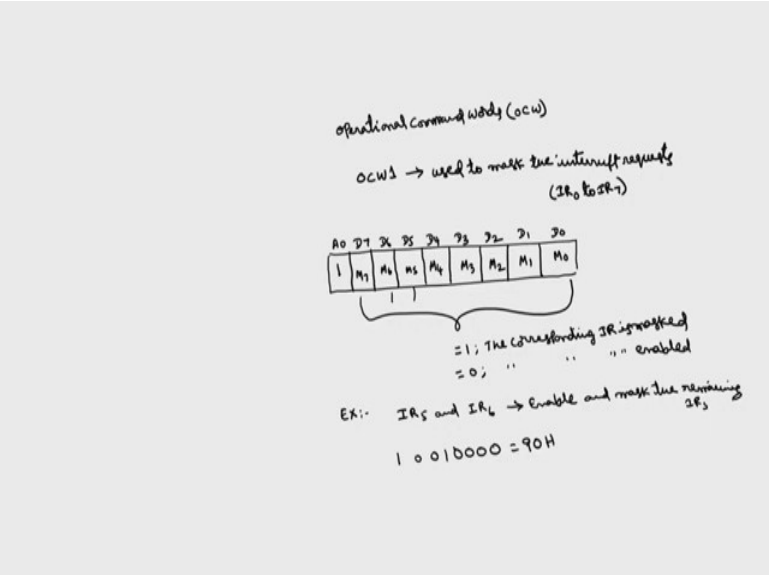
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture - 30
Operational command words of 8259

Okay, in the last class we have discussed about the initialation command word of 8259; the programmable interrupt controller. So, once this 8259 is initialised, then it is ready for processing the interrupts ok; now the way in which the 8259 is going to handle the interrupts.

(Refer Slide Time: 00:57)



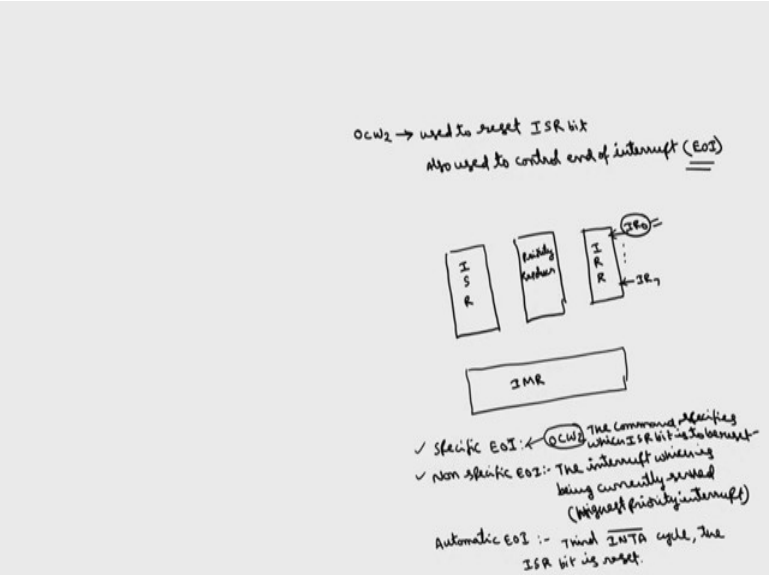
So, we have a 3 operational command words OCW. If you take OCW 1, the operational command word 1; this is used to mask the interrupt requests. We have 8 interrupt requests IR 0 to IR 7. We can mask any interrupt request or you can enable any interrupt request ok. Further this command word format will be used this is D 0 is directly for M 0.

M 1, M 2, M 3, M 4, M 5, M 6, M 7 and this is A 0; this will be always 1 for OCW 1. So, this basically this is D 0, D1, D 2 so on up to D 7; so, this M 0 is correspond to IR 0; M 1 is correspond to IR 1 and so on ok.

So, if we want to mask any particular bit here, if this bit is equal to 1; the corresponding IR is masked; corresponding IR is masked. If this is equal to 0, the corresponding IR will be enabled. If I take a example, I want to enable IR 5 and IR 6; and you mask the remaining IRs.

Then what is the OCW 1? So, IR 5 is corresponding to this IR 6 is corresponding to this. So, these two bit should be 0 remaining all should be 1s; 1 0 0 1 then all 0 s; this will be 90 H. If I load 90 H into operational command word register 1 then, IR 5 and IR 6 will be enabled whereas, the remaining all will be masked; this is about the first operational command word and the second operational command word OCW 2.

