Microprocessors and Microcontrollers

Prof. Santanu Chattopadhyay
Department of Electrical & Elctronics and Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture-53
Interfacing

So, another very interesting topic that we have or in this microprocessor microcontroller course is about the interfacing techniques, like we have seen these microprocessors and microcontrollers so they can produce digital outputs they can take digital inputs. Some of the microcontrollers can take analog input as well which is a VR series we have seen they can take analog inputs now apart from that. So, if we want to connect the devices a number of devices which are required for making a system for example, you may need to connect some digital switches on off switches or you may like to connect some your some LEDs for display or some 7 segment displays are there then we want to connect some keyboards for some basic input output.

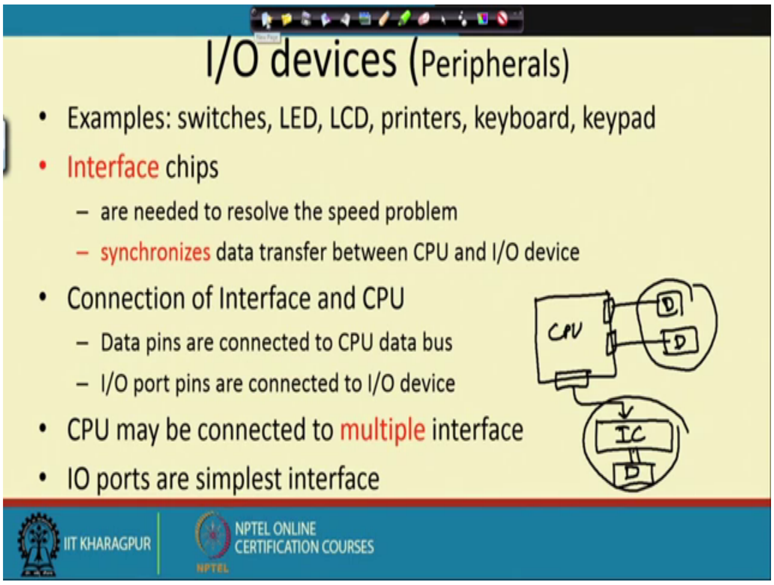
So, may not be a detailed keyboard like that we have with the PCs, but it may be a small keyboard by which you can enter the basic, basic digits of a basic digits and say characters particularly the hexadecimal characters ok. So, the hexadecimal keyboard you can interface or you can have interfacing with say LCD displays ok, so like that so, a number of devices that we need to connect ok.

So, how those devices are generally connected with these processors say microprocessor or microcontrollers. Now, we will see that this connection pattern is more or less standard and whichever processor that you are connecting to. So, from the device side the pins remain same or the controls remains same. So, if you are connecting to 8085. So, in that case you may have to connect it through some other chip like 8255 which is which has got a number of ports in it or if you are connecting to say 8051 then you have got already these port pins available ok. So, you can directly connect through the port pins the p 0, p 1, p 2, p 3 like that.

So, all those port pins are there or AVR microcontroller pic microcontroller. So, we have got those pins available. So, we can connect through those pins. So, to make our discussion simple so will not be considering it the examples with 8085. So, we will take examples of connection with 8051 microcontroller with understanding that if you want to connect it to say it is 085. So, in most of the cases what we have to do is that we have to put another chip 8255 which is a peripheral interface chip that provides a number of ports and through which it has to be connected of course, we will see how to connect the 8255 chip to the to your system, so that we will see. So, rest of the thing will remain same. So, programming part there will be small changes that you can understand, hardware side there is not much change ok.

So, this is the portion so, will be talking about the interfacing.

(Refer Slide Time: 03:18)



So, why I o devices input output devices or peripheral devices they are commonly known as. So, examples like say switches LED, LCD, printer, keyboard, keypad. So, you can have you can think about different types of input then, let us say digital to analog converters analog to digital converters. So, like that you can think about a number of such IO devices that you need to connect to a system. Interface chips they are needed to resolve the speed problem because the device and this CPU or this processor they were not operating at the same frequency so somehow to meet the gap between their speed.

