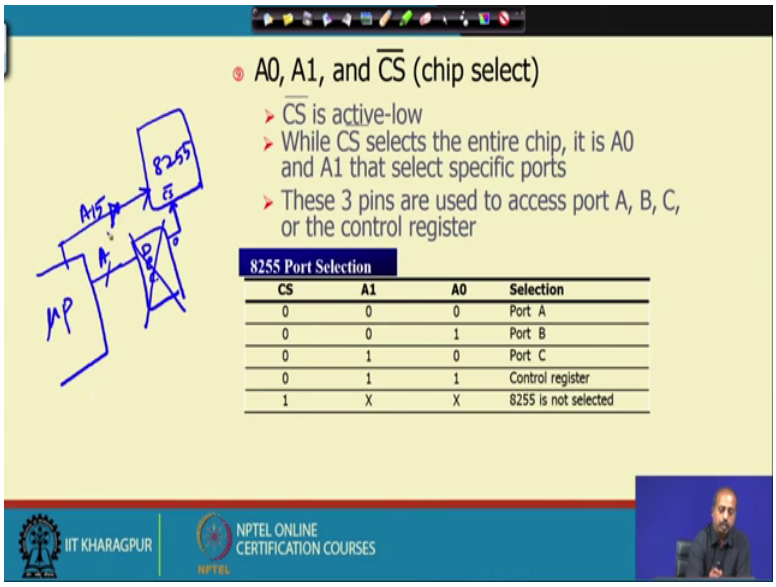
Microprocessors and Microcontrollers
Prof. Santanu Chattopadhyay
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture - 55
Interfacing (Contd.)

So, other important pins that you have in 8 0 8255 are this A 0, A 1, CS bar pin. So, A 0 and A 1.

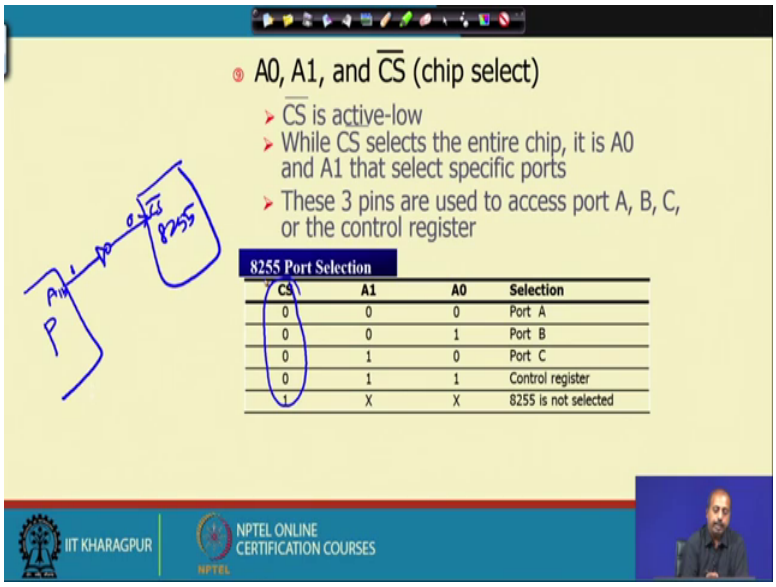
(Refer Slide Time: 00:32)



So, they are used for selecting individual ports of 8255 and CS bar is the chip select. So, if CS bar line is high. So, 0 8255 actually not activated. So, it will not do any work. So, the file CS bar selects the inter. So, we can; so, if you if you so, your decoding logic. So, should select this CS bar properly. So, what I mean is that if suppose, this is the microprocessor or microcontroller that you have and this is your 8255, then this address bar lines that are coming out from 8 from your microprocessor. So, that should go through some decoding logic. So, this may be a decoder, this may be a decoder and this decoder should select this 8255 chip. So, this is the CS bar line. So, for selecting this line; so, decoder should put A 0. So, decoder should be programmed in such a fashion that for a particular address bit combination, this 8255 will get selected and accordingly. So, maybe I can say that a; if this if I say for a simple chip case, if I do not have this decoder. So, I can just take the A 15 line of this from this microprocessor and connect it

to 8255. So, this A 15 may be connected through some inverters. So, put a put a inverter here let me draw it fresh diagram. So, what I am telling is that this is the processor and this is the 8255.

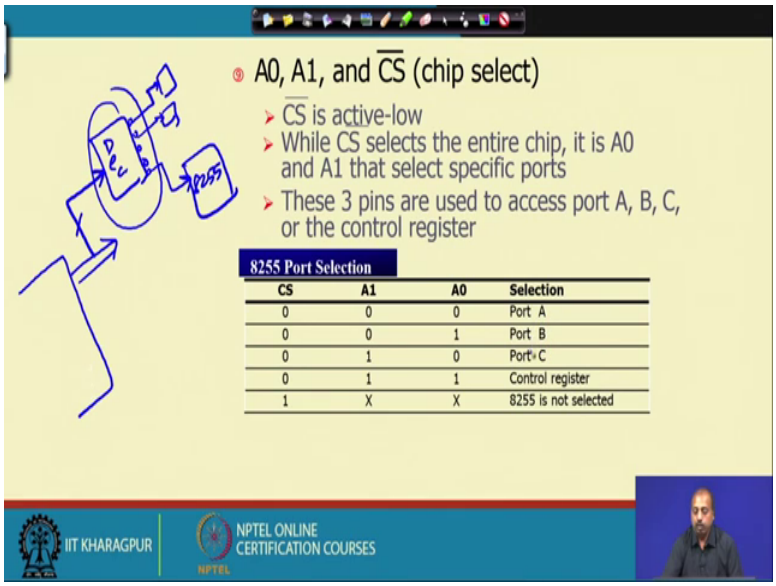
(Refer Slide Time: 01:49)



