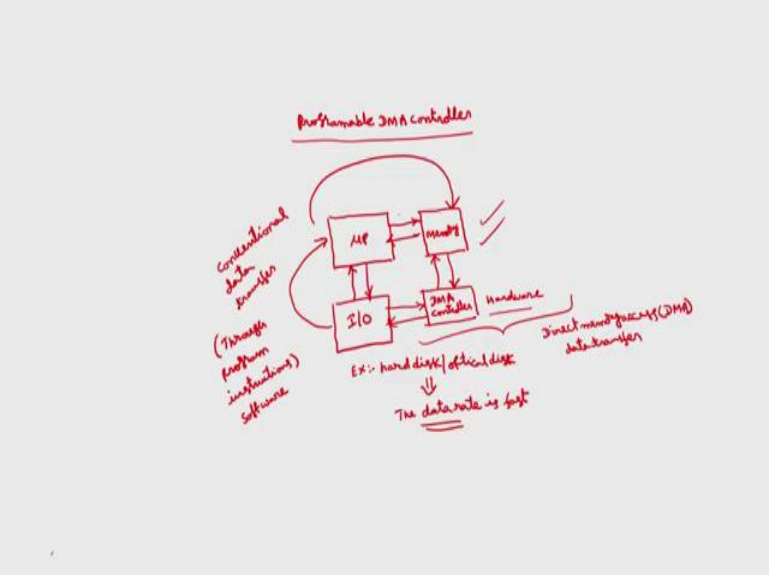
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture - 31
8237 Architecture, Interfacing and Programming

(Refer Slide Time: 00:51)



In the last class, we were discussing about various peripheral devices such as 8255 programmable peripheral interface, 8254 programmable interval timer, 8259 programmable interrupt controller. The next, I mean peripheral device is programmable DMA controller, programmable DMA controller.

So, before going for this programmable DMA controller; I will discuss first, what is necessity of DMA controller. So, we know that so, the data transfer between the memory and peripheral will take place through the microprocessor. If we take the microcomputer block diagram, this is the microprocessor, then we have the memory, then we have I/O. We have bidirectional signals from memory to I/O and I/O to microprocessor.

So, the normal way of data transfer will be. So, if you want to transfer the data from I/O to memory or memory to I/O say, for example, I/O to memory. So, I/O will write onto the microprocessor then microprocessor to memory. This is the conventional way of data transfer. But, the drawback of this conventional data transfer scheme is if I use this I/O

device such as a fast device if I/O device is, if I use the disk, floppy disk or hard disk or it can be an optical disk such as a DVD CD.

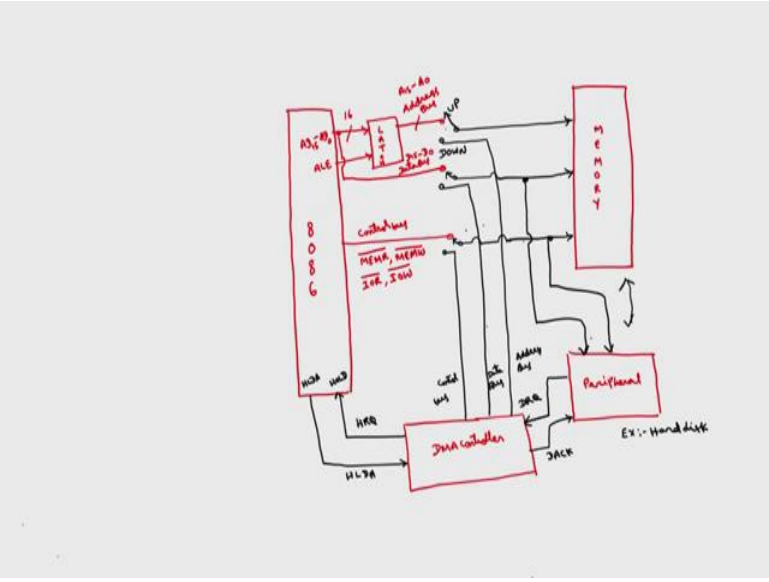
The rate of data transfer from the I/O is very fast; in such cases the data rate is very fast. So, data rate is fast so, normally in case of conventional data transfer how this data will be transferred from I/O to memory is through the program instructions. So, in this conventional data transfer, the data transfer takes place through program instructions that is software; it is software type of data transfer. So, the drawback of this type of program instructions or software type of data transfer is.

So, in order to fetch the instructions and to execute the instructions, it will take some time. So, the time that is taken to fetch the instructions and execute the instructions is less than that of the data rates of the hard disk or optical disk. Since the result of that it become difficult to transfer the data from I/O to memory through microprocessor using software instructions. So, in order to avoid this problem; we need a hardware element, dedicated hardware element which is faster than the program instructions which can enable the data transfer between the I/O to memory directly.

So, for that we require a Direct Memory Access controller, DMA controller. So, this is the hardware device unlike in case of conventional data transfer where the data transfer takes place through software instructions. The drawback is execution of this instruction will take more time and the data rate of I/O devices is large in case of a hard disk and optical disk so, it becomes difficult to transfer. Now using this DMA controller, what we will do is so we will transfer the data from I/O to memory through DMA controller. This type of data transfer is called direct memory access data transfer; Direct Memory Access, DMA data transfer.

It is obvious that because this is a hardware device. So, this is faster than the software type of data transfer. This is the advantage of DMA control and need of the DMA controller. Now if I draw this diagram in a somewhat detailed manner. So, that we can understand what is the step by step procedure that will take place to transfer the data from I/O to memory directly without involving the microprocessor. There is some step by step procedure is there. So, if you consider this detailed diagram of DMA controller this will be something like this.

(Refer Slide Time: 06:31)



There is a microprocessor so, this can be any microprocessor here I will consider 8086 then we have memory, the same previous diagram I am going to show in detail. We have memory and then we have peripheral, then we need a DMA controller, direct memory access controller.

The connection details are 8086 will be having address bus, data bus and control bus. So, in order to transfer the data from peripheral to the memory or memory to peripheral, the DMA controller must contain this address data and control bus, but these are with 8086; for that we need a switch type of configuration. So, through this switch configuration we can transfer the control of this address data and control bus to the DMA controller whenever needed ok.

So, we will be having address bus see only we are assuming that only 16 bit address is enough. So, we will take A15 to A0, but this is multiplexed in case of 8086; we know that address bus and data bus are multiplexed. This is actually in fact, will be available as AD15 to AD0, this is AD15 to AD0. This is 16 lines, in order to demultiplex this address and data bus, we require a signal called Address Latch Enable, ALE and this is LATCH.

So the output address will be latched, here this will be address bus which contains A15 to A0. In the first cycle, we latch the address. So, in the subsequent cycles we will use the same bus to carry the data. Now these will act as data bus D15 to D0 then, this will be having some control bus this is control bus the signals that are contained in control bus

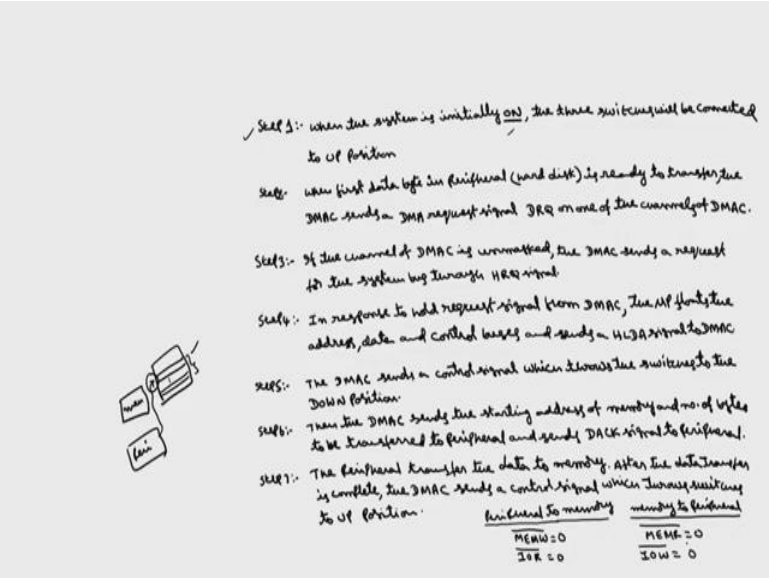
are memory read, yes I have discussed in the very first class of this course. The microprocessor can perform 4 operations memory read, memory write, I/O read, I/O write.

