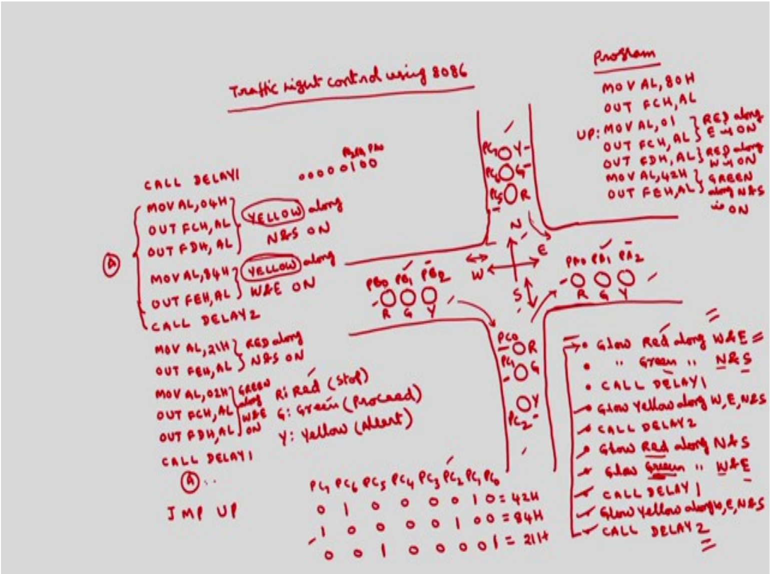
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture – 23
Traffic light control, A/D converter

So, in the last class we are discussing about this Traffic Light Control using 8086. So, I have given the arrangement.

(Refer Slide Time: 00:35)



So, now we have to write the program. So, let the program here. So, basically here I am going to use all these ports as output ports. Here you are using port A, here you are using port B, here you are using port C, because you have to connect LED to all these ports, you have to program all the ports as output ports for that the instructions are you have take 80H into AL register. Then you have to OUT on to port A, I will assumed that the port A address is FCH. So, these two instructions will program all the ports as output ports.

Now, coming for this first operation which is glow red along W and E. So, red along W and E means; red along W E is connected to PB 0, red along W E is connected to PA 0. So, you have to make PB 0 and PA 0 as 1. So, that red will be glow along West and East directions. So, far that I am taking MOV AL with 01H I will output onto port A as well as port B OUT port A address is FCH , AL. So, with these two instructions will glow RED along East direction it is will glow RED along East.

Similarly, RED along West direction you have to output onto port B, because red is connected to PB 0. So, you have to OUT because the content of AL remains same FDH, FDH is the address of port B. So, this instruction will glow RED along West is ON. Now what you have to do? You have to glow green along North and South so, that the passenger can pass from South to North as well as North to South.

So, green is connected to PC6, green is connected to PC1 here only port C. So, what I am going to do is, here port C, PC1 and PC0 should be 1, PC1 and PC6 should be 1. So, PC7, PC6, PC5, PC4, PC3, PC2, PC1, PC0 among this green is connected to PC1 and PC6, PC1 and PC6 should be 1s, remaining all will be 0s, because some of these pins we have used for some other LEDs along some other directions. So, that I do not want to glow those LEDs.

So, what is the hexadecimal equivalent? 42H, so, if I write MOV AL , 42H if you output onto a port C port C address is FEH , AL. So, these instructions will glow GREEN along both North and South. Now, you have to call the delay. So, I will write the remaining program here.

CALL DELAY this delay depends upon the traffic conditions of the particular junction, then what you have to do is you have to glow yellow along all the directions. So, that it will be alert for the vehicles crossing from North to South and South to North they will assume that. So, red is going to be glow along these two directions and the people along west and east they will alert and it is a signal for them that it is going to be green on in short duration.

So, I have to glow yellow along all the four directions. So, yellow is connected to PB2, PA2, PC7 and PC2. So, PC7, PC2 if you want PC2 should be 1, PC7 should be 1. So, remaining all will be 0s. So, the hexadecimal equivalent of this one will be 84H, if I output 84H on port C the yellow along North and South directions will be ON. And in order to glow yellow along West and East, PB2 and PA2 has to be ON.

