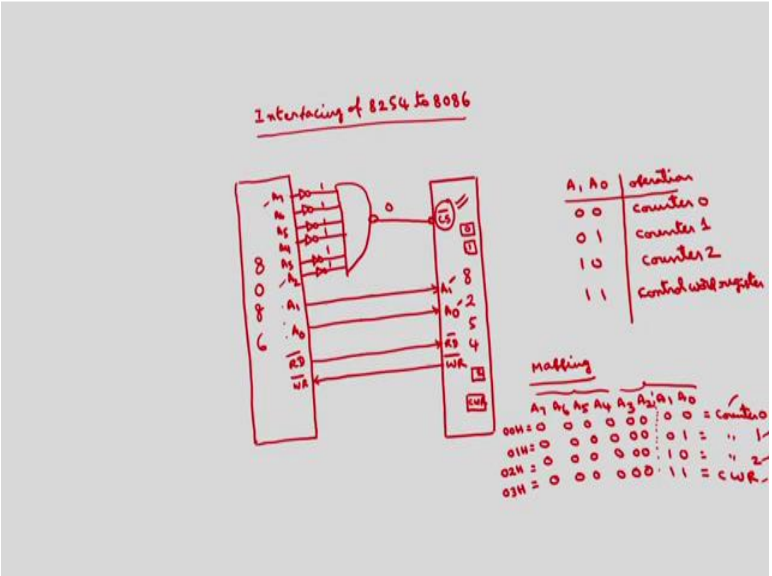
Microprocessors and Interfacing
Prof. Shaik Rafi Ahmed
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture – 26
Multiplexed 7-segment Display Interface

In the last class we have discussed about the architecture of 8254 and the control word format of 8254. So, today we will discuss about how to interface 8254 to 8086. And then we will discuss some of the modes of 8254.

(Refer Slide Time: 00:49)



Interfacing of 8254 to 8086. So, this 8254 also will be interfaced to 8086, in a similar manner to that of 8255. So, we know that here also in 8254, A1, A0 is going to decide the counters or control word register. 00 means counter 0; 01 means counter 1; 10 means counter 2; 11 is control register.

So, if I take this 8086 and 8254. This is 8254, this is 8086. So, this A0, A1 of both the devices will be connected together. A0 of 8086, A0 of 8254 will be connected together, A1 to A1. Similarly, we have read bar write bar, then you have to generate the chip select of this particular interface for IC. I will use only these addresses of these 3 counters inside these there are 3 counters; counter 0, counter 0, counter 1, counter 2 and control word register. I will use 8 bit addresses of these counters, for that I have to use

A7 to A0 of 8086. Out of that A1 A0 I have connected to A1, A0 of 8254, then I will use A2 to A7.

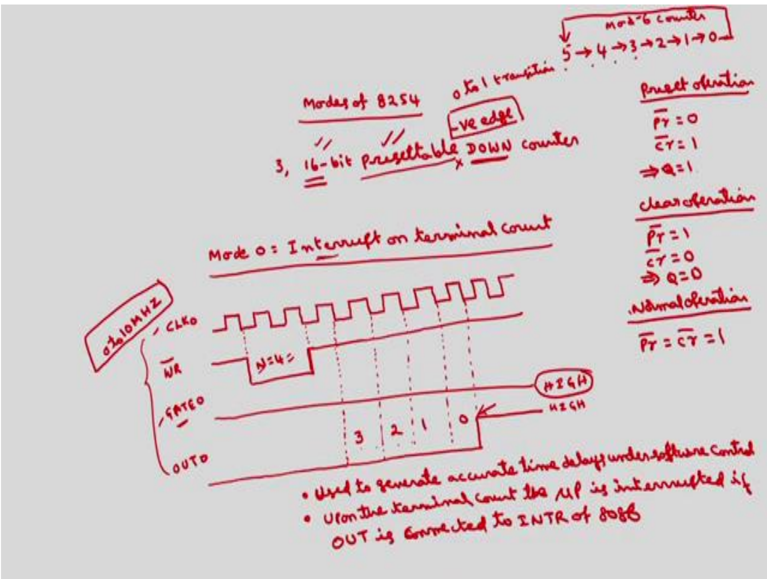
This is A2, A3, A4, A5, A6, A7. If I connect through a NAND gate and if I connect these to chip select bar of this one. Now, if I can connect directly this A7 to A2 to this inputs of the NAND gate. Then the address of counter 0, counter 1, counter 2, control register will be FC, FD, FE, FFH which is similar to the 8255 addressing. To have a different feeling I am assuming here some inverters are connected. Now, what will be the address of counter 0, counter 1, counter 2 and control word register. This is mapping A7, A6, A5, A4, A3, A2, A1, A0.

