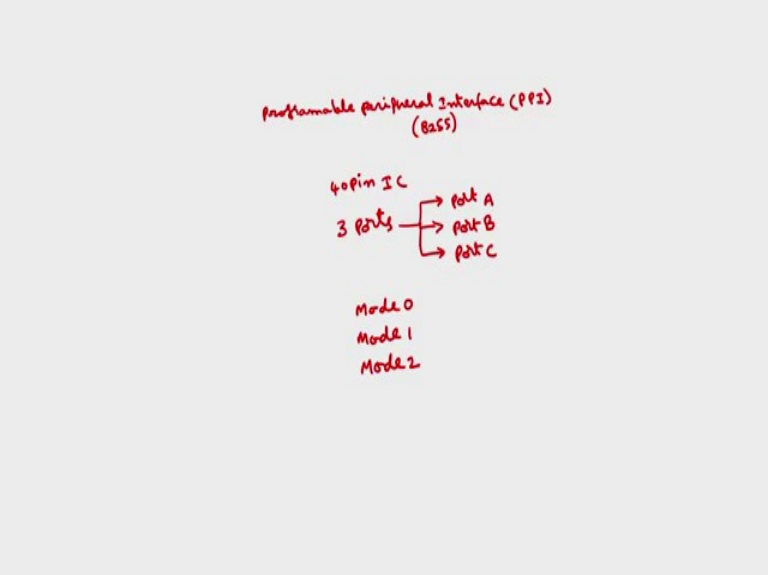
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture - 18
Architecture, Interfacing to Simple I/O

Okay, In the last class we are discussing about the memory interfacing to 8086, next in order to interface the IO devices, we need a peripheral chip call programmable peripheral interface 8255.

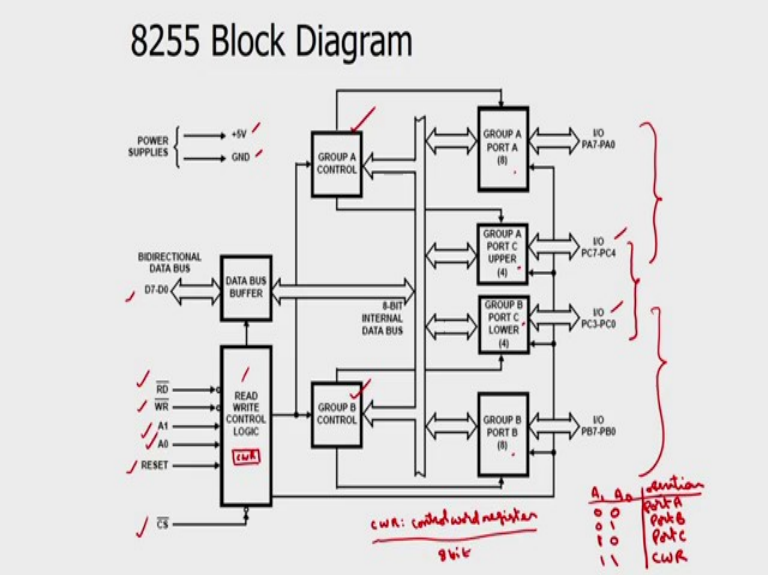
(Refer Slide Time: 00:41)



Programmable Peripheral Interface PPI, the IC number of this one is 8255. So, using these 8255 we can connect any input output device to the microprocessor ok. So, 8255 is basically 40 pin IC with the 3 ports called port A, port B and port C.

So, each port can be programmed as input or output ports, because we can use input device as well as output device to communicate with microprocessor. And here this can operate in 3 different modes called mode 0, mode 1, mode 2. Port A can operate in all the 3 modes, port B and port C can operate in only 2 modes ok. So, the detail pin diagram of this one and the architectural diagram of this one. So, we will discuss in the next slide.

(Refer Slide Time: 02:17)



So, if you take the block diagram of this 8255 as I have told there are 3 ports which is port A; port A, port B and port C. But port C is again divided into 2 sub ports here called port C upper and port C lower ok. So, this port C upper is PC 7 to PC 4, port C lower is PC 3 to PC 0 ok. So, these two are called; these two are called group A; group A and these two are called group B ok.

