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

Lecture - 31
8051 Microcontroller (Contd.)

8051 microcontroller has got a number of timers inside it, and those timers can be also used for counting purposes. So, when it is used as timer; so it is driven by the clock that we have in the microcontroller, and if you are counting some outside event. So, you can use it in counter mode, and in that case the events occurring in the outside world will be giving as pulses to the microcontroller, and the microcontroller will count the number of such events.

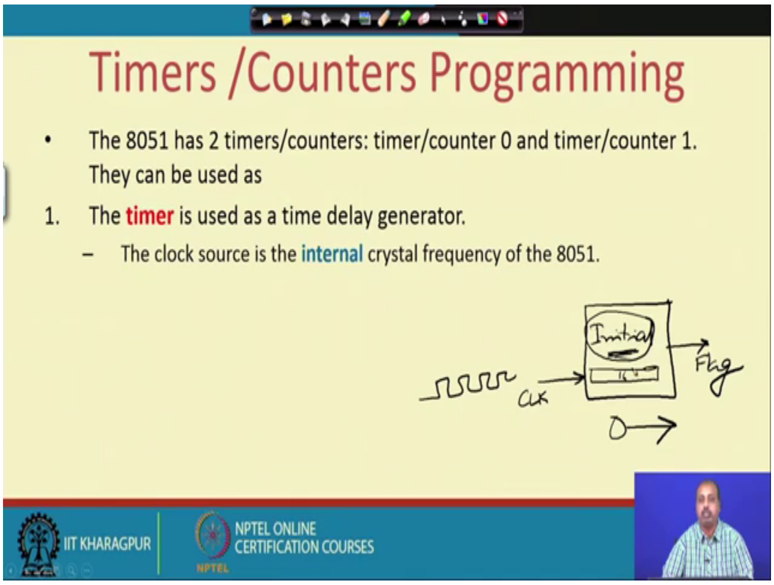
So, we will see that this is very much useful particularly when you are designing embedded systems. So, it is also the we need to interact with the environment, and the second important thing is that the delay part that we have seen so far, the delay routines they have got they are they are soft delay that we have seen.

in the sense that we are using some software instructions, initializing some registers to some value, and then decrement jump not 0 type of instructions, are used to see that it will introduce some amount of delay, but as you can understand that you cannot produce any arbitrary delay by those by those instructions, because the instructions have got a fixed number of machine cycles that are needed, and each machine cycle has got a delay. So, when you are using those instructions.

So, you can get only some of the delay values that are possible, whereas in if you need a very precise delay like suppose you are operating a robotic hand. So, this hand has to move in a particular sequence and that delays are fixed, or you are doing say a traffic light controller system, then the lights should be turned on and off for some specific period of time.

So, those values may not match exactly with the soft delay values that can be produced using these 8051 instructions. So, in those cases we need to use this precise timers and this. So, we will see that this timers there the precisions are very high.

(Refer Slide Time: 02:15)



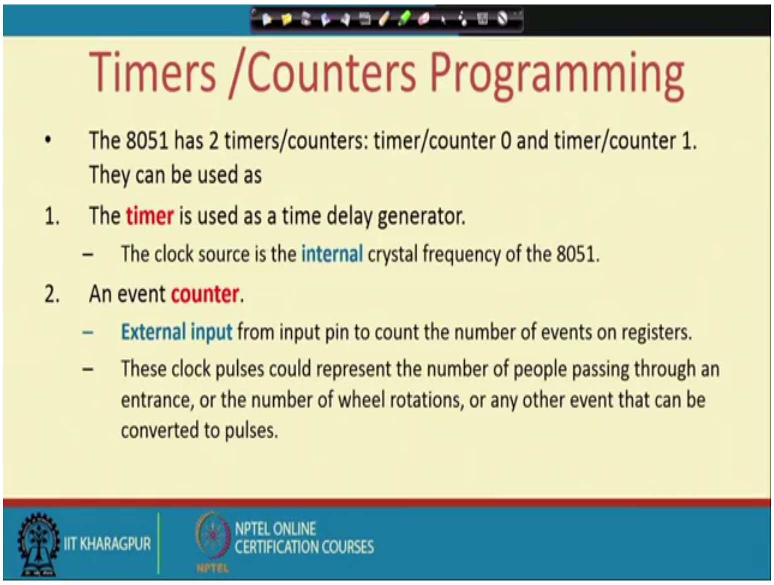
So, the as far as 8051 is concerned. So, they it has got 2 timer counters. So, the timer counter 0 and timer counter 1 they can be used as either timer to produce some delay time delay, in that case the source of the timer is clock source is the internal crystal frequency like whenever we have got a timer module or a counter module. So, this is basically if this is the module. So, it is driven there are certain things like first thing is, that it has to be operated with some clock signal.

