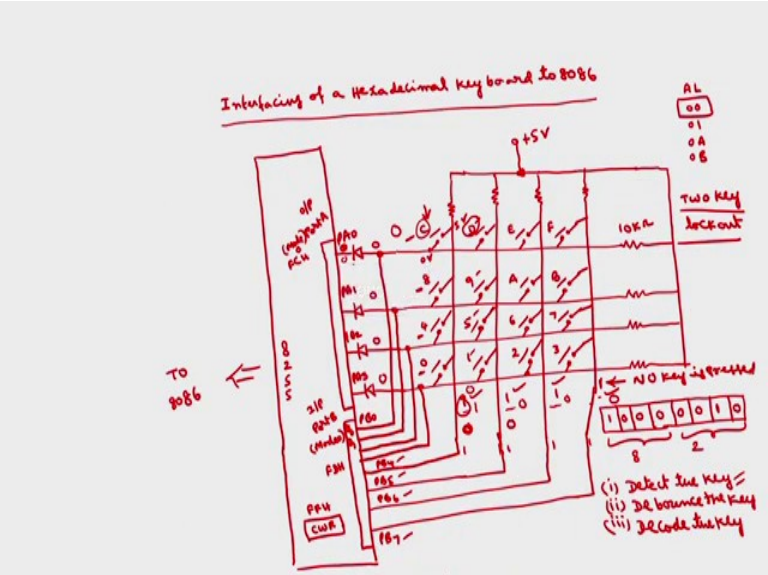
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture – 19
Keyboard Interface

In the last class we have discussed about the Interfacing of a simple LED: Light Emitting Diode to 8086 and then a simple switch ok. So, I have given the example of one input device such as switch and output device LED. Today we will discuss about how to Interface Keyboard which is one of the major input block for any microcomputer ok.

(Refer Slide Time: 01:01)



How to I mean interface a keyboard to the 8086; interfacing of hexa decimal keyboard, I am discussing hexa decimal keyboard. As the hexa decimal keyboard implies 16 keys will be there. So, we have a different types of the keys normally we will use the mechanical switches push button type of thing ok.

So, here we have to connect the 16 keys in 4 rows and 4 columns. So, we have to use 4 rows and 4 columns. So, these are 4 rows and we have 4 columns these 4 columns will be connected to power supply of plus 5 volts. Then these switches will be arranged here at the cross sections, this is 1 switch you press this push button this switch will be closed. See now its 0 switch, 1, 2, 3, we have 4, 5, 6 and 7; 4, 5, 6, 7, 8, 9 A because

hexadecimal 10 will be A B, this will be C D E F, this has to be connected to the ports of 8255 ok.

So, I am taking 8255 here, this is 8255, this will be connected to 8086 this side I am not showing these connections which we have already discussed in the earlier classes. So, to one of the ports of this you have to connect through diodes. This you have to connect to one port, this port has to be programmed as output port. So, you can use any port say port A, then another port you have to use for inputting this you have to use 8 pins of this one.

So, these 4 again we have to take the status of this one also we are going to take here as 4 inputs you have 4 rows and 4 columns. The entire information will be taken into this port, this can be any port I will take this as port B this is PB 0, PB 1. So on up to PB 7 this is PB 7, this is PB 0, PB 1 and so on this I will take as PA naught, PA 1, PA 2 and PA 3.

So, how to program this port A as output port and port B as input port? Let us take both in mode 0, this also in mode 0. So, let us take the address of this one as which we have derived in the last class as this is FCH; FCH is the port of address of this one, FDH is port B address, FFH is the control word register. Port C is FEH which we are not using here ok.

This also has to be connected to this common. Then what will be this control word to program this port A as output port in mode 0 and port B as input port in mode 0, what will be the control word? So, we have a 8 bit status of the control word for IO or mode selection the first bit is 1, then this last one is port C lower. So, which we are not using I am just taking as output port, this is port B; port B how to use as input port, so this should be 1.

And this is mode selection for port B, I want to operate in mode 0, this is port C upper. So, I am not using I am taking as output port. This is port A you have to program as output port, so 0. These two bits are mode selection of port A. So, I am operating in mode 0, so both are 0s. So, what is the hexadecimal equivalent of this one? 8 2, 82 H. So, you have to take first 82 H into control word register ok.