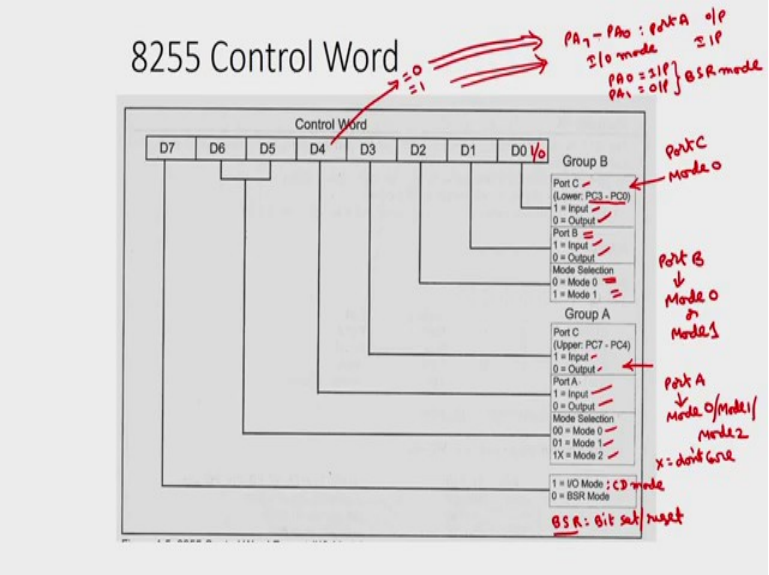
And there will be a some data buffer will be having 8 bit data bus and then we have read bar signal, write bar signal, A 1, A naught which are used for selecting these ports, there is one reset signal, chip select bar signal. To select these particular chip we require chip select signal and we require plus 5 volts and ground to operate this IC.

So, here this is the read write logic, using this read write logic we can select the type of operation whether it can be reading from the input device or writing into the output device. This A 1, A naught is going to find out there is a control word register inside this read write logic there is a control word register which plays an important role CWR; CWR stands for control word register.

By programming this control word register we can program the ports as input or output port and any one of the modes mode 0, mode 1 and mode 2 ok. So, this A 1, A naught is going to decide A 1, A naught if they are 0 0, then the control word refer to port A, this is 0 1 port B, 1 0 port C, 1 1 control word register. This is how A 1, A naught will be using for selecting the ports or control word register ok.

So, the format of this control word register is 8 bit register; this is 8 bit register, the format of this control word register is as follows.

(Refer Slide Time: 05:02)



This is the I mean format of the control word register. So, we have this D naught LSB bit which will be used as a port C. So, you can program these port C lower this port C lower which is PC 3 to PC naught is either input port or output port. If this bit if you make as 1 this will acts as a input port, if I make as 0 this will acts as output port. The next pin will be for a selection of port B again same thing 1 means input port, 0 means output port. So, the next bit will be mode selection of this port B mode selection of port B. So, port B can operate in only two modes; port B can be operated in either mode 0 or mode 1.

Where as port C can be operated in only one mode which is mode 0. So, there is no bit to select the mode of port C, because only by implied it is mode 0 ok. Where as port B can operate in mode 0 and mode 1 so, we have one bit for selecting this mode 0 or mode 1. Then coming for the group A signals port C upper this is by default mode 0, 1 means input, 0 means output. Port A again we can program these port A as either input port or output port by selecting 1 or 0 by placing 1 or 0 into corresponding bit position, but port A can operate in 3 modes mode 0, mode 1 or mode 2.

That is why we require at least 2 pins to select the mode of this port A. So, here we have used this D 4, D 5 and D 6 for selection of the mode 00 is mode 0, 01 is mode 1, either this can be 10 or 11 is mode 2, 1 X means X is a do not care which can be either 0 or 1.

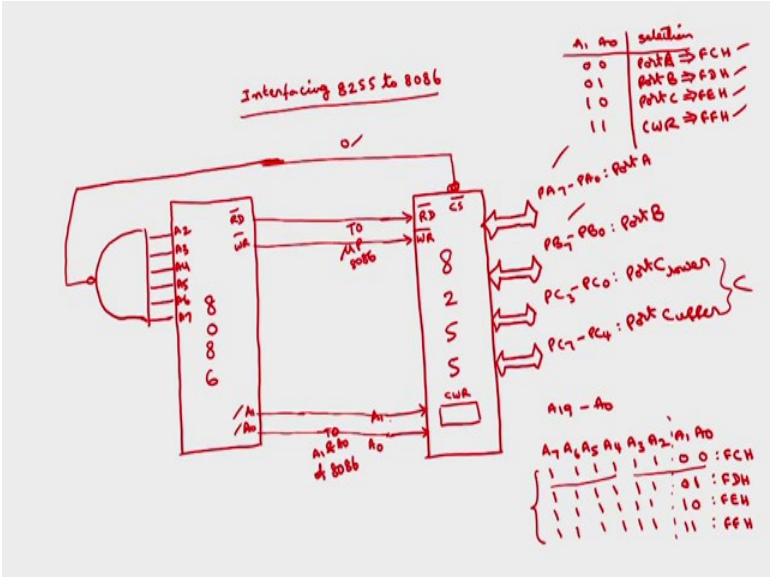
Then coming for the MSB bit D 7. So, this 8255 can operate in two modes overall these 8255 which is called IO mode or BSR mode, this is sometime also called as control word mode CD word; CD mode control word mode or BSR mode BSR stands for Bit Set Reset.