So, these 2 pin will be connected to A1, A0 of the corresponding microprocessor pins. This is 00 means; this will select counter 0, 01 means; counter 1, 10 means; counter 2, 11 means; control word register. Now, this will be A1 A0, first of all you have to select the chip after that only based on this A1, A0 we can select counter 0, counter 1, counter 2 or control word register.

So, how to select the chip for that CS bar should be equal to 0. For what values of A7 to A2 this output will be 0? If all are 0s then, only output is 0, 0 0 0 0 0 0, this should be all 0s whether you want to select counter 0, counter 1, counter 2 or control word register. So, you have to fix this values to 0 0 0, 0 0 0, 0 0 0, 0 0 0, 0 0 0, 0 0 0. Because these are 0s output of inverter will be ones if all the inputs of NAND gate are 1 then only output is 0, if any one of the input is 0 output is 1.

So, what will be the address of this counter 0? This is these 4 bits will be 0, these 4 bits will be 0, this is 00 H. So, this will be 01 H, this will be 02 H, this will be 03 H. So, this is about the interfacing of 8254 to 8086. Now, we will discuss about the modes of 8254.

(Refer Slide Time: 07:23)



So, this 8254 can operate in 5 modes. So, before going for these modes; we know that that this 8254 is having 3 counters, 3 16 bit presetable, down counters which operates on negative edge, they are also called as presetable negative edge down counters.

So, what is meant by presetable? Presetable means; so, we can preset the flip flops. Basically a counter will be having flip flops. So, each flip flop can have two operations preset and clear operations ok. So, in order to preset there will be 2 signals called preset bar and clear bar for any counter. If this preset bar is 0, clear bar is 1, then implies Q is equal to 1, this is called preset operation. For a clear operation preset bar should be 1 and clear bar should be 0 implies Q is equal to 0, Q is equal to 1 is called preset operation, Q is equal to 0 is called clear operation. For normal operation; we will keep both the clear and preset values as 1.

So, we can preset these counters and they are down counters. Suppose, if I want to count a 5, initially you have to load 5, then 5 will after first clock signal it will goes to 4, then 4 to 3, 3 to 2, 2 to 1, 1 to 0, then 0 to it will go to 5, this is the timing diagram or state diagram of mode 5 down counter. Then this is all 16 bit counters, we have 3 such which is counter 0, counter 1, counter 2, negative edge. So, this counting will takes place at the negative edge of the clock. Negative edge means which is a transition from 0 to 1, negative edge means transition from 0 to 1. So, if I take mode 0, there are 6 modes of the operation; mode 0 is interrupt on terminal count. So, we know that this 8254 is having different signals like clock, suppose if I want to use counter 0.

So, clock 0 you have to use clock 0. So, I am taking this square wave this clock of 8255 can operate between 0 to DC to 10 MegaHertz. Depends upon the application we can operate this clock with frequencies ranging from DC to 10 MegaHertz. This is the clock of course, I am plotting this ideal clock practical clock will have some slopes at the transitions, this is clock 0 of counter 0. Then we have our write bar signal; 8254 is having write bar signal.

So, this write bar you have to make low in order to initialize the counter. So, after making this we can load any count value say N is equal to 4, I am loading N as 4. This N can be any 16 bit number ok. Because these are all 16 bit counters, so we can feed from 0 0 0 0 H to F F F F H. For the sake of simplicity I have taken just only N is equal to 4 for explanation purpose. And gate of this one in mode 0 has to be kept always high. Gate 0 has to be made high.