All these signals are normally active low signals because NAND gate is a universal gate. Then, this memory also will be having address bus, data bus and a control bus. So, we are going to connect this memory address bus to one of the switches. So, this is coming from DMA controller. So, we have address bus of DMA controller, data bus of DMA controller and control bus of DMA controller. So, these are to be connected through switches to memory. This will be connected to memory through switches. There will be a switch now here, there will be switch here, there will be switch here, this is up position, this is down position.

So, this is control bus of DMA controller, this is address bus of DMA controller, this is data bus then, this data and control bus will be connected to peripheral also. Then, between the DMA control and 8086 there will be two signals. So, one is called hold request signal HRQ, HLDA signal this will be connected to hold signal of 8086. We know that in minimum mode, there is a hold signal for 8086 and this will be connected to a HLDA, hold acknowledge, these two are minimum mode signals. And between these peripheral and DMA controller there is a DMA request signal, then DMA acknowledge signal. This is called DMA acknowledge signal and this is DMA request signal.

So this is the complete diagram of the DMA controller with 8086 memory and peripheral ok. Now what is the step by step procedure that need to be performed to transfer the data directly from peripheral such as hard disk to memory? So, you want to directly transfer the data from peripheral to memory, but this will happen through DMA controller.

(Refer Slide Time: 13:15)



So, if you coming for this step by step procedure; Step 1: when the system is initially ON the 3 switches will be connected to the UP position. So what does it mean? The address data bus and control bus will be connected between the memory and microprocessor, its clear from the diagram that. So, initially whenever the system is on, there will be a physical connection between the microprocessor and memory and DMA controller will be bypassed ok. So, whenever you want to transfer the data from peripheral to memory. So, the next step is; when first data byte in peripheral which was hard disk is ready to transfer to memory.

So the DMA controller, I will call DMA controller as DMAC sends. You can see that there is a connection between DMA controller and peripheral. There are two signals; the first signal is DMA request signal sends a DMA request signal DRQ on one of the channels of DMA controller.

So DMA controller can have any number of the channels. So, later we are going to discuss a specific DMA controller 8237 that contains 4 channels. So, this DMA (Refer Time: 15:56) is a peripheral can place the request on any one of the channels. The next step is step 3; if the DMA channel of DMA controller if the channel of DMA controller is unmasked, DMA controller is unmasked. If it is masked, then that service will not be recognized, if the channel on which the peripheral device has placed the request if that channel is unmasked.

Then in response to this DMA request because they have to transfer the data, the DMA controller needs address bus, data bus and then control bus. But now, initially when the system is on this switch is connected to the up position as a result of that the address bus, data bus and control bus will be connected between the memory and microprocessor. Now in case of DMA, we need to connect the address bus, data bus and control bus between DMA controller and memory ok.

So, for that the DMA controller needs the use of the system bus which is address bus, data bus and control bus. So, it will request for the system bus through hold signal. If a DMA corresponding channel is unmasked, the DMA controller sends a request for the system bus, system bus consist of all the three buses address bus, data bus and control bus through hold request signal HRQ signal. You can see this signal in the block diagram that I have drawn.

This HRQ signal is connected to hold signal of 8086. So, once this microprocessor receives hold request signal from DMA controller, then the microprocessor floats the buses. In response to the hold request signal from DMA controller, the microprocessor floats the buses, address data and control buses and sends the hold acknowledge signal to DMA controller. So, once the DMA controller receives hold acknowledge from microprocessor, the DMA controller sends a control signal to the switches.

So, then the switches will be thrown to the lower down position. The next step is the DMA controller sends a control signal which throws the switches to the DOWN position. So, it is clear from the diagram that when the switches come to the DOWN position, there will be a physical connection between the memory to DMA controller. No longer this microprocessor will be available. Now, the connection for the microprocessor will be bypassed.

Now once the DMA controller takes the control of the system bus, address bus, data bus and control bus, then it is ready to transfer the data between the peripheral and memory. Then the DMA controller sends the starting address of memory and the number of bytes to be transferred to DMA peripheral and sends acknowledge signal say DMA acknowledge signal DMA, DACK signal to peripheral.

Now, I want to transfer the data between this peripheral and memory; this is peripheral and this is memory. So, in the memory we have lot of locations from which location

onwards you have to transfer the data, this address will be send by the DMA controller and how many bytes of the data you want to transfer totally. So, these two are the important factors ok, these two will be sent by this DMA controller through this DMA acknowledge signal. It will tell this DMA peripheral device to be on for the first byte of data transfer.

So, once this peripheral receives DMA acknowledge signal, then the peripheral places transfer the data to memory. After the data transfer is over is complete, then the DMA controller sends a signal control signal which throws switches to the UP position. So, again the microprocessor regains the control of the system bus; address bus, data bus and control bus. This is how the exact DMA data transfer will take place; this is the step by step procedure.

So, initially the first step is. So, the system whenever the system is on, there will be physical connection between the memory and microprocessor that time DMA controller will be bypassed ok. So, whenever the first data byte in peripheral is ready. So, this peripheral will first request the DMA controller through this DMA request, then in response to that the DMA controller because this requires the control bus, data bus and address bus; it will request the microprocessors through hold request.

So, in response to that the microprocessor will respond through the hold acknowledge and then it will I mean tri states or it will floats, floats in the sense tri states. So, we know what is meant by tri state we have discussed in the earlier classes. The control of the address bus, data bus and a control bus to the DMA controller, then the DMA controller will send a control signal to throw the switches from up position to down position. Now your physical connection will be established between the DMA controller and memory.

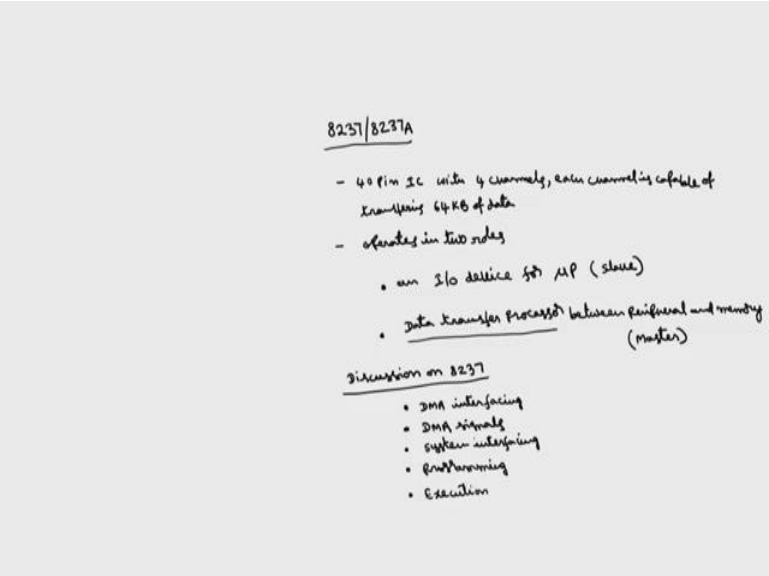
Then these DMA controller will places. So, which memory location onwards you have to transfer the data from peripheral to memory or how many number of bytes to be transferred. This data will be placed and it will be sent to the DMA controller. Now DMA controller will transfer the bytes, once the data transfer is over again the switches will be thrown to the up position. So, that they again the physical connection between the memory and microprocessor will takes place. This is the step by step procedure that will takes place to transfer the data between the peripheral to I/O directly without involving the microprocessor.

