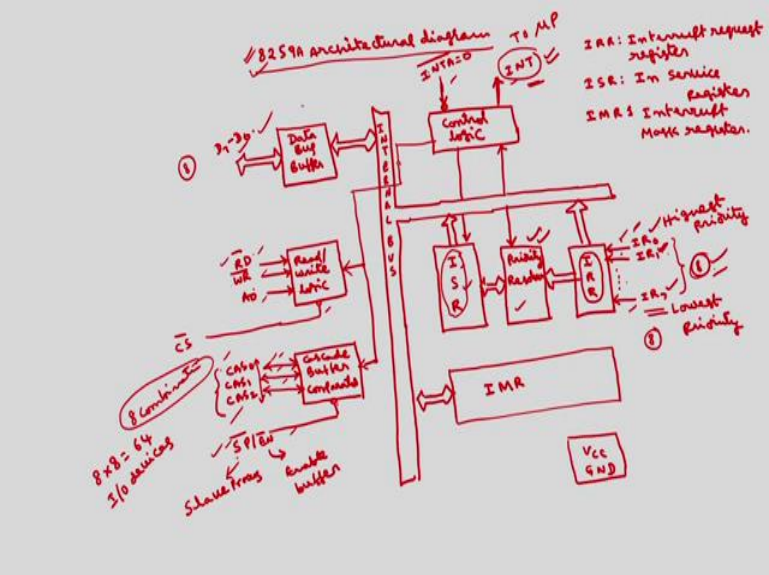
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

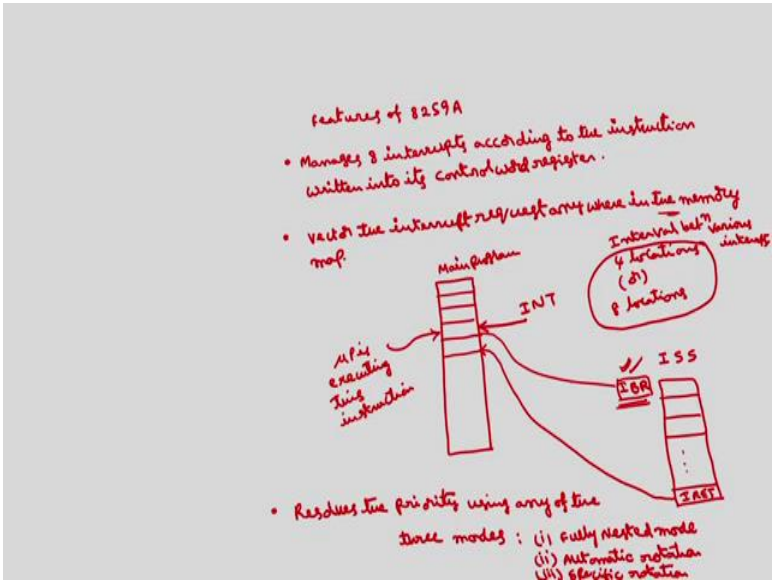
Lecture – 28
Architecture of 8259

(Refer Slide Time: 00:37)



So, ok, in the last class we are discussing about the programmable interrupt controller 8259. In fact, this is 8259 A, 8259 is not accessible to the 8086. 8259 A is the latest version of 8259. So, this 8259 A is compatible with 8086. So, this is the I mean the block diagram of 8259 A. So, if I take this salient features of this 8259, features of 8259 A.

(Refer Slide Time: 01:05)



So, let us begin how this salient features, first this manages the 8 interrupts according to the program that is written in the control word registers of this 8259 A, 8 interrupts. According to the instructions written in it's control word registers. I will discuss what are the control word registers of 8259, there are 4 initialization registers, this is 1 feature which manages 8 interrupts as you have seen this in the diagram also.