So, what you can do? You from the address bar, you take this A 15 line and connect it to the CS bar line.

So, whenever this A 15 bit is 0, whenever this A 15 bit is 1. So, this will be 0. So, 8255 will get selected. So, provided your entire memory is located in the lower half of the total address space ok. So, you can say that I will be setting this A 15 equal to 1, it will mean that I am going to do the 8255 operation. So, this CS bar line. So, this selection has to be generated by the decoder. So, here it is a very simplistic one, ideally, if you have got a more detailed situation like if this is the processor and from here, the address bars is going some of the address bars lines are going to the decoder.

(Refer Slide Time: 02:36)

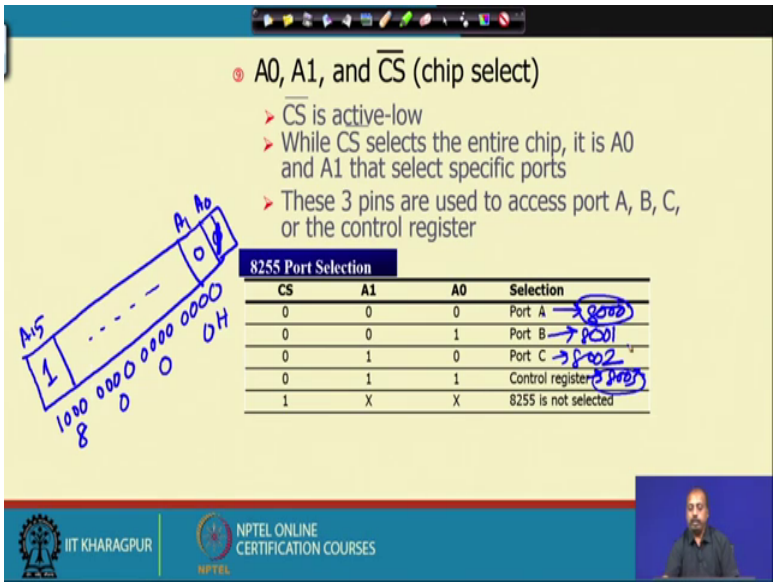


So, this is the decoder; some of the address bars line are taken here and then this has got a number of chip select and this individual chip selects are going to individual memory chips. So, they are going to different memory chips and some of these decoder lines may be dedicated for connection to 8255. So, this way also, you can figure out like; what is the exact address of this 8255 that based on the based on this decoding logic that we have incorporated here the address can be found out.

However this A 0 and A 1; so these are the 2 bits that has to be that are to be set for selecting the registers within 8255. So, within 8255, we have got 3-4 main registers; one is the control register and the rest are the port different port registers. So, if you are; so, control register is required for configuring the 8255 chip in different modes and this port registers A, B and C. So, they are actually for putting the values onto the ports or if you want to read the content from the port onto 8 to want to the processor. So, you can use those addresses; so depending upon your address setting. So, the individual port will get selected.

So, we can ok; I will take an example and try to explain what may be the port addresses like say if I have got that 15 bit sorry 16 bit address.

(Refer Slide Time: 04:21)



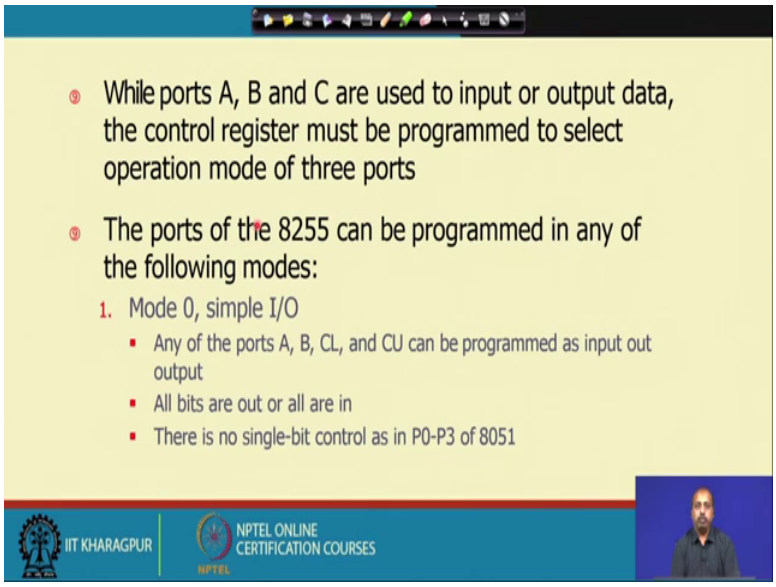
And I can say that ok. So, these 2 bits are say A 0 and A 1, these 2 bits are A 0 and A 1 and this rest of the bits are there and this is the bit A 15. So, I can do it like this, I can put if I say that in my case, whenever A 15 is equal to 1, it will mean that 8255 will get selected that is for the rest of the for rest of the modules rest of the memory modules and all that. So, I have got this A 15 bit equal to 0, the decoder will not select those modules if A 15 is equal to 1. So, this selects the 8255.