So, PA2 PB2, PB2 means this is PC if I use this as PB, then this bit should be 1 so; that means, 04H. So, MOV AL , 04H, 04H means 0000 0100. So, I am going to output onto port A as well as port B, port A is FCH ,AL, port B is FDH , AL.

So, in this if I assume port A PA0, PA1, PA2; PA2 will be ON, if I ON output onto the port A and PB2 will be ON if I output onto the port B. So, these three instructions will glow the YELLOW along North and South directions is ON and to ON this YELLOW along East and West directions is 84H has to be sent to port C. So, it take this AL with 84H OUT along port C which address is FEH. So, this will ON YELLOW along West and East will be ON.

Then you have to CALL another DELAY. I will call this as CALL DELAY the previous one as CALL DELAY 1. So, CALL DELAY 2, DELAY 2 is normally less than delay 1, then you have to reverse the operation. So, now, this step is over this step is over. Then you have to glow red along North and South, green along W and E, then again you have to CALL DELAY you have to glow yellow along all the 4 directions CALL DELAY and the same classes has to be repeated.

To glow red along North and South, North and South red is connected to PC5 and PC0. So, PC0 should be 1, PC5 should be 1, remaining all will be 0s this should be 21H. So, if I output 21H along the port C the red along North and South will be ON, MOV AL ,21H, OUT FEH, AL. So, with these two instructions red along North and South is ON.

Now, you have to ON the green along West and East. So, green is connected to PB1 and PA1. So, for PB1 or PA1 the content that you have to take into AL will be last, but one bit should be 1 means it should be 02H. So, it take 02H. MOV AL , 02H OUT onto port A as well as port B, FCH , AL OUT onto FDH , AL. So, these 3 instructions will ON green along West and East. So, this is over; this is over, then you have to again call the DELAY, this DELAY will be same as the previous delay which is CALL DELAY 1.

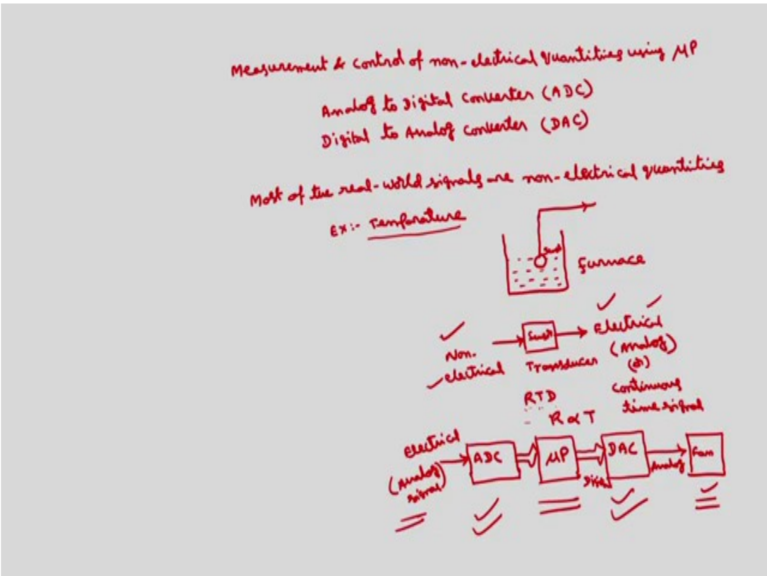
So, this step is over now you have to glow yellow along all the 4 directions. So, for that the instructions are these are the five instructions you have to use to glow yellow along all the 4 directions. So, if I call this instructions as some A, I have to repeat the A here also even including the CALL DELAY 2. So, the set of six instructions you have to repeat here, this A is set of six instructions that A operating means here you have to replace six instructions.

Then this will be over, this is over, then you have to repeat from the starting. So, an conditional JUMP to UP, where should be this UP? UP should be again start with this point. So, this will continuously perform the task that I have given in the this algorithm. This is how we can control the traffic lights along the four ways of a junction.

Here I am taking only a simple problem where the passage is allowed from West to East or East to West, North to South or South to North. So, West to South and South to East these conditions I have not considered. So, in a similar manner you can extended to so any I mean possible combinations that depends upon the traffic conditions and the I mean junction. So, this is about the traffic light control using 8086.