So, in IO mode or CD mode they cannot program the single bit of a port as a input port or output port, the entire port can be programmed as either input port or output port. It is not possible to I mean program a bit of port as input port, a bit of the same port as output port and all, whereas, in case of bits reset mode as the name implies. So, one pin of this port A so, the port A will be having 8 bit pins PA 7 to PA naught is port A.

In case of IO mode IO mode so, the mode selection for port A is this D 4 bit if D 4 is 0 the entire port A will be programmed as output port. If this is 1 the entire port A will be programmed as input port. We cannot program PA naught as input PA 1 as output and so on ok, but this is possible PA naught we can program as some input port, PA 1 you can program as output port in case of BSR mode as the name implies bit set reset mode.

So, that is the difference between the IO mode and BSR mode. In case of IO mode we can program the entire port as either input or output port whereas, in bit set reset each bit of this port can be programmed as either input or output modes ok. So, that is about this control word register. So, depending upon this control word register the 8255 will program the ports as either input or output ports and any of the modes mode 0, mode 1 and mode 2 ok. So, now, using these 8255 we can connect any input output devices, first I will take a simple example of connecting a LED Light Emitting Diode a simple output device.

(Refer Slide Time: 10:01)



Before that I will discuss the interfacing of 8255 to 8086. So, what are the connections required between this, okay. So, we know that 8255 we have discussed the pin diagram. So, different pins of 8255 are we have 3 ports, this is PA 7 to PA naught is port A, then we have port B PB 7 to PB naught and port C is available as port C upper as well as port C lower.

All these are bidirectional because they can be programmed as input or output ports this is called port C lower, then we have PC 7 to PC 4 port C upper; port C upper, this is 8255. And in addition to this we have read bar signal, write bar signal then we have a chip select bar signal, chip select is active low signal, then we have A 1, A naught.

So this A 1, A naught has to be connected to the A 1 A naught of 8086 ok. So, here this port can be either 8 bit port or 16 bit port. So, this we have to connect to A 1 and A naught of 8086, these two corresponding read and write bars of microprocessor 8086 reset also you have to connect, the only thing is how to generate this chip select signal for this 8255. So, here we have 8086 this is 8086 you simply connect these to A 1 of 8086, this you connect to A naught of 8086 this you connect to read bar of 8086, write bar of 8086.

Now, how to find out this chip select signal? Ok. So, here we have totally 20 bit address lines A 19 to A 0 out of this A 0, A 1 I have connected to this A 1, A naught of 8255. I will use this port addresses as only 8 bit we can use 16 bit also. So, in that case I will use

only lower order 8 bits of this address bus. So, what is lower order 8 bits of the address bus A 7, A 6, A 5, A 4, A 3, A 2, A 1 and A naught anyhow we are going to connect to A 1 and A naught of the 8255.

As I have discussed the function of A 1 and A naught of 8255 is 0 0 it will select port A, 0 1 port B, 1 0 port C, 1 1 control word register this is the selection. Here this A 1, A naught I am going to connect to here, suppose if I make this remaining A 2 to A 7 as all 1s, it is up to you can correct to either 0s or 1s, here I am assuming that they are connected to 1 ok. Then what is the address of port A? If A 1, A naught is 0 0 then port A will be selected. So, what is the 8 bit address? This will be C this will be F. So, this address will be FCH.

And if you want to select port B this I am going to fix for all the ports as all 1s all 1s. Now, what is different is A 1, A naught, for 0 0 you are getting FC address so, this will be the address of port A. And for this 0 1 what will be the corresponding hexadecimal? FDH, this F is 4 4 4 1s and 1 1 0 1 is 13 D. And this is for 1 0 this will be FEH, for 1 1 this will be FFH. So, the address of this port A becomes FCH, this will be FDH, this will be FEH and this will be FFH.

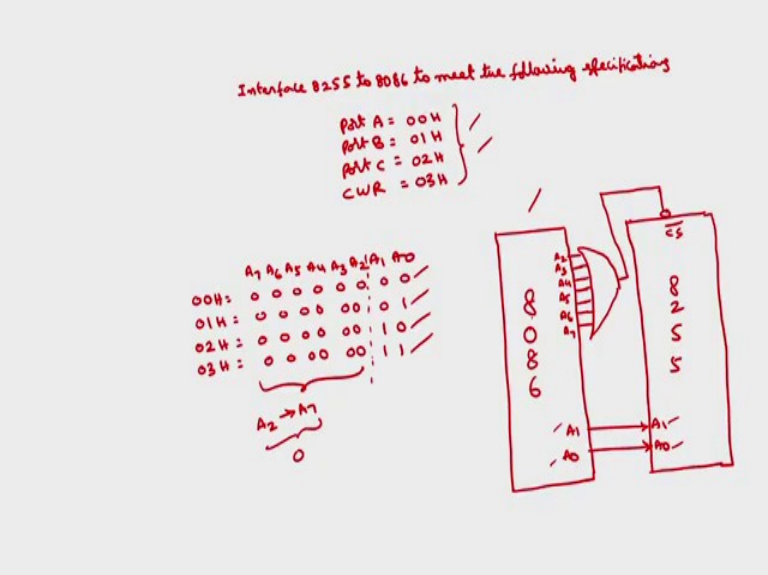
Now, how to select this chip select signal so you have to connect this A 2 to A 7. So, I am taking A 2 to A 7 of this microprocessor A 2, A 3, A 4, A 5, A 6, A 7 if all are 1s this chip select should be 0, this should be 0 to select the chip. So, if A 2 to A 7 all 1s then the output of the gate should be 0. So, which gate you have to use here is NAND gate. You connect the output of this NAND gate to the chip select of this.

