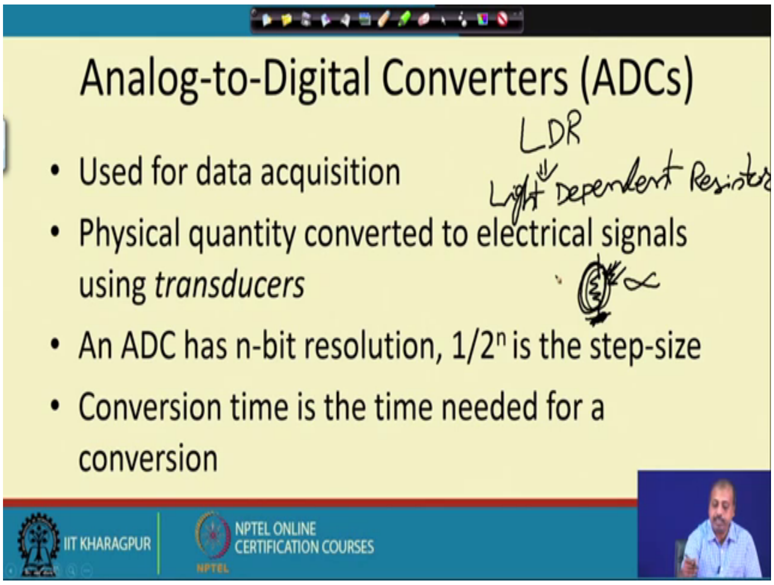
Microprocessors and Microcontrollers
Prof. Santanu Chattopadhyay
Department of E & EC Engineering
Indian Institute of Technology, Kharagpur

Lecture- 56
Interfacing (Contd.)

So, the next interface that we will look into is for the analog to digital converters Or ADC's.

(Refer Slide Time: 00:19)



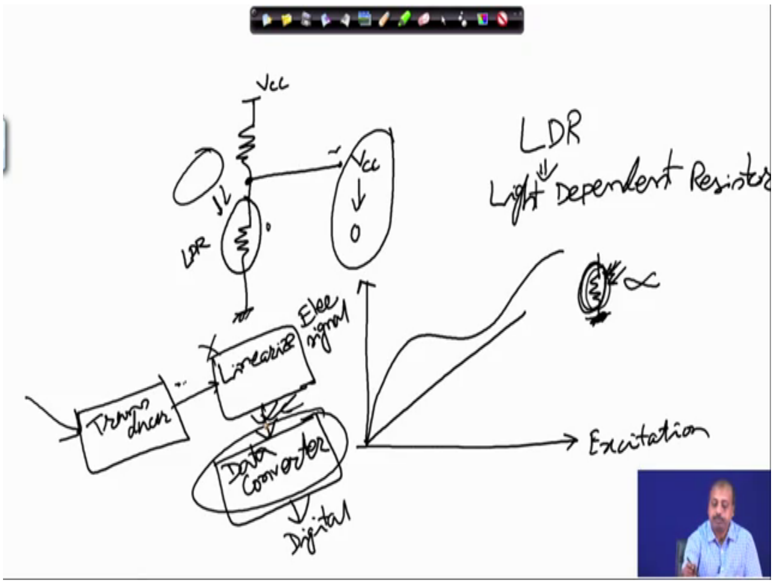
As you know that real world is analog in nature. So, though we try to get the signal values inside the microprocessor or microcontroller in ti in digital terms, and we do the processing in digital terms. So, the real world is analog so, the signals that you get from the environment so, they are analog in nature. So, to get the values from signal from the nature so, we need some facility by which we can read those values.

For example, if you are say reading the temperature of a room so, that is the (Refer Time: 00:50) even if it is converted into some voltage so, what you get is a continuous range of voltages. Or if you are monitoring the flow of a say speed of a strain or say speed of a belt conveyer belt so, that is also a continuous quantity that you are getting. So, but unfortunately the microprocessors and the microcontrollers they are digital in nature and they cannot process this analog data directly. So, what is required is that we should convert this analog data into some digital values, ok. So, these are the, these are done by the chips which are known as analog to digital converters or ADC's.