So, here we can have 8 interrupts. So, the 8259 will manages this 8 interrupts. So, how it manages is the second point, is the second feature of this one is, this vector the interrupt request anywhere in the memory map. So, the meaning of this one is so, correspond to each interrupt we know that the interrupt process is; we have discussed this interrupt process in earlier classes. So, whenever the microprocessor is executing some program, this is the main program.

Suppose, the microprocessor is executing this instruction. So, at that time if some interrupt occurs. So, what happens is the microprocessor will complete the execution of the current instruction, then it saves the contents of the instruction pointer onto the stack and then it will jumps to a location called interrupts branch address. Interrupt branch address, which is the starting address of interrupt service sub routine.

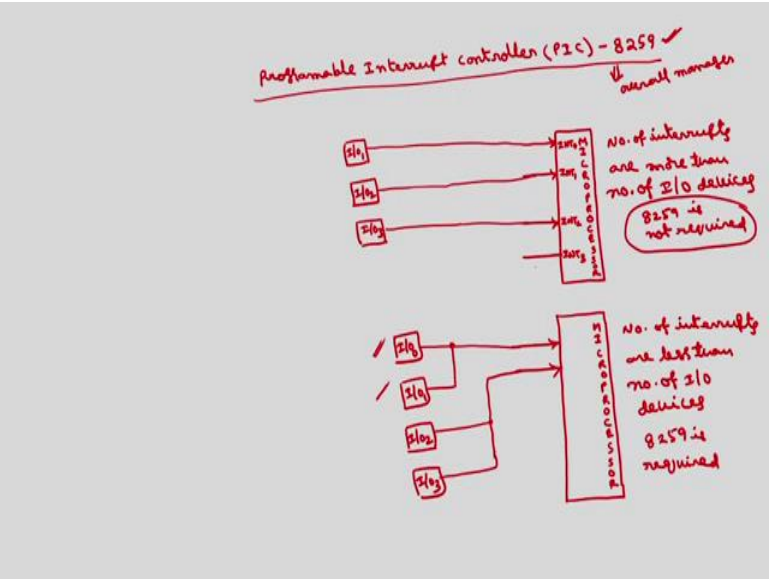
And the last instruction IRETURN it will pop the contents of instruction pointer and it will come back to the instruction which is next to the instruction during the interrupt. So, this interrupt branch address is the starting address of the interrupt service sub routine,

this is called interrupt service sub routine. So, this address can be anywhere between the entire memory of the 8086, that is the meaning of this. This vector the interrupt request anywhere in the memory map.

So, this address can be any address that can be loaded it into the control word registers so, during the programming. So, we will discuss how this interrupt branch address will be loaded into the control word registers while discussing about the programming of 8259 A. This is the second feature, but this address can be loaded anywhere in the total memory map, but the address of the consecutive interrupts they will be separated by either 4 locations or 8 locations.

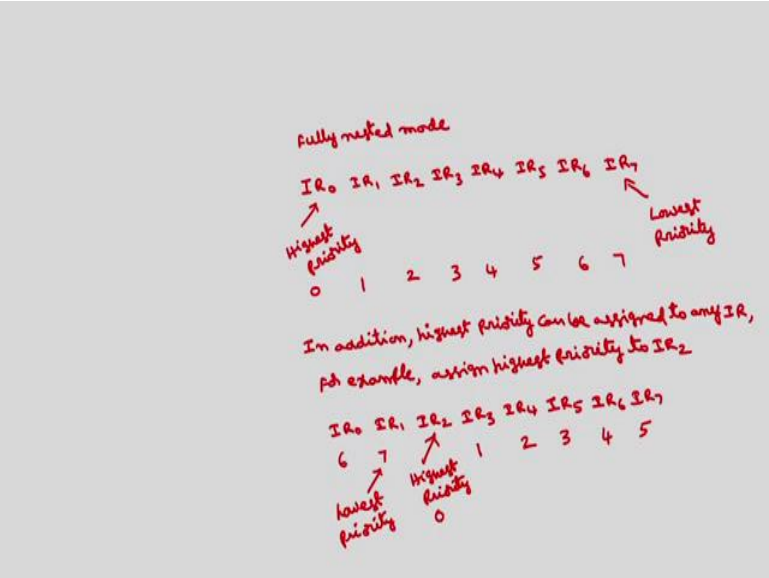
The interval between various interrupts. This is also can be programmed, this can be either 4 locations or 8 locations. So, you see this next feature. Third feature is it resolve the priority using any of the three modes. As you can see in a 8259 A architecture, there is one priority resolver is there. So, this is going to be resolve the priority of the interrupts interrupt requests. How this priority resolver will resolve the priority, that some three modes of this resolving the priority. So, one is called fully nested mode.