Now, here the data transfer will take place through the DMA control which is a hardware device because of that the rate of data transfer will be fast when compared with the software data transfer which takes place through the microprocessor. This is about the DMA controller. In the similar way if you want to transfer the data from the memory to I/O also the same process will repeat. The only difference between this peripheral to memory or memory to peripheral that depends upon the 4 signals.

So, in case of peripheral to memory so reading the data from the peripheral which is I/O, you are writing onto the memory I/O. This DMA controller sends 2 signals. We know that totally we have 4 signals control signals which is a memory read, memory write, I/O read, I/O write. So, in peripheral to memory; memory write will be enabled and I/O read will be enabled because you are reading from the peripheral which is I/O. So, I/O read is 0, we are writing onto the memory, memory write will be 0 whereas in case of memory to peripheral other 2 signals will be low. Memory read will be 0 and I/O write will be active low.

Except for this there is no difference between the data transfer between the memory and peripherals. So, with this discussion, now we will go to the a dedicated DMA transfer controller which is called 8237 or 8237 A.

(Refer Slide Time: 28:09)



So, 8237 or 8237 A. There are some slight advanced version ok. So, you see a dedicated DMA controller. So, a discussion that they have made in the earlier slide is a general DMA controller. Now we will come to the specific DMA controller.

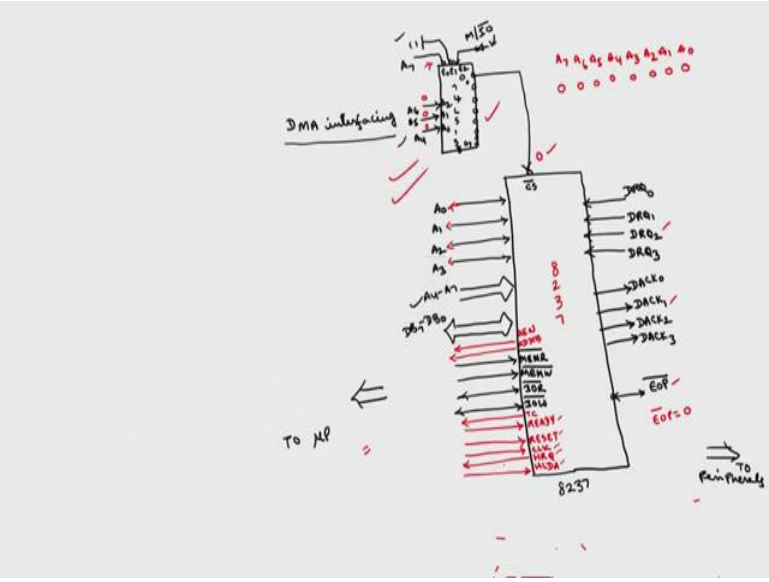
So, the features of 8237 or 8237 A is this is 40 pin IC with 4 channels. As I have told in the previous procedure so, this peripheral device will place the request on one of the channels. The number of channels in a DMA controller varies from the device to device. In case of 8237; there are 4 channels and each channel is capable of transferring 64 kilobytes of the data. So, this is the 1 feature of 8237 and another feature is 8237 or any DMA controller in fact, 8237 can operate in two roles. So, one is I/O device for microprocessor and another one is, it can also be used as a data transfer processor between peripheral and memory.

So when the switch positions are up, it will it acts as just simply I/O device. Whereas, in case of DMA it will acts as a processor, because the processor is no more available microprocessor will be bypassed in case a DMA data transfer. So, DMA controller itself will acts as a DMA processor, this mode is called slave mode because here the master is microprocessor. So, in the second role this DMA controller will acts as a master; this is called master mode. So, this DMA controller can operate in two roles one as a slave in case of I/O data transfer, in case of DMA data transfer this will acts as a data transfer processor.

Then because this 8237 is a complicated device for clarity, I am going to divide these discussion on 8237 into 5 parts I am going to divide into 5 different parts for clarity. Otherwise we actually this is a complex device so, it becomes a difficult to follow this ok. So, the first I will discuss about DMA interfacing then, I will discuss about DMA signals then, I will discuss about system interfacing; how to connect in a system then, I will discuss about the programming and then execution.

So, I will divide the entire discussion into these 5 parts; DMA interfacing, DMA signals, system interfacing, programming and execution. Coming for the DMA interfacing.

(Refer Slide Time: 32:53)



So, if I take the DMA device 8237 or 8237A this is 8237. So, total 40 pins are available ok. So, one is chip select signal for any chip, we will we require a chip select signal then we have the other side. This side is actually we will connect to the peripherals this side, we will connect to microprocessor.

As I have told there are 4 channels, each channel will be having 1 DMA request 1 DMA acknowledge. So, these are 4 DMA requests from the peripheral as I have discussed in the earlier step by step procedure. So, DMA request DRQ this is available as active high signal only, 0 DRQ 1, DRQ 2, DRQ 3. Similarly acknowledgeable signals 4; DACK 0, DACK 1, DACK 2, DACK 3, then we have one signal which is called end of process signal. I will discuss what is the function of this one EOP which is also active low signal, bidirectional signal.

Coming for the other side; so, we have 4 signals which are going to control 16 registers inside the 8237. These are bidirectional signals sorry this is unidirectional this is A0, A1, A2, A3, then we have a 3 to 8 decoder, 74LS138, this we have discussed in the memory interfacing also, this is 3 to 8 decoder 3 inputs, then 8 outputs O0 bar active low signals all, this is O0, this is output 0 signal.

So, this I will connect to chip select signal, accordingly we will get some addresses. If you want you can connect O1, O2 also O7 also this will we have total O7. So, this is A0,

A1, A2, then we have 2 active low signals E0, E1 active low signals and E2, there is active high signal. These are the different signals for a 3 to 8 decoder 74LS138.

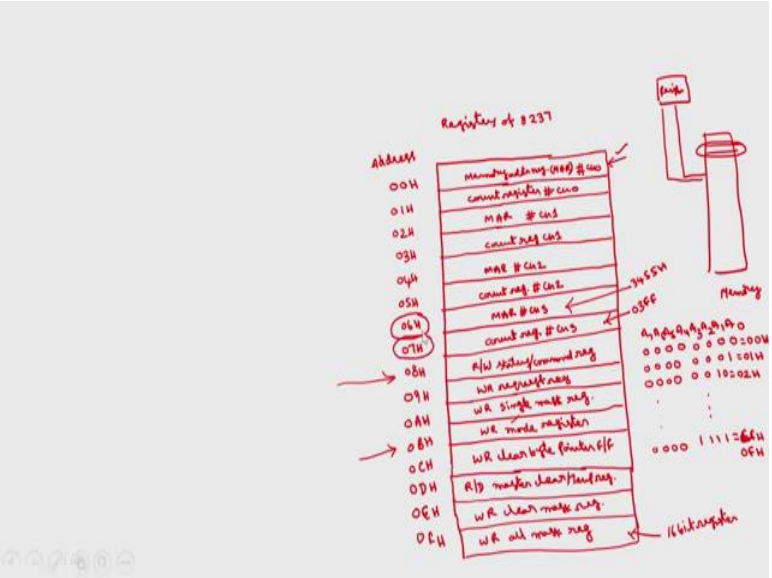
So, this I am going to connect to this A0 to A4 is there I will connect here A4, A5, A6 then, I will connect this to A7, this I will ground this is active high signal so, I will connect this to M by I/O bar through inverter. So, for I/O, this will be zero. So, I will connect a inverter here, this is m by I/O bar this signal is M by I/O bar. In case of 8085 is I/O by M bar we can connect directly. So, you have different signals then the remaining signals are we have a data bus total this A4 to A7 will be another A4 to A7 this actually in fact, are connected here ok.