They are used for data acquisition that is for collecting data from the environment. Physical quantity converted to electrical signal using transducers. So, transducer means, this is some device by which you can convert physical data into a say into a electrical signal. For example, a very simple transducer are the light dependant registers or LDR's light dependant register.

So, if we want to detect the light so, it is a structure like this, so, this is simply a register, but this value of this resistance will dependent on the light that is falling on to it. So, a typical design may be that when there is no light falling. So, this resistance is say infinity or very high, and when some lights are falling on to it the resistance will decrease. So, you can have a very simple design like this.

(Refer Slide Time: 02:32)



So, you put a resistance, a current limiting resistance, and this is the LDR this is the LDR. So, this is connected to some supply voltage Vcc, this is grounded, and from this point I am getting the value. Then what will happen is that when this when light is not there, then resistance is very high. So, you will get here voltage almost equal to Vcc. There will be very little current flowing through this device. So, you will get the voltage around Vcc. But when some light is falling on to it, then it is resistance will decrease, and depending upon the amount of light that you have so, this resistance will be varying.

So, if I go for the maximum possibility, then this resistance will be close to 0. As a result, this point will be almost grounded. So, you will get almost 0 volt there. So, this is the, this is. So, this is the electrical signal converted for of the light that is falling on to the system ok. So, that that way there are many such transducers like we have got this pressure transducers, then different type of sensors gas sensors and other sensors that we have. So, they all they all come under the heading called transducers.

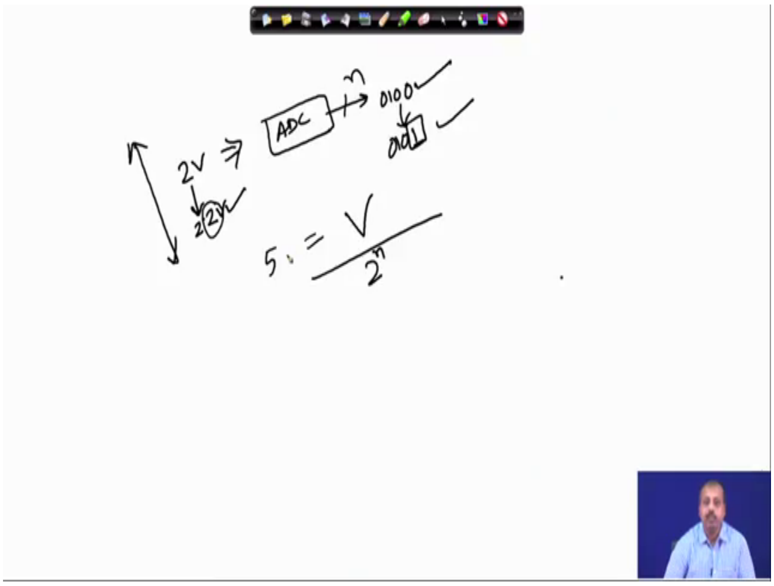
Now, this transducers may be linear or the transducers may be non-linear. So, linear transducers means that the amount of excitation that we have. The corresponding electrical signal that is produced is proportional to that. So, it is ideal that I have a behaviour like this ok. So, if this side is the, this side is the excitation, and we have got say the electrical voltage electrical signal there may be voltage may be current ok. So, whatever it is so, I would ideally like to have a straight line like this. It may not be at 45 degree, but somewhat it is more or less the straight line.

So, I would like to have a behaviour like this. But this seldom happens and normally we get different types of behaviour, may be it will go like this, it will have a behaviour like this. So, what is required is that we have to identify the range and depending upon the range we have to do some sort of piece wise linearization or thing something like that, but ultimately. So, this say what I mean is that this transducer alone is not sufficient for this purpose ok. This transducer output it goes through some circuitry which we call in general the linearization circuitry.

And from this linearization circuitry it goes to the convertor. So, this is the data converter that we will be talking about in our course. So, that is the analog up to this much I have got the analog signals. So, at this is the environmental excitation that we are getting the transducer. So, it is producing some analog values that analog value is linearized, and then we get some linearized analog values here, but this is also analog. So, ultimately it goes to the data converter, and here what you get are the digital quantities, they are the digital quantities.