(Refer Slide Time: 07:21)



So, this modes are one is fully nested mode and the second one is automatic rotation mode and third one is specific rotation mode. So, what is this fully nested mode, automatic rotation and specific rotation? We will discuss now.

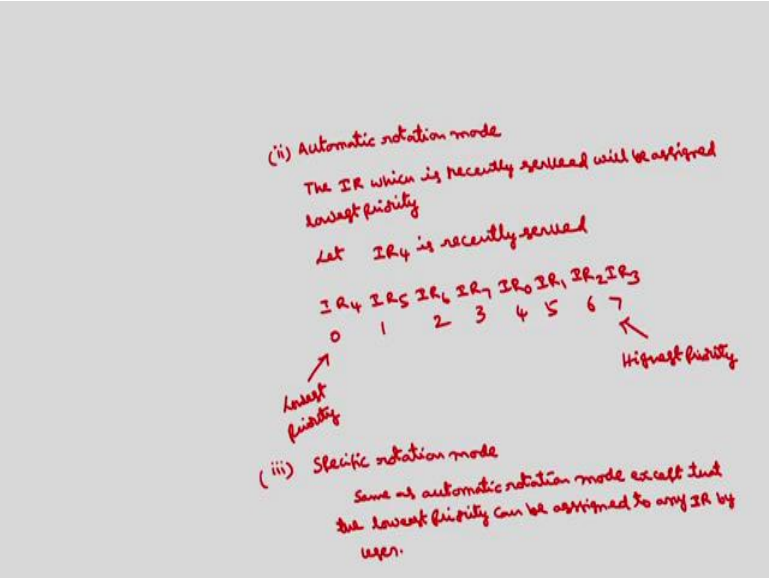
(Refer Slide Time: 08:18)



In full nested mode; by default IR 0 will be having highest priority IR 0 to IR 7. IR 1, IR 2, IR 3, IR 4, IR 5, IR 6, IR 7. So this IR 0 will be having highest priority. So, I will retain the highest priority with number 0, then the next one will be having next highest priority is 1 is IR 1, 2, 3, 4, 5, 6 and 7, IR 7 is having lowest priority, in addition, we can also set highest priority to any of these IRs. In addition highest priority can be assigned to any IR. For example, if I want to assign the highest priority to IR 2. Then what happens is? The sequence, this IR 0, IR 1, IR 2, IR 3, IR 4, IR 5, IR 6 and IR 7.

Now, I am assigning the highest priority to IR 2 means 0. Automatically, the next IR will becomes priority 1, priority 2 is IR 4, priority 3 is IR 5, IR 6 is priority 4, IR 7 is priority 5 then IR 0 is priority 6, IR 1 is priority 7, which is the lowest priority, this is called nested mode. So, in the second mode which is called as automatic rotation mode.