(Refer Slide Time: 04:32)



This is basically used to reset the ISR bit; also we can use this for control of end of interrupt; end of interrupt is EOI. We know that this 8259 is having 3 registers; this is IRR which is interrupt request register where these IRs will be applied IR 0 to IR 7; then we have priority resolver; then we have here in service register ISR; of course, we have here this interrupt mask register also.

So, once a particular interrupted say IR 0 is serviced ok; so how to represent the end of the interrupt. The start of the interrupt will be whenever this IR 0 requests this IRR. If this is having highest priority then priority resolver will send the corresponding bit of ISR to 1; then this will be requested here through INTR signal to the microprocessor;

then microprocessor will send the INTA bar signal interrupt acknowledge signal; then the corresponding call instruction will be executed ok.

Now here, how to represent the end of the interrupt? So, we have discussed about the start of the interrupt; how this interrupt service will be started; and how to represent the end of interrupt this is also equally important. So, in order to I mean represent this end of interrupt we will use OCW 2. So, there are 3 ways to I mean represented end of interrupt ok.

Before going for this a format of OCW 2 so, I will discuss the ways in which end of interrupt can be performed; one is called specific, another is nonspecific, another automatic. In case of specific EOI, we are going to specify the bit which is to be reset in the ISR as the name implies.

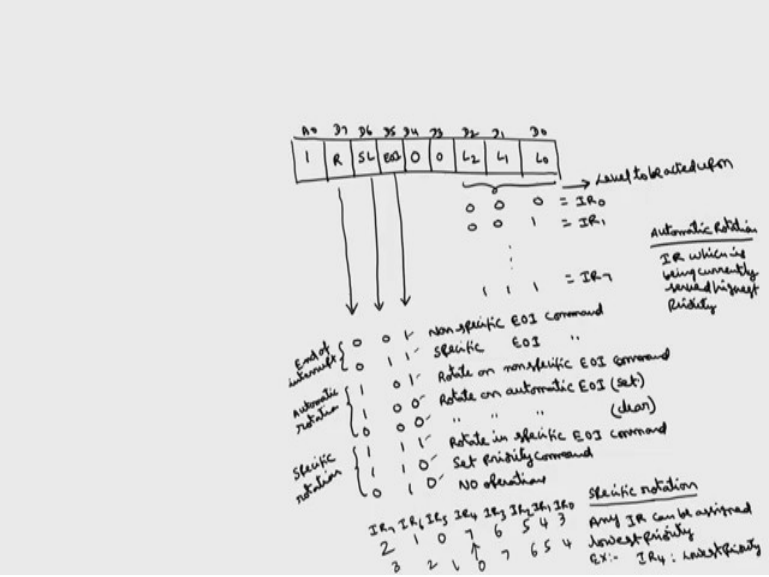
So, whenever this 8259 receives this command this specific EOI will be send specific EOI will programmed by the OCW. Whereas, in case of nonspecific EOI as the name implies by default the interrupt which is being currently served are highest priority interrupt which is being currently served; by default this is highest priority interrupt.

So, the corresponding bit of this ISR will be reset in case of nonspecific EOI. So, in case of a specific EOI, the command specifies here we have to use OCW. So, the command specifies as the name implies which ISR bit is to be reset.

So, I think now the difference between specific EOI and nonspecific EOI is clear. So in case of nonspecific EOI, by default the interrupt which is currently being processed or being served will be reset. In case of a specific EOI, the command specifies we have to I mean give through this OCW 2; you have to which bit is to be reset. There is one more EOI which is called automatic EOI.

Here there is no need of command; by default in the third INTA bar cycle in third INTA bar cycle; the corresponding ISR bit will be reset. But, the drawback of this Automatic EOI is so the ISR does not have any information on the interrupt which is being processed that is the drawback of this one. So, normally we will use either specific or nonspecific EOI.

(Refer Slide Time: 10:44)



Now, coming to the format of OCW 2; so the format of OCW 2 will be here we have L 0, L 2, L 1 bits. This is L 0, L 1, L 2 these three bits are going to decide the level to be acted upon. So, if this is 0 0 0 by default this is IR 0; IR 0 level will be used 0 0 1, IR 1 and so on up to 1 1 1 means IR 7.

Then the next bit will be by default 0 0. We are not going to use this and the next 3 bits is going to decide the mode. This is called R bit rotation; it is going to decide the rotation then SL bit EOI bit; and this is of course is A 0 bit which is 1 these are D 0 to D 7.