Now, here the basic process to interface a hexadecimal keyboard to the microprocessor and then we have to read the corresponding hexadecimal value into say some AL

register, if I want to press 0 I am taking this such that AL will be having key value ok. So, if I press 0 00 has to be taken here, if I press 1 01, if I press A 0 A, if I press B 0 B and so on.

So, whatever the key that we are going to press that hexadecimal value, so, I want to take into accumulator AL A. So, here we have some hardware as well as I have to write the program also ok. So, before going for this program this basic steps in interfacing a keyboard to the 8086 is we have three steps; one is called first we have to detect the key. Then second one is debounce the key, third one is decode the key.

So, I will explain these operations with the help of flowchart, then I will convert the flowchart into assembly language program ok. So, the basic process for detecting the key will be your output a 0 on any of these ports ok. If I output a 0 here let us first assume that if all the keys are open no key is pressed ok. So, what will be the value here? Because these 5 volts is there. So, these 5 volts through this resistor; resistor will be normally 10 kilo ohms through this resistor 10 kilo ohm this will be connected here.

So, here we will get 1 1 1 1 whenever you have 1 1 1 1 here means 1 1 1 1 here; whenever you have 1 1 1 1 here means no key is pressed because these are directly connected ok. Here there is no connection here at the intersect let us assume that here there are no connections; connections will be represented by a thick dot here there is no thick dot, so there is no connection at this junctions ok.

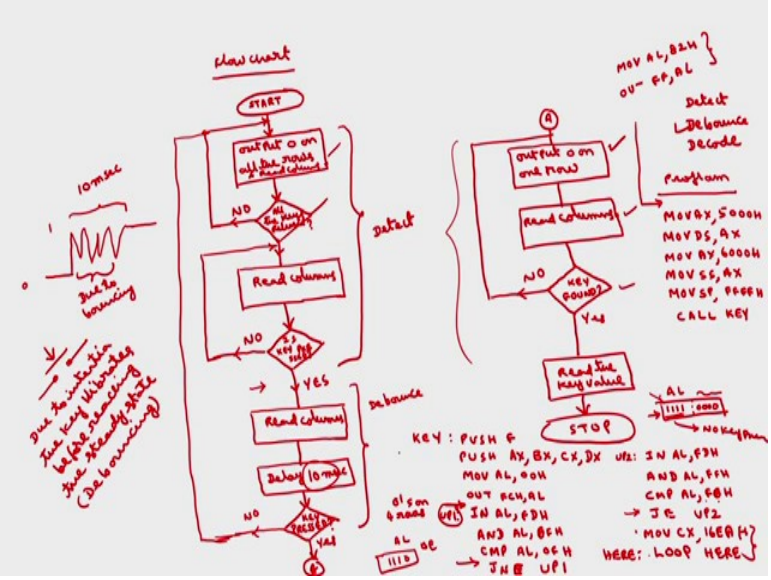
So, if no key is pressed these 4 bits will be 1 1 1 1 because these 5 volts through after some drop across this resistor that same 5 volts a value will comes at this I mean columns there by we will get 1 1 1 1 here ok. If any key is pressed what happens? If any key is pressed suppose let us assume that key C is pressed.

So, what happens? This PA naught value will be connected here, if PA naught is 0. So, this is 0 volts this is 5 volts; so this is 0 volts this is 5 volts. So, because of the short circuit this becomes 0 volts and a 0 will be appeared here ok. If all are 1s means no key is pressed, if any one column is having 0 means there is a one key pressed in that particular column.

So, suppose if I get 0 here the key pressed can be either 0 or 4 or 8 or C, the key pressed will be either of these 4. If I get 0 here the key pressed can be either 1, 5, 9 or D. So, like

that say 0 on any column represent there is a key pressed in the along that particular column ok. So, with these basics I will go to the flowchart and then program.

(Refer Slide Time: 13:30)



So, the flowchart will be for this as I have told there are three steps; first one is detect flowchart, first step is detect, then debounce, then decode ok. So, in order to detect the first thing is we will start and this is start of the flowchart. Normally, I will start with this start, output 0 on all the rows. So, in the previous slide. So, first I am going to send 0 on all the rows. So, these diodes will be I mean short circuited because of this 0, this cathode is connected to 0 anode is connected to some positive voltage ok.