This is A 2, A 3, A 4, A 5, A 6, A 7 if all are 1s for the NAND gate if all inputs are 1s then only output is equal to 0, if any one of the input is 0 output is 1 ok. So, only for this particular 4 combinations only output of the NAND gate becomes 0 there by this chip will be selected. After selecting this chip this A 1 and A naught is going to decide within this 8255 they are going to select the port A or port B or port C or control word register which is stored here.

Now, with this interfacing diagram the address of the port A is FCH, port B is FDH, port C is FEH and control word register is FFH. Suppose if I want to I mean connect in any address locations then accordingly you have to set this connections this is how we can

connect this 8086 to 8255 ok. So, we will discuss about how to I mean connect this 8086 to 8255 for a given address locations.

(Refer Slide Time: 18:25)



Suppose you will be given to interface 8255 to 8086 to meet the following specifications. For example, if I want port A as it is up to you we can take any 4 consecutive locations 00H, port B as 01H, port C as 02H, control word register as 03H. So, this should be consecutive locations, then what will be the connection diagram? This is similar to this earlier one except for that depends upon the address map we have to select the chip select signal.

So, this is 8255, 8086 remaining all connections are same A 1 to A 1, A naught to A naught. I am going to just show how the chip select will be generated. So, what is the address map for port A? A 7, A 6, A 5, A 4, A 3, A 2, A 1 and A naught. So, what is 00H? Is equivalent to 0 0 0 0 0 0 0 0.

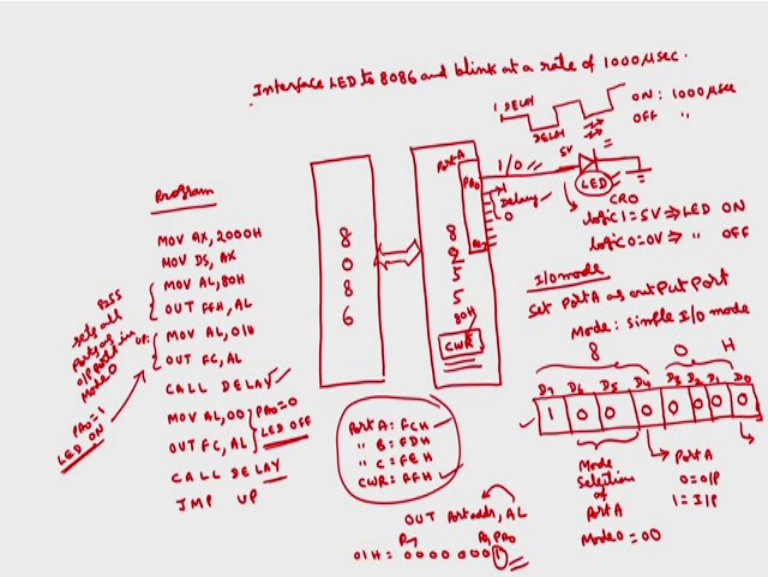
So what is 01H? 0 0 0 0 0 0 0 1, 02H 0 0 0 0 0 0 1 0 H, 03H 0 0 0 0 0 0 1 1 H. You see here for all these combinations this A 2 to A 7 all are 0s whereas, A 1 A naught changes this is for port A 0 0, port B 0 1, port C 1 0, control word 1 1 anyhow I have connected this A 1, A naught to A 1, A naught. So, that will select the ports and the control word register accordingly.

Now, using this address lines of this 8086 so, if I want to interface in this address. So, what should be the hardware required to select these chip select of 8255 is. So, whenever these all are 0s, then the output should be 0, if anyone of the input is 1 output should be 1. So, what is the gate? So, you take this A 2 to A 7 connect to a gate A 2, A 3, A 4, A 5, A 6, A 7 you choose a gate such that whenever all the inputs A 2 to A 7 are 0 output should be 0, if any one of the input is 1 output should be 1 ok. So, what is that gate? OR gate right.