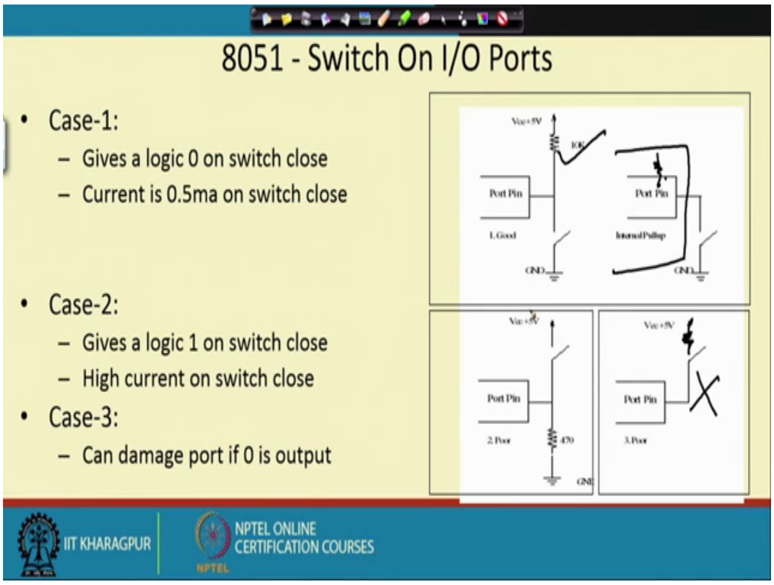
So, we will like to put some intermediary chip. So, that they can do some sort of buffering and all that ok. So, that way the transfer the operation becomes smooth, like these devices are mostly for a human interface ok. So, human beings are involved in operating those switches or looking on to it on to the display and all . So, this type of things we have to take care ok.

So, this interface chips are provided that can resolve these problems. So, it also synchronize data transfer between the CPU and IO device. So, that is there. So, connection of interface and CPU so, data pins are connected to CPU data bus and IO port pins are connected to IO device. So, that way we are going to connect the CPU with the IO devices now the CPU may be connected to multiple inter multiple devices and IO ports are the simplest interface, IO ports are simple port bits. Like say we can have a situation like this that we have got this are the CPU and then the CPU directly has got this number of ports.

So, this may be one port, so this may be another port, so like that. So, from this I can connect some device directly. So, that is one possibility, another possibility that we have is that we have got some other interface chip and through say this is a port. So, from this port we have got connection to another interface chip, so this is some interface chip and with that interface chip. So, we have got the device connected, so the devices whose functionalities are complex. So, they will be connected like this and the devices whose functionalities are simple that they will be connected directly with the ports of the processor.

So, we look into many such examples of these devices and their connections.

(Refer Slide Time: 05:52)



So, first of all we see how can we connect a switch on the IO ports. So, these are the various facilities by which you can do this say so, the first configuration so it gives logic

0 on the switches. So, when the switch is closed so this point will the voltage will be 0. So, the as a result you will get a 0 here. So, if the switch is closed you get a 0 here if the switch is open. So, if you measure the voltage at this point you get a 1 and there is a resistor here. So, this is the current limiting resistor. So, the current will be not more than so it is 10 k. So, the current will be 5 volt divided by 10 k so the current is limited. So, this is good because it will not the current that will be going through the port pin is limited by this resistance ok. So, this gives a current of about point 5 milliampere when the when the switch is closed.