And then we have data bus DB7 to DB0 data bus, then we have 4 control signals we know memory read, memory write, I/O read, I/O write. This is memory read bar, memory write bar, I/O read bar, I/O write bar. Similarly, we have the other signals READY signal, RESET signal, then we have clock signal then as you have discussed in this step by step procedure, there will be a hold request signal HRQ and the hold acknowledge signal HLDA. And there is one more called terminal count TC.

So, these are the different signals available because these are bidirectional signals. So, there are different signals available on 8237. So, total we have 40 IC or 40 pins. So, this side will be connected to microprocessor this side will be connected to the peripheral devices. Now how to generate the addresses of the different register present in the 8237 and, how to connect these to the microprocessor?

So, in order to connect to the microprocessor it requires two more signals called address enable signal and address strobe signal. Address strobe it is STB and address enable signal AEN, these are the different signals available. So, if we connect to the microprocessor so, I will show that connection of the microprocessor to 8237 in the next slide. So, coming for this the internal registers totally we have 16 registers inside the 8237.

(Refer Slide Time: 40:42)



So, the register so, of 8237. Corresponding to each channel we have two registers; one is called memory address register, another is called count register. So, this is memory address register corresponding to channel 0 memory address register, I will call as MAR corresponding to channel 0 and corresponding to this same channel 0 we have count register.

So, MAR of channel 1 and count register of channel 1 MAR of channel 2, total we have 4 channels and count register of channel 2, MAR of channel 3, count register of channel 3 so, total 8 locations. So, I will generate the addresses using the interfacing circuitry. So, I will write here the corresponding address.

So, we know what is the purpose of this memory address register, if we want to transfer the data from peripheral to memory peripheral such as hard disk, peripheral to memory. How many bytes of data have to be transferred and what is the starting address of the address? So, the starting address of this location where the first data has to be send we have to place in memory address register and the number of bytes to be transferred we have to place in count register.

So, the remaining 8 locations of this 8 registers of 8237 are this is read write, read write signal, status slash command signal, command register, then you have write request register, write single mask register, write single mask register, write mode register. We are going to define the various formats of these registers. Write clear byte register flip

flop, we have read write master clear slash temporary register. So, total we have 1, 2, 3, 4 here up to here its 8, 9, 10, 11, 12, 13, 14, we have two more registers.

So, write clear mask register, write all mask register. So, these are the different 16 bit register all are 16 bit registers. Then how to know these addresses of these registers? That will be decided by the previous decoding circuitry here in order to select this chip select should be 0, if it is if I write this I mean A7 to A0, A7, A6, A5, A4, A3, A2, A1 and A0.

So, in order to select this chip the output of O0 should be 0, so that these 3 should be 0 0 0, then only the first output will be selected. So, A4, A5, A6 should be 0 0 0, A4, A5, A6 and this A7 should be 0 because this is connected to E0 which is active low signal. So, with these 4 0s this chip will be selected after that this A3 to A0 is going to select the channels if this is 0 0 0 0 the first address, so the first address of this one will be this memory address register of channel 1. So, this address will be now 0 0 H. So, this A7, A6, A5, A4, A3, A2, A1, A0.

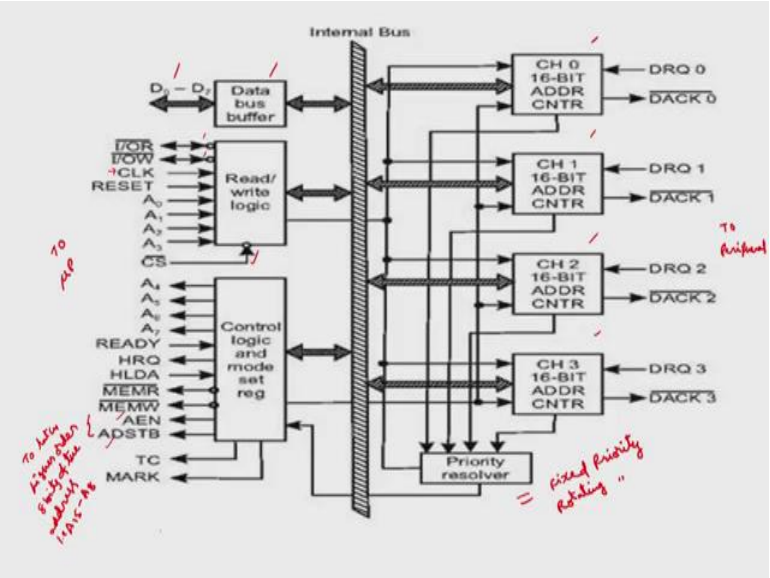
So, these 4 are 0 0 0 0 to select the IC this varies from 0 0 0 0 to so, if it is 0 0 0 0, 00 H. This can be 0 0 0 1, so there 0 1 H, 0 0 1 0, 0 2 H. So, on up to 1 1 1 1 is FF H because this remains 0 0 0 0 only for all the registers because if they are not 0 0 0 0, the IC controller itself will not be selected.

So, the address of this one is 0 0 H, address of this one is 0 1 H and so on. 02, 03 up to 0F H this is 0F H. 04 H, 05, 06 H, 07 H, 08 H, 09 H, 0A H, 0B, 0C, 0D, 0E, 0F H, this is how we can generate the addresses of all the registers.

Now, this is the second step as I have told for the clarity, I have divided this discussion on 8237 into 5 sub groups. So, you see the interfacing circuitry and second one is system. So, DMA signals, DMA signals also we have already discussed. So, we have discussed all the functions of this DMA signals ready signal is so, whenever this 8237 is ready.

So, this will be one signal will be generated similarly reset clock these are default signals and these are this DMA request and DMA acknowledge signals end of process. So, whenever this process is completed or by name, by name it implies that if EOP is 0 means process is over ok. So, these are the different signals then system interface.

(Refer Slide Time: 50:01)



So, if I connect this come to this internal diagram of 8237. So, whatever I have drawn here in the earlier slide is only just a block diagram. Now, we have the complete details here. So, we have all the signals so, this is the data bus buffer data bus buffer say which will be having 8 data lines this is read write logic. So, this chip select we have generated then we have RESET signal, the microprocessor can reset the 8237. This is clock signal the clock can be given from the microprocessor or we can give from the clock generator also. This I/O read, I/O write are bidirectional signals active low signals because in I/O we can have input operation as well as output operation.

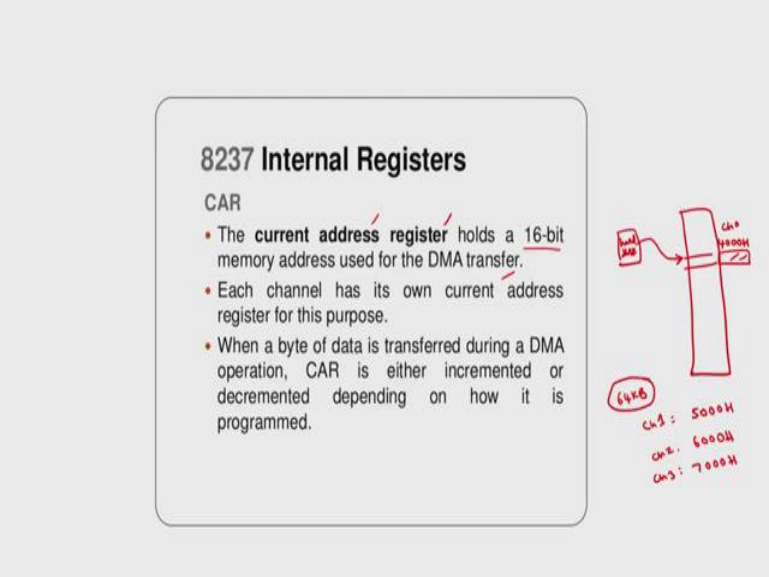
This A0, A1, A2, A3 will be used to generate the address of the registers just now I have discussed then we have A4, A3, A5, A6, A7, these are used for generating the chip select signal, then we have ready signal that will be controlled by the microprocessor. This side as I have told this side will be connected to microprocessor this side will be connected to peripheral such as hard disk.