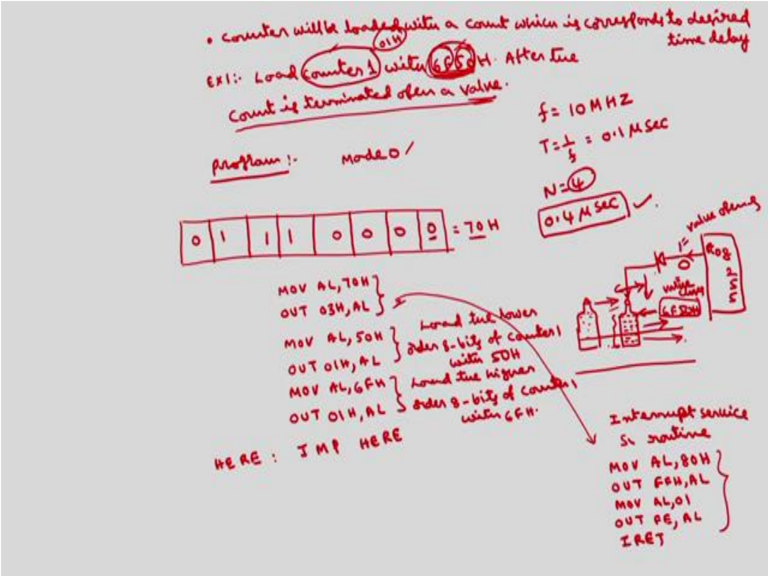
Here I am using only counter 0 signals we can explain the same thing for counter 1 as well as counter 2 also. So, gate signal has to be made high. Then once this write bar goes to high; so the next immediate negative edge signal negative edge transition. So, initially OUT 0, initially, OUT will be 0 only. Now, here onwards when N is equal to 4 and write bar goes from low to high.

So, after becoming high; the next negative edge of the clock this one. Here it will start counting the this counter. So, as I have told this counter is a down counter. So, this N is equal to 4 becomes 3, this is 4 over this clock signal over 4, then 3, 2, 1. I mean, this becomes 3 because 0 to 3 will be modulus 4. So, normally this is mod 6 counter means 0 to 5, mod 6 counter, so this will be 3, 2, 1, 0. After 4 clock signals, till 4 clock signal this will remain low only. Once the count is over, this OUT goes to high. As long as this gate is equal to high so, this will be remains high only.

Once if I make gate is equal to again low, so you have to make a new value you have to load into the counter. N is equal to 4, you have to load into counter 0 here if I take these signals ok. So, where this particular mode 0 will be useful? Ok. So, as the name implies interrupt on terminal count once the count is over, I can interrupt to the microprocessor.

So, basically, this will be used for generate accurate time delays under software control. Because, we see under software control this will produce accurate time delays. Upon the terminal count the microprocessor is interrupted. If OUT is connected to interrupt of 8086 that is why the name interrupt on terminal count. So, I will give some examples, so that this operation will be clear.

(Refer Slide Time: 17:14)



So, example 1. So, what value has to be loaded? The value that is to be loaded will be here. So, depends upon this the value that is to be loaded depends upon the time delay that we want. So, we have to load the appropriate count value. So, this desired delay will be set by count value. Counter will be loaded with, counter will be loaded with the count which is correspond to desired time delay.

However, this time delay will be decided this depends upon the clock. Suppose, if the clock frequency is clock frequency is 10 mega Hertz, the highest frequency and after every 4 clock signals if I want to interrupt, then the OUT signal will be having 4 clock signals means this will be time period will be the time period of this one will be 1 by f 0.1 microsecond.

So, if you want to I mean load the count with N is equal to 4, if N is equal to 4, for every 0.4 microseconds; this microprocessor will be interrupted. So, this depends upon the time that is required. If I want some other time accordingly I have to choose this N value. See here I want to I mean load counter 1 with a value of say 6F50 H ok. So, I want a delay which is corresponding to this count value. After the count is terminated, open a relay or valve. This is something like if in a bottling industry if I place the bottle here and if you want to feed this bottle with some liquid which is controlled by some relay here or some valve.

So, this valve can be on or off, this I am going to control by using 8255, this if I connect to the 8255 of the 8086 kit, if I connect to say PC0. If this is 1, valve will be open. So, liquid will flow into the bottle. If this is 0 valve will be closed so, there will be no liquid into the bottle. So, I want to feed this bottle with some level. So, this level in bottling industry if you want to fill the bottles with some medicine, some syrup.