Next is, if you want to connect any physical quantity, if you want to measure any physical quantity such as pressure, temperature, any other physical quantity, because this microprocessor will be very much useful in industrial applications. Nowadays, also there are lots of I mean industries which uses the microprocessors as a controllers. So, in order to I mean measure non electrical quantities for the measurement and control of non-electrical quantities using microprocessor any microprocessor.

(Refer Slide Time: 13:31)



So, you have to use two devices two ICS called Analog to Digital Converter and Digital to Analog Converter ADC and Digital to Analog Converter DAC. So, this ADC and DAC play an important role in industrial control systems. So, why you have to use this analog and digital converters analog to digital and digital to analog convertors?. So, what is the reason is, most of the physical quantities are non - electrical quantities most of the real word signals are non electrical quantities for example, if I take temperature.

So, I want to control the temperature of a furnace, this is the furnace I want to control the temperature inside this, I want to operate over a particular range of temperatures. This type of operation is important in many industrial applications. So, temperature is a non electrical quantity. So, what are the various operation that you have to do is, we can convert somehow this temperature to so, you can place a sensor here temperature sensor which also can be called as a transducer.

A transducer is a device which converts non electrical quantity to electrical quantity. So, you have the output of this one will be electrical quantity, input for the transducer will be our sensor, this can be a sensor or transducer, input will be non - electrical quantity, output will be electrical quantity.

So, this if I take this examples as RDT is one of the temperature sensor which is called Resistance Dependent Temperature. So, this actually the principle of this one is the resistance varies with the temperature, resistance which proportional to temperature. If I measure this change in resistance which is measure of temperature. How to measure change in resistance? We can use some bridge after that you will get some electrical quantity, still these electrical quantities analog another thing is here

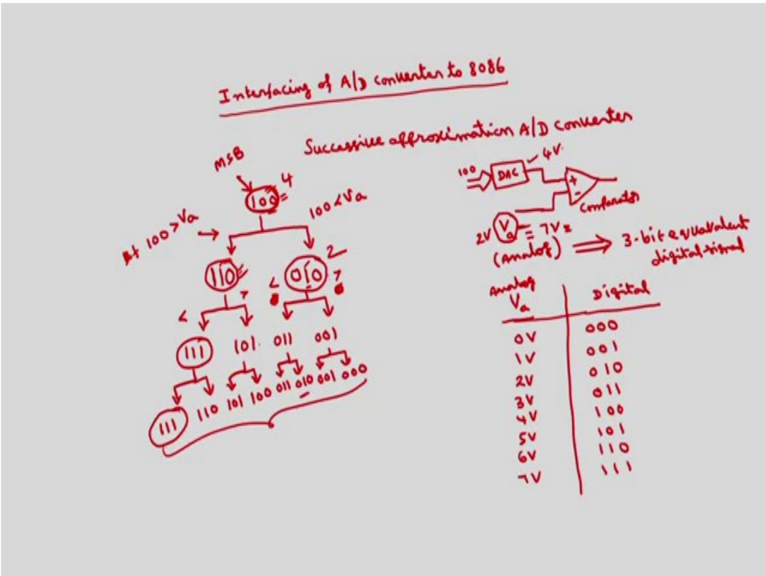
So, most of these things are non-electrical quantities we can convert the non electrical into electrical using transducer again this electrical quantities analog in nature analog are called as continuous time signals. If you want to process this continuous time signal or analog signals using microprocessor, microprocessor is a digital device. Now, you have converted non electrical into electrical by using transducer or sensor, I cannot apply this electrical signal which is analog in nature, I cannot connect this directly to the microprocessor. So, for that what is required is A to D converter analog to digital converter. Now, we will get digital data this will be bits this I can connect to the microprocessor.

This microprocessor can be any microprocessor. So, in this course we are discussing about 8086, but this can be any microprocessor. Similarly, after performing the various operations in this microprocessor finally, I have to I mean control a non electrical quantity. So, I can send this digital signal from here if I want to control non electrical quantity. So, this is something like if I want to I mean control a fan or heater, if I have a fan or heater here I want to drive the fan. So, this fan cannot be driven by using digital signals. So, what you have to do is, you have to convert this digital quantity into analog by using DAC.

So, this is analog now, this is digital this is analog which can be used to control the various analog appliances. That is why in between the processing is in digital domain, but the input side and the final output will be analog signals. So, we need to use A to D converter and D to A convertor. Now, the question is how to connect this A to D converter to the microprocessor and D to A convertor to the microprocessor? Ok.