Now, if you look if this part this A 0 A 1. So, A 0 A 1 when it is 0 ok. So, 1; so, if one possible address is say 1, 0, 0, 0, 0, 0, 0, 0, 0. So, that way, I have got say this A 15. So, that is the 8 bit, then all these are 0. So, this is the 16 bit address bus. So, you see that this is basically 8000 in hexadecimal. So, I can say that this is basically 8000. So, in hexadecimal; so, I can say that if I if I connect like that the A 15 line connects to the CS bar of this inverted A 15 line connects to the CS bar pin of 8; 8255, then this port A address is turning out to be 8000.

Similarly, if this bits; this bit is A 0 is 1, if A 0 is 1, then the port B will be selected. So, this address is 8001, port C is 8002 and this one is 8003. So, this way, based on your decoding logic and this A 0 and A 1 bits of the address bars connected to A 0 and A 1 of your 8255 chip. So, you can find out the port addresses after that if your microprocessor write something to this particular memory location 8000 so; that means, it is accessing porting. So, if it is writing something 8003. That means it is doing it is writing to the

control register. So, this is the technique. So, we have got this CS bar as active low and when CS bar selects entire chip A 0 and A 1 will select the specific ports. So, these pins can be used to access ports A, B, C or the control register.

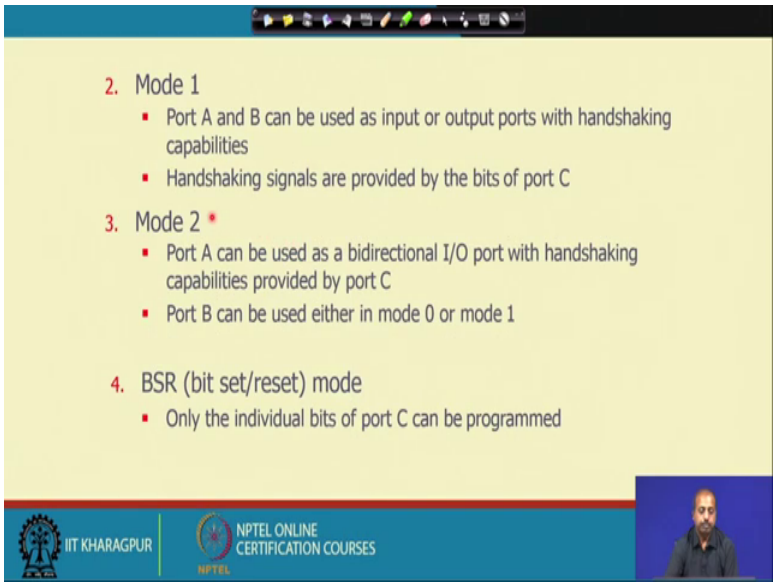
(Refer Slide Time: 06:57)



Now, this A, B, C port. So, they can be used for input or output data and control register is used for controlling the operation of these 3 ports ok. So, we can have A, this 8255 operate in various different modes the first one, the first mode is the mode 0 or simple. So, this is the most popular mode and almost all the applications that you have may mostly. So, that they will be using this port this mode 0 of 8255. So, in mode 0 any of the ports A, B, C lower and C upper, they can be programmed as input or output port.

So, input port or output port and all bits are all bits are out or all are in. So, we cannot do it individually. So, I either the end all the 8 bits port A or output and all are all the 8 bits are port A are input. So, there is no bit selection, though we have set previously that port C is port C is bit addressable in some mode, but in mode 0 that is not true that is true for a particular mode only. So, so in all the entire port is configured in the same way, either it is input port or output port unlike 8051 where these ports P 0 to P 3, they have got this bit controllability. So, here we do not have that feature.