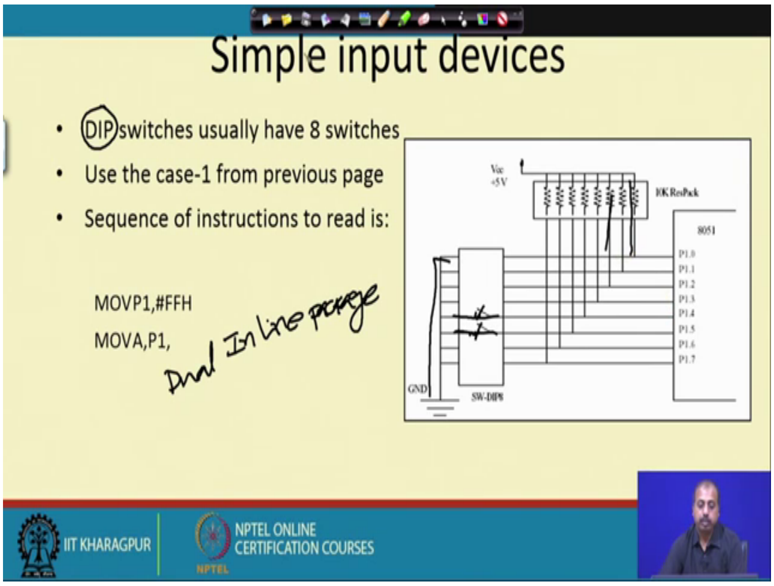
And in case 2 it gives logic one on this switch is closed and high-current the servers. So, this is when the switch is close so say this configuration. So, this configuration we have got this when the switch is closed. So, there so the, you will get a high voltage here. Now, the difference from this configuration to this configuration when this we here when the switch is open. So, you are getting a voltage value, but here the voltage value will be, when the switch is close to the voltage value is determined by the current that we have here the drop across this resistance. So, that will determine this thing, but over the current flow will be high because this is 470. So, they are 470 ohm. So, this compared to this 10 k. So, as a result when the switch is closed there will be high current flowing into the device. So, case 3 is this one.

So, here I have got this situation. So, we have if we close this pin. So, close this switch then this line is directly connected to port pin and this is a bit risky because this port pin may. So, you see that there is no resistance here so that their very high current will flow through this line. So, it is it this connection is not advisable. So, we should this is not advisable. So, whenever we have got this type of configuration so it is better to have it like this. So, I have a current limiting resistance at this point and then make the connection ok. So, it is always said advisable like this.

So, and this one you see that when this, the switch is closed. So, you get a 0 if the switch is open so it is it depends on that technology. So, maybe it is open so this line is this line is open. So, as a result that open may be taken as logic high. So, if that is the case if there is an internal pull up available so if the if it says that there is an internal pull up. So, inside the chip so if this is the chip. So, if there is an internal pull up to this pig then of course, there is fine, but otherwise this will be having difficulty because the current flow will be a problem, but this case 3 is not advisable. So, we now we try to avoid this case 3.

Next we see into some simple input devices like this.

(Refer Slide Time: 09:13)