So for OR gate if all the inputs are 0 output is 0, if anyone of the input is 1 output is 1 ok. So, this is how we can connect 8255 in a given locations or for a given connections we can find out what are the address of port A, port B, port C and control word register ok. So, this is how we can connect 8086 and 8255. Now, assuming these connections then how to connect a LED? First of all what is the need of this if I want to connect a LED here, why do not to connect LED directly to 8086 or input device such as keyboard or 7 segment display.

So, these devices why do not you connect directly to the 8086 without using 8255; without using this 8255? The reason is this I mean input output devices operates on some electrical characteristics whereas 8086 will operate with some other characteristics, because of that there will be some mismatch ok. So, in order to match the electrical characteristics of 8086 with input output devices we need interfacing circuitry such as 8255 ok.

(Refer Slide Time: 23:43)



Now, let us take a simple example interface LED Light Emitting Diode to 8086 and blink at a rate of some 1000 microseconds.

So, this 8086 or 8255 connections I have already discussed this is 8086. So, I am going to program this LED through 8086 only, but we are going to connect interfacing device in between. So, the connection between this I am not showing how to select port A port B port C and all that I have already shown in the earlier slides ok. Let us assume that the connections are such that the port A address is you can choose any addresses, but consecutive addresses you take same this first example FD, port B FE, port C f sorry this is FCDEH, FC FD, FCH, FDH, FEH and control word register FFH.

Then in order to connect a LED this is an LED Light Emitting Diode, if the anode operates at 5 volts, cathode is grounded this will emit the light this is the symbol of ordinary diode if I add this light symbol then this will be LED. So, I am going to ground the cathode, if I give 5 volts here LED will glow 5 volts means logic 1. If I give 5 volts or logic 1 here implies LED will glow, if I give logic 0 here which is equal to 0 volts LED will be off.

Then in order to connect this we know that 8255 is having 3 ports port A, port B, port C and inside this we have of course, control word register. So, I need only one port bit ok. So, you can connect to any port suppose if I want to connect these two PA 0 of port A, I have this entire port A this is port A. So, in that we have PA 0 to PA 1, PA 2, PA 3, PA

4, PA 5, PA 6, PA 7, but only one port pin a will be enough. So, I am connected to PA 0 ok, I am not using port B and port C I have not shown here, you can use pin of port B as well as port C also so, for these connections.

So, I want to blink this LED at a rate of 1000 microseconds, it should be ON for 1000 microseconds; ON for 1000 microseconds, then OFF for 1000 microseconds and we have to repeat this same thing ok. For that very first step in order to connect any input output devices you have to set this control word register to meet the requirements. So, what is the requirement here? Out of the 3 ports we are using port A only, but in port A also I am using a single bit. So, you can use bit set reset mode also and you want to use IO mode. So, you have to set this port A as output port.

And which mode ? we will take simple Mode 0. So, I will discuss the different modes what are the things in different modes. So, I will take simple mode 0 which is called as simple IO mode any port can be programmed as either input or output mode. And what about port B, port C? I do not care port B port C can be either input ports or output ports and port B can operate in either mode 0 or mode 1 okay. I am not interested in the those ports because I am not using those ports ok. Then what will be control word register? I am using this in IO mode. So, this D 7 pin D 6, D 5, D 4, D 3, D 2, D 1 and D naught.

So this D 7 should be 1 for IO mode if it is a bit set reset mode this will be 0 ok. So, in this D naught this stands for port C lower; this stands for port C lower as I have discussed in the earlier slides. So, I am not using port C lower. So, I can program this as any other input mode or output mode. So, for the sake of simplicity I am representing this as output mode. This is port B; port B also I am not using. So, I can use either 0 or 1. So, I am using 0 means port B also will acts as output port you see mode selection I want to operate in mode 0.

And this is port C upper I am going to program as output mode and whatever D 4 bit, this is for port A because I am using only port A this is important I want I have to set this as 0 for output mode, 1 for input mode, because I am using this port A as output port I have to set this as 0. So, the remaining last 4 bits can be either 0 or 1 whereas, this should be 0 and these 2 bits are for mode selection of port A.

So, I am using mode 0 means this will be 0 0. So, I am using 0 0. So, what is the control word now? This is 8 and this is 0, 8 0 hexadecimal. So, if I place this 8 0 H into this

control word register according to this control word format this will set all the ports as output ports in mode 0 and in IO mode ok. So, this is the first step we have to find out what is the control word register required, then you can write the program ok.

So, in order to write the program; so I have discussed two instructions called IN and OUT instructions, this IN and OUT instructions are used for reading and writing the data from the ports ok. So, first of all you take this MOV data segment register with some address the same as that of the other instruction that I have written MOV data segment AX.

Now, for IN and OUT instructions so, you have to output the data. So, you take some data into AL register. So, we know that there are 2 instructions IN and OUT, OUT instruction is nothing, but OUT port address comma AL. This will output the data from the AL on to the port address that is specified in the instruction.