So, this digital quantities can be processed by microprocessors and microcontrollers. So, our part of discussion is around this. So, we do not talk about this linearization circuitry. So, that you can find in some analog electronic course so, we will not go into that. So, we assume that this linearized electrical signal values are available, and then we are going to convert it from analog quantity to digital quantity. So, the device that is mostly used for this purpose known as the analog to digital converter or ADC so, an ADC can have a n-bit resolution in 1 upon 2 power n is the bit step size. So, like if I say that this transducer is producing some voltage. Now what is the difference in voltage that it can convert it into digital? Like see if I have got if there is a if there is some electrical signal of say 2 volt.

(Refer Slide Time: 06:50)



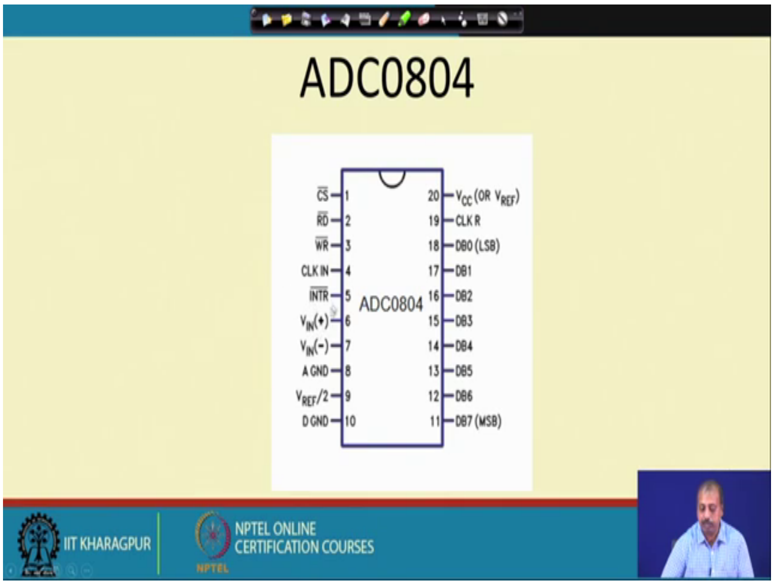
Accordingly suppose this device that I have this ADC, it produces some output, say it produces 4-bit output so, 0 1 0 0. Then the next closest output that it can produce is 0 1, 0 1 the single lsb change, fine. Now single lsb change, now for what change in this voltage it can produce this single lsb, lsb change, ok? So, if the corresponding the next voltage is 2.2 volt, such that for 2-volt output is this 1, for 2.2-volt output is this 1, then I can say that it has got a resolution of 0.2 volt. So, it is it can sense the change of 0.2 volt, and convert it into a different digital value, the next digital value ok.

So, in general I can say that if the total range of voltage that you have here is say V and the total number of outputs that. So, if this ADC is producing an n-bit output, then you can understand that it can produce to power n different values. So, those 2 power in values will correspond to this 2-power n different voltage levels. So, I can say V upon 2 power n. So, this may be taken as the resolution of the lec le of the, this ADC because ideally so, this entire range is uniformly divided. So, it is not that say V equal to say 5 volt; so, it is not that 0 to 1 volt it produces values which are close to each other from 4 to 5 volt it produces values which are far away from each other. So, so the ave the corresponding change in input voltage that is required to change the lsb is different over different range of voltages values. So, that if that is that is happening so, we are not really happy with that situation because then this conversion does not have much belief in that. So, ideally, I would like to have a system where this range is uniform,. So, each of them are converted into proper values within the range.

So, it has got an n bit resolution, and then this n bit resolution is having this n bit resolution has 1 upon 2 power n as the step size. And conversion time there is a quantity called conversion time so, which is the time needed for the conversion. So, it is not that analog value changes continually and still this digital ADC will be following that value (Refer Time: 09:27) the analog value at the same rate that may not happen. So, normally this devices they take some time to for doing this conversion. And there are different types of ADC's ok.