Now, coming for these 3 bits; we have 8 combinations and we have 8 different modes. If this is 0 0 1, 0 1 1 this is not in the proper binary order 1 0 1. So, there is some reason for why this is having some nonspecific order I will explain 0 0 0, 1 1 1, 1 1 0, 0 1 0. So, this side these two will represent end of interrupt end of interrupt; these two will specify automatic rotation and the remaining which is going to determine specific rotation; this we have already explained.

The other side the first one represents nonspecific EOI command. I have to explain the what is meant by nonspecific and 0 1 1 represents specific EOI command then rotate on nonspecific EOI command; rotate it automatic EOI command; rotate on automatic EOI command; rotate in automatic EOI clear; this is set and this is clear both are same, but the difference is only set and clear.

Rotate on specific EOI command and set priority command no operation. So, because these two represents end of interrupt, these three will be automatic rotation, these three

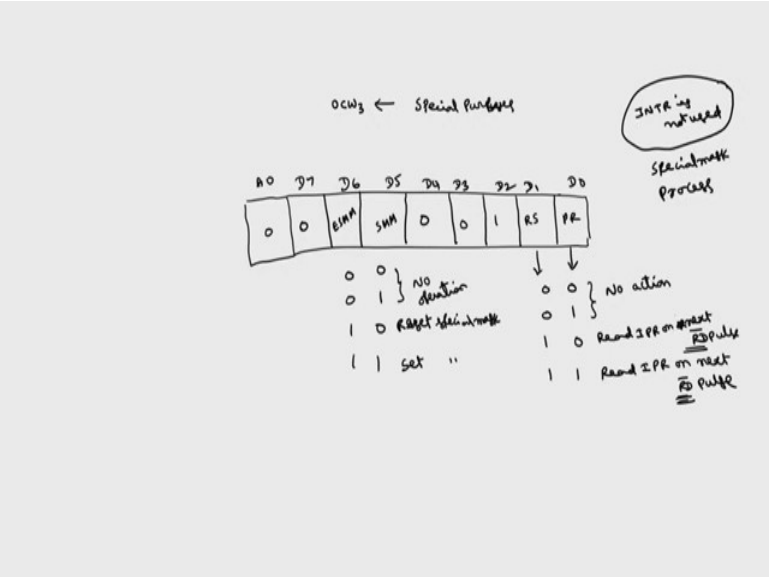
will be specific rotation that is a given this in the random order ok. And this nonspecific and specific automatic EOI just now I have explained and coming for this as specific rotation and automatic rotation this we have discussed in the earlier class. So, in case of a specific rotation as we know that so in case of specific rotation, we will be having this IR s will be assigned priority ok.

In specific rotation, any IR can be assigned lowest priority. For example, if I assign IR 4 as lowest priority; this will be clear if I give an example how this IR will be assign lowest priority. Then by default what happens is this IR 7, IR 6, IR 5, IR 4, IR 3, IR 2, IR 1, IR 0; so, if IR 4 is lowest priority; the lowest priority will be represented by 7. 0 is highest priority; then automatically the next one IR 5 will be having highest priority; 0, 1, 2, then 3, 4, 5, 6 these are the level of the priorities; in case of a specific rotation.

Whereas in case of automatic rotation, by default the IR which is being currently served will be having highest priority ok; so in case of automatic rotation, the IR which is being currently served will be having highest priority. Similarly the order of the priority will be for example, this if IR 7 is currently serving, so this will be having highest priority will be 0; then the next highest will be which is 1 will be assigned to this 2, 3, then 4, 5, 6, 7 ok.

This is the difference between the automatic relation and specific rotation there is one more rotation which is called as specially specific rotation ok. So in that case, so a slave if it request the master so while the interrupt is being served this slave can request for the another interrupt also. So, this is basically used with cascading mode. So, I will discuss that while discussing OCW 3. So, this is about the OCW 2. Now, OCW 3 will be used for only some special purposes operational command word 3; this will be used for only some special purposes.

(Refer Slide Time: 19:28)



So, in this the format of this OCW 3 will be this is PR, RS; then we have 0, 1 by default; then we have E, SMM, ESMM; then D 7 is 0; this is D 0, D 1, D 2, D 3, D 4, D 5, D 6, D 7; then by default A 0 is 0; D 4 is also 0 here. This is ESMM this is simply SMM; this is 0. So, these two bits will be 0 0 means 0 1, 1 0, 1 1.