So, these resistors are so selected that if I send 0 here. So, this voltage is greater than cut in voltage of this diode thereby these diodes will be short circuited. So, before going to press any of this key. So, what you have to do is first you have to check that all the keys has been released ok. So, after all the keys has been released, then only you have to press the key otherwise we will get a problem of two key lockout there is one problem with switches is called two key lockout is whenever a particular key is struck already.

If I press the second key then two keys will be pressed simultaneously that particular problem is called two key lockout. To avoid this first I will ensure that no key is pressed all the keys are in released position if I ensure that then they are going to press the key ok. So, to check that, so what we are going to do is we have sending first 0s on this and

we will check this status at the columns ok. So, if no key is pressed what will be the column status?

If I send 0 if no key is pressed here you should get all 1s means no key is pressed ok. So, it is the first step. Output 0 on all rows, now you have to check whether all the keys has been released or not; all the keys released or not. If no until the key is released we cannot press the another key, so it will go here. If it is yes, then I will check for a key that is to be pressed.

Then I will read columns and read columns also you can write here; read columns key is pressed is key is pressed. If yes we have to go for debouncing this is for this entire thing is for the first operation which is detect. If yes it will go for debounce the second operation ok, if no it will wait until the key is pressed fine. Then debounce, what is meant by debouncing?

So, this is one of the important phenomena in keyboards; keyboards normally will be having nature of bouncing ok. So, normally because what happens is if we have this push button type of push button type of the switch you have two positions and if I press here. So, before making the final connection it will vibrates because of this mechanical movements of these switches inertia and all.

Due to inertia the key vibrates before reaching the steady state, this phenomena is called debouncing of the key that is if I made from low to high this is low to high transition. So, instead of going into high, so what happens is it will come it will vibrates after some time it will settle at high this is logic 0, logic 1 this is due to bouncing. The reason for bouncing is as I have told inertia because of the inertia and keys there will be de bouncing. So, it is why even if a key is pressed.

So, I have to wait for some time until it settle downs this de bouncing time we have to set. So, after this also key is pressed means its original key otherwise this may be due to noise also ok. So, to avoid this noise effect. So, you have to wait for some time that will be normally 10 milliseconds ok. So, for that I have to read columns again, you obtained a delay of set a delay of 10 milliseconds. Still key is pressed I have to check still key is pressed, if no this will be due to noise, if still key is pressed this original key ok.

If no because of the noise. So, you have to go to this, if yes then you have to go for decoding I will continue this here to connect the big flowcharts. So, we can use some letter this is ending of this part. So, the second operation debounce is over and the third operation is decode ok.

So, I am starting here again with A. So, this is A; this is A means this is linking between these two, this is the notation of the flowcharts ok. So, a then again what I will do is I will read the columns before that to decode the column output 0 on one row what I will do is I will output 0 on one row. Suppose if I output a 0 on this row and if I check 0 on this row, if I check these columns 4 columns, if I get 0 here what does it mean?

So, in the first row and first column 0 if remaining three are 1s, so key C is pressed ok. So, if I get a 1 here these three 0 0 0, then key D is pressed is it clear? So, you send a 0 on any particular row, we check all the columns at any column if I get 0 means in that column and that row the intersection is the whatever the key is present in the intersection that key has been pressed this is the logic to decode the keys. Then read the columns is key is found, if no we will go for outputting the row on another row.

Initially we will keep this 0 on for the first row then second row, third row, fourth row we will check for all the rows. If yes then we have to read the key value, then STOP this is the flowchart to interface a hexadecimal keyboard to the 8086 microprocessor. Now, if I know this flowchart I can map this flowchart on to the assembly language program ok.

So here I will right main program as well as sub routine ok. So, for that the program will be initially setting is because we are going to use the sub routine you have to I mean initialize stack segment, data segment is any how necessary. So, you take some contents into A X 5000 MOV DS comma AX, we are setting data segment register to 5000 means the starting of the data segment will be 50,000 H. And MOV AX comma some 6000, MOV AX contents onto stack segment means the starting address of the stack segment is 60,000H and whenever a stack segment is used stack pointer has to be initiated.

The stack pointer I am initiating with say some FFFFH. So, that the stack top will be 6 FFFFH 60,000 plus FFFFH ok, then the first step is. So, these are this I mean in the main program that I will just call the subroutine call key. So, it key sub routine I will write a part of this program here because I have to explain this flowchart.

So, the KEY sub routine. So, in this the first instructions will be you have to PUSH flag register whenever you want to use the sub routine its always good practice to PUSH all the registers and pop all the register at the end, PUSH AX these are 4 instructions I am writing in the single instruction BX, CX, DX ok. This is the order while popping you have taken the reverse order ok.