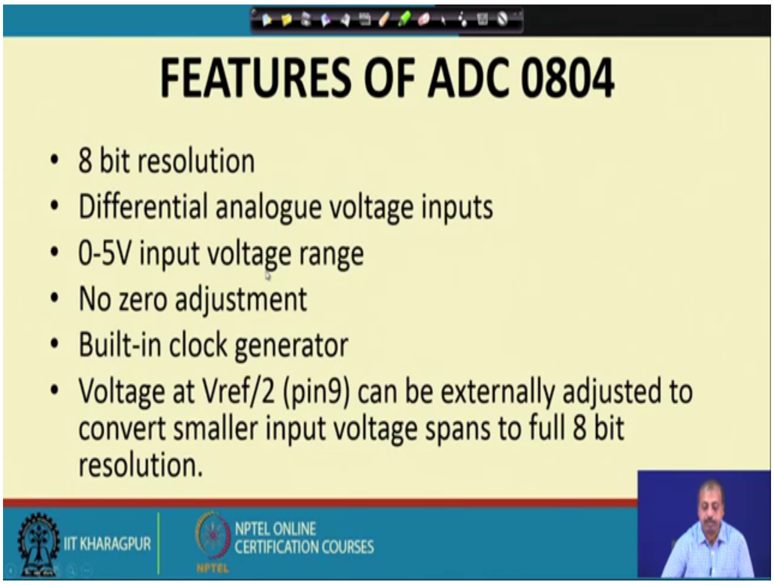
So, we have got simple successive approximation type, where it changes just go on change goes on changing the bits one by one, first it changes the lsb then compares the resulting value with the corresponding analog value. And if it is not sufficient then it will be again changing the next bit so, it will do like that. So, we have got successive approximation type upto say flash type where this conversion is parallelly done. So, all the bits are checked parallelly for the possible values ok. So, that way we have got different classes of led ADC's so, the conversion time will be different for each of them.

(Refer Slide Time: 10:20)



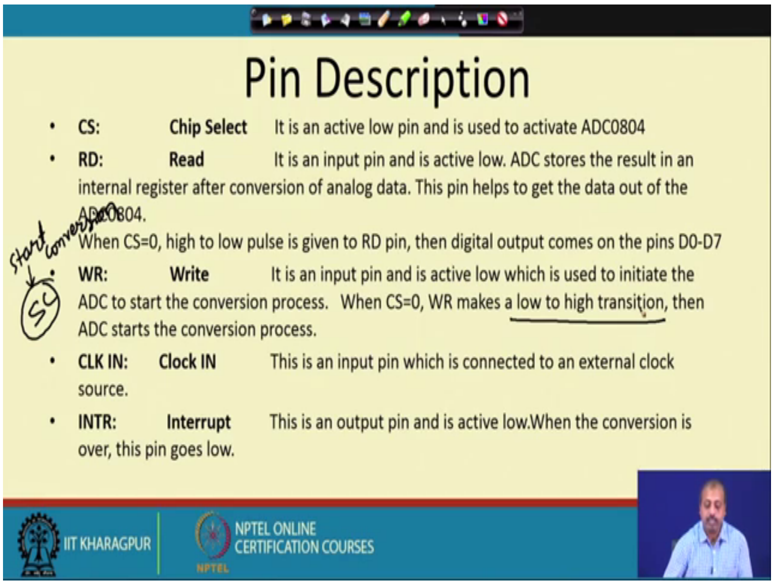
So, this is one example ADC, ADC 0804 so, it has got a number of pins there. So, we will see about this pin this name of this pins and there meaning in successive slides.

(Refer Slide Time: 10:26)



So, this has got 8-bit resolution. So, that n equal to 8 bit so, if you look into this diagram, you can see that it has got DB 1 to DB 6, and this DB 0 to DB 6 DB 7 so, this is the ADC output. So, this may be producing an 8-bit output. So, that n equal to 8 so, total resolution is, that is why it is said an 8-bit resolution. So, we have got this 8-bit resolution an differential analog input voltage inputs.