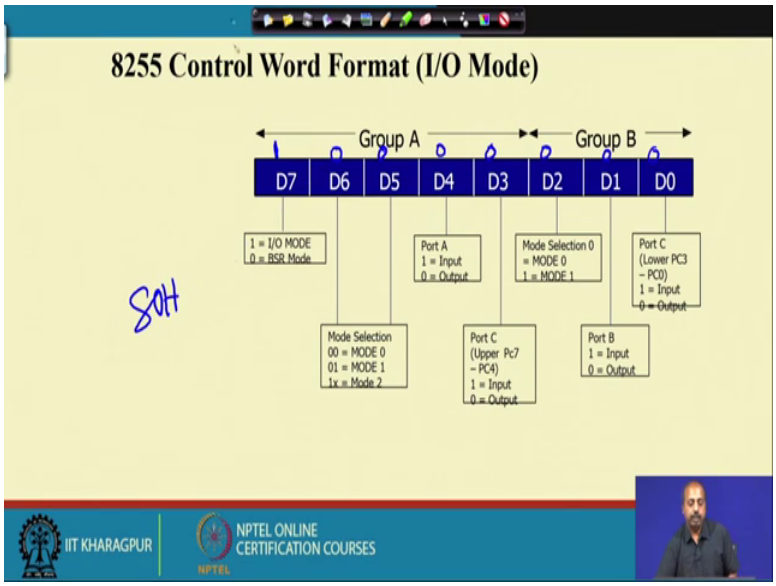
(Refer Slide Time: 08:24)



Then, we have mode 1, in mode 1, port A and B can be used as input or output with handshaking capabilities and this handshaking signals will be provided by bits of port C. So, handshaking capability means, you can you can say, you can put the data on to the data bars and from one master device, when you are putting into slave device, then when the data is put accordingly the strop signal is given, then the receiver; it will receive that value and it will put, it will put another acknowledgement and all that. So, that way it will go. So, all those things are controlled by port C, this handshaking signals are given by port C and port A and port B. So, they are used as the data values they are by data ports.

Then we have mode 2, we have port A can be used as by directional I O port with handshaking from port C and port B cannot be used in mode 2. So, they can be ok, they can be configured in mode 0 or mode 1 only. So, we cannot have this mode 2 with port B and there is another interesting mode which is known as bit set reset mode or BSR mode, here you can control the individual bits of port C ok. So, that way if you are really looking for the bit level addressability. So, you have to go to mode 4 which is the mode 3 which is the BSR mode or bit set reset mode.

(Refer Slide Time: 09:53)



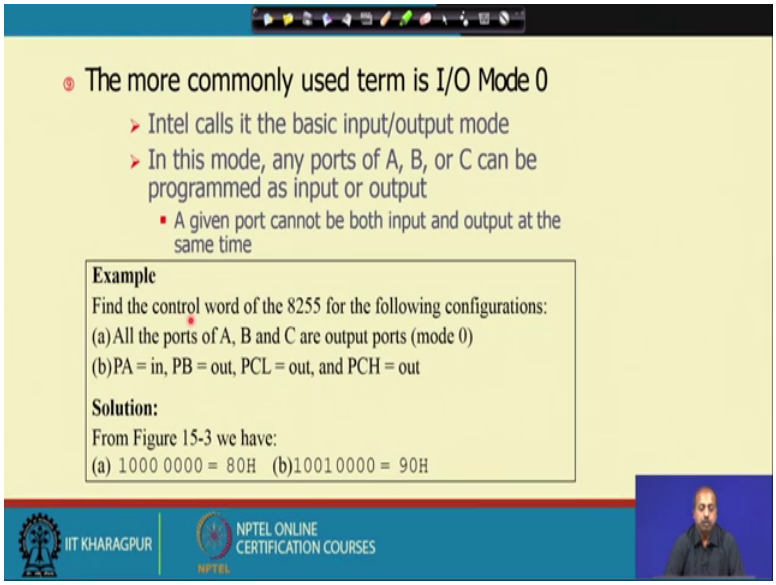
So, the first job that we have to do is to set the control register. Now I have set that 8255 has a control register where you have to programme properly. So, that you can it can be the port can be utilized by the microprocessor. Now in this case, we have this entire control register can be divided into 2 groups. So, group A and group B, in group A, we have got port A and port C upper bits ok, you see the port A and port C upper bits are control by this group A setting and group B setting controls port B and port C is lower bits and there is another bit in the group A register which is called bit D7.

So, here if this bit is 1, it will mean I O mode and if it is 0; that means, it is BSR mode if it is in I O mode. In that case, this bits D 6 and D 5, they will identify the mode like it is 0 is mode 0 0, one is mode 1 and one x is mode 2. So, that way we can select like in the previous slide, we have seen in that we can have mode 0 1 2 or BSR mode we did not say explicitly mode 3 ok. So, that is BSR mode. So, for setting BSR mode, we have to set this D 7, D 2 0, this D 7 1, then you can select this other bits and accordingly the mode will be set.

So, if you want to set this port A as input mode then this bit D 4 has to set to 1, if you want port A as output, then this D 4 bit should be 0. Similarly port C upper, if you want to set it as input port, then the bit should be one D 3 bit should be 1, otherwise D 3 bit should be 0, then this group B, we have got 3 bits D 2, D 1 and D 0 out of that D 2 is for mode selection for group B. So, it is mode 0 or mode 1 and then in mode 1 in mode 0,

you have got this you. Now, after this after the mode selection; so you can control individual ports like port B can be set to input or output and port C lower part PC 3 to PC 0, they can be set as input or output by controlling this D 0 pin. So, this way by setting proper control over. So, you can configure all the ports from 8255 and for different input output purpose.