Now, what is the first step? You have to output 0 on all the rows and you have to check whether all the keys has been released or not, this is to avoid two key lockout problem ok. So, initially before going to I mean press a any key you have to make sure that all the keys has been released previously. So, that no two key lock out problem will occur ok.

So, you have to output 0 on rows; rows are connected to this rows are connected to port A whose address is FCH you have to send 0 0 0 0. So, I will take MOV MOV AL comma 00H OUT rows are connected to port A. So, port A address is FCH; FCH comma AL. So, with this instruction what happens all the port A pins will be having 0 output ok. So, in that 4 pins we have connected to the rows 4 rows.

So, 0 will be outputted on 4 rows. Then what is next step? You have to check any keys released or not. So, if all the keys are released what does it mean as I have told? So, all the 4 bits should be 1111. So, where we have connected these 4 bits? This 4 columns you have connected to PB 4, PB 5, PB 6 and PB 7 ok. So, I have to check these PB 4, PB 5, PB 6, PB 7. If all the 4 these bits are 1s means no key is pressed all the keys are released ok.

So, I have to check this last 4 bits of port B. How to check; how to check the 4 bits of port B? For that you have to write IN you have to take the data into AL, IN AL comma the port B data whose address is FDH. So, the very first instruction in this program is I have forgotten. So, you have to initialize the ports as I have computed this this control word register I have obtained as 82 H this I have to take into control word register.

So, we can take here anywhere we can write here also. So, the first two instructions are MOV AL comma 82H OUT FF comma AL. So, this will program port A as output port and port B as input port in mode 0 ok, but this is just one initialization only. Now, in AL comma FDH in AL FD comma H AL will comes the data of all the columns 4 columns ok. What I am doing now is I am ending that AL with FFH; ending AL with FFH if you have any the first 4 rows will be 0 0 0 because you have inputted 0 0 this will be 0.

If any one of this particular 4 column status is 1 ending operation will result non zero value ok. So, that is why I am comparing with compare AL comma 0 FH this can be either FF or 0 F, its up to you because the remaining 4 bits we are not using compare AL with 0 FH. So, after this what happens? In AL the 4 bits remaining 4 bits I am not caring if the last 4 bits of this 4 columns, if any one of this column is 0. So, for example, 1110 ok. So, this is E 0 E first will be a 8 bits will be 0 E. So, its not 0, jump on not equal.

So, when this jump on not equal is true what does it mean? Jump on not equal. If these two are not equal what does it mean? So, in the 4 columns we do not have all the 1s ok, one of this column is 0. If all the 4 columns are 1, so, the last 4 bits will be 1111. So, 0 F AL also will be having 0 F and we are comparing with 0F itself. So, result should be equal jump not k not equal has to be true. So, because if this is not true go to UP 1. So, you have to again input this data and you have to check. So, whenever this is in loop means all the keys has not been released. So, you have to wait until all the keys has been released.

So, whenever this program come out of this loop means all the keys has been released, if all the keys has released 4 column bits will be 1 1 1 1, we are comparing with 1 1 1 1 only. So, if they are equal this jump not equal will be true. So, it will come out of the loop. Then after that so, this first part is over these two parts are over. Now, again you have to read the columns. So, for that again in IN AL comma where you have to read the columns? Columns are connected to port B in that port B also the last 4 bits that is PB 4, PB 5, PB 6, PB 7 MSB. So, in port B whose address is FDH.

So, a reading columns then you have to check if any key is pressed, if any key is pressed we have send the 0s on the rows, if any key is pressed that particular column should be 0 ok, if all the column bits are 1 1 1 1 no key is pressed do the same thing MSBF. So, I am going to I mean end with this AL contents in that MSB bits, LSB bits you have send the 0000, MSBs if all are 1s means