(Refer Slide Time: 11:11)



So, if we so, we have got this analog input voltage V in plus and V in minus ok. So, this differential voltage so, this is useful because it will have less error in the input that we are getting. So, if you are giving a single input only V in and that with respect to ground. So, that way possibility of error will be more. Like, if there is some error which some error which effects both of them equally, then since this pins are very closely located. So, if they there is an electromagnetic interference or something like that which effects pin number 6. Then it is very much likely that effect pin number 7. So, the input is taken as 6 minus 7 so, as a result those noises they will get cancelled out so, common mode noise they will go. So, that is the purpose of having this differential input.

Then 0 to 5 input voltage range. So, as I was telling that the range of inputs that it can convert from analog to digital. So, analog signal value can be in range of 0 to 5 volt, there is no 0 adjustment. Some of the ADC's that you have so, there is a pin by which you can do A 0 adjustment, means if you give A 0 volt at the input then what is the corresponding output so, we expect that the output should also be 0. So, 0 for that matter the minimum possible re re range value that you have.

So, if you give that it is expected that the output should be 0. So, it should be if it does not happen then there is a problem. So, in this case this 0-adjustment facility is not there. There is built in clock generated, because when it this ADC is operating so, it requires a clock signal. So, for some of the ADC's this there is a built-in clock circuitry generated because when it this ADC operating. So, it requires a clock signal. So, for some of the ADC's there is built in clock circuitry so, which can feed it so, clock directly. Or you can have some some ADC's we have to connect externally some clock generated may be coming from the microcontroller, or may be coming from some circuitry say may be say 5, 50 timer-based circuitry from which the clock is generated and fit to it.

Then the voltage at Vref by 2 can be externally adjusted to convert smaller input voltage spans to full 8-bit resolution. So, this is the pin Vref by 2 so, Vref by 2. So, if your input voltage range is not in the input voltage is not in the range of 0 to 5 volt or something less than that. So, you can connect some value here by which you can say that my input is not in that range. So, it will be less than that that way it can be set.

So, important chips that we pins that we have in this case, first one is the CS which is the chip select. So, chip select is an active low pin and is used to activate the ADC. So, just like any other device that we connect with microcontroller so, there has to be a chip select line. Because the databus line that have from this microcontroller from the ADC will also be driving the databus of this microcontroller to which other devices are also connected. So, this chip select line should be selec should be low for the chip to before the ADC to get selected.

Then there is a rd read rd pin. So, it is the input pin it is active low. So, so ADC will store the result in an internal register after conversion of analog data, and this pin will help to get the data out of ADC. So, so you apply an analog voltage and tell the ADC to convert it. So, it will convert the value, but it will not put it on to the databus immediately. So, it will hold the value in some internal register so, to read the value of the value from that register on to the databus.

So, what is required is, you should make this read bar pin low so that you get the value in that register on to the databus. So, when CS bar CS equal to 0 or CS bar is activated, a high to low pulse will be given to the read pin, and then the digital output will come on to the dg D 0 to D 7. So, this is read line so, this is a this requires a high to low pulse, that is low falling h triggered operation.

Then there is another pin which is write WR, this is also an input pin and active low, and this is this initiates the conversion process. So, in some of the processer some of the ADC's so, there is another pin which is known as start conversion or SC start conversion. So, in this particular ADC so, you do not have any separate start conversion line so, we have got this write line so, if this low to high to low pulse is given on to this sorry sorry when we make this low to high transition on this write pin, then this ADC will start converting. Because once the data has been put on to the once the analog signal has come so, this microcontroller should tell the ADC to start doing the conversion.