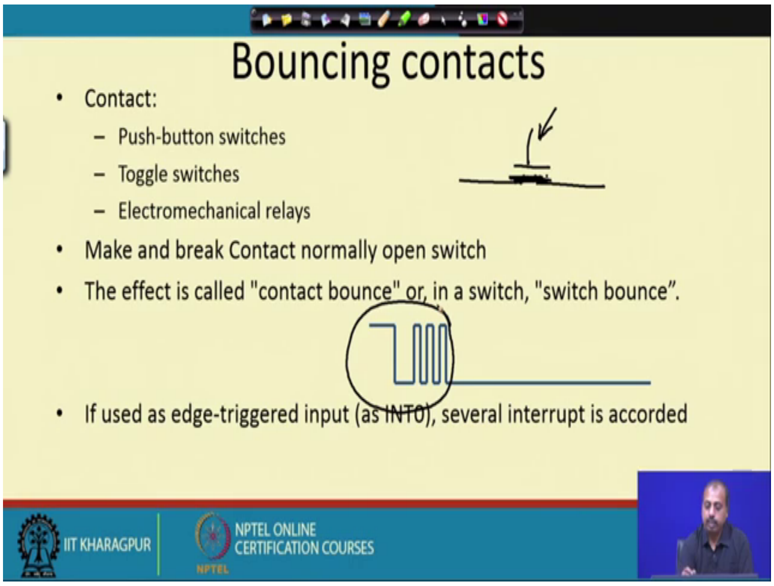


So, which is known as the dew DIP switch so, this DIP stands for dual inline package so dual, dual inline package or dual inline package situation. So, here we have got, so there are switches. So, switches are you have got this these poles are there. So, if these poles are pr if these switches are pressed. So, it will establish a connection like this ok. So, here we have connected so every point has got this type of poles now. So, this side everything is grounded. So, and we have pulled all these lines through this pull up resistances to one. So, if you are connecting say A 1 8051 sport 1 to one such DIP switch then what will happen is that when none of them are closed so you will get all one here. If any of the switch is closed if any of the switch is closed. So, you will get a 0 at this point ok. So, that way we can have this DIP switch connection.

So, the instruction sequence should be like this first we have to move p 1 comma FFH because port one I will be using for input purpose and in 8051 particularly you know that whenever you want to if you want to read the values from a in an input port first you have to write a one there. So, this statement this statement will do that. So, this will write once to the port to port bits and then move a comma p 1. So, this comma is not required.

So, move a comma p 1. So, it will read the value of the port p 1 into register a. So, this way we can have this simple input device connected.

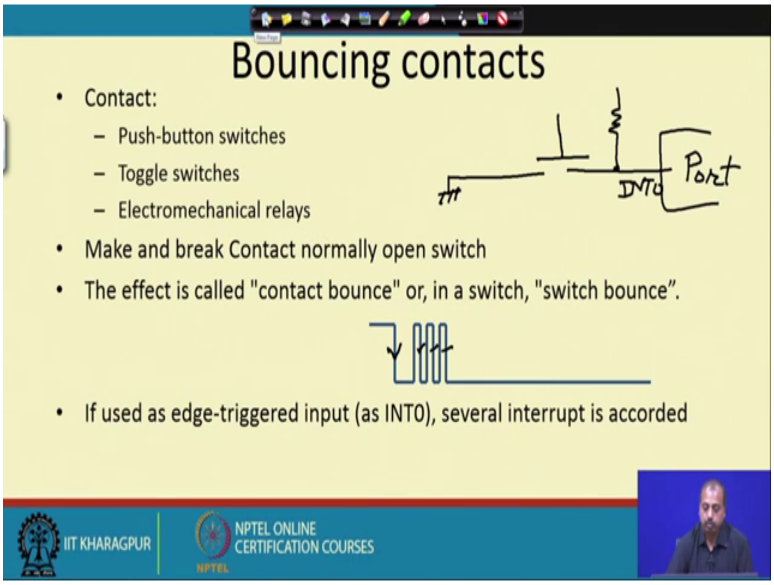
(Refer Slide Time: 11:16)



Next we have some. So, the problems with this type of switches are like this so this contact that we have. So, we have got pushbutton switch, toggle switch and electromechanical relays. So, these are the different types of switches that we have and these switches they make and break contact. So, so that way it will normally works ok. So, what happens is that if you have got say a connection like this say this is say one line and this is another line and we have got a switch here.

So, if I press this switch then these 2 the switch touches this point as a result the connection gets established. So, these 2 lines get sorted and we the connection gets established, but when you are doing this what happens is that you get this type of situation that is the. So, the initially the line is high. So, if you if you if you have got a pull up like this. So, if you have got a situation like this that this side is say grounded and this side we have got this switch and this side I have got a pull up and this is the input to the micro controller.

(Refer Slide Time: 12:15)



So, this is the port of the micro controller.