(Refer Slide Time: 11:22)

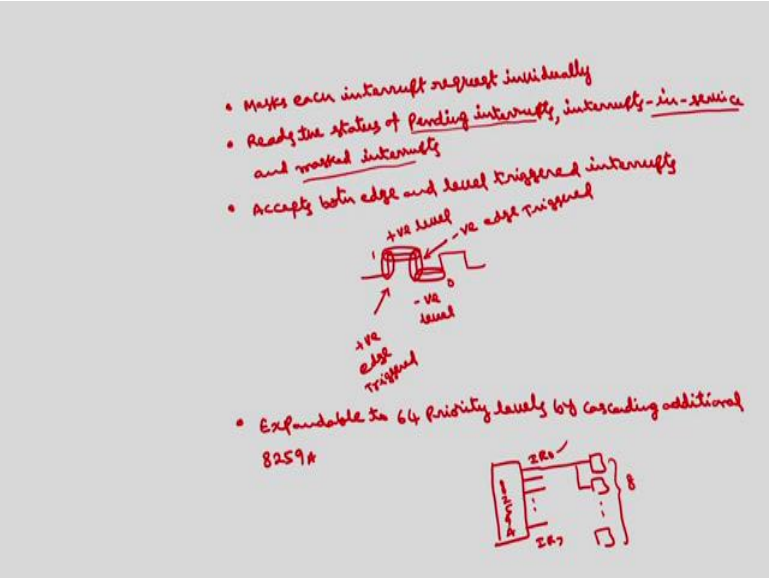


In automatic rotation mode, so the interrupt which is being serviced, recently serviced that will be assigned the lowest priority. The interrupt the IR which is recently served will be assigned lowest priority. For example, let IR 4 is recently served. The order of priority will be this is IR 4 then IR 5 IR 6, IR 7 then, IR 0, IR 1, IR 2 IR 3. So, this will be having lowest priority 0, then this will be 1, 2, 3, 4, 5, 6 and 7, this is the highest priority. Then the third mode this is second mode which is automatic rotation mode, third mode is specific rotation mode.

So, this is same as automatic rotation mode except for that this IR lowest priority can be assigned by the user. This is same as automatic rotation mode except that the lowest priority can be assigned to any IR by user. So, in the previous mode, automatic rotation mode, the lowest priority will be assigned to the IR which is recently served whereas, here the lowest priority can be assigned by the user. What is the difference between the automatic rotation mode and specific rotation mode?

So, you have a 3 modes using which the 8259 A resolves the priority of the all interrupt requests. Then there the other features of this 8259 A is mask the each interrupt request individually.

(Refer Slide Time: 15:10)



If your particular interrupt is serviced, then the remaining interrupts can be masked. It can mask each interrupt request individually, then read the status of the pending interrupts. It can read the status of pending interrupts means; at any time if more than 1 interrupt occurs the interrupt with highest priority will be served, the remaining will be kept pending.

So, it can read the status of all the pending interrupts as well as interrupts in service and maskable interrupts, masked interrupts. There are 3 types of the interrupts you can see here; pending interrupts, in service interrupts and then masked interrupts. So, the in service interrupt is obvious from the definition itself, the interrupt which is being currently serviced.

And there are some pending interrupt, there are some masked interrupts. Pending interrupts will be in the queue. So, after the in service interrupt process is over then it will take the pending interrupts maskable interrupts means; they have masked, they cannot I mean interrupt the 8086 so, while this interrupt are in service. So, after the servicing of the particular interrupt, so, the maskable interrupt can be unmask. So, that they can request 8259 for the interrupt service. So there are 3 types of interrupts. So, this 8259 can read the status of all the 3 types of the interrupts.

So, next feature of this 8259 A is it accept both edge as well as level triggered requests. There is a bit in control word register to select whether edge triggered or level triggered.

So I have already discussed about what is edge triggered, what is level triggered. If you take a ideal clock, this is called positive level, this is negative level say if interrupt is level triggered interrupt it can be either positive level triggered or negative level triggered means in order to I mean interrupt a processer if it is a positive level triggered that interrupt you have to make as logic 1.