First I will discuss about the interfacing of this A to D and D to A convertors to the microprocessor after that using these interfacing I will consider some examples where the physical quantities I will take the same examples of say temperature control and measurement using microprocessor. So, first I will discuss about the interfacing of ADC to the 8086.

(Refer Slide Time: 20:18)



You can also call it as ADC or A to D converter. There are basically many types of A to D converters. There are basically four types of A to D converter, the most commonly used A to D converter in microprocessors is successive approximation type A to D converter. Because the speed of successive approximation A to D converter is more, microprocessor is a fast device. So, we have to use a faster converters. So, among the all the converters successive approximation type A to D converter is faster so, most of the microprocessors use just successive approximation type A to D converter.

So, what is the principle of this successive approximation type A to D converter? Basically, we will be having analog signal which has to be converted into digital ok. So, I will take the analog signal this is analog signal Va, which has to be converted into digital. So, what I will do is here, I will start with if suppose if I want a 3 bit equivalent of this I am going to use 3 bit A to D converter.

So, in general in most of the practical applications we will use 8 bit or 12 bit for the explanation purpose I am assuming a 3 bit A to D converter. So, I want 3 bit this is analog signal, what is the 3 bit equivalent of digital signal. I will assume here for the sake of simplicity only Va can assume only the integer values from 0 to 7 and the corresponding digital value that I am going to expect is this is analog.

I will assume that 0 volts correspondingly 3 bit means 0 0 0, 1 volt 0 0 1, 2 volts 0 1 0, 3 volts 0 1 1 this is basically binary equivalent of this decimal value, 4 volts 1 0 0, 5 volts 1 0 1, 6 volts 1 1 0, 7 volts 1 1 1. So, initially what I am going to use here I am going to use D to A converter also inside the A to D converter there will be a D to A converter ok.

So, I will take a value to D to A converter this is DAC I will compare this with the analog value using a comparator. This digital value this is 3 bit equivalent digital value this will be given to the comparator, depends upon whether this analog equivalent of this digital signal is greater than or less than this Va you will accordingly set or reset the bits. So, the algorithm will be something like this, first I will start with whatever the analog voltage I will start with the 100 MSB bit I will make as one.

On this MSB 1 0 0 will be converted into analog signal which is 4 volts using DAC now we will compare with Va. So, if this 4 volts is greater than Va, what I will do is there are two conditions arises here after converting this 1 0 0 in to analog voltage that is 4 volts. So, this can be greater than or less than Va. So, if this is greater than Va I will keep that first MSB bit as it is, because this is still greater means the value is more than this 1 0 0.

So, I will keep this MSB bit as it is, then I will make a second MSB also 1, third MSB will be 0, this is the case if analog equivalent of 1 0 0 is greater than Va. If analog equivalent of 1 0 0 is less than Va; that means, this 1 0 0 is a larger value. So, I will reset the first MSB bit to 0 and I will make next MSB bit to 1 then 0.

Now, coming over here I will compare again the analog equivalent of this value which is 6. So, if this value is greater than the analog value which is to be converted into digital I will keep these first two bits 1 1 I will make third bit as 1 if this value is greater than if this is less than then I will reset the second value and I will make third value as 1, because this is already less than. So, this 1 1 0 is less than means it is less than 6. So, I will make this second bit as 0 and then I will make this as 5.