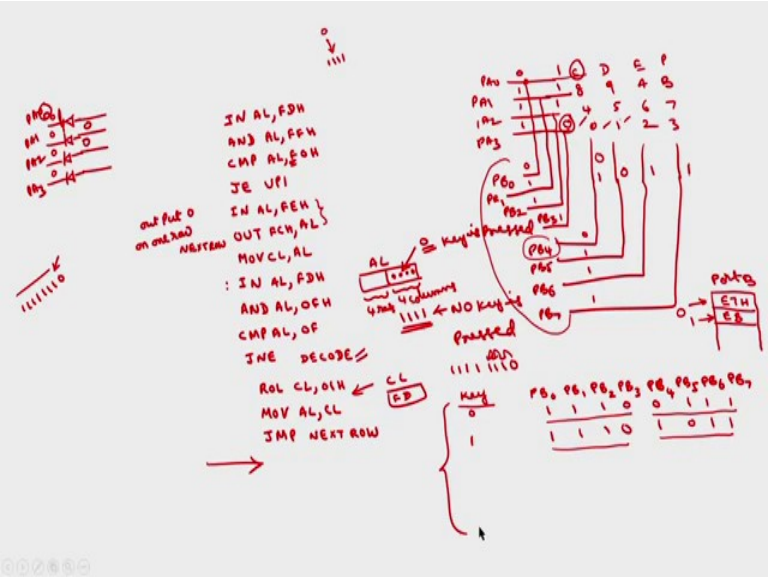
This represent no key pressed, if any one of these bit is 0 means there is 1 key pressed till any one of this bit is 0 we have to wait in a loop okay and AL comma FFH, so if the result is compare AL comma FFH here this ending this becomes 0. So, F 0 H; F 0 H. So, this is any how 0, if this 4 bits are 1s, then this will be F 0 .

If these two are same means no key is pressed. So, jump on equal jump on equal UP 2, if this is true if these two are equal means no key is pressed. So, you have to wait until the key is pressed for that I have to wait here in the in loop UP 2. So, if it is false jump equal is false means in this one of the bit will be 0.

Then only this jump equal is false, it will come out of this jump equal means one key is pressed. So, it will come out of this one, so this will be this point ok. Then we have to provide a delay after the delay we have to again check the read columns ok. So, for delay MOV CX comma 16 EA in the last class we have discussed about how to compute the delays.

So, with this count value if I loop here itself; loop here itself loop is here itself then this program will generate a delay of 10 milliseconds this program is basically these two instructions are basically to provide a delay of 10 milliseconds. Now, after this again I will check still the key is pressed, if still key is pressed means that is the original key. Otherwise its due to noise or debouncing of the key again I have to read the columns.

(Refer Slide Time: 37:44)



So, for that IN AL comma port B address is FDH AND FFH AL comma FFH compare AL comma F0H jump on equal again the same logic. So, equal means if AL is having the upper 4 bits of AL is having 1 1 1 1 means no key is pressed means that previous key pressed that we found before the delay was due to noise.

So, I have to wait until a valid key is pressed ok. So, I have to jump to again the starting of UP 1 this UP 1 because we found that the key pressed was due to noise or debouncing effect if this is I mean false means one of these bit is 0; that means, a valid key is pressed. So, I will come out of the loop, then I have to decode that particular key the third operation is you have to decode the key. So, this is coming out of the loop. So, this again to decode the key these are the steps; these are the steps you have to decode to find the correct key ok.

So, what is that next operation? We have to output 0 on one row, then you read the columns if key is not found then you send a 0 on the second row, third row, fourth row like that we have to check for all the 4 rows. So, you have to output here now you have to output 0 on one row, so I am going to on the output a 0 on the first row which is connected to PA 0.

The first row we have 4 rows, this is connected to PA 0 of course, the diodes are there here this is connected to port 1, PA 0 this is connected to PA 0, PA 0 PA 1, PA 2, PA 3 first I am sending 0 on the first PA 0. So, what is the instruction to send the 0 on PA 0? What are the instructions required? IN AL comma FEH ok. So, FEH means 1111, 1110. So, only this least significant bit is 0 remaining all bits are 1s this I am going to output onto port A; port A address is FCH AL. So, with these two instructions a 0 will be send on PA 0 ok.

Then I have to check any column is pressed if not I have to go for the next key to be pressed ok. So, we have this FE I have to use this 0 here now I have send 0 here next I have to send 0 here, next I have to send 0 here, next I have to send 0 here, but this AL I am using for some other purposes. So, I am keeping this value of AL in CL also. So, that even if I use this AL for some other purpose the original contents of FE will be present in CL ok. So, after sending a 0 on one row what do you have to do? You have to read the columns, if key is not found you have to repeat the same process read the columns.