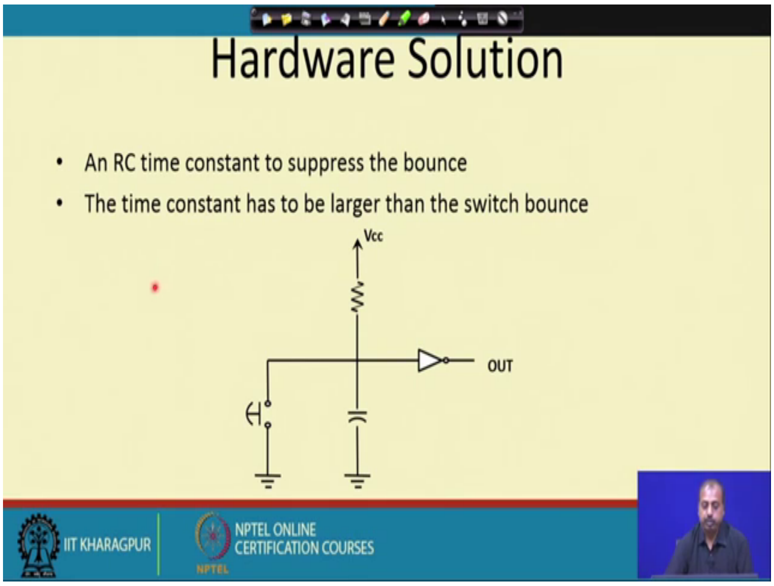
Now, you see that when you make this connection, about push this switch and make the connection then you have then this connection will be established, but initially the value of the line was high so it will become low but since it is a mechanical process due to vibrations, so this contact will not be farm. So, it will jump a few times before it get it it gets settled to the stable value, so that way. So, that when we are pressing a point here pressing a switch so by human being. So, this due to this mechanical make and break of contacts, so if you read these ports port bit and since the microcontroller will be operating at a very high frequency. So, compared to the speed at which we are pressing the switches and all.

So, you will get a large number of ones and zeroes. So, as if this is one place they say this is it will be taken as this one as one place then this one will be taken as another place, this one will be taken as another place, this one will be taken as another place. So, as if the switch has been pressed so many times which is not correct. So, this is called contact bounce or the, or in a switch it is called switch bounce. So, if we are using say h triggered interrupt for the so if we if we say that instead of connecting to port.

So, this is actually connected to the INT 0 pin of the processor. So, INT 0 pin of the processor which is h triggered. So, this high to low transition so this will be affecting the interrupt the to be to be get acknowledged by the processor. So, all these points will be taken as separate interrupts like this one, this one, this one. So, they will be taken as separate interrupts so which is misleading ok. So, so many times interrupt service routine will get invoked whereas, the keep rest is only ones. So, we want it to be invoked only once.

So, how to solve this is like this one possibility is to have a hardware solution.

(Refer Slide Time: 14:45)