(Refer Slide Time: 12:24)



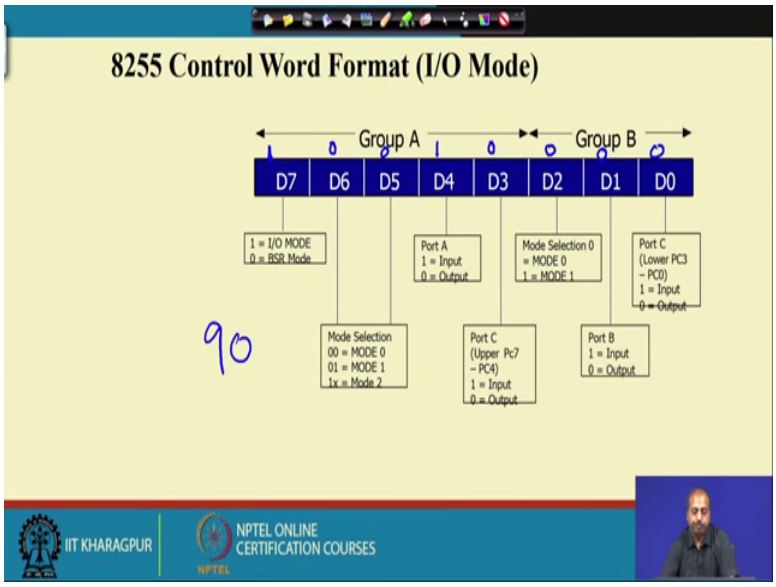
The most commonly used term is I O mode that is 0 I O mode 0 ok. So, that BSR mode is also not that common. So, 8255 the applications that we will see mostly it is using this I O mode and in mode 0 so, that is why it is a Intel, they call you the basic input output mode. So, they so much used that it is called it has got the name basic input output mode, in this mode, all ports of A, B and C can be programmed as input or output and a given port cannot be both the input and the output at the same time ok.

So, these are this is an example. So, suppose we want that all ports of A, B, C, they are output mode ok. So, in this case if all ports of A, B, C are in output mode and in mode 0. So, you can see in the previous setting. So, this is I want I O mode. So, this bit should be 1. So, this bit should be 1 and then. So, it is mode 0. So, these 2 bits are 0 and then this port A is all this is output mode port C is also in output mode. So, this is again mode selection.

So, 0 is mode 0. So, this is 0 and this is also output mode. So, this is also 0 and this is also 0. So, if you do that you see the control what we are getting is 80H ok. So, that way

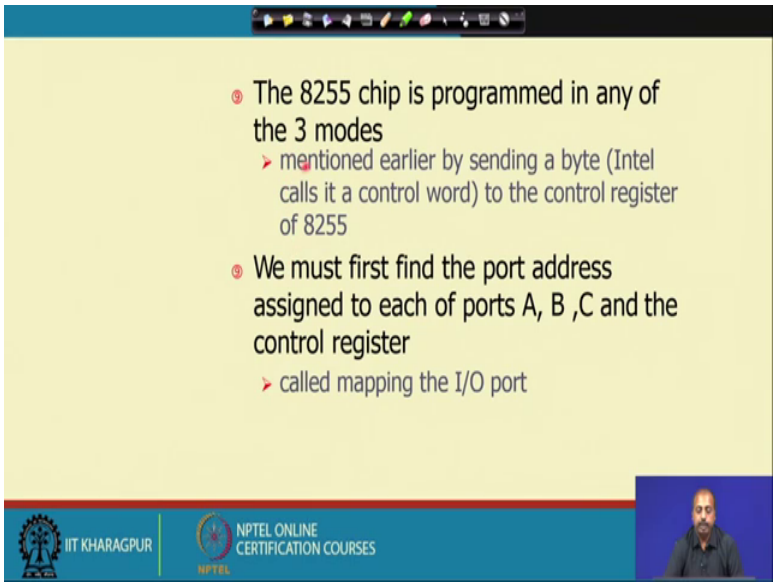
we can say that when we are setting when we are doing this setting when we are doing this setting that all port bits are in output all ports are in output mode, then this is the controller the other hand, if you say, that port A will be input port B will be output PC low should be output and PC high should both of them are output, but port A should be input mode. So, port A has to be in input mode. So, again this is I O mode. So, this has to be one this has to be one.

(Refer Slide Time: 14:25)