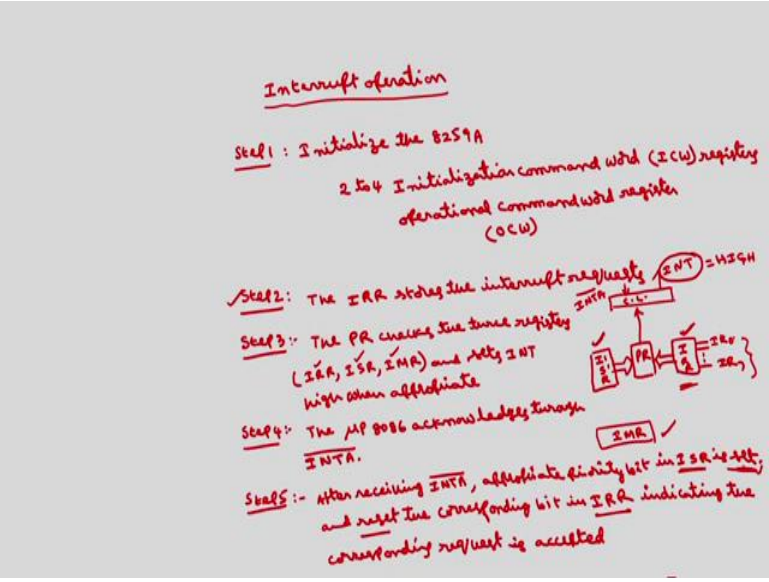
If it is a negative level triggered you have to make that interrupt as logic 0 then the microprocessor will be interrupted. So, if it is edge triggered, this is called positive edge triggered, a transition from logic 0 to logic 1. So, in order to I mean in interrupt the processor, this interrupt level has to change from logic 0 to logic 1. Similarly, we can have negative edge triggered. So, interrupt will occur whenever there is a change on this line from logic 1 to logic 0 this is called negative edge triggered.

So, 8259A accept both type of triggered interrupts. the next feature is it can be expandable to, 8259 enables to expand to 64 priority levels by cascading additional 8259. When cascade to 8259 using single 8259, we can have 8 interrupt requests if I cascade 2 8259 A is. So, we can have 64 priority levels ok.

Because we have this one 8259 is having 8 IRs, IR 0 to IR 7. So, by cascading these IR again we can have 8 different we can have 8 different here devices. So each line is IR 0 can connect to 8 devices, IR 1 can connect to 8 devices. So, total will be 8 into 8, 64 devices can be connected to this microprocessor using cascading of 8259 s.

So, there are some important features of 8259. Now coming for the interrupt process. What is the interrupt operation? How these interrupt operation will be takes place.

(Refer Slide Time: 21:36)



What are the steps in this interrupt process? So what is the step by step procedure for interrupt operation. So, first you have to initialize the 8259. So, step 1 is initialize the 8259 A for initialization we can have 2 to 4 initialization command word registers. There are 2 to 4 initialization command word registers, which is called ICW registers. 2 registers are must and 3 and 4 depends upon the application we can choose 3 and ICW 4.

So, I will discuss this format after discussing about this interrupt operation procedure. So, the first step you have to initialize using this interrupt command word registers. In addition we can have one operational command word register also. The other type of register control word register which is operational command word register called OCW.

So, after the initialization; the next step is the interrupt request register stores the interrupt requests. We can see in the architecture. So, here this is the interrupt request register for which we are going to apply IRRs. As the name implies the interrupt request register IRR. So, this stores the interrupt requests the IRR stores the interrupt requests and the next step is the priority resolver.

So, we have 3 registers in order, this is IRR, this is priority resolver, priority resolver PR is priority resolver, then we have ISR, In Service Register. And then here we have control logic bit which gives the INTR, INTA bar. This is a part of that 8259 architecture that we have discussed in the earlier classes. So, these are these IRR, this is priority resolver this is ISR which is In Service Register and then we have IMR which is

Interrupt Mask Register then we have control logic for which the output INT and input is INTA bar.