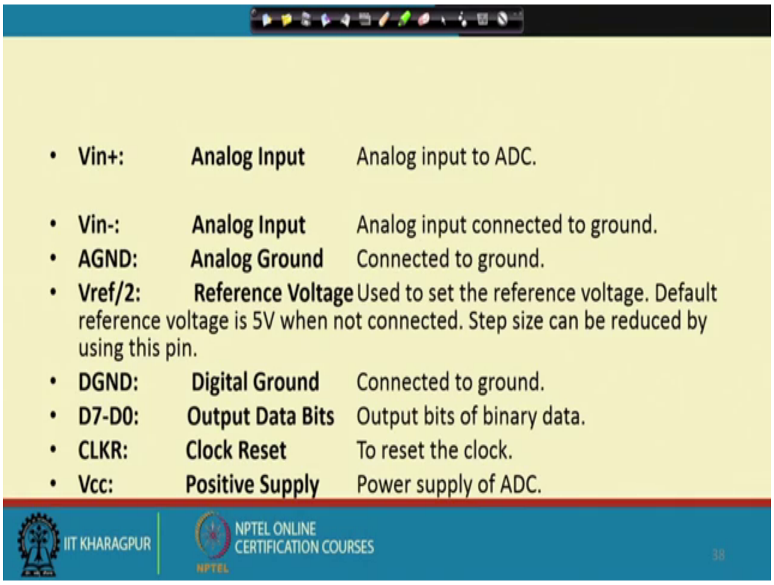
Otherwise there is no point converting all the values that are coming on to the analog line. So, analog line will continually change it is value because it is coming from the environment so, it can continually change it is value. So, the micro controller may not be interested in getting all those converted values. So, whenever it feels that I should get the corresponding digital output. So, it should set this start conversion signal or the right signal and give a low to high transition on that with CS equal to 0, ok. Then the ADC conversion process will start.

Then we have got clock in so, if you have got an external clock source, then you can connect this clock near the line there so, this will be acting as the external clock source. As I said that this 0804 has got internal clock source as well. So, if you think that I will be working with an external clock so, you can do that. Then this there is an INTR pin. So, which is an output pin and active low, when the conversion is over this pin will go low.

So, when this you think about the situation like you have connected one ADC with a microcontroller, and you have microcontroller has given this CS bar and write bar CS bar signal and given a low to high transition write so that it starts converting. Now depending upon the type of ADC so, it will take some amount some amount of time to do the conversion. So, what does the microcontroller do in that period of time? So, one possibility is that the microcontroller will be continually checking whether the conversion is over or not, and other possibility is that microcontroller will go to some other job it will do some other job.

And when the ADC is done so, it will give an INTR signal to the to the microcontroller the interrupt signal to the microcontroller telling that the conversion is over. Accordingly, so, it may be this particular line may be connected to say for 8051. So, it may connected to int 0 or int one pin, and accordingly it will be jumping over to to the corresponding ISR. In that in that ISR so, we can activate this read bar line, and accordingly we can get the content of this converted value.

(Refer Slide Time: 18:51)

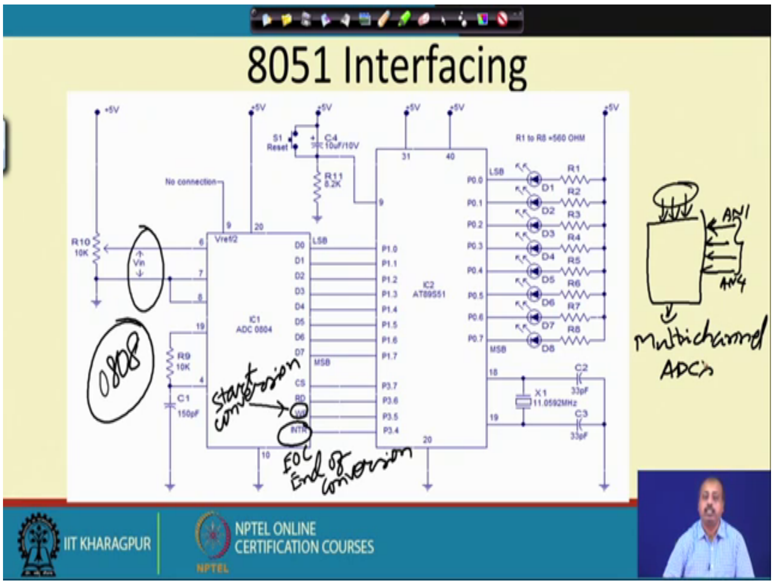


So, this is other pins so, Vin plus and Vin minus. So, they are the analog input so, this differential input has I say. So, analog may be you just connect if if you don in the simplistic case Vin minus is connected to ground, and the Vin plus is the signal line then we have got analog ground. So, which is connected to ground and there is a Vref 2 reference voltage so, we used to set the reference voltage. So, default reference voltage is 5 volt when not connected if you keep this Vref by 2 pin open then the reference voltage will be 5 volt. And when if it is connected then the reference voltage will become 2.5 volter, accordingly step size is you can be reduce by using this pin.