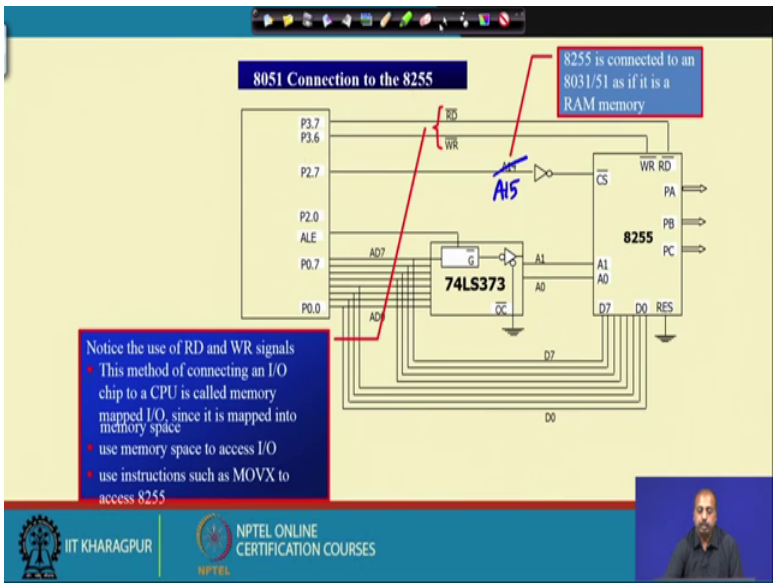
Then mode 0; so they are mode 0, now port A is in input mode. So, this has to be 1 and port C is output mode. So, that is 0. So, this is again mode 0 port B is also in mode 0 and this is also 0. So, the pattern that you get is 9 9 0. So, let us see what pattern we have got there. So, you see that pattern is 90H. So, this way you can have these individual ports settings by controlling the controller setting.

(Refer Slide Time: 15:42)



Now, we can now we have some connections in which this 3 8255 chip can be programmed in any of these 3 modes that we have seen. So, first of all we have to see the port address assign to the ports A, B, C and the control register. So, this is the called the mapping of I O port. So, this is the first thing that we have to do like to which address is it connected the how to how to access this individual ports.

(Refer Slide Time: 15:36)



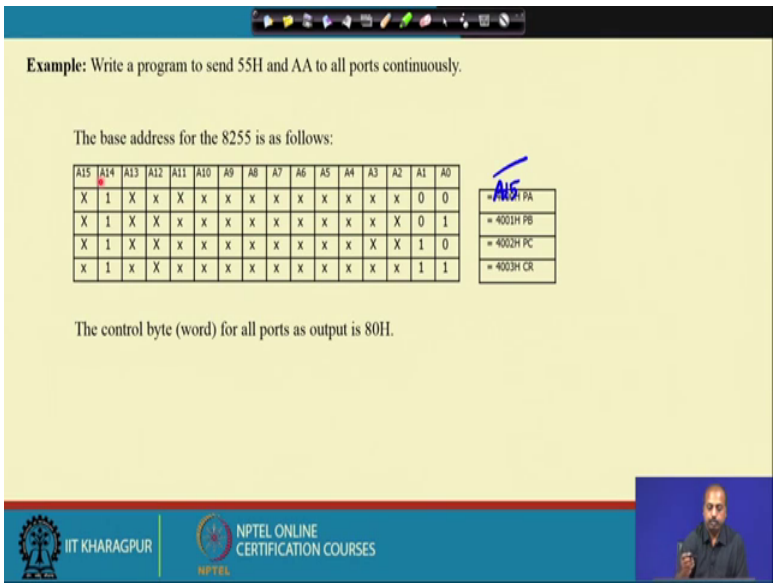
So, this is a typical example, suppose the connection is like this that we are having this, we are having this 8255 and this A 0 and A 1 lines from our address databus, you see that

port A port 0 bits are connected to 7 4 3 7 3 and port 0 bits, they have got alternate function of this multiplex address databus lower order address and databus. So, they are actually connected to this through this buffer, through this buffer, I am getting this A 0 and A 1 lines stable.

So, those lines are connected to the A 0 A 1 pins of 8255 and this P 2.7; from port 2.7, I am getting the line A 14 and that is connected to this 2.7 is actually A 15. So, this is not A 14. So, it is connected to the CS bar line. So, this A 15 line; so if it is P 2.7 A 15. So, A 15 line is basically going to inversion of that is going to CS bar. So, whenever this bit is 1. So, this 8255 chip will be selected, then this read bar and write bar lines are going and these are the setting. So, this is the connection. So, this reset of 8255 should be connected to ground read bar write bar line should be connected to this.

So, this particular connection; so, this type of connection is called memory map I O because the 8255 will be visualized by 8051 as a memory location only. So, whenever this A 15 bit is whenever this A 15 bit is, it will mean that it is 8255. So, this is A 15 not A 14. So, whenever A 15 bit is 1. So, it will be going to 8255. So, we can use it for memory access even this 8255 ports can be used as memory access and we can use the MOVX type of instruction for getting access to the 8255 locations ports.