Similarly, you can have if it is a 16 bit data we can have OUT port address comma AX, but here we are assuming only 8 bit of the port addresses. So, first I have to set the control word register I have to load with 80H ok. So, whatever the data that I want to output first you have take into AL register by default. Then OUT AL to which port what is the port address of control word register, I have assumed that control word register is FF so, this should be FFH. So, with these 2 instruction the 8255 sets all the ports as output ports in mode 0, but I have used only port A. So, anyhow port A also will be used as output port.

Now, what we have to do? Now, we have to send a logic 1 on this. So, that LED will glows then you call it delay then again you send logic 0 here send a logic 1, call delay, then send logic 0, then you repeat the loop, again you send the logic 1, delay, logic 0. So, continually this LED will blink at a rate which is set by the delay program ok. First I want to set one logic 1 and PA naught. So, what do you have to take, MOV AL comma 0 1. So, what will be 8 bit of this 01H? 01H is nothing, but 0 0 0 0 0 0 0 1 ok.

If I write OUT, what is the address of the port A? Port A address is FCH I want to output this data on to FCH port A. So, after these two instructions what happens? This port A this is PA naught, this is PA 1, this one PA 7. So, this PA naught becomes 1; PA naught becomes 1. So, PA naught is connected to LED so, LED will glows, with this two instructions we can ON the LED, then I have to call the delay program.

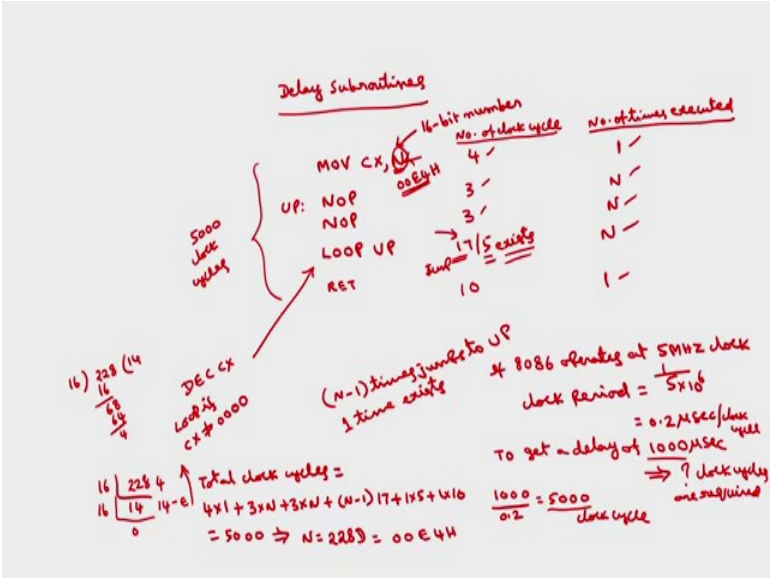
I will discuss how to write the delay program after this program ok. So, this delay the rate at which you want to blink this LED that you can set that delay using the sub routine. Then after this delay what you have to do? You have to make LED off means PA naught should be 0. So, for that you take MOV AL comma 0 0 then OUT onto the same port A.

So, with these two instructions what happens? PA naught of course, PA 1 to PA 7 also becomes 0, but I am not interested because I have connected the PA naught to the LED. So, PA naught becomes 0 which implies LED is OFF. So, one cycle is over ON, then CALL DELAY OFF I have to repeat the same thing you call the same delay again then unconditional jump to UP.

So, this UP will be at this instruction. So, that again AL will be taken as 0 1 it will be outputted onto port A. So, LED will be ON after some delay, LED will be OFF after some delay, again because this will go here LED will be ON, OFF. This will be ON and OFF with a rate of the delay that is provided by this sub routine ok. This way we can connect a simple LED to the microprocessor 8086 through 8255 and you can blink the LED ok.

Now how to find out this delay? Even if I want to I mean generate a square wave also the same program instead of connecting this LED if I connect to a CRO because here we are getting 1 for some time this is 1 logic 1 some delay. Then you are making logic 0 the same delay and then you are repeating this 1 and 0 1 and 0 which is nothing but square wave ok. So, instead of this LED if I replace with CRO we can observe a square wave on the CRO and what is the frequency of this CRO that will be decided by this DELAY sub routine ok. Then how to write the delay program? Ok.

(Refer Slide Time: 38:04)



.

So, you will load some CX with some number MOV CX comma some number this is 16 bit number, you perform some NO operation. So, basically it does not perform any operation as I have told while discussing about the instruction set this simply provide some delay. Then LOOP UP; UP will be here after that of course, the return instruction will be there and the last instruction is return.

So, in case of MOV instruction the number of cycles will be 3; number of cycle clock cycles is 3. The 8086 is designed such that this MOV instruction requires 4 clock cycles sorry 4 clock cycles and NOP each requires 3 clock cycles this LOOP UP will take 17 slash 5 clock cycles ok.