And this is hold signal hold acknowledge signal which I have already discussed memory read memory write. This AEN and AD STB these signals are used for latching the higher order 8 bits of the address they are used to latch higher order 8 bits of the address, that is A15 to A8. Then terminal count; once if we load the count into the count register and you have to decide whether their memory locations has to be auto increment or auto decrement that you can do using the mode control word or the command control word.

Then, once this count is 0 after each transfer the count becomes automatically decremented by 1. Once the count becomes 0 it will indicate through this terminal count signal. And here these are the 4 channels; channel 0, channel 1, channel 2, channel 3 each will be having 1 request signal 1 acknowledge signal and we can assign the priority to this signals, similar to this a programmable interrupt controller where you can assign this priority here also you can assign the priority in two ways; one is called fixed priority another is rotating priority.

Fixed priority; we will fix the priority whereas, in case of rotating priority we can follow some rule like the channel which recently served, we can assign the lowest priority ok. This also can be programmed using the mode word registers. This is the internal details of 8237 and coming for these the various control words which we are going to select the modes of DMA, we have different control words.

(Refer Slide Time: 53:37)

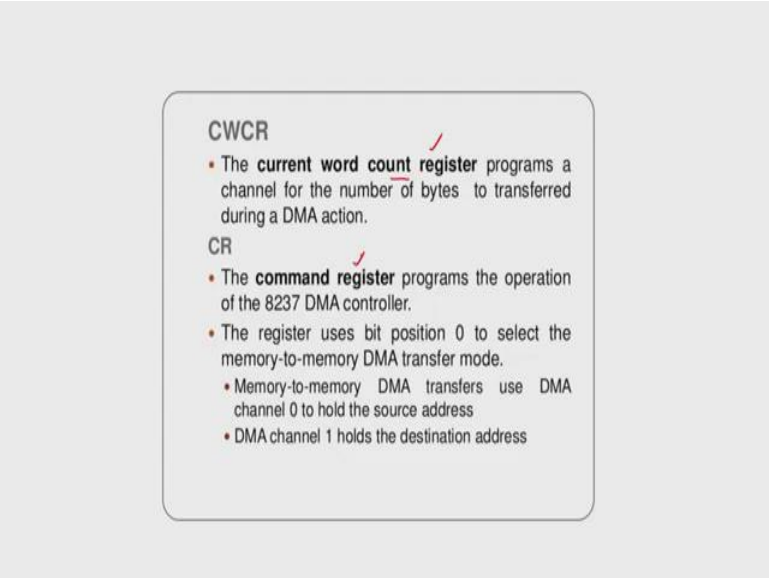


So, first one is a current address register. As I have told so, if we want to transfer the data from peripheral to such as hard disk to memory. So, where does the first byte has to be transferred? What is the address of this one? So, this will be decided by this current address register. This holds 16 bit memory address used for the DMA transfer, the starting address.

So, this each channel will be having its own address, this may be corresponding to this some channel is some 4000 is corresponding to channel 0 and for channel 1, you can have some 5000, channel 2 you can have 6000.

As I have told each channel is capable of transferring maximum of 64 kilobytes of the data. Channel 3 will be having some other that is say 7000 H. This will be decide by the current address register which is present inside the 8237 and this can be either auto increment or auto decrement that will be decide by the control word registers that we are going to discuss in the coming slides.

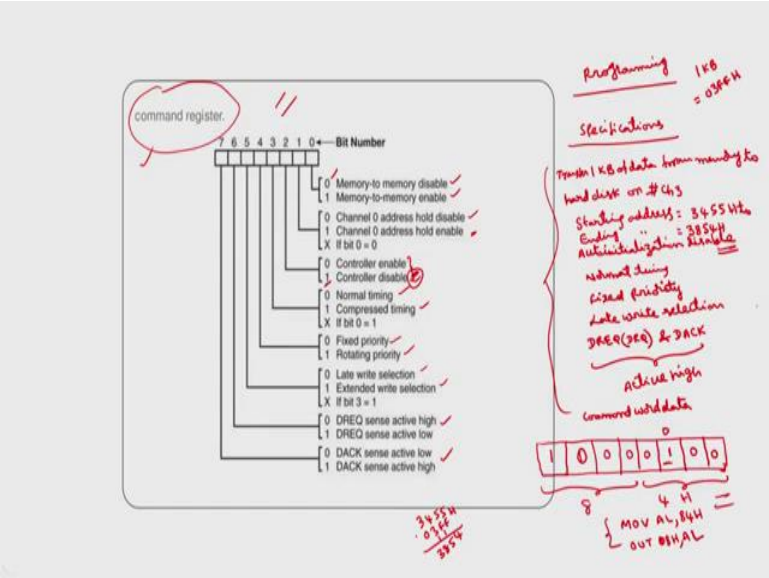
(Refer Slide Time: 55:03)



Then we have count registers as I have told so, if you know the starting address of the memory. Then the next question arises is how many number of bytes you want to transfer that will be stored in this count register. Just now I have told that each channel will be having two registers; one is called memory address register which stores the starting address of the memory and the current word count register which stores the number of bytes to be transferred.

Then we have command register so, which will decide the operation of DMA controller. So, whether this is from peripheral to memory or memory to peripheral or even memory to memory operation also possible. So, I will see the format of this command word register in the next slide.

(Refer Slide Time: 55:57)



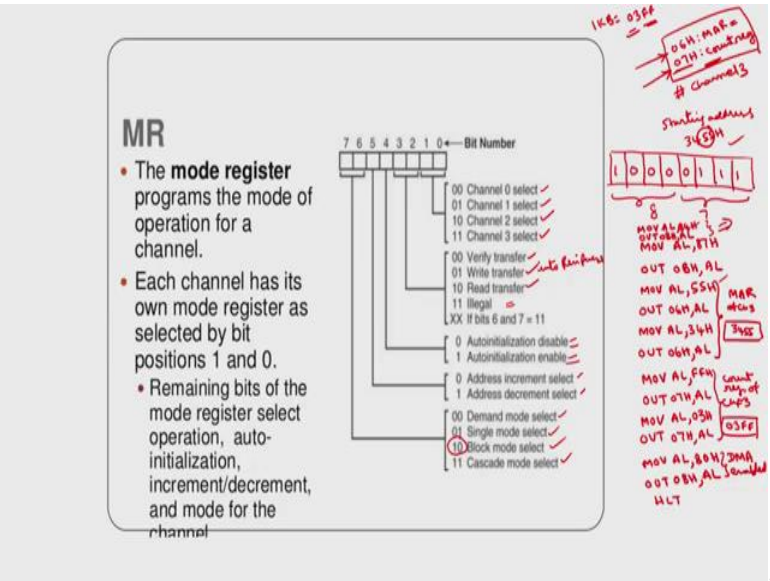
So, this is the format of command word register. This we have to load into 8237 before you are going to transfer the data this is self-explanatory. So, the last bit represents, if this is 0 then memory to memory transfer will be disabled, if this is 1 memory to memory transfer will be enabled this is self-explanatory. Similarly, second bit; second bit can be 0 or 1 depends upon whether channel 0 address has to be hold or we have to disable. If this is 0 channel 0 address will be hold if this is 1 channel 0 address will be enabled. The third bit will be DMA controller; whether you want to enable or disable the DMA controller.