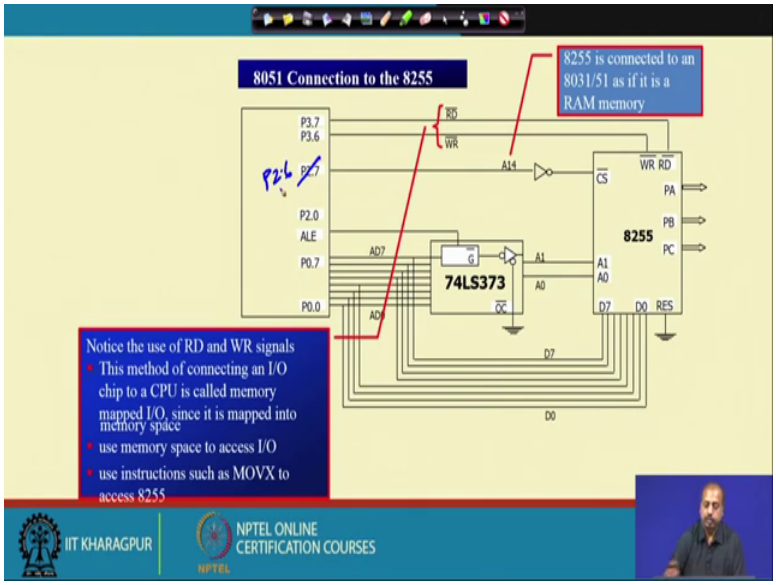
(Refer Slide Time: 17:59)



So, this is the thing that we have got. So, so if we have connected a 14. So, actually there is a problem with the previous diagram when this particular table. So, this A 14; so if you

connect then the A 15 line, this should be if it is A 14, then this one should be P 2.6. So, this one should be P 2.6.

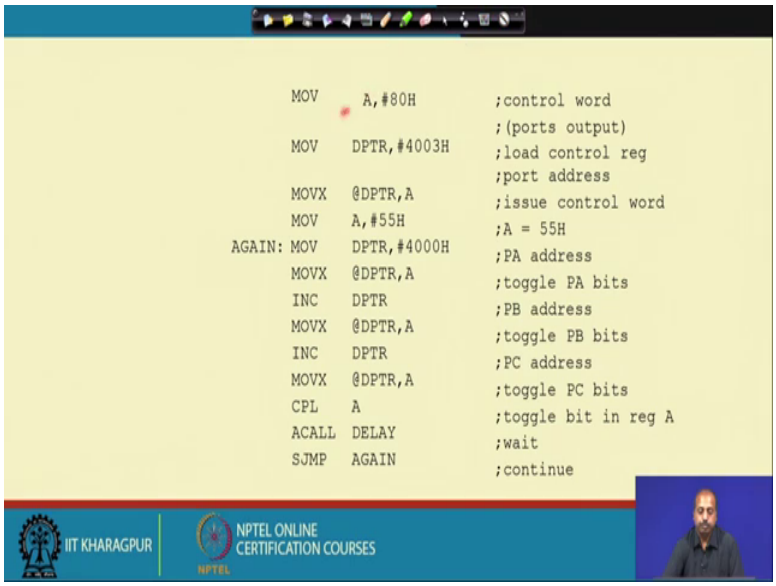
(Refer Slide Time: 18:33)



So, for that assignment, the next table is valid ok. So, this table you see that we have set the this A 14 to 1, then our 8255 chip will be accessed and this A 0 A 1, those bits are 1. So, I can say that port A is the corresponding location is 400H there may be many locations. In fact, the folding concept that we have discussed previously; so, they will have have this, there will folding because all this bits are. So, all of them will be coming to the port A of 8255.

And then this A 0 one are being A 0 and 1. So, 4001 that is port B 1 0 is that is 4002 is the port C and this control register will be at 4003 the control byte for if you want to put all port output then we have seen that the control setting is 80H. So, with this, we can we can try to write a program where we will sending the pattern say 5 5 x and 22H, 55H and AAX to all the ports continually that we want to complement the port bits.

(Refer Slide Time: 19:55)

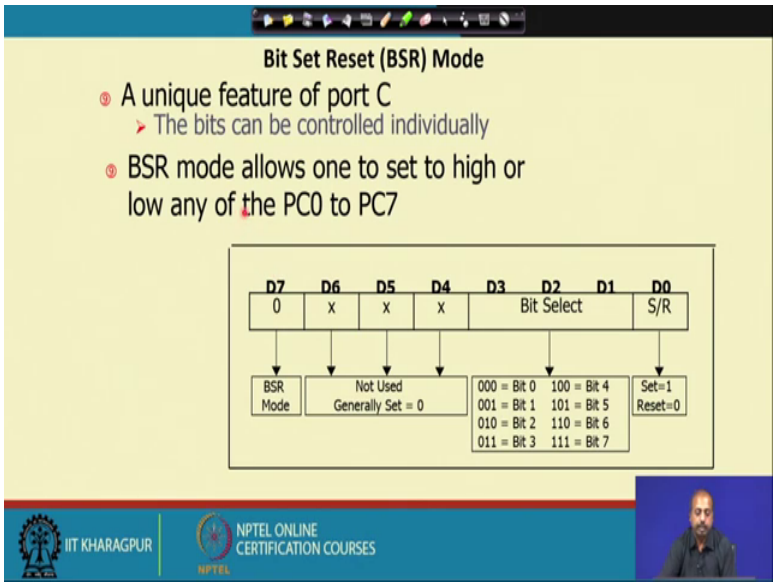


So, how to do this first we have to put the control mov to the control register. So, this move a comma 80H. So, this will configured this will move the pattern 8 0 2 this a register then we load this DPTR with the control register address control register port address 4003 for the particular decoder setting that we have done; so, now, this MOVX at the rate DPTR comma A. So, this 80H which was there in a register will go to the address 4003 as a result the control register will get 80 and all ports will be configured as output, then we get the value 55H into the a register and we move that and get in the DPTR value 4000. So, 4000 was the address of a register of port A and so, in the next instruction when we do move at the rate MOVX at the rate DPTR comma A, so, this 55H which was in a goes to the location 4000.