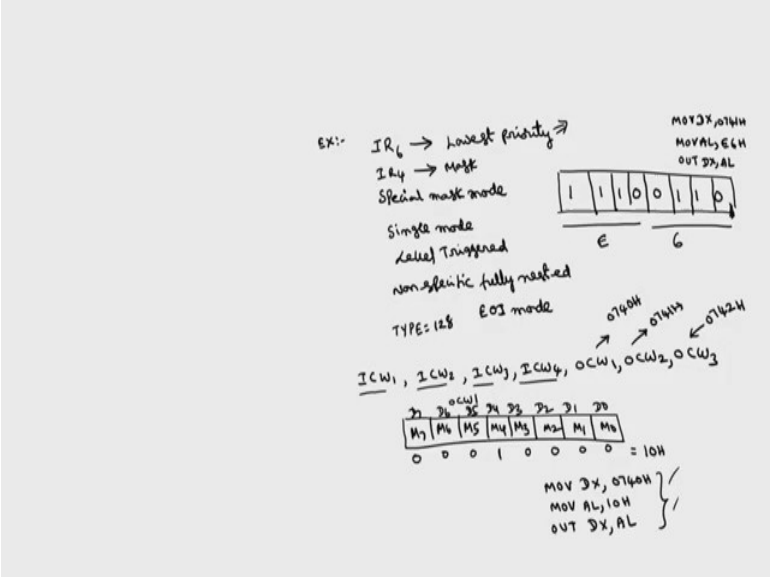
So, 0 0 and 0 1 there is no action; in case of a 1 0 read IPR on RD pulse next RD pulse read IPR and next RD pulse if it is 1 1. This is read bar pulse actually. So basically in this OCW 3, the INTR is disabled INTR will not be used. During the read bar cycle only the interrupt process will be takes place. So, in normal operation the IR 0, IR 1 so on up to IR 7.

If any request comes on this one, the IR with the highest priority will be serviced through INTR. So, this 8259 will send INTR to the microprocessor then the microprocessor acknowledge through INTA bar thereby the type of interrupt will be specified means the tape of call instruction; after that that particular interrupt will be served.

Whereas in case of OCW 3, in case of a special mask this is called special mask this INTR is not used in special mask process only read bar signal. So, during the second pulse, during the third pulse they are going to send this 1 0 and 1 1; and corresponding to these two bits this is the read specific mask set bit mask. This can be 0 0, 0 1, 1 0, 1 1; this is the reset specific mask reset special mask; this is set special mask and here there is no operation.

This is about the OCW 3 but, the OCW 3 will be rarely used in normal applications ok; so, here about the 3 operational command words of 8259. So, we have discussed about the initialation command words then operational command words. The initialization command words as the name implies they are used for initializing the 8259s and operational command words is going to decide which I mean ISR bit is to be reset and how this end of interrupt operation will be takes place.

(Refer Slide Time: 24:34)



So, I will take one example how this ICWs and OCWs will be formed for a given specifications and I will end this 8259. So, I am giving an example. So, here I want to use 8259; I want to assign IR 6 as lowest priority; then we have special mask mode and the single level triggered; single mode and level triggered mode.

These are the specification that I want to use and I have to program this 8259 ok; nonspecific fully nested then specific EOI mode. So, what are this ICW 1, ICW 2, ICW 3, ICW 4; Similarly OCW 1, OCW 2, OCW 3. There are totally 7 command words; 4 initialation command words and 3 operational command words ok.

So, if I go for this operational command word 1 OCW 1. Suppose if I want to mask IR 4; so, this D 0 to D 1, D 2 so on up to D 7; this is M 0, M 1, M 2, M 3, M 4, M 5, M 6 and M 7. So, I want to mask only IR 4 so, IR 4 means this correspond to this bit will be 1; remaining all will be 0s. So, the hexadecimal equivalent will be 10H ok.

Now, if I want to use a port address say OCW port address is from 0740H; this is 0741H; this is 0742H; then in order to load this one. So, what you will do is MOV DX comma 0740H; then MOV AL comma 10 H; OUT DX comma AL.