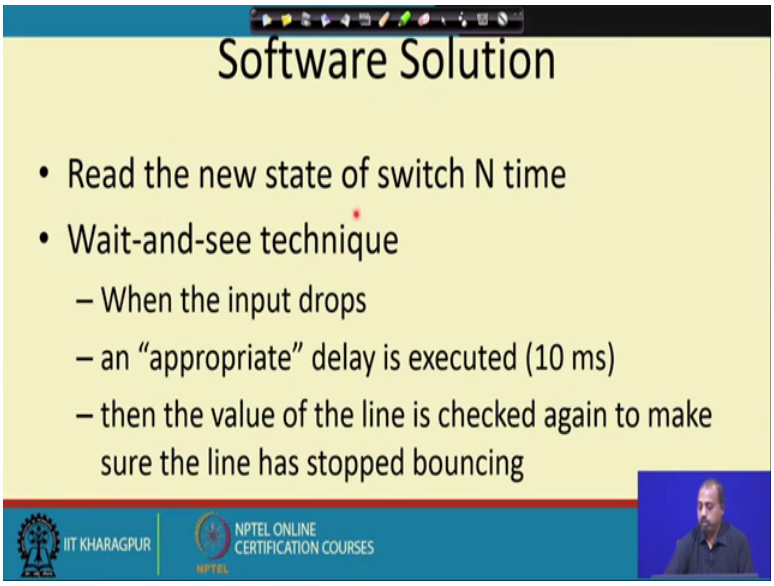


So, what is the hardware solution? So, this RC, we put a resistance and so up to this much was there ok. So, if we just forgive forget about this capacitor connection. So, up to this much we have seen in our previous diagram, now we put a capacitor here. So, that if this when the switch is not pressed so this capacitor will be charged and you will get a one here. But, when this is when this switch is pressed then this capacitor will get discharged and you will slowly get a so this capacitor will be discharged and then depend when the depending upon this RC values so this capacitor will be charged again slowly.

So, this time constant if the time constant is larger than the switch bounds then it will not be taken taking it as the another value ok. So, so switch bounds time so. So, if this is the switch bounds time. So, if that you are the capacitor for charge takes more than this time then it will not be a stable one like when the first time this has occurred. So, the capacitor will get discharged. So, you get a 0 here.

So, capacitor starts charging, but it will not reach to the high value till up to this time because capacitor will take some time to charge. So, capacitor will require depending upon the RC constant value. So, it will take some time to charge, but it will not charge that first because of the reason that this RC time constant is reasonably high compared to the contact bounds. So, this way we can avoid this bouncing thing the bouncing activity so we can avoid.

(Refer Slide Time: 16:28)



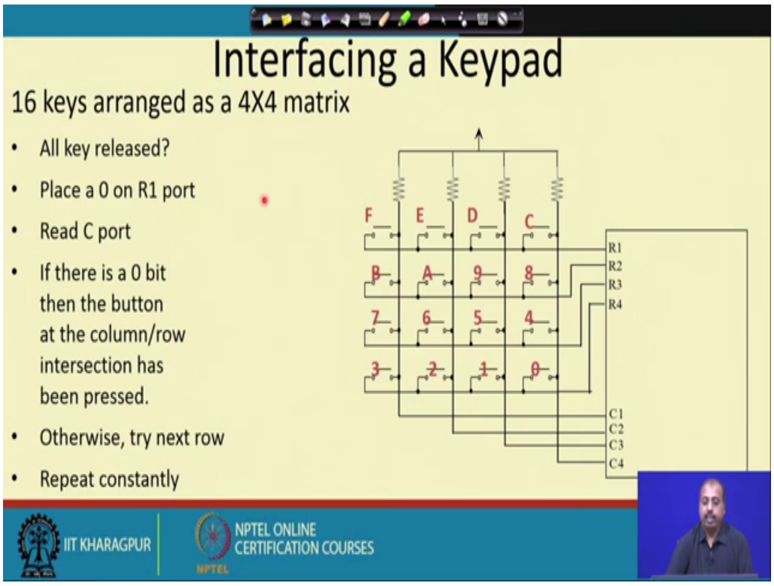
Another solution so we that we have is a software solution. So, what we do we read the new state of switch n time so we. So, maybe n equal to 2 at least ok. So, what we do we go once we detect that a key has been pressed. So, accordingly you have they found a 0 on the port beat. So, you wait for some time and then again read the value. So, if again if it is, if it is if, if it is not stable then possibly you will get a one at that time so, as a result you will you can take it at that this is actually a bouncing. So, if you get a 0; that means, you have crossed over that bouncing region so like here. So, if you first time I read the key. So, it is basically here then if I know that, what is the typical bouncing time.

So, typical bouncing time may be say 10 milliseconds. So, after 10 millisecond I again read the value of the switch ok. So, if now, if I now the value of the port bit so, here also if I find the value is 0; that means, the key has been placed and if it is just some flicker noise etcetera. So, that the key is not actually placed so it was just due to some vibration it came and all that then you it will be it will not come to 0 here. So, it will remain it will become high after sometime.