So, that goes to the port A we increment DPTR. So, that now it points to port b. So, again the same thing we put the 55H value on to. So, that port B increment DPTR again to come to port C and we put this value DP, this a value into DPTR. So, that is that 55H goes to the port C, then we complement a. So, all bits of A, they are toggled, now in the A register, you will have the pattern A. So, we put a delay routine and then SJMP again. So, we come back to this point where we will be again outputting the pattern AA to port A, then AA to port B, AA to port C, again, this compliment will take it back to 55.

So, this way, this programme will be doing the operation. So, it will putting the proper values on to individual bits of the port.

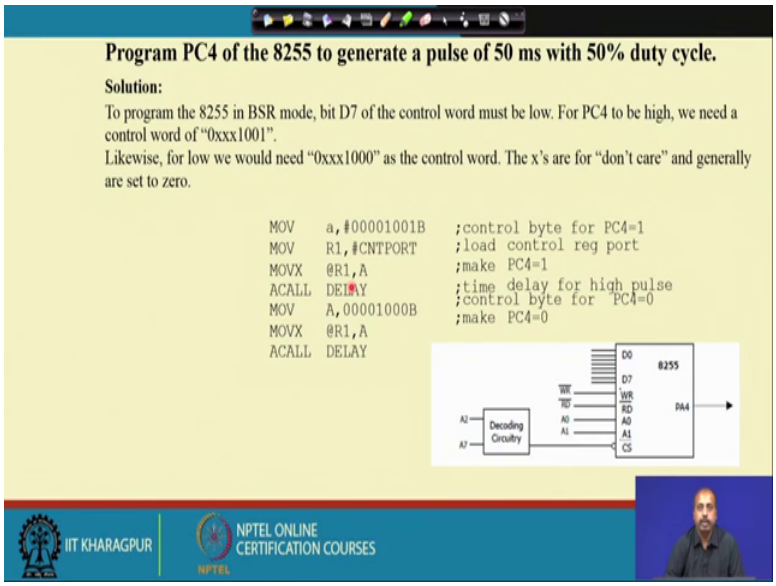
(Refer Slide Time: 21:58)



Another important feature that this 8255 has is the bit set reset mode. So, this is unique feature of port C and the bits can be controlled individually. So, if you it will allow you to set allow you to set the high or low any of the PC 0 to PC 7 bits. So, you can see first of all this D 7 bit this is the controller register setting. So, the D 7 bit must be 0 then because in that case it will mean that it is a BSR mode bit set reset mode now this D 6, D 5, D 4.

So, they are they are not required because this mode settings are meaningless if bit set reset D 7 is 0. So, it is BSR mode. So, this mode selection of port group A that does not have any meaning; so, they are by default normally we are setting it to 0, then the next 3 bits, they will select the bit that you want to set or reset ok. So, there are there are 8 bits. So, you can select this bit numbers by putting this bit select accordingly. Then, if you set if you want to set this bit, then this D 0 should be 1, if you want to reset the bit it should be 0. So, this way you can control this individual bits of this port C and you can make you can set or reset those bits.

(Refer Slide Time: 23:24)



So, suppose we want to write a program in which we want to generate a pulse of 50 millisecond with 50 percent duty cycle on PC 4 that is port C bit number. So, first thing that we have to do is put 8255 in BSR mode we and D 7 will be low and PC 4 be high. So, that is the setting that this is that D 7 bit which is 0 next 3 bits do not have any meaning and. So, then I have to put the bit number. So, this is bit number four. So, it is 1 0 0 and then we want to set it to high. So, this is setting to high. So, PC after doing this a register has got this value. So, if you move this value.

So, if you move this control port value control. So, load control register port. So, this will be move to this control register address move to r one and then MOVX a comma at the rate r one. So, this control port will be writing this value. So, that way this PC 4 bit will be set when we put a delay then we have set the now we have to make the PC 4 0. So, this next this control setting there should be a hash here the hash is missing; so this control setting. So, this particular control setting will like this PC 4 equal to 0 and then the same thing r one has got the control port address; so, at the rate MOVX at the rate r one comma A. So, this will be moving this control of content of a register on to the port C then a call delay.

So, it will be delaying this, after this, there should be a loop around this that is not shown here. So, we can we can do that it will become continually looping this operation and if this entire body has to be repeated. So, this way we can use this 8255 to generate pulse

with 50 millisecond delays, 50 millisecond delay can be controlled by this delay routine. So, that is not shown here explicitly. So, you can know how to generate these delays by using this timers or using soft delay, etcetera. So, we can do that. So, it is 50 millisecond pulse width and 50 percent duty cycle, 50 percent duty cycle is ensured here, but 50 millisecond is not shown explicitly. So, we can do that.