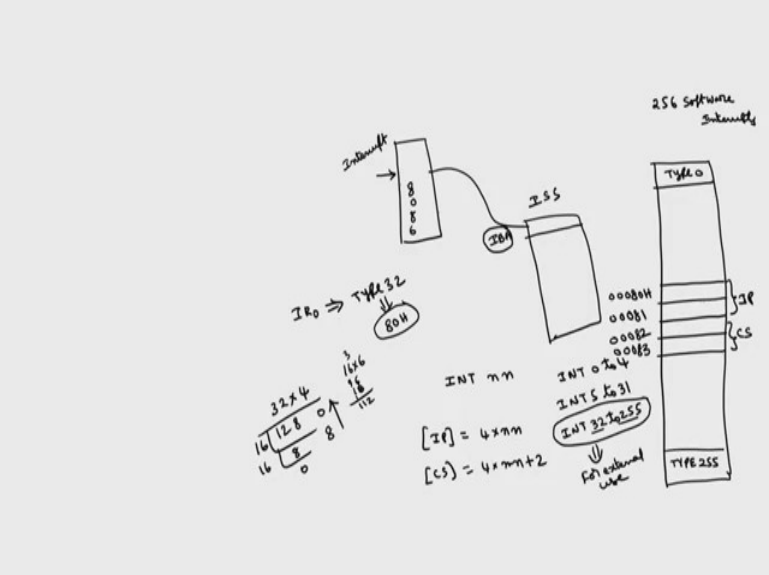
So, these are the instructions used to load this 10H on to OCW 1 whose address is 16 bit. In case of 16 bit address you have to take into DX register and then you have to specify in the OUT instruction DX register. So, similarly we can form OCW 2; if you form OCW 2 here. So, I want to assign lowest priority to IR 6. So, this L 2, L 1, L 0 bits will be 6 means 1, 1, 0 ok.

So, then corresponding to this, this is nonspecific EOI mode fully nested. So, the corresponding bits of this the remaining three will be this is 0, this is 1, 1, 0 and this is 1. So, what will be the control word? This is E 6 ok. So, the corresponding instructions for this one will be MOV DX comma 0741H; MOV AL comma E6H; then OUT DX comma AL.

So like that, we can initialize these registers. Similarly you can have the formats of this ICW 1; you take the corresponding addresses and you take the format based on the specifications. We can generate the hexadecimal equivalent of that ICW 1, ICW 2, ICW 3, ICW 4 and you can load those values into corresponding registers by using similar type of instructions ok.

So, say I told the type of this program also has to be mentioned. So, if I use this TYPE as 128 say. So once this IR is requested, a call instruction will be executed because interrupt process is nothing but so, whenever the microprocessor executes.

(Refer Slide Time: 30:42)



So whenever interrupt occurs, it completes the execution of the current instruction; it will go to a sub routine called interrupt service sub routine this is called interrupt service sub routine; and this first address is called interrupt branch address this is called interrupt branch address.

So, this interrupt branch address will be different for different IRs; totally, we can how 64 different IRs in a cascading mode ok. So, this will be one of this TYPE 0 to TYPE 256 interrupts. We know that in case of 8086, we have 256 software interrupts. So, TYPE 0 to TYPE 255 each will take 4 locations. So, the contents of IP will be so, in general if you have INT nn contents of IP will be is equal to 4 into nn; and contents of CS is equal to 4 into nn plus 2 ok.

And in this actually INT 0 to 4 we have some special purpose and INT 5 to 31 they are reserved; so, only INT 32 to 255 are allocated for the user for external use. So, corresponding to this IR 0 to IR 7 even again in in cascade mode IR 0 can be connected to eight more interrupts. So, we have to use one of this INT 32 to INT 255 ok.

For example, if I use INT 32; 32 type of interrupt for IR 0; say IR 0, I want to operate in TYPE 32. So, one of the corresponding address is 32 into 4 is 8, 128 is decimal hexadecimal will be 16; 16, 6 will be 96; 16, 7 will be 112. So, 16, 8s will be 128; 0 is the reminder; 16, 0s 8 is the reminder; so, 80H; so, this is corresponding to TYPE 32 means this is correspond to 80H hexadecimal value. So, in 0080H here we have 00080H and 81H you have to store IP; 00081H and in 00082H, 00083H you have to store CS;

then you have to jump to the corresponding address. So, here this way IP values will be there; here CS values will be there. So, using this IP and CS values it will compute this interrupt branch address.

So, this is also another specification that you can give to program the 8259 ok. So, these are all the different specifications. Using these specifications, you can write the programs for 8259 ok. So, this is all about the 8259. So, we will discuss the next I mean peripheral device which is a DMA controller in the next class.

Thank you.