So, if say if you are instead of converting 0 to 5 volt if you are converting the range. So, 0 to 2.5 volt then what will happen? Is the same 8-bit ADC so, you are getting same 256 levels of inputs converted into digital values. So, if the range is less; that means, you are your resolution will be higher because now you can have less reference in the lsb in the analog voltage that will produce change in the lsb so, we have got this Vref by 2 pin.

Then there is a digital ground pin so, dgnd so, that is also connected to ground. And this D 0 to D 7 these are the output databits, then we have got clock R so, the for resetting the clock, and this is a Vcc power supply so, it is the power supply for the ADC. So, these are the important pins that we have for this ADC.

(Refer Slide Time: 20:24)

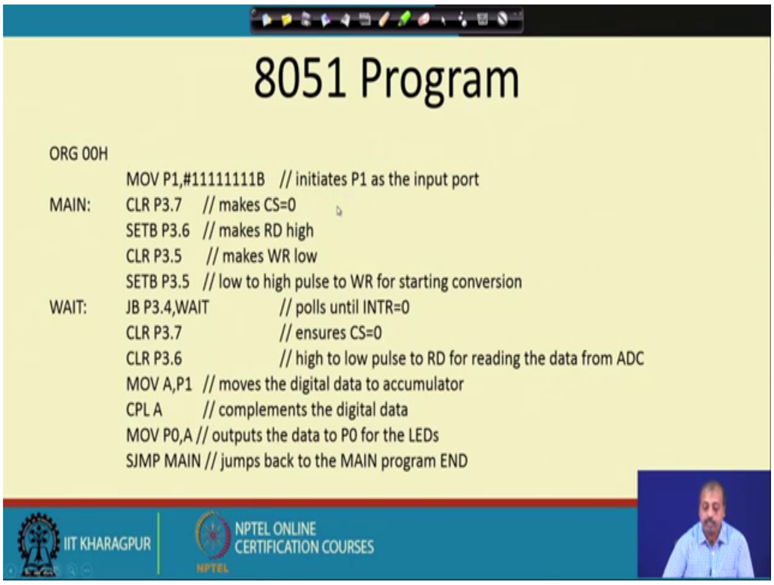


So, this is a schematic diagram in which we have connected one 8051 chip with this ADC 0804, as you so, in this case this port one has been used for a getting the data. So, this port one is connected to this lsb is D 0 is connected to P 1.0 and this D 7 is connected to 1.7. Port 3 has been used for driving the other signals like the chip select read write and interrupt. So, these are the signals given for this operation and other values like this Vin is coming from say the Vin is connected.

This is input to the microcontroller by means of this it is connected to this connected to this ad ADC by means of this 10-kilo pot so that we can change this pot position to give different analog input values to the ADC. And this ADC will convert the value accordingly, the, this Vin minus and this analog input ground so, analog ground so, they are sorted. And we have to connect a resistance and a capacitor here this is required for that clock generation circuitry.

So, internally it will generate clock so, this circuitry will do that. So, Vref by 2 is not connected and supply voltage is 5 volt so, that is the side we have got for this IC iip 0804 ADC. On this 8051 side, I have already said that port one is used for getting the data bit lines. Port 3 is used for giving the control signals. And suppose in port 0 we have connected a number of LED’s number of LED’s are connected. So, whatever digital value will be obtained here will be put on to port 0 so that on port 0 you can see the corresponding bit pattern glowing ok. So, that is done here so, this is interfacing circuitry.