So, when you read after a delay so you will get a high there so you will not get this low. So, this way we can think about doing a software debouncing, so we read the new state of the switch n time. So, these I have talked about only 1 time. So, you can do it several times if you to get to be more confident. So, you can read the switch several times and then you can do this thing.

So, wait and these is the wait and see technique. So, when the input drops and appropriate delay is executed. So, when if the port bit has come to 0 so, you wait for a time of say 10 millisecond and then the value of the line is checked again to make sure that the line has stopped bouncing. So, this way we can take care of this software of this debouncing, of the bouncing of the keys and these are the debouncing solution.

(Refer Slide Time: 18:42)



So, with this idea so we can look into the interfacing of a keypad, ok. So, suppose we have got a 4 cross 4 matrix keyboard. So, this is a keyboard where we have got the keys marked as 0, 1, 2, 3 up to say f ok. So, this is this is also known as hexadecimal keyboard and things like that. So, what we are essentially doing is, so if we press the key. So, we need to read the value we need to get the key number into this. So, every key k, every key has got a corresponding location you see that this particular key. So, if I number if I name the rows. So, this is say in row 0, row 1, row 2, row 3 and this is column 0, column 1, column 2, column 3.

So, every key it has got a location in terms of those row index and column index. So, how should we, how should this how can I detect that whether any key has been pressed suffer first of all we have to check that whether all keys have been released. So, if all keys have been released then you see that if you if you if you read these columns. So, C 1, C 2, C 3, C 4 so this R 1, R 2, R 3, R 4 and C 1, C 2, C 3, C 4. So, they are connected to say some ports of the microcontroller. So, now, when you if you if you read this value of this port C 1, C 2, C 3, C 4 and if you find that all these bits are one ok.

So, and also we ensure that so this R 1, R 2 on the lines R 1, R 2, R 3, R 4 we put 0 ok. So, all these bits are 0 and in that situation if you read this C 1, C 2, C 3, C 4 then if you get all 1; that means, no wire is the key place. So, if any key is pressed since say suppose this 9 key has been pressed the senses are 2 bit is 0. So, if 9 key is pressed. So, this value of this point will become 0 as a result on the on the beat C 3 you will get a 0. So, this way we can take care of this rows and columns. So, these are and we can identify whether all keys have been really. So, have to check whether all keys are released. So, we put all 0 here and then read this C 1, C 2, C 3, C 4 if I get all the bits to be equal to 1; that means, no key has been pressed.

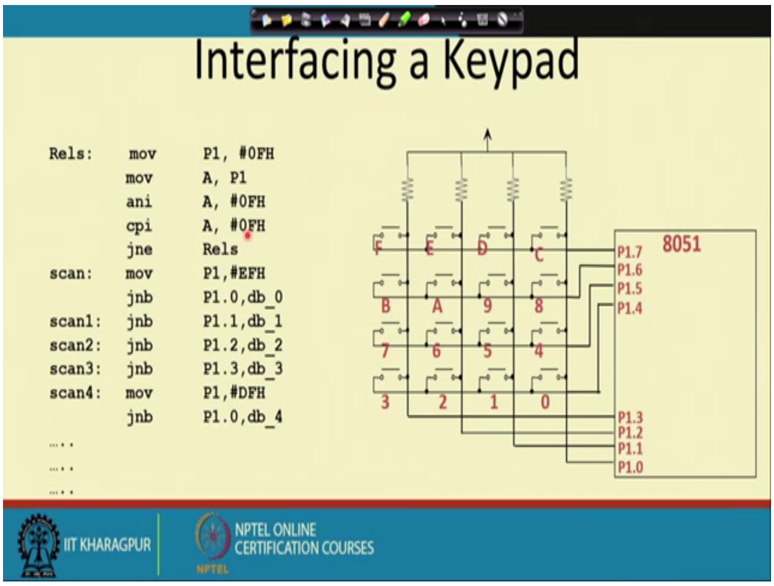
After that we place a 0 on the R 1 port. So, on now we have to now say after some time we have to check that whether some key has been pressed, if any key has been pressed. So, first we first we check the keys on the row 1 the R, R equal to 1. So, R 1 we are checking. So, here you see that we will put a 0 here and this R 2, R 3, R 4 we put once. So, this is R 2, R 3, R 4. So, these are 1. So, even if these keys are pressed any of these keys are pressed we are not going to sense it now ok. So, we will go those at C 2, C 3, C 4.