So, we have some prescribed limit. So, the time taken to fill the bottle up to here you say let us assume that if I feed the counter with 6F50 H, the corresponding time. So, within that corresponding time the bottle will be filled with this. So, for that much amount of time you have to open the valve and then again you have to close then you have move this conveyor belt. So, you have to allow the next bottle to come this position and that also has to be filled with the same amount of the time. Like that we are going to fill the bottles in a sequence.

This is the second bottle, once this moves in this direction we will move this here just below the valve again we will open the valve for the same time which is decided by this count. So, that this also will fill up to this point like that so, we will fill this bottles in a industry. So, I want to open a valve once this end of the count is reached.

So, what is the program for this one? Before writing the program, what will be the control words? I want to use counter 1 in mode 0, because this is on the terminal count we have to interrupt. So, I want to use in mode 0. So, what is the control word register which we have discussed in the last lecture. This is 0 for binary counting we want only binary counting only, so this is 0 these 3 bits will decide the mode. So, because we are operating in mode 0 this 3 will be 0 0 0. So, these 2 will decide RL1 or RL0. So, this is 11 is total 16 bit count first it will take the lower order 8 bits then higher order 8 bits.

So, here because our count is 16 bit. So, I have to load with 11, so that it will first take the lower order 8 bits and then higher order 8 bits of the count. Then we have these 2 signals or select counter for counter 1; this will be 01. 00 means, counter 0, 01 means counter 1, 10 means counter 2, 11 is illegal here.

So, what will be the control word? 70 H. And the throughout this program I will assume the same addresses which is 00 H is counter 0 address, 01 H is counter 1, 02 H is counter 3 and control word register is 03 H. So, because I am using counter 1 whose address is 01 H. First of all we have to feed this 70 H into control word register. MOV AL comma 70 H, OUT, 03 H comma AL.

So, this 03 H is the address of control word register. So, with these two instructions so, counter 1 will be selected in mode 0 for 16 bit count and this is binary counting. Then you have to load this counter with the 16 bit value. First LSB, then MSB into counter 1 who address is 01 H, similar type of the instructions. MOV AL comma the lower order count will be in this 6F50 H, 50 H. Where you have to feed this? Into counter 1 whose address is OUT 01 H comma AL. This will lower the counter 1, lower order 8 bits of counter 1 with 50.

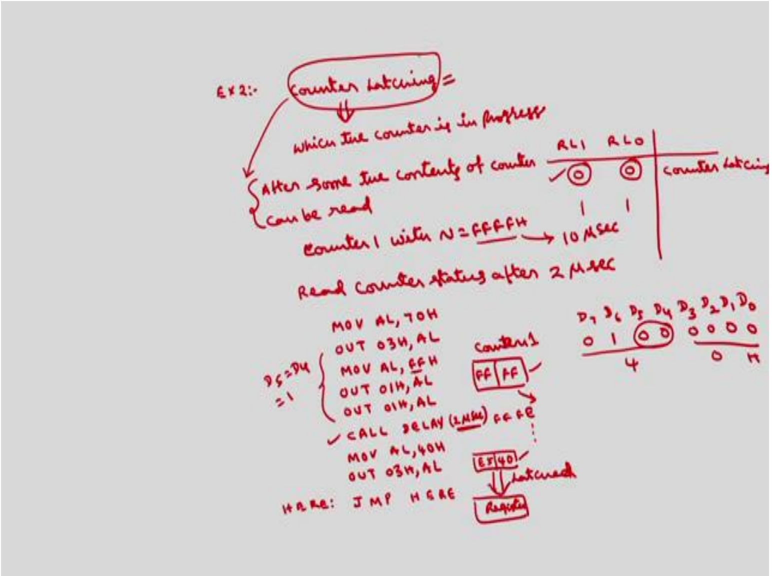
Here higher order 8 bits you have to load with 6F. So, you take MOV AL comma 6F, then OUT 01 H comma AL. This will load the higher order 8 bits of counter 1 with 6F. So, you have loaded the count, so then the operation will starts. Once if we make this gate is equal to high so, after this read bar goes to high. In the next I mean negative edge onwards the count will starts. So, at this point count will end, so here this will be terminated.