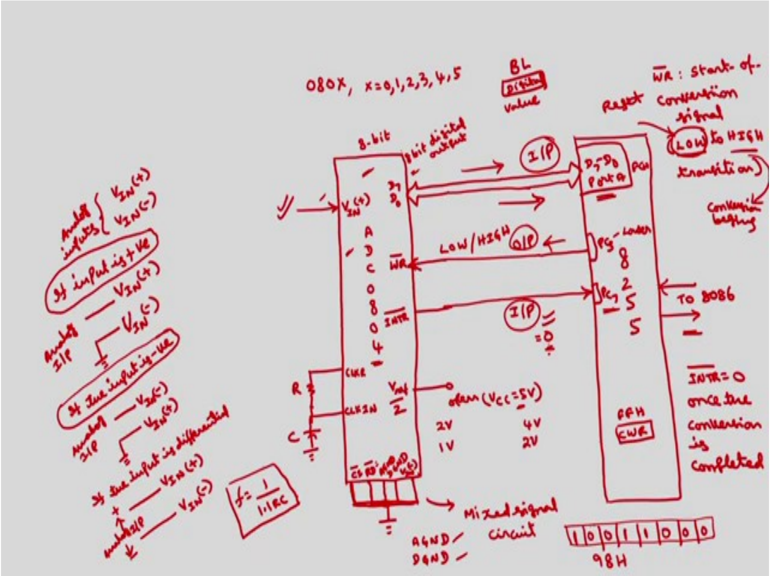
Here also we have two possibilities this can be greater than or less than this can be greater than, greater than means I will keep this one as it is and I will make the next bit also 1, less than means this 2 is less than means I will make this 2 as the second bit 1 as 0 and I will make 1.

Similarly, now you have three things, two here. So, if it is greater than then I will make 1 1 1 as it is, because I have only these three options only this option only. Then the other option is if it is less then I will make 1 1 0, here also we have two things either I can keep this 3 bits as it is 1 0 1 in case if it is greater than or equal and otherwise I will make third bit as 0. Similarly, here I can keep this value as it is 0 1 1 or 0 1 0, here also I can keep this value as it is which is 0 0 1 or I will reset the last value 0 0 0.

Now, here you see all the 8 possibilities are there. So, like that it will converge here let us assume that the given analog voltage is say for example, 7 volts ok, now what is the flow? First, I will take this 4, 4. So, 4 is less than ok. So, I have to make this is less than and this is greater than, this is less than, this is greater than similarly this is less than, this is greater than. So, 4 is less than so, I will go to 1 1 0, 6; 6 is also still less than I will go to 7 this will be stopped here.

If I take say for example, this is some 2 volts then first I will take the 4, 4 is the I mean greater than 2. So, I will go to this one 2, 2 is equal so, I have to stop here. So, this will remain 2 only. So, after that we will not go and we will stop here itself. So, that 2 is coming here as 2. So, like that we can find out the 3 bit binary equivalent of the analog signal this is the principle behind successive approximation type A to D converter.

(Refer Slide Time: 29:28)



So, the conversion time here will be more, now there are different I mean ICs for this A to D converters, the ICs of A to D to converter will be of the order of 080X, X varies from 0, 1, 2, 3 so on up to 5, means it can be 0800, 0801, 0802, 0803, 0804 or 0805 ok. So, the if I take the interfacing diagram of A to D converter to microprocessor this A to D converter, I am considering here 0804 is a 20 pin IC with the signals represented by this.

This is ADC 0804. So, the difference signals or we have a data bus because this is 8 bit A to D converter this is 8 bit A to D converter. So, input is analog value output will be digital value this is 8 bit D7 to D0 this is 8 bit digital output and you should have some analog signal. So, analog signal will be we have two analog signals; this is one is called VIN plus, another is called VIN minus.

Here this thing is VIN plus and V IN minus two inputs are there, analog signals two VIN minus these are called analog inputs, why two signals are required is. So, the given input signal can be either positive or negative or it can be a differential signal, we can apply the sinusoidal type of signals also. So, if the input analog input signal is input is only positive I want to I mean convert only the positive values into digital equivalent. What I will do is, I will connect the input to the VIN plus at VIN plus we will connect the input analog input which is to be converted into digital and simply VIN minus is to be grounded.

This is the case if I want to use the positive input if the input is negative, I want to only convert the negative analog signals into corresponding digital equivalent. Then the reverse analog input has to be connected to VIN minus VIN minus and VIN plus has to be grounded. If I want to I mean use input is differential if the input is differential means it will have positive values as well as negative values then I will connect that between VIN plus and VIN minus.

Here I will connect the input signal between VIN plus, plus to plus minus to minus. Here I am assuming that VIN minus I am going to ground these along with some other signals means that I want to use only for positive signals VIN is input signal is positive. Then in addition to this A to D converter requires some clock the time taken to convert this analog signal to digital signal is decided by clock.

Clock has to be connected between the two pins, one is called CLKR clock R you have to connect externally R and C which is going to decide the frequency of this particular clock CLK IN. If this is R and this is C then the frequency of this clock is decided by frequency is equal to 1 by 1.1 RC, by properly choosing these R and C values we can find out the frequencies.