What are the instructions to read the columns? Columns are connected to PB 4, PB 5, PB 6, PB 7. So, in simply basically to read the columns in AL comma port B address is FDH. Now, IN AL we will be having port B in this the last 4 bits will be this will be 4 columns; this will be 4 columns, this will be of course, 4 rows itself ok. So, in the 4 columns, so I will check if any column is receiving a 0 means in that particular column a

key is pressed ok. So, if any one of these 4 bits is 0 means key is pressed, if all these 4 bits are 1 1 1 1 no key is pressed ok.

Then AND AL comma this is of course, whatever this 0FH compare AL comma 0 F after ending also if I get 0F, what does it mean? These are 4 bits are ones only, so NO key is pressed. So, jump on equal if this is true what does this mean? So, this is NO key is pressed. So, you have to go for the next row jump on equal go to next row.

If they are not equal means key is pressed we have to decode. So, next row means where should be this next row? Otherwise I will write jump not equal, so that this will be easier jump on not equal. So, if these two are not equal means key is already pressed you have to go for DECODE otherwise you have to go for next key.

So, before going for the next key what I will do is because I have to send 0 here this contents FE was there in CL. So, I will take again that MOV CL comma rotate accumulator PA 0 becomes PA 1 means to the left, rotate left CL comma 0 1. So that in CL what will be present? This FV 1111, 1110 you are rotating this this will come here, this will come here, this will come here like that.

So, this will goes to here this becomes F0010 becomes E F sorry D FDH, I am taking this into AL again CL unconditional jump to next row here this equal means already key is found you have to just decode and if not you have to go for the next key check. So, where should be this NEXT ROW? NEXT ROW will be here.

Now, you have sent you have to output this value sorry here NEXT ROW you have to output and you can save that value FD because in the next round I have to shift this FD by one more bit. So, I am saving into CL again and then I am going for the this inning of this columns. So, this process will repeats. So, once this come out of this loop when does this will come out of the loop if any one of the column is found to be 0 ok.

So, then whatever this row that I have sent and the column these two you have to use for encoding or decoding the program this is I am calling as decode, but the correct word is encode ok. So, you have to encode that particular key. So, for that I can use the lookup table this is something like. So, I have 4 rows and 4 columns ok.

So, along these row we have 0 1 2 3 4 5 6 7 8 9 A B C D E F any how this we are reading in port A, in port A we are taking this as in port A we are taking this PA 0, PA 1 PA 2, PA 3 and this you are taking in PB in addition to this we are taking this also ok, we are taking this also in PB also this you are taking as PB 0, PB 1, PB 2, PB 3. This is PB 4 PB 5 PB 6 PB 7; PB 0 PB 1 PB 2 PB 3 PB 4 PB 5 PB 6 PB 7. Now, this lookup table will be suppose if I send on this port B itself if I send 0 1 1 1 and if I receive same 0 1 1 is available here also 0 1 1 1 if I receive 0 1 1 1 here itself what does it mean? In 0th row 0th column key is pressed which is C ok.

So, like that if I want to press key 0. So, what will be these values of PB 0 PB 1 PB 2 PB 3 PB 4 PB 5 PB 6 PB 7? Ok. So, 0 key pressed means 0 is here, so this row should be 0 and this column should be 0. So, what is the corresponding code? This will be 0 0 0 0 because this row is PB 3 PA 3 as well as PB 3 and this row is PB 4 if PB 4 is also 0 0 1 1 sorry this this 3 should be 1 1 1 you have to send 0 in any one of the rows, 1 1 1 0 if I send 1 1 1 0 means PB 3 is 0 and PB 4 is also 0, the remaining 3 are 1s.

So, this code is corresponding to key that is pressed as 0 in port B this we are going to take this 8 bit of the data into this port B. In port B if I receive these code as E7H; in port B if I have E7H means a key 0 is pressed this is correspond to key pressed 0. Similarly, what about for key 1? Key 1 means 1 is here. So, this should be these three should be 1 1 0 only this will be 1 0 1 1. So, what will be this one? This PB 0 to PB 3 remains same because 1 is also in the same row as 0.

So, 1 1 1 0 column will be this 0 will be available here 1 0 1 1. So, now, what will be the code? E EB, this is the code corresponding to 1 ok. Similarly you can derive for 2, 3 so on up to F then that value I want to take into accumulator ok. So, this only this thing that is remaining is how to I mean encode, how to read the key value into the AL register that part we will discuss in the next class.

Thank you.