So, it will be operating on some clock and there will be some initial value loaded into the timer. So, this initial value will be loaded, and then this when this clock pulses are coming. So, based on this clock pulses this; so, initial value that we have. So, it will go on decrementing, and when it underflows or so if it is a decrementing timer; so, when the value underflow that is when the value becomes 0, then when it is trying to reduce from going down from 0 onwards.

So, one flag will be generated which is overflow or underflow flag, which is basically the timeout flag. Now similarly you can also have the concept of upcounting or uptiming, in the sense that we initialize the value with some initial value, and from that point onwards it starts going upwards till some maximum value

So, that maximum value depends on the register that you have here to hold the timer or the counter value, if that value is a 16 bit. So, it can go up to 65535, and after that there will be an overflow, and the when that underflow occurs then this flag will be set to say that the timer has overflown. So, essentially you get the time value. So, we will see that very precise time values can be generated by doing this think. So, we can use this timer as it the time delay generator, and the clock source is the internal clock frequency of 8051 or it can be used as a counter.

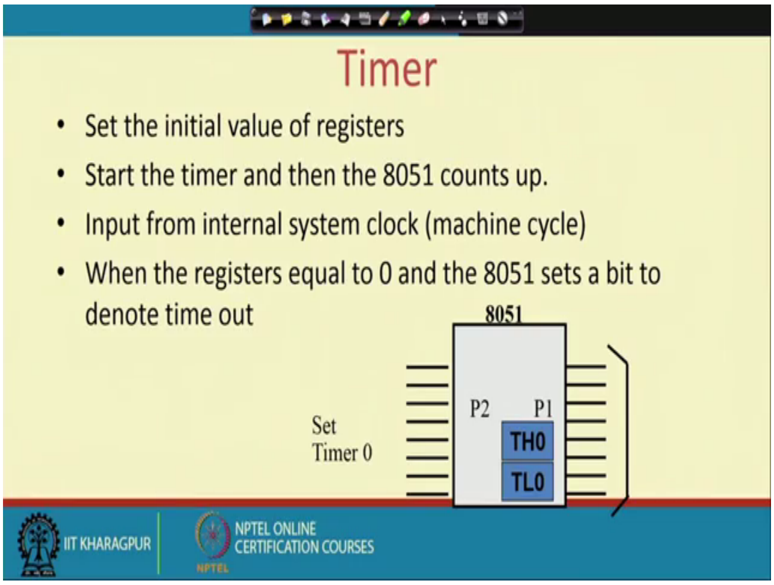
(Refer Slide Time: 04:25)



So, in that case some external input from an input pin, you will be used and that we will count the number of events that have occurred in that pin from in the outside world. So, and that count value will be kept in the register; now this clock pulses that we have; so, this maybe this may represent many things like, say on a conveyor belt some items are passing. So, you can just you can note how many items has passed, or if you have installed a system at the entry of a room in the door like as persons are entering and exiting; so, this values may be updated.

So, that way we can have many such examples. So, number of wheel rotations any event that can generate some pulse. So, the essential requirement is that whatever event we want to count. So, it should ultimately it should be converted into an electrical signal in the form of pulse. So, each event should correspond to a pulse. So, if we can do that then, we can use it for this counting purpose.