Now, the question is when you want to disable the DMA controller. So, depends upon the initialization whether it is auto initialization or we are going to I mean initialized by the programmer. So, if the programmer want to initialize that time we have to disable this controller. This will be clear if I give an example. So, 0 means controller will be enabled 1 means controller will be disabled. And the timing also you want the I mean normal timing or compressed timing.

So, we can compress the time also by scaling factor, but normally we will use normal timing. So, normal timing 0 for compressed timing 1. As I have just now told if this can be having fixed priority or rotating priority. So, this will be decided by this bit and the next bit will be, next bit decides whether late write selection or extended write selection. So, if this is 0, late write selection otherwise extended write selection.

So, this depends upon the speeds of the devices, if it is late you have to select the late write it this depends upon the specifications of the peripheral that we are going to connect. Then we have D request and D acknowledge can be either active high or active low this is up to you. So, we can program the status or the level of DMA request and DMA acknowledge signals. This is the format of command register.

(Refer Slide Time: 58:31)



So, I will write the program using this command register after some time, then the mode register. There are two important registers; one is command register, another is mode register. So, mode register as the name implies it will select the mode of operation. So, this will decide the channel to be selected. So, if this is channel 0, 0 0, channel 1, 0 1, channel 2, 1 0, channel 3, 1 1.

So, we can have the three operations in addition to read write we can verify the data also. Suppose the some data is present in the memory, I want to verify whether the same data is present or not that operation also possible through DMA controller. So, for that we will use this as 0 0 for 0 1 write operation, 1 0 read operation whereas, 1 1 is illegal and the next bit is auto initialization enable or disable.

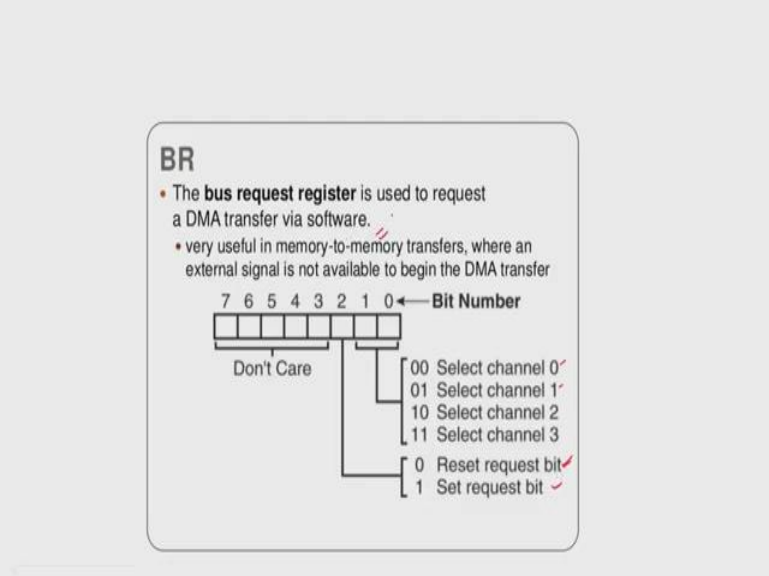
As I have told in case of auto initialization enable we have to enable the DMA controller in the previous mode. If auto initialization is disabled then I will disable the DMA bit in the previous this 1 here this control will be enable or disable this depends upon whether we are going to provide auto initialization or the main initialization provided by the

programmer. And the memory address; we are going to specify the starting address of the memory. So, whether the data need to be transferred in the ascending order or descending order, whether the memory address has to be incremented or decremented that will be decided by this bit. 0 means; address increment, 1 means; address decrement.

Then we have different types of the modes. So, demand mode select and demand we can transfer, we can transfer only single byte and we can transfer a block of bytes and this is also possible to cascade 8237, similar to 8259 we can also cascade different 8237s to handle different peripheral devices.

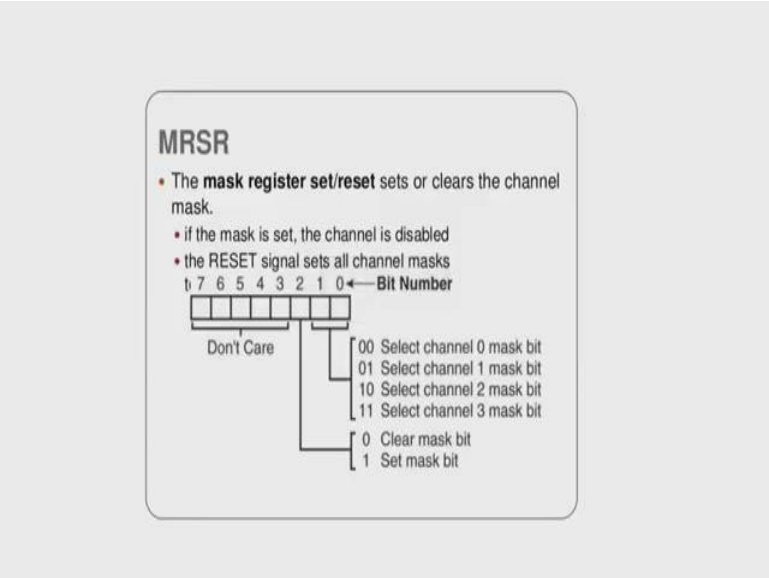
So, these two are the important registers which we have to use for programming. As I have told the next step is programming so, in that I have to use this mode register and command register right. And there are some other registers which will be normally we will not use, I will just we will just have a quick click on the quick look.

(Refer Slide Time: 60:59)



So, this is called the bus request register. So, this signals are I mean obvious from this slide. So, these are the channel selections and this bit will be 1 if it sets requests 0 reset the request bit and the remaining all are don't cares. This will be useful only in memory to memory transfer. But memory to memory transfer is not that important because memory is the speed of the memory is almost means it is compatible with the speed of the microprocessor where the DMA controller is important this is in case of peripheral to memory because peripheral is fast device.

(Refer Slide Time: 61:46)

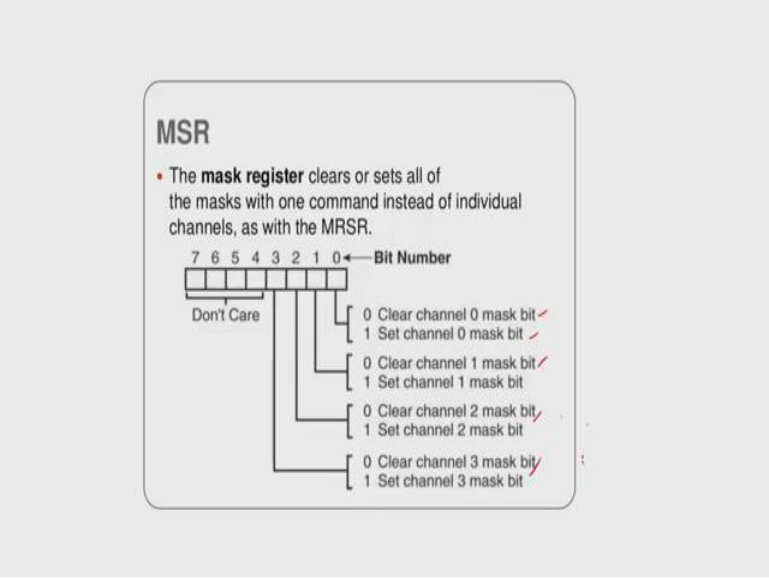


.

And we have mask register set reset these are all I have I have been discussed in the 16 registers the tables that I have given. So, these are the formats. So, these are all self explanatory.