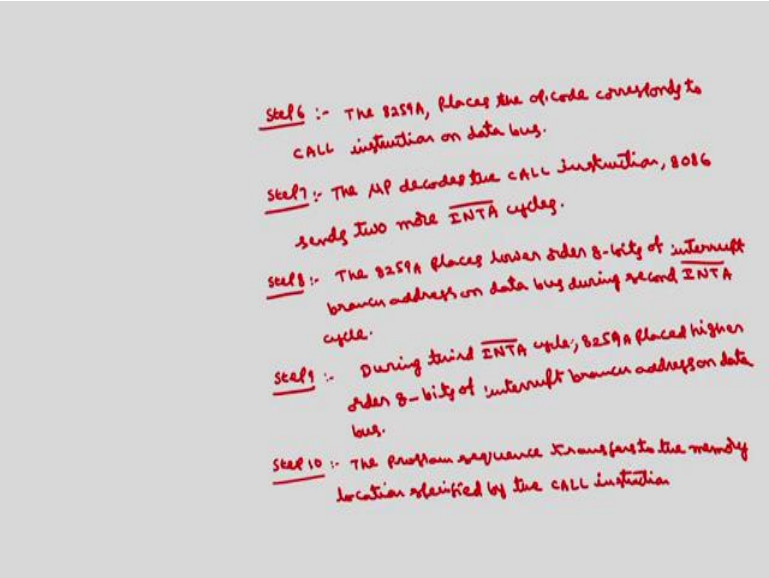
This is control logic and here you have interrupt mask register IMR. So, this will take IRS, 8 IRS. So, in this 2 nd step this IR, IRR stores the requests of all the interrupts, then the priority resolvers. The next step is the priority resolver PR stands or priority resolver. Checks the 3 registers, which 3 registers? We have this IRR, ISR, IMR, 3 registers; IRR, ISR, IMR these 3 states are important to assign the priority.

One is which interrupt has been requested and which is in service and which is masked. Based on this the priority resolver will assign the priority to the interrupts and sets INTR INT high when appropriate after checking these 3 registers, the priority resolver will request this control logic and it will makes INT is equal to high. This INT will be connected to microprocessor. This INT, INTA bar will be connected to the microprocessor. Means; this 8259 A is now has requested the microprocessor for interrupt.

So, then the next step is step 4; the microprocessor acknowledges through INTA bar the microprocessor 8086 is acknowledges is through INTA bar. So, whenever 8259 receives INTA bar, the first cycle of INTA bar, the next step is. So, in this IRR, the corresponding bit of that interrupt request will be make 0. So, indicating that particular request has been accepted after receiving INTA bar the appropriate priority bit in ISR is set.

So, in ISR if this is IR 0, then IR 0 bit will be set if it is IR 1, IR 1 bit will be set like that and at the same time it will reset the corresponding bit of IRR indicating that the request has been accepted. Set and resets the corresponding bit in IRR. So here this will be set corresponding bit will be set in ISR indicating that particular interrupt is in service and the corresponding bit in IRR will be reset indicating that the particular interrupt is accepted. Indicating that the corresponding request is accepted.

(Refer Slide Time: 30:21)



Then, after that 8259 places the opcode correspond to call instruction, because in a step in step 6 as I have told this interrupt operation is exactly similar to the call instruction. So, the 8259 A places op code correspond to operation code corresponding to each instruction we have some opcode corresponds to call instruction on the data bus. Then the data bus is connected to the microprocessor so the microprocessor decodes the call instruction.

Once it is decoded it knows that it is a call instruction it has to jump to some other address. So, it will send 2 more interrupt acknowledge cycles. So, after decoding 8086 sends two more INTA bar cycles. So, that it is requesting for the instruction the interrupt branch address where the interrupt has to be jumped. So, through this 2 more INTA bar cycles the microprocessor request the 8259 A to send the call address.