Similarly, this will have some chip select signal that also has to be grounded in order to I mean perform any operation chip select has to be grounded this is chip select signal this is also I am connecting to ground common ground. And we have read bar signal this also we have to ground for read operation whenever this conversion is over you have to read the digital data here for that RD bar should be 0 and then we have two grounds; one is called analog ground, another is called digital ground.

This is A ground, this is D ground, totally five signals we are going to ground here, one is VIN minus, chip select bar, RD bar, analog ground and digital ground. So, what is the reason for using the two grounds, analog ground and digital ground, because this is a mixed signal device this is a mixed signal device, we have analog signal as well as digital signal mixed signal circuit.

So, that is we have analog ground as well as digital ground, if I use two grounds it will reduce the noise that is the reason why we will use two grounds. Coming for the other side, we have one write bar signal also. So, this write bar signal we will add say start of conversion signal I will explain this rate of write bar signal and then we have INTR signal, this write bar is output signal, INTR is input signal and then we have V reference/2.

This is normally made open, if this is open this will assumed VCC which is normally 5 volts. If I want to use other volts other voltage here you have to connect these two. If I want to I mean operator a maximum voltage of 4 volts you have to connect here 2 volts, if I want the maximum voltage of 2 volts you have to connect to 1 volt this is V reference by 2 ok. Now, coming for the connections to this 8086 so, you have to connect to through 8255 we know that say I mean I/O device has to be connected through 8255 this is 8255.

How to connect this 8255 to 8086, we discussed in the earlier classes to 8086, then this digital has to be connected to 8255 is also having 8 bit data bus I will connect just simply these two data bus together. So, that the data transfer can be takes place between ADC and 8255, this is D7 to D0 of 8255. So, earlier you have discussed there is a 8 bit data bus there also.

I will connect these two one of the ports of 8255 I will say use port A I will connect these to port A of 8255 whose address is FCH. Then I will connect these to one of the pins this is a single signal I will call as this one as some PC3 this I have to connect as input port this is going to be input port and this has to be output port this I will connect to PC7 it is up to you.

Here the operation is basically first the microprocessor has to be initiate from here microprocessor connection is there. So, this will start conversion through this 8255. So, this is reverse this WR bar is output pin and INTR is input pin, this is output, this is input.

So, this is input pin. So, the operation here is WR bar will acts as start of conversion signal; start of conversion signal. So, whenever the microprocessor wants to read some data from the A to D converter. So, what the microprocessor will do is, first the microprocessor will initiate ADC through start of conversion signal. So, here in this particular ADC 0804 if I apply a low to high transition then the conversion will start.

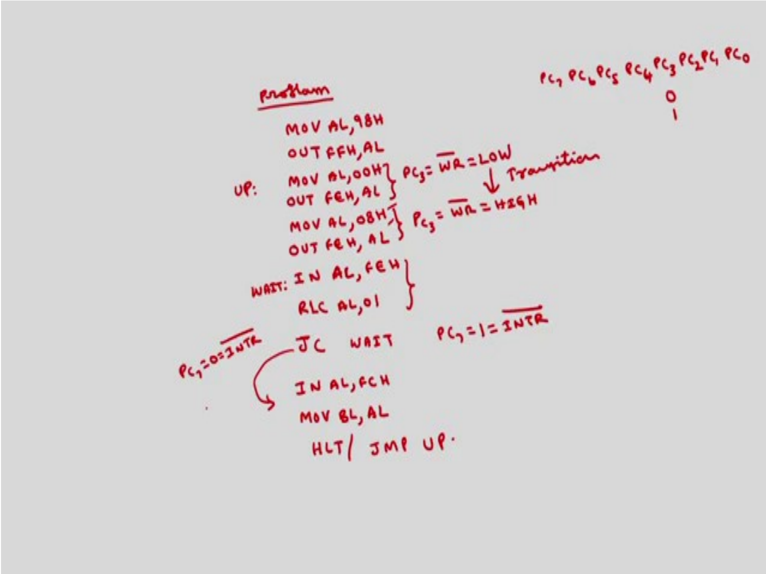
Whenever this WR bar is low this ADC will be reset whenever it goes to high this will start conversion. Low means this will reset the ADC this will reset the ADC and whenever this is high, conversion begins. So, after the conversion is over this INTR signal becomes 0; INTR bar becomes 0; INTR bar becomes 0 once the conversion is completed this is the process.