So, written also it will take some clock cycles say this is some 10. So, if I operate this way 8086 can operate at 3 different frequencies, if I use that 8086 operates at 5 mega Hertz clock. So, what is the period of each clock period? Is equal to 1 by 5 into 10 to the power of 6 which is nothing, but 0.2 microseconds. To get a delay of 1000 microseconds, how many clock cycles are required? How many?

So, this is 0.2 microseconds per clock cycle ok. So, for 1000 microseconds 1000 divided by 0.2 which is nothing but 5000 clock cycles are required. If this programs gives 5000 clock cycles, then this will give a delay of 1000 microseconds, this 1000 microseconds delay is required in the program that we have discussed in the previous slide ok. So, this is how many times this particular instruction will be executed? This is number of clock

cycles number of times the instruction is executed, because this MOV instruction is not there in the loop. So, in the loop we know that so, this loop nothing, but it will decrement the CX decrement the CX register it will a loop if CX is not equal to 0 ok.

So, because this is outside the loop this will be executed only once and this how many times this will be executed because N is loaded here. So, this will be executed N times this also will be N times, this also will be N time, this is also only 1 time ok. So, here this 17 by 5 is whenever it jumps to the loop 17 cycles are required, whenever it exists the loop exists, then 5 clock cycles are required. So, in the loop and jump 17 cycles are required for exist 5 cycles are required. Here how many times this will jump? N minus 1 times it will jump to up and 1 time exist ok.

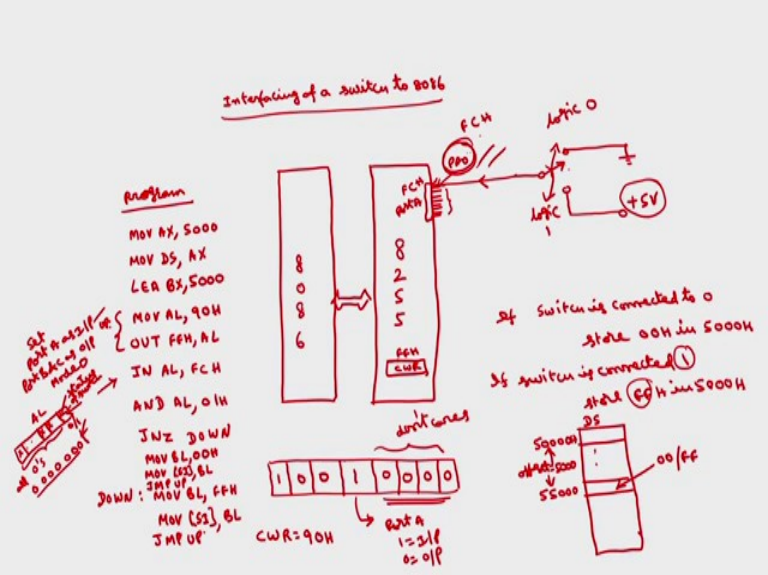
So, then what will be the total delay of this program is number of clock cycles into number of times plus this number of clock cycles into number of times, number of clock cycles into number of times it is like that you have to add. So, here total clock cycles is equal to 4 into 1 plus 3 into N plus 3 into N because this these 3 instructions in fact, will be executed N times plus out of this N times N minus 1 times this will be jump to UP. So, N minus 1 times 17 or you can write 17 into N minus 1 plus 1 time 1 into 5 plus 1 time 10 if the return is 10.

So, this should be equal to you see the total number of clock cycles if this is equal to 5000; this is equal to 5000; then it will probably delay off 1000 microseconds ok. If I solve this for N so, we will get something like N is equal to some 228 value decimal. So, the corresponding hexadecimal you have to place here. So, what is hexadecimal equivalent of 228? That value 228 is 16 1s are 16 68 16 4s are 64 4. So, these 228 if you want to convert into hexadecimal 16 14s are 4 is the reminder 16 0s 14 is the reminder. So, 14 is nothing, but E and you have to read from bottom to top. So, this is equal to 00E4H.

We simply replace this N with 00E4H this will give you the delay of approximately 1000 microseconds. If I change this value the delay mount also will be changes so, like that we can generate any delays ok. So, this is about simple I mean interfacing of the output device which is such as LED or CRO. So, we can generate either square wave of any desired frequency using 8086 and 8255, similarly you can blink the LED at any desired rate, similarly you can connect a switch also input device. So, I have discussing about the

output device now you want to discuss or I want I mean connect input device to the 8086 interfacing of a switch to 8086.

(Refer Slide Time: 46:11)



So, we cannot connect this switch or LED directly to the 8086 because of different electrical characteristics. So, we need 8255 in between. So, I am not going to show this 8086 and 8255 connections because that I have already discussed. So, this is 8086 and 8255 will be connected using the interfacing circuitry that I have discussed it in the earlier slides. So, the connections are such that I will take the same 8255 control word register.