So, that address is nothing, but the interrupt branch address. So, in the next step, the 8259 A places lower order 8 bits of the call address or you can call as interrupt branch address on data bus then it places higher order 8 bits during the third cycle. This is during the second cycle during second INTA bar cycle. So this total 16 bit address it will send in 2 INTA bar cycles.

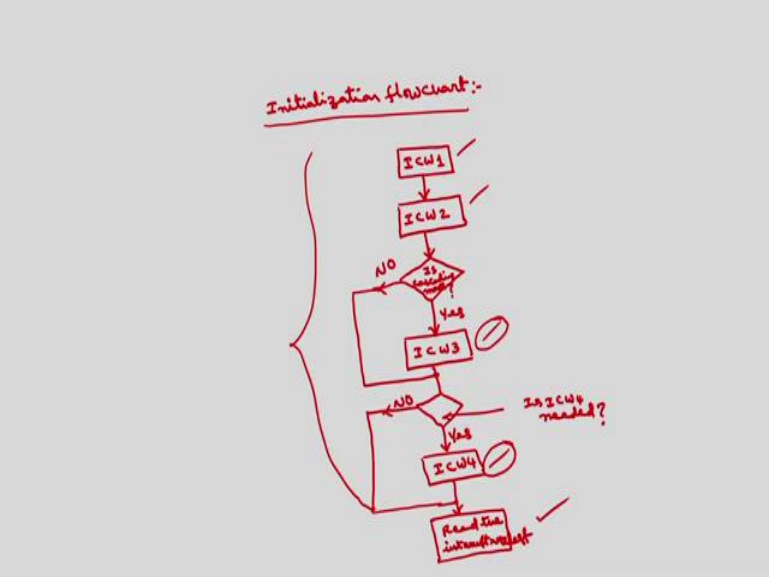
So, in the first INTA bar cycle is a acknowledgement to that particular interrupt request in response to second INTA bar 8259 A places lower order 8 bits of the interrupt branch

address. So, next one is during the third INTA bar cycle. The 8259 A places higher order 8 bits of the call address higher order 8 bits of interrupt branch address on data bus.

So, once the address is known then the microprocessor control will transfer to that particular address interrupt branch address mean interrupt service subroutine after completing the execution of the current instruction. Then the last step is the program sequence transfers to the memory location specified by the call instruction. So, this is the step by step procedure to I mean recognize the interrupt and to service the particular interrupt.

So, coming from this initialization of 8259. So, the first step is initialization of 8259. So, how to initialize this 8259? As I have told there are 4 initialization control word registers in 8259 A. So, what is the format of this initialization control word registers we will discuss now. So, as I told depends upon application it will choose 2 or 4 initialization control word registers.

(Refer Slide Time: 36:33)



The flow chart or this initialization will be, but first IC Initialization Control word IC 1, ICW 1 from that it will go to the ICW 2, these 2 are mandatory. So, I will draw the formats of this ICW 1 and ICW 2. Then the third one ICW 3 will be initialized in only the cascading mode. So, it will check now whether this is the cascading mode or not. Is it cascading mode? If yes; it will go to the initialization control word 3, ICW 3.

If no it will skip ICW 3, ICW 3 is not required then it will check for the next condition which is ICW 4. I am writing this here is ICW 4 needed. If yes, it will initialize the control word 4, if no it will move here and here read the interrupt request as I told the second step is reading the interrupt request. So the first step is this initialization. After initializing this ICW 1, ICW 2, ICW 3, ICW 4 depends upon the application. It can be 2 to 4 as I have told in the earlier procedure. 2 to 4 this depends upon the application.

So, after that the next step is reading the interrupt request and after that the step by step procedure here I have already explained. So, this ICW 3 is useful only for cascading mode and the ICW 4 depends upon the application. So, this may or may not be required. Whereas, ICW 1 and ICW 2 are mandatory. So, what are the control word formats of this ICW 1 and ICW 2? And, how to program 8259 A? So, we will discuss in the next class.

Thank you.