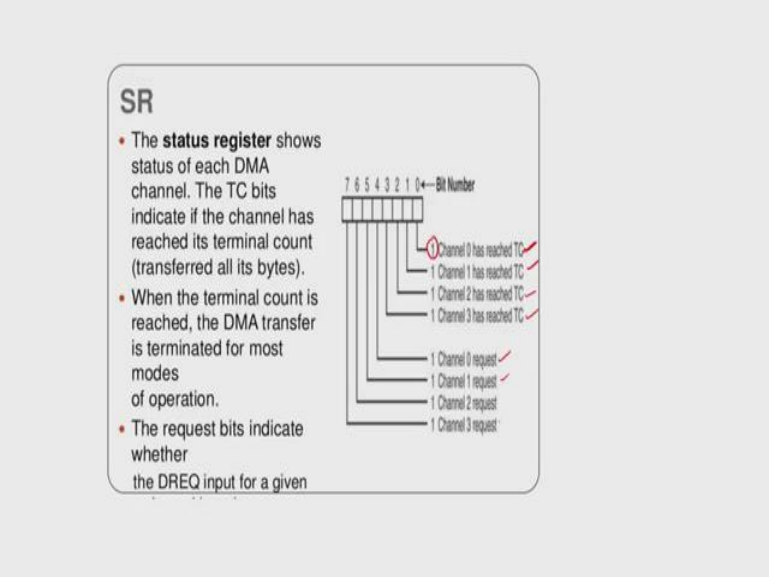
.

(Refer Slide Time: 62:01)



This is mask register so, we have 0 for clearing channel 0 1 for enabling or setting the channel 0 similarly for channel 1, channel 2, channel 3 and the remaining all are don't cares.

(Refer Slide Time: 62:15)



This is the status register. So, after loading this starting address of the memory and then the number of byte to be transferred ok so, after each byte transfer the count will be decremented by 1. So, whether this count becomes 0 or not that status you can verify through this status register. So, if channel 0 reached terminal count TC stands for Terminal Count, then this bit will be 1, otherwise 0 means; the I mean the data transfer is in progress. So, this is 1 implies that the terminal count has been reached. So, the DMA controller can relinquish the control of address bus, data bus and control bus the main control of the three buses to the microprocessor.

Similarly, this is for channel 1 terminal count, channel 2 terminal count, channel 3 terminal count. So, which channel has been requested? So, we can request the peripheral can be connected to channel 0, channel 1, channel 2 or channel 3. So, if you want to request through channel 0, this bit you have to make as 1, channel 1 this bit we have to make 1. So, this is also possible that at a particular time more than 1 channel becomes 1.

So, in that case you have to assign some priority, according to the priority you have to decide to whom this I mean request has to be granted. So, these are the different registers. Now coming for the programming; I will discuss so, on these registers itself the

command registers ok. Suppose these are the specifications of the program that I want to write.

So, I want to transfer 1 kilobyte of the data transfer 1 kilobyte of the data from memory to hard disk on channel 3 and the starting address of the memory should be say some 3455 H and I want to use auto normalization auto initialization disable. And I want to use normal timing, fixed priority, late write selection and I want to use both D request.

So, in fact, this is DRQ or DREQ, in some books it is written DRQ and in some books it is written DREQ DRQ both are same only and D acknowledge I want to use this as both are active high signals. So, these are the main specifications for that you have to write the program. So, the first step is we have to find out what is the command word register. So, what will be command word?

So, what will be these bits. So, what about the last bit it is not a memory to memory transfer. So, I have to disable the memory to memory transfer. So, this will be 0 and then coming for the next bit, this is channel 0 address hold disable because we are not using the channel 0 so, I want to disable this then because I want to disable auto initialization. So, this bit also will be controller we have to disable.

So, this will be 1 because I want to disable the controller, then only I can initialize the process otherwise if I enable this controller it will do the automatic initialization of the process. Then I am using fixed priority so, this will be 0 then late write this will be 0 I want to make both high D request high means; this will be 0, D acknowledge is high means; this will be 1. So, this is corresponding to 8, one more bit is this is last one is 0 then last, but one is 0 controller is disabled it is 1.

So, normal timing is 0, fixed priority is 0, late write is 0, then we have one more bit here this D request is this is 0 for active high and this is 1 for active high D acknowledge. So, what will be this one 8 4 H. So, what is the address of this command register if I use this circuitry which I have used in the earlier slides? So, for this connections, I will assume these connections. For these connections, what is the address of command register?

So, in this where is the command register, this is the command register the corresponding address is 0 8 H. So, what are the instructions? So, the instructions are MOV AL I am discussing the programming here itself this is what is called programming. MOV AL

comma 8 4 H I have to place this into the command word register whose address is 80H, if I assume the interfacing connections that I have discussed in the earlier slide.

Then we have to write OUT 0 8 H comma AL these 2 instructions are used for setting this command word register then, coming for the second register which is the mode register. So, what will be this data of mode register? So, the last 2 bits because I am using channel 3, these two are 1 1 and I want to use write operation write into peripheral because I want to transfer from memory to peripheral so, this is 0 1.

Then, I want to disable auto initialization 0 and I want to increment the address. So, this is another specification you can write in the previous specifications increment the address. So, 1 byte of the memory starting from this address to so, what is the ending address if I want to increment this? 1 kilobyte is equivalent to 0 3 F F H in hexadecimal.

So, ending address will be now 3455 H plus 03FF H. So, this is F plus 5 will be F plus 1 is 1 0 1 4 and this is 1 5 this is 8 3 8. So, I want to transfer the data from this is the starting address and ending address will be 3854. So, that the memory has to be increment after each transfer.

So, this will be now auto increment means 0, then I want to transfer a block of data so, 1 0 because I want to transfer 1 kilobyte. So, this will be block of data. So, this is 1 0. So, the corresponding this one is 8 and this one is 7, 87 H we have to load onto the mode register whose address is the address of the mode register is here this one 0B H this is mode register.

So, MOV AL comma 87 H, OUT 0BH is the address of mode register comma AL, this is for the initialization. Then I have to load this count what is the count that I want to load 1 kilobyte using channel 3. So, what is the address of channel 3 which holds the memory starting address? Again coming for the table so, I want to store the starting address in channel 3, this is channel 3 memory address this I have to load with 0 3 F F H.

And this is count register is 1 k sorry, this is 0 3 F F H because count and the starting address is we have to load here and what is the starting address we have taken is 3 4 5 5 H, we have to load here 3 4 5 5 H. So, the address of channel 3 MAR will be 0 6 H and count register will be 0 7 H. So, you have to use these addresses 0 6 and 0 7.

So, 0 6 H is memory address register and 0 7 H is count register for channel 3, these are the addresses of channel 3. So, first you send the lower order address onto this memory address register this is 16 bit register. So, the address is 3455 H the starting address is 3455H.

So, the previous two instructions I have written here which are MOV AL comma 84 0 8 MOV AL comma 84 I will write here for the completeness of this program MOV AL comma 84 H 08 OUT 08 H comma AL then MOV AL comma, what is lower order byte of the starting address? 3455. So, 3455 H this is lower order byte.

So, you take that lower byte on to AL OUT. So, what is the address of this MAR? It is 06 H OUT 06 H comma AL. This is possible actually to transfer the entire 16 bit also on to 16 bit port here I am assuming only 8 bit addresses. Then MOV you take the higher order which is 34 H onto same memory address register whose address is 0 6 H comma AL.

These four instructions will load memory address register of channel 3 with 3455H then you have to load the number of byte to be transferred ok. So, the number of bytes is 1 kilobyte and what is the address of that count register is 07 H again you take MOV AL comma for 03FF, 03FF is 1 kilobyte is 03FF.