Let the address of this one is FFH and port A is this is port A, this is PA naught, this port A address is FCH port A and port B port C will be having FD and FEH, here also because only single switch is there I need only one port ok. So, I am going to connect this port A this will be programmed as input port I have a switch. So, the switch have two positions, in one position this will be connected to ground, in another position it will be connected to VCC ok. This is a diagram switch if I connect to this this is logic 0 if I connect this switch to this this is logic 1 ok. So, I want to read the status of this switch I have connected this as PA 0 I am connected this as input port.

Let us write a program this is interfacing simple interfacing let us write a program that if switch is connected to logic 0. So, you do some operation if switch is connected to logic 1 you will do some operation ok. Let us assume that so if switch is connected to logic 0

store 00H in the data segment pointed by 5000H offset. If switch is connected to logic 1 store FFH and the same offset 5000H that is in the data segment this is data segment say 50000 H is the starting address the offset of 5000 will becomes this is 55000 offset is 5000.

Here I want to store either 00 or FF depends upon whether switch is connected to logic 0 or logic 1 ok. So, what will be programmed for this now? So if I take the control word I am using IO or CD mode only. So, it must be bit should be 1 and what about the LSB? I am not using port C upper or port C lower. So, simply this is a do not care. So, I will take for the sake of simplicity 0, this is also do not care because I am not using port B at all. So, I will just for the sake of simplicity I will take 0, this is for mode selection of port B. So, this also I want to operate in mode 0.

These 3 are in fact, do not cares, then port C upper this also I want to program as output mode this actually in fact are do not cares because of, we are not using these ports for the sake of simplicity I have taken as 0s, then what is important is here the remaining 3 bits. This is for port A, 1 means input port, 0 means output port, because you are using this port A as input port. So, this should be 1 and if I want to operate this in mode 0 this will be 00. So, what is the control word register? This is 1 0 0 0 is 90H.

So, initially you have to load this 90H onto if I write the program. So, the first two instructions are say MOV AX comma 5000 starting of the code segment, then MOV DS comma AX and the offset is 5000 that you can point to you can take LEA BX also you can take SA also you can take say BX 5000 because offset is also 5000. Then MOV AL comma 90H first I have to set the control word register this I have to output to control word register whose address is FFH comma AL.

So, with these two instructions it will set port A as input port, port B port C as output port, in mode 0, but we are not using port B and port C. So, this will set port A as input port, port B and port C as output ports in our mode 0 all will be operated in mode 0 ok.

Here we are interested in only this port A port only, then I have to read these to read this what is the instruction? In AL comma from which port this a switch is connected to port A whose address is FCH. So, FCH so, here whatever the data 8 bit data so, this remaining 7 bits are do not cares we are not using anything, this only the PA 0 bit is

having the original information of this switch whether this can be logic 0 or logic 1 depends upon whether the switch is connected to this position or this position ok.

So, here in order to check the last bit without affecting the other bits other bits can be either 0 or 1 we do not know ok. So, with this the status of the switch also will be present in the last bit of AL this is the status of switch. And this is 0 means switch is connected to 0, this is 1 means switch is connected to 1, the reminding all the 7 bits are do not cares. So, I will just make this one as all 0s for that I am ending END AL comma 0 1.

So this data whatever is the data you are ending with 0 0 0 0 0 0 0 1. So, what happens? This bit remains same whereas, the remaining all bit becomes 0s ok, after this ending operation if the resulting AL is 0 means the switch position is 0, here the resulting AL is non-zero because the remaining 7 bits are 0s means this switch is connected to logic 1 ok. After ending this one jump not carry no carry means the result of the operation is or you can write the jump not 0 also because the result can be either 0 or non-zero.

Jump Not 0; not 0 I mean the sense what happens? AL contents are not 0 means AL contents are 1 only I have 2 possibilities either 0 or 1, AL content is one means switch is connected to this plus 5 volts. So, if switch is connected to plus 5 volts I want to store FF into 5000H okay DOWN otherwise. So, I will write here DOWN I have to store FFH into 5000H for that you take some MOV BL comma FF and MOV contents of SI comma BL.

So that FF will be stored in the location whose offset is 5000 H if this is true, if this is false means I have to store 00, so for that you how to take MOV BL comma 00, then MOV SI comma BL. So, in any case we have to halt either the program can be halted here or program can be halted here, if I want to check only once. If I want to check continuously then instead of this halt you can write unconditional jump you can write unconditional jump to UP.

And also you can write unconditional jump to UP, this UP will be here. So, that it will continuously check the status of switch ok. This is how you can connect a simple switch which is called input device to the microprocessor a simple output device such as LED to the microprocessor this will be possible through 8255 the programmable peripheral interface ok. Later we will move onto the some complicated input output devices such as has 7 segment display, keyboard, these things we will discuss in the next class.

Thank you.