(Refer Slide Time: 22:36)



Now, the structure the programme structure will be something like this. So, org origin 00H means that the programme will be complied from location 00H. So, first of all this P 1 has to be put in the input mode ok, this if you look into this port P 1 is connected as input from D 0 to D 7. So, P 1 has to be converted P 1 has to be configured as input volt. So, this is done here, then we will select this chip select line and make it equal to 0. So, this clear P 3.7. So, this will put this CS bar CS line equal to 0, then set bit P 3.6.

So, this will make this read line high and clear P t P 3.5 it will make the write line low. Then we put set bit 3.5, so, this will be giving a high pulse to the write for start conversion. So, this read line should not be set now, because a high to low transition on the read line will be needed for leading the content of the ADC the converted value. But for starting the conversion so, we need to put a low to high pulse on this write bar on the write line ok. So, that is done here by set by clearing this P 3.5 and then setting P 3.5.

Then we wait till the INTR becomes equal to 0 ok. So, this is JB, jump on bit 3.4 WAIT, so, if you look into this diagram P 3.4 is connected to the Intel line from ADC. So, it can be programmed as interrupt, but in this sample programme so, we have not done that. So, we are just waiting on a busy loop, till INTR becomes equal to 0. So, once INTR becomes 0 so, it will come to the next statement. It will clear P 3.7 so, it is chip select is so, this is done again.

So, we have already done we have already cleared P 3.7. So, this may be an extra caution so, we are clearing c CS bit by clear P 3.7. Then we clear P 3.6 so, that gives a low to high high to low pulse on the read line. So, read line was previously set to here it was set to high by set bit 3.6. So, it was set to high, but here it is it is cleared. So, we get a high to low pulse on the read bar line. After that the data is available on the databus on the. So, it is the D 0 2 D 7 line of the ADC, there it is containing the converted value. So, that I have to read on to the port P 1 to on the port P 1.

So, that is done here move a comma P 1 so, from that digital data will come to the accumulator. Now we want to display it now for the display part, you see the way this LED's are connected so, if this P 1.0 was say equal to 1 so, to glow this led I have to put A 0, here similarly if this bit is 0, I do not want to glow this led. So, accordingly I should value put a value of 1 at the this point. So, this way you see that we need a compliment of this value that we have read from here to be put on to port 0 for getting the corresponding pattern.

So, this is done here so, after getting that P 1 port content into according to accumulated A. So, we are complementing A and this complementing A will complement the bits. So, all the bits are complemented and then move P 0 comma A. So, move P 0 comma a so, this will output the data to P 0 for the LED's. So, this will be outputting this P 0 A so, this will be out this pattern will go to P 0. And then SJMP main so, again go here. So, you can again I have to make this read line high ok, they gave another by this time I am ready to sample the next data from the ADC so that I should give another RD's signal another write signal to the ADC.

So, that is given here by high to low pulse on the read line on the write line low to high pulse on the write line, and then again it will do the conversion so, this process goes on. So, in some cases so, we have got an this INTR signal, in some of the processers you will find that this Intel line so, it is also known as end of conversion. So, this pin in many many cases you will find it written as eoc or end of conversion, end is the end of conversion pin. And this write line this I have already said that this is the start conversion. So, this type of conventions are there so, if you looking if you are looking on to a different ADC. So, you may find that this pins are names are slightly different.

So, this way you can connect ADC's with that. So, this particular ADC this is a single channel ADC. So, there is only one channel on which you are feeding this Vin input and this is converting that. So, in some of the ADC's like there is a next ADC that we have is 0808 ADC. So, it will have a more it will have a more number of analog channels. So, that chip instead of having a single analog channel.

So, it can have a number of analog channels. So, there may be say 4 such analog channels, they are named as a A N 1 to A N 4, and then there will be corresponding some select lines will also be required from the microcontroller. So, you can tell which analog channel you are going to convert by setting these select lines. And accordingly, the conversion will be done and the converted value will be available on the databus. So, that way we have got multi-channel ADC's also so, these are commonly known as multi-channel ADC's. So, they are useful for if we have got several analog quantities to be converted into digital value so, they can be useful for that.