The lower order address will be F F higher order will be 0 3 you take the lower order FF H OUT the address of the count register is 0 7 for channel 3 so, 07 H comma AL MOV AL comma 03 H OUT onto 07. So, this will load the count register of channel 3 with 1 K 03FF. So, I have given all the specifications I know the starting address I know the number of bytes to be transferred then you have to enable the DMA.

So, here in the previous this command word here we have disabled the DMA. So, by disabling the DMA only you got this this one. So, here we have disabled the DMA by putting 1 on the corresponding pin. Now I have to enable for the data transfer. So, what will be this if I enable this? This will become 80 H so, MOV AL comma 80 H OUT 08 H comma AL. This 08 H is the address of command register.

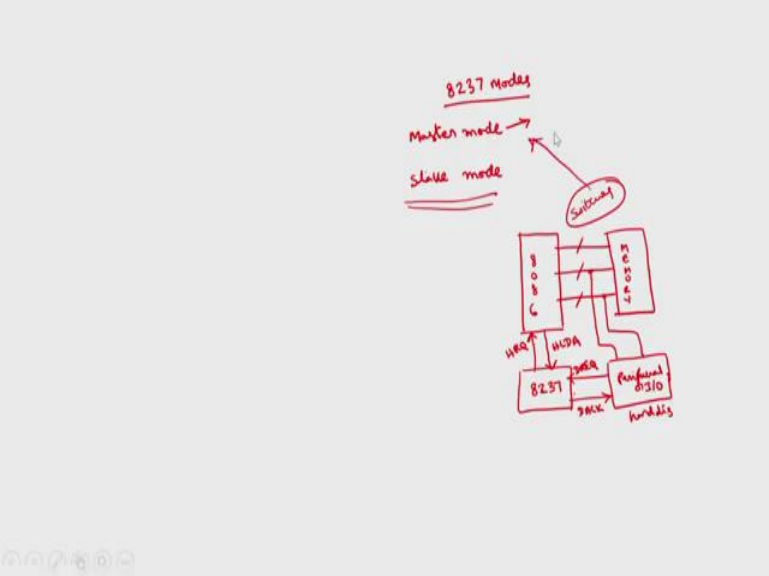
So, I have now enabled the DMA, DMA will be enabled. Then the 1 kilo bytes will be transferred after that the terminal count will be raised it will stop the process, this is about the programming of 8237. So, the important task here is; first you have to decide

command word register using this format. So, you have to select these are the specifications.

So, it depends upon the specifications, you have to find out what is the corresponding data to be loaded into command word register then, based on these specifications you have to decide what data has to be loaded into mode register. And so, you have to first of all interface the 8237 to microprocessor to generate the address of mode register as well as command register, then you take the corresponding data and you load onto the corresponding addresses.

Then the next step is you have to load the starting address of the memory in the selected channel and then the count onto the channel, then you enable the DMA, so this is how we can program the DMA now corresponding to these two modes; slave and master mode.

(Refer Slide Time: 79:49)

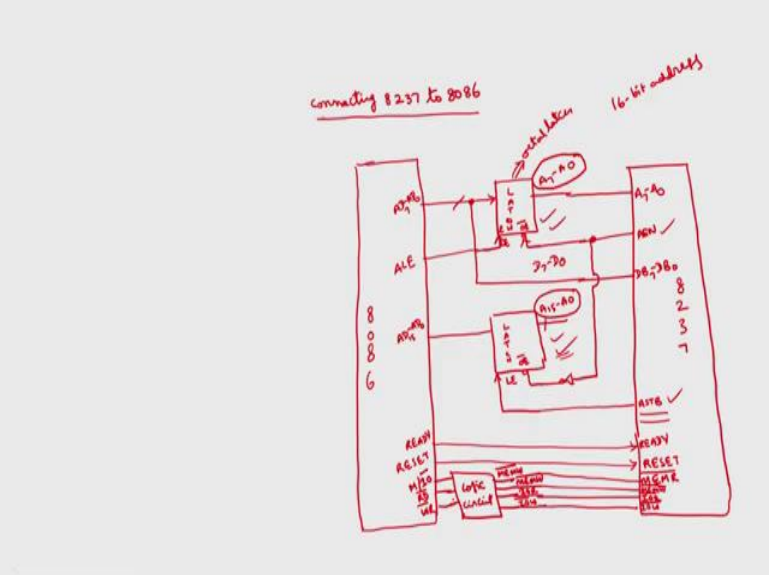


We can operate this in master mode as well as slave mode 8237 modes. In master mode, I have discussed these 7 or 8 steps in the earlier lecture. So, the step by step procedure that follows to operate this 8237 in master mode. In slave mode simply so, this DMA controller will be acts as a slave mode. So, in that case simply this 8086 will be connected to memory address bus data bus control bus.

Now this is DMA controller 8237 and this will be DMA hold request hold acknowledge and here we will be having I/O peripheral or I/O which has hard disk this will be connected to the address and control bus of this one and this will be DMA request this will be DMA acknowledge. This is DMA request, DMA acknowledge and this is hold request and this is hold acknowledge. This is the connection diagram in case of slave mode, in case of master mode we will be having switches here switches in case of master mode ok.

These are about the different modes of 8237. Now coming for the final the connections between the microprocessor and 8237.

(Refer Slide Time: 82:02)



The other side I am not showing because there peripheral will be connected this is 8237 so I am calling as 8237 only, but this is this can be 8237 A also ok. This will be having only 8 bit address bus, these are two latches and this is your 8086. So, we know that 8086 having multiplexed address data buses. So, I am considering only the 16 address lines, this is the latch this will be having two signals; one is called latch enable signal and output enable signal this will be available as active low signal and this is latch enable active high signal this is LE latch enable.

This if I consider this AD7 to AD0, the lower order 8 bits of the address this is multiplexed bus this is octal latch, this will have 8 bit flip flop inside this. This will be

connected to address latch enable address latch enable. So, the output will be here address bus A7 to A0 and this will be now connected as data bus D7 to D0.

This is how we can de multiplex the address and data buses. So, this has to be connected to the address bus of corresponding address bus of 8237, this is address bus this is DB7 to DB0 and this output enable bar this will be connected through address enable AEN of 8237. Now coming for the higher order 8 bits, there will be another latch. So, this also will be having latch enable.

Latch enable and output enable, output enable is normally active low signal. So, this will be address latch enable will be connected, but this will be connected through NOT gate if this is enable, this will be disabled, if this is enable, this will be disabled, because AEN signal is connected through NOT gate and this LE latch enable signal will be generated by now address strobe.

So, in case of lower order address I mean latch enable will be generated by the ALE signal whereas, in case of higher order address bus. So, this input will be now AD15 to AD0. Here this output will be A15 to A0. So, this and this together will acts as 16 bit address; because we have the memory address is 16 bit, the starting address of the memory MAR is 16 bit. So, you have to place 16 bit address on to this bus.

So, this is how we can use this address strobe signal and AEN signal to de multiplexer address and data buses ok. So, this address strobe is basically used for de multiplexing the higher order bus and AEN is of course, required for both the buses and the lower order bus will be de multiplexed by using AE a address latch enable. The remaining signals are directly connected ready to ready reset to reset, these are ready, this is reset, this is ready.

And similarly, we have memory read bar, memory write bar, I/O read bar, I/O write bar, memory read bar, memory write bar, I/O read bar, I/O write bar. But here we have only M by I/O bar read bar write bar. Using these 3 signals using some circuitry this we have discussed in the earlier classes some logic circuit we can generate four signals ; memory read bar, memory write bar, I/O read bar, I/O write bar.

So, these are correspondingly connected to the corresponding signals. Memory write bar, I/O read bar, I/O write bar, these are about the connections between 8237 to 8086. So, these are all the complete details of the DMA controller 8237.

Thank you.