(Refer Slide Time: 05:37)

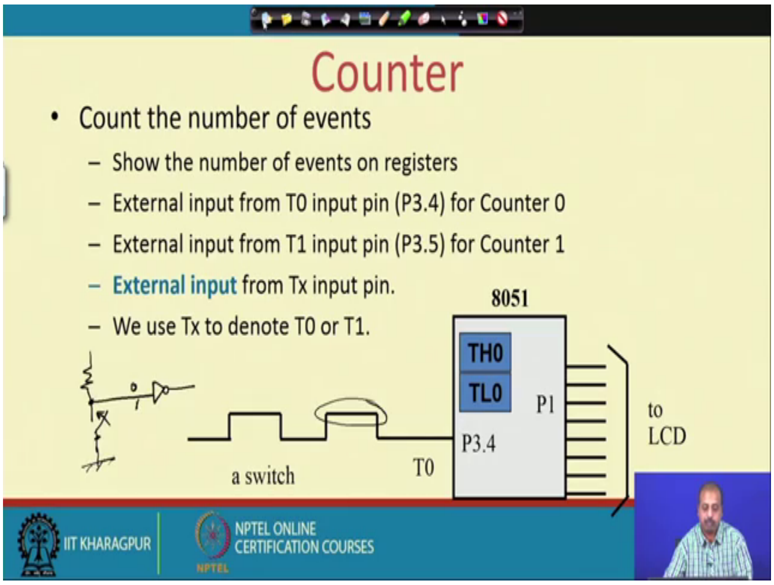


So, we will first look into the timer operation. So, timer it is a set to some initial value of registers. So, set the initial value for the registers; so, this TH 0 and TL 0. So, these are the 2 registers that are used for this timer 0. So, it is time count high and the time count or the time value high, and time value low start the timer. So, then we there some way by which you can start the timer and 8051 will count up.

So, previously we were talking about countdown or count up type of timers; so, in 8051. So, this is a count up type of timer. So, it will start upcounting from the value that you have noted in this TH 0 TL 0 register pair, and now on based on this input from internal system clock. So, it will be updating the value of those registers, and this when these registers will equal to 0 so that means, there is an overflow.

So, depending upon this size of this TH 0 TL 0 registers. So, this register will overflow after some time. So, after going the maximum value it will come down come to come back to 0, at that time it will set a bit to denote that there is a timeout. So, this is the generic operation of the timer.

(Refer Slide Time: 06:58)



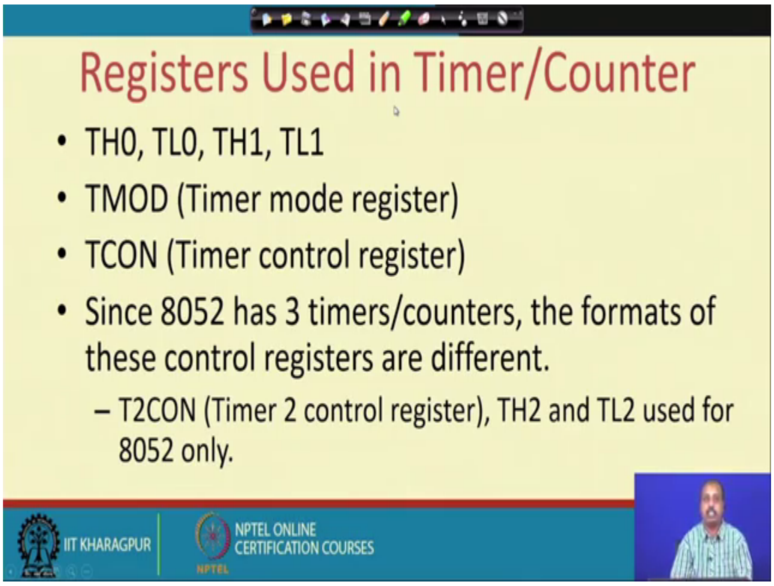
On the other hand, when you are looking it as a counter in that case this 8051. So, this pin number P3.4 and the pin T0. So, this is a multiplexed operation. So, when we discussed about the port 3, we said that bit number 4 is actually multiplexed with T 0. So, T 0 is an external input it is basically for timer 0 and this is for the events that are occurring.

