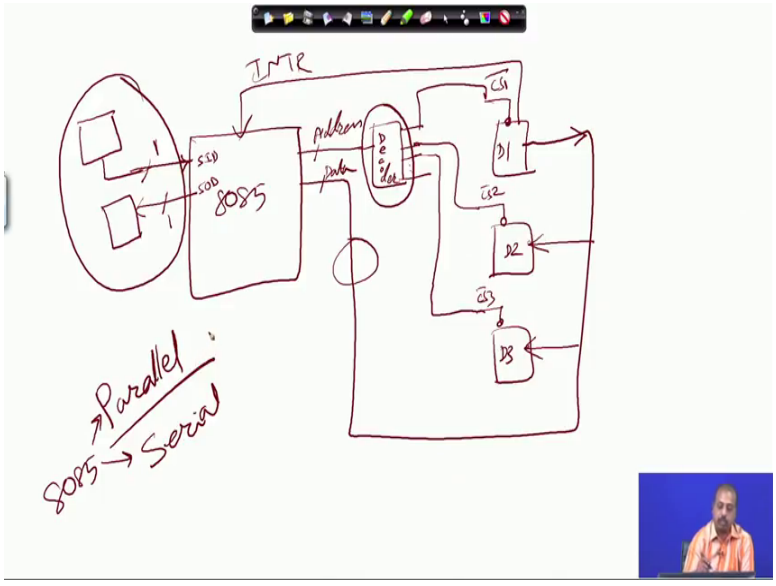
So, this is how this is serial data transmission will take place, so we need to have some number then we the to identify the device then we should have some read write control for reading from the input device writing of the output device and we have to have only one be only one data line can be used for transferring the data.

(Refer Slide Time: 19:49)



So, will take some example; so when it is microprocessor controlled so it is unconditional polling or status check so these are the things to be done, so maybe by microprocessor transfers the data from one IO to memory to the device or it may be interrupt driven.

(Refer Slide Time: 20:38)



So, this is this will be true when we are having say interfacing using say normal operation so say when we are having say this is they.

Suppose we have got this as the microprocessor 8085 and we have to we have to con we have connected a number of devices, so this is D1, D2, D3 like that. So, each device will have some address and that address is identified by something like some sort of say chip select signal, so this chip select signals will becoming so CS bar 1, CS bar 2, CS bar 3 some chip select signals can be there.

So, the address lines that are generated from the microprocessor the address bus, so this address bus it has to go through a decoder address decoder and accordingly it will generating a number of enable lines and this enable lines are to be connected to the chip select lines this enable line has to be connected to the chip select line so that based on the address that is generated so one of these devices will get selected and from the date of bus i can connect to all these devices this data bus can be connected to all these devices.

So, it may be if it is i say if D1 is an input device so the data bus this direction will be like this, if D2 is an output device direction will be like this, so that D3 maybe another output device direction will be like this. So, this is the data bus from the microprocessor and also we have to give the read write signals the read write signals are to be activated, so i am not showing it explicitly that, but that read write signals will be there.

Now, so this type of transfer so how this can happen? So, this 8085 so can the processor can determine i will send some data to D2, so it will send some it will put the appropriate address on the bus and this D2 this chip select will be activated and the data will be sent through the D2 line to the database to the D2 device, similarly when it decides to send something to D3 so it can do that way, so that be one possibility.

So, other possibility is that instead of connecting it like this so this devices when they are ready there sent interrupt to the 8085, through interrupt it can they can be connected through the interrupt line and as we have seen that if i have got multiple devices then i can have in between an interrupt controller and that interrupt controller maybe this it can resolve the priorities and sent interrupt to the processor so it can do it like that. So, that ways in case of interrupt driven what happens is that? When the interrupt come the processor will go to the interrupt service routine and then it will do that transfer.

So, this type of communication that we are talking about so they are they are basically the parallel type of communication they are parallel communication because we are sending 8 bits of data be simultaneously between the processor and the device, but the

interfacing this devices the challenge that we have is first of all we have to have this decoder plus we have to have these 8 lines running so many lines need to run through the chip. So, instead of that; so what we have in 8085 is there is another facility which is the serial input so SID and SOD pins are there.

So, anything that you want to input from a device so if you can connect it to the SID the serial input data, so it can be connect it can transfer data it can read data from the device serial in 1 bit at a time so this line width is 1 bit, similarly we can have the SOD serial output data if I want to output something to the device so again the 1 bit connection can be there. So, instead of parallel connections of this communication where facility that we are talking about so this is the serial communication, so 8085 it supports both the types of communication parallel communication and serial communication and this parallel communication can again be classified into classes so this like control this transfer of data.

So it can be unconditional, so unconditional means that whenever the processor feels like; so it will transfer the data to the device, pulling means the processor will ask the device one at a time that is it ready for the transfer so then the data will be transferred that way, status check so it will check the devices so that the device is ready and then pulling and status check more or less same they actually the device the processor will regularly pull the device to check it is status and if the status is the device is free when it can be transferred so this is under the microprocessor controlled facility.

Under the device controlled facility so there can be interrupt, so device will send an interrupt to the processor and when that interrupt comes there are the processor will transfer the data. So, we have so these are the so challenges that we have are like this; so we have this requirement to identify the interface so for that we have to have some port number through which the device are to be accessed, so we will see this memory mapped peripheral mapped concept later. Then we have to activate this read write control signals and we can transfer only one data line they can be so instead of doing that instead of doing this parallel transfers if you use a single bit transfer then will come to the serial IO operation.