So, I will be get as. So, that if that effect will not come, but since this is 0 if any of these keys C D E or F has been pressed. So, the corresponding bit will become 0. So, if the see if the key C has been pressed then you will get this bit C 4 to be equal to 0, if d has been pressed you will get the key bit C 3 equal to 0. So, that way which he is pressed. So, you will get the corresponding thing a 0. So, I have made R one equal to 0 and suppose I find that C 1 is also equal to 0 after reading C 1. So, what will happen so you take the intersection of R 1 and C 1 that is this one. So, it will mean that the key f has been placed.

So, after putting this 0 on the R one port we read the seaport and if there is a 0 bit then the button at the column low intersection has been pressed. So, that is the intersection of row and column to that button has been pressed, if not if I find that all of all the bits are one only then within our one nothing has been pressed then we try to check the second row ok. So, now, we put R 2 equal to 0 and R 3, R 1, R 3, R 4 equal to 1 so putting R 2 equal to 0.

So, now in the same logic so I can check these keys 8, 9 A and B and they will be if any of those keys have been any has been pressed then accordingly this the bit C 1, C 2, C 3 or C 4 will become equal to 0 and we can do we can determine that the key has been placed. So, this way we can go on trying out different keys and we can go on checking the unique key is pressed so we can check it. So, this process is repeated. So, if this placing of 0 at R 1 port after that we will place a 0 at R 2 port, then place a 0 at R 3 port so like that we can continue.

(Refer Slide Time: 23:43)



So, this is the this may be a kind of possible connection like in 8051. So, we use a port p 1 for the connection the rows are connected to bit number 4, 5, 6 and 7 and the column bit that the C are connected to bit number 0, 1, 2 and 3. So, first I have to see whether all keys have been so all keys are released or not. So, for that you see that this part this part these 4 bits of the port. So, they will be acting as output port whereas, this bit. So, they will be acting as input port so, to ensure that we write p 1 comma hash 0 fh. So, this is the configuration of p 1 port. So, after doing this these bits will be acting as input and these bits will be acting as output.

So, that is then we are reading the value of p 1 into a. So, so we are getting the value there then after that. So, we have put in 0 here so 0 values are available and this is there. So, we are after that if I if we read the value of p 1 into a. So, as a result I will get this bits 0, 1, 2, 3 set properly according to the placing of the key. So, I end immediate with 0 FH. So, I am not bothered about whether the, whatever be the values I have got here. So, I am bothered about the values that we got here. So, if I do an end immediate with 0 F then this part will be coming and the upper part will be neglected, then we compared with 0 FH, so compared with this with this 0 FH.

So, it will be comparing with that and if it is not equal if it is if it is not equal to 0 then it will be if it the value that I have read in the a register is not equal to say 0 FH then it will be jumping back to again the release. So, again it will be putting 0 here and f here and then it will be reading the value of port p 1 into the a register. So, this way this first part so this will be doing the check that all keys are released from the operation the rest of the program. So, they will be actually selecting individual ports this bit 4, 5, 6 and 7 put a 0 there and rest of the rest of the things as 1 and then it will be checking whether the cursor in any key has been pressed in the in the corresponding columns of that particular room.