Now I will write the program to read the digital equivalent of this analog value into say some register or some location it is up to you. Suppose if I want to take digital equivalent of this one in to some register say the digital equivalent of this value I want to take into register say BL, BL I want to store the digital equivalent of the given analog signal. So, for that we have to write the program using 8086 instructions.

So, the first part of the program will be initialization. So, you have to program this port A as input port this is going to take the data port A should be input port and this should be output port, port C this comes to lower port C lower, port C upper has to be input port. So, what is the control word register for this? So, if you take the control word register the last bit is port C lower this should be output 0, then port B we are not using I am using as output port mode selection for port B I am using mode 0.

Then port C upper this should be 1 because port C upper you are going to used as input port and this is port A port A has to be used as input port so this should be 1, mode selection for port A is I am going to used in mode 0 and then this is your mode control which signal which is equal to 1. So, what is the corresponding hexadecimal, this is 98H if I enter this 98H into the control word register of this 8255 whose address is FFH. Then this will set port A as input port through which the data will be read and port C lower as output port, port C upper as input port, this is the first instructions.

(Refer Slide Time: 43:58)



So, MOV AL , 98H OUT control register address is FFH , AL. So, these two instruction will set port A as input port in mode 0, port C lower as output port, port C upper as input port all in mode 0.

So, after that what you have to do, you have to start the conversion means you have to first make this signal you send here low. Then after that you have to send high, then the conversion will start, after that you check the signal input signal. If this input signal is 0 means conversion is over, if this is 1 you have to wait until this is equal to 0, once this is equal to 0 is you simply read the data that is the procedure. So, first I want to send a low over this PC3 and then high over PC3.

So, what is the instructions for that, low over PC3, PC3 means PC7, PC6, PC5, PC4, PC3, PC2, PC1, PC0. So, because I am using PC3, PC3 should be initially 0. So, I can make all 0s. So, we can take MOV AL , 00H, OUT port C address is FEH , AL. This will make all the pins of port C as low, but I want only the PC3 which is equal to which is connected to write bar this will becomes LOW, a LOW to HIGH transition is required.

So, what is the time between the LOW to HIGH transition? A small amount of the time which is the time taken to execute these instructions is enough. So, no need of delay routines here. Then I will make MOV AL , I want only PC3 is on means this should be 08H, OUT FEH , AL. Now what happens? PC3 will be WR bar which is equal to HIGH. So, there is a HIGH to LOW transition.

What is the transition time, is the transition time is to execute this and this instruction that time is enough to I mean activate the A to D converter. Now the conversion will start I have to check whether the conversion is over or not. So, conversion end of the conversion signal which will acts as a INTR signal which is connected to PC7, I have to check PC7, if PC7 is 0 conversion is over, if PC7 is 1 conversion is not yet over.

So, input the contents of port C. So, the instruction will be in IN AL , FEH and you have to rotate left ROL AL,01, in AL will be having port C contents. So, in that MSB bit is PC7 which is connected to. So, now, we are checking here PC7 bit will comes into carry, JUMP on CARRY. If carry is there what does it mean, if this is to jump on carry WAIT.

Jump on carry means PC7 is 1 means conversion is not over, PC7 which is equal to INTR bar, 1 means conversion is not yet over. So, I will just simply wait, where you have to wait, you have to input these again you have to input this and you have to rotate for the MSB bit we can I use here ROL AL or RLC also through carry. So, when does it will come out of this loop is whenever PC7 is equal to 0 which is equal to INTR bar. Whenever this is equal to 0 this will come out of the loop means conversion is over.

Now, where the data is available? In port A so, you have to read that port A, in AL , port A address is FCH, I want to store this data into BL register. So, we can move contents of AL on to BL, you can stop if you want to read only once if you want to continuously read instead of all you can write jump to UP, this UP is again this. So, in this way you can read the digital equivalent of the analog signal using ADC.

So, you will see the more details like how the conversion has to be takes place and what are the exact digital equivalent of the analog signals and what is the resolution of A to D converters in the next class.

Thank you.