So, you have to wait until the count is terminated so, jump here itself. Once that count is terminated the microprocessor control will move from here to interrupt service subroutine. Because, I am going to connect this out signal to the interrupt of 8086. Here, what is the problem that I have to write? So, I want to open the valve say which is connected to PC0. So, what are the instructions MOV AL comma because, I want to I mean program this PC0 as output port. So, we can use 80 H to program all the ports as output ports OUT FFH. If I assume the same addresses FFH comma AL, this will program all the ports as output ports in mode 0, then MOV.

So, if this is 1 means valve opens, 0 means valve closes. So, 1 you have to take AL comma PC0 means 01. So, that the last LSB bit becomes 1 out port C, port C address is FEH, then I return, interrupt return. So, whenever this terminal count is reached this will go on to open the valve again we will come back to the main program.

In the main program automatically again you have to load that value this 6F50. So, this will generate a time which is corresponding to filling of this bottle ok. So, like that we can use this mode 0 operation to control the industrial process. In mode 0 itself I will take another example. I will take another example in mode 0 itself, where the counter latch. This is an example where I will discuss about counter latch.

(Refer Slide Time: 29:38)



So, I have discussed about this RL1 and RL0 of this control word register where we have explained that 00 is corresponding to counter latch. Means; what is this counter latching is while the counter is counting the count, while the counter is in progress. So, I want to I mean read what is the count value in between before it reaches to 0 after some time delay. I want to read what are the contents of the counter. So, that is also possible that particular operation is called counter latching. After some delay the contents of counters can be read.

So, this particular operation is called counter latching. For that; so, initially I will load count same counter 0 or counter 1 with N value of FFFFH, say this is corresponding to a delay of say some 10 microseconds. But I want to read the counter status after 2 microseconds right. So, for that the program will be MOV AL comma 70 H, as you have discussed for 70 H is for programming the counter 1 in mode 0 and 16 bit count value, OUT 03 H comma AL so, control word register. Then we have to load FFFFH, MOV AL comma FFH OUT, 01 H, because this is counter 1, 01 H whose address is 01 H comma

AL. Again you have to write because FF remains same. So, no need to load this the other value into AL. I will write another time out 01 H comma AL. So, that the same AL will be transferred to counter 1 twice. So, FFFF.

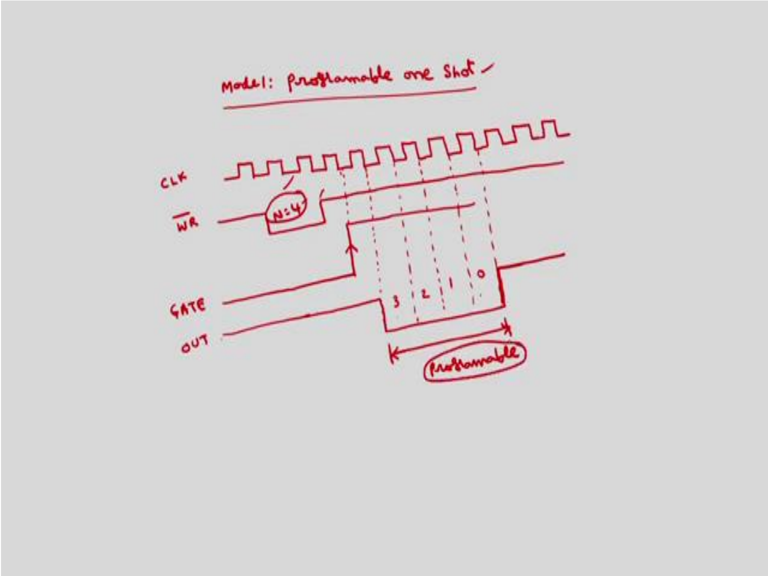
So, the counter 1 will be loaded with; counter 1 will be loaded with FFFF. So, this goes on decrementing the count because this will be a down counter after 1 clock signal this becomes FFFE. So, like that upon every clock signal at negative edge this count is going to be decreased. So, in between I will write a call sub routine call delay. So, this delay I want to write as for 2 microseconds. Because I want to read what are the contents of the counter after 2 microseconds.

So, that is why I am writing this CALL instruction. Then I have to latch this counter for latching of the counter this RL1 and RL0 should be 0s. For reading the 16 bit count RL1 RL0 should be 11. So, what are these bits now? So, we have out of these 8 bits of the control word register D7, D6, D5, D4, D3, D2, D1, D0, this is binary count. So, this remains 0 and this will be 3 bits will be for mode 0 0 0 0 and these 2 bits will be for RL1 comma RL0.

So, for loading this 16 bit value here I have taken these D4 and D5 as 1, but now to latch these 2 should be 0s and because this is counter 1, these 2 bits will be 01. So, now, what will be control word? This is 40 H. For latching operation after this many seconds I want to latch.

So, what is to be loaded into control word register is MOV AL comma 40 H, then OUT 03 H comma AL. Now, I have loaded this control word register with 40 H which corresponds to latching of the counter. So, with these 2 instructions, counter will be latched whatever the contents of this counter at that time, whatever the count it can be a say sum E540 H something after the second. This will be latched onto a resistor; latched onto the register. So, this register contents we can read into the microprocessor, then you have to jump here itself. This is how we can read the contents of the counter while the counting process is in progress. This is another application of this mode 0 which is called counter latching.

(Refer Slide Time: 36:35)



So, the next mode is mode 1, it is called programmable one shot. So, in mode 1; we have clock corresponding to any of the counters, this is same as mode 0. You have to take the clock with some particular frequency which ranges from DC to 10 mega Hertz depends upon the application, this is clock signal. Clock I am not writing clock 0, clock 1 or clock 2. So, it depends upon the counter which we are going to use. Then here also this write bar signal has to be make low to load the count say N is equal to say 4. For explanation purpose I am taking this lower smaller values here otherwise this can be a 16 bit count varies from 0000 H to FFFF H.

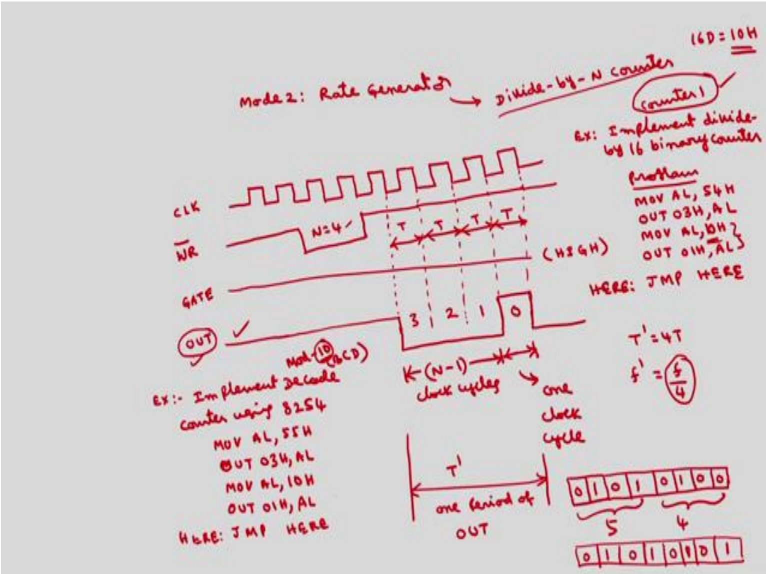
Now, the additional point here is unlike these mode 0 where the gate signal will be always high here the gate signal initially low. After this, this is gate signal; gate signal is initially low after this write becomes high, this will go to low to high. So, at the rising edge of this gate after that whatever this negative edge of the clock. So, this is the negative edge of the clock after this, this is after the negative edge of this.

So, this negative edge of this clock onwards the output, OUT was initially high, OUT signal is initially high, then at this point it starts counting. So, this is 1 count, so 4 becomes 3, after this another clock signal 3 becomes 2, after another clock signal 2 becomes 1, after another clock signal 1 becomes 0. Once a 0 is reached this will go again to the high and it will wait until again the new value of N is loaded.

So, OUT will be initially high, then it will goes to low for this many clock signals 1, 2, 3, 4, then after that again it will go to high. Again this will be reloaded with N is equal to 4. So, if I make this gate as low to high transition again, N will be reloaded with the same value and again it will goes to low for these 4 clock signals again it will go to high, this process will repeats. So, why they named programmable one shot is here this width of the signal is programmable, this is programmable.

If N is equal to 4 this is the width, if N is equal to 5, we will get one more clock signal. So, like that this width is programmable, so that is why the name programmable one shot.

(Refer Slide Time: 40:52)



Next coming for the mode 2; mode 2 is called rate generator. So, this basically will be used for dividing the frequencies, this will acts as divided by N counter. Here also we will be having clock, this is the clock of a particular counter, this is clock. And also you have to make write bar is equal to 0 and you have to load with some value, N value say N is equal to same 4 only. This is write bar and here also you have to make gate signal like mode 0, gate signal will be always high and OUT signal will be initially high. So, after this write bar goes from 0 to 1 the next immediate negative edge onwards. This is negative edge, this is 1.

So, this was initially high, then it goes to the low for N minus 1 clock signals again it goes to high for 1 clock signal. This is 3, 2, 1, 0, this is for N minus 1 clock signals; here

N is 4 means; 3 clock signals, it will go clock cycles this will go low then for 1 clock cycle it will go high. Now, if I compare the frequency of this. The clock; how many clock signals are there in one period of this one? From here to here; this is 1 period of OUT. So, this is 1 period of clock, if I call this one as T, T, T, this if I call as 1 period of this one as some T dash. This is T, another T, another T.

So, what is the relation between T dash and T? T dash is equal to 4 times T ok. So, time period is 4 times means; frequency if I call this F dash as the frequency of out signal. This will be equal to if F is the frequency of a clock signal F by 4. So, the frequency of this OUT signal is divided by 4, that is by the name divided by N counter. This basically used for dividing the frequency by some factor. So, by changing this N value we can obtain any counting value. If I take an example suppose, if I want to generate binary as well as BCD. I want to implement; implement divided by 16 binary counter.

So, what is the program for this one? So, you have to first operate one of the counters in mode 2. Let us assume that I am using counter 1. I want to operate counter 1 in mode 0 and 16 binary means you have to give this 16 number in hexadecimal form. 16 decimal is equal to 10 hexadecimal, you have to give 10, which is 8 bit only. So, for the 8 bit and this is binary counting. So, the last bit will be 0, then mode is mode 2 means 010 these 3 signals will decide the mode then RL1, RL0.

This is only the lower at 8 bits only 8 bit operation so, this will be 01, then select counter, I want to use counter 1 which is 01. So, what it is the control word? This is 5, this is 4, 54 H. So, you have to load 54 H onto the control register whose address is 03 H, 03 H comma AL. So, these 2 instructions will load control register with 54 H, so that the counter 1 will be operated in mode 0 and it will act as only 8 bit counter.

Now, the count has to be loaded. Count will be 10 H, because I want to operate as divide by 16 counter MOV AL comma 10 H, 10 H, OUT, because we are going to use this as a counter 1 whose address is 01 H comma AL.

So, these two instructions will load counter 1 with 10 H. So, that if I connect this OUT to this CRO, this is just you have to jump here itself. If you connect this OUT to this CRO, so you can see that the frequency of this OUT signal will be 1 by 16 times that of the clock frequency. On the I will give the second example where I want to use as BCD counter implement decayed counter. Decayed is nothing but, marked in BCD counter using 8254. This is nothing but mod 10 counter BCD counter.

So, here we have to give directly a 10 value not hexadecimal value like here ok. Because this is BCD count. Changes here will be so, here this 0 becomes 1 because this is BCD count. This becomes 1, so that I can directly give this 10 into the counter then, I mean for these 3 bits these 3 are 010 only because mode is mode 2 and this 1 is also only 8 bit only.

So, this will be 01, 01 because, 8 bit and these 2 also 01 because this is select counter is counter 1. So, this control word will be 55 H. MOV AL comma 55 H, 54 H means binary counting, 55 H is BCD counting. OUT 03 H comma AL, so that they might 8255 will be operated in decimal mode. Then what is to be load 10. MOV AL comma 10 H OUT onto counter 1 whose address is 01 H comma AL. So, this will divide if I connect this OUT signal to CRO here this will divide by 10 which is decayed counter. You have to wait. So, that continuously this operation will repeats jump here itself.

So, this is about this mode 2 operation which is a rate generator basically we can divide the frequency by any factor. So, this type of the frequency division will be useful in even microprocessor also. Microprocessor will be operating at higher frequencies and if I want a peripheral device which can operate at lower frequencies, we can directly take the microprocessor clock and we can pass through the, this divided by N counters to generate the lower frequencies. So, the remaining 3 modes we will discuss in the next class.

Thank you.