So, if we have some sort of switch here, the connected to the T 0 such that when whenever we close the switch. So, it generates the signal value becomes high, and when we open the switch it becomes low. So, basically; so, in this way we can generate this pulse, a very simple way to generate this switch maybe like this. So, we have got some register and then we have got say switch here, and then this point is say grounded, and if you if we measure the voltage if we take the voltage at this point then, whenever this switch is closed you can understand that the value will become 0 and whenever the switch is open the value will become 1.

So, whenever this; so, it is just the reverse; so, of this operation, so here if we want that whenever it is 1. So, in that case whenever it is whenever the switch is close the value should be one. So, what you can do? You can put an inverter and then feed it to the T 0 line. So, that way we can do. So, basically, we can count the number of times this switch has been closed. So, shows the number of events on register. So, this counter will this now this TH 0 TL 0 pair. So, it will hold the number of events that have occurred and the value will be stored in the register, for 2 timers timer 0 and timer 1 or in that way counter 0 and counter one, we have got 2 pins T 0 and T 1 that correspond to the this external events.

So, T 0 is for counter 0 and T 1 is for counter one. So, correspondingly they are multiplexed with port 3s bit number 4 and bit number 5. This external input comes in the there the T x input pin. So, T x equal to 0 or x equal to 1 either of them; so, we can have something like this. So, you are just having this value, and this is available in this register pair and you can possibly connect to one LCD display here; so, there. So, that this whatever be the count value, that is available on the LCD. So, that way we can have a system that can display the number of events that have occurred.

(Refer Slide Time: 09:52)



Now what are the registers that are going to use for this timer counter. So, TH 0 TL 0 TH 1 and TL 1. So, these are used for holding the time timer values or the counter values. So, TH 0 TL 0 for timer 0, and TH 1 TL 1 for timer 1, there is one more registered TMOD, which is timer mode register.

So, this will tell this timers will see that they can be operated in various different modes, and this TMOD is basically holding the mod value. So, we can set these timers to some particular mode before starting it. So, that it will operate accordingly then, there is a TCON register, which is a timer control register. So, this is also useful and we will see that some part of it is multiplexed with interrupts. So, it will control the timer operation and 8 0 5 2, is an advanced version of 8051 and it has got 3 timer counters

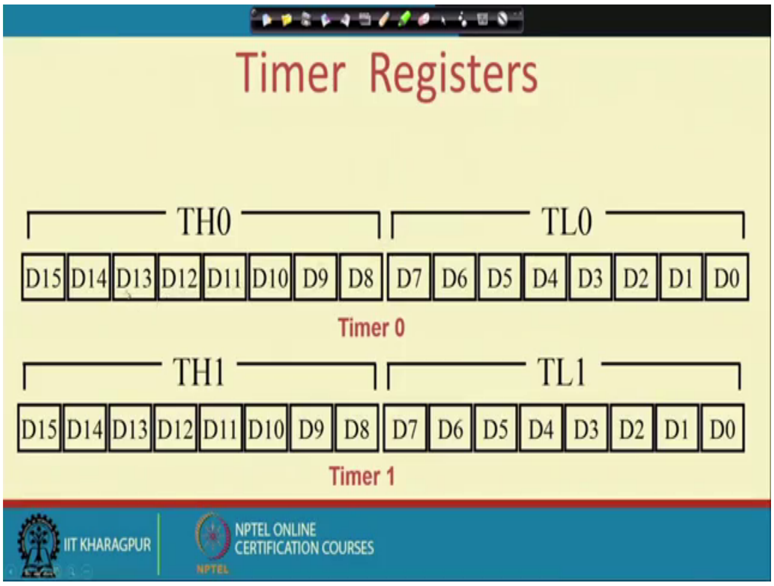
So, instead of having 2 timer 0 and timer 1 it has got timer 2 also, and this formats of this control registers will be slightly different, and we have got T2CON. So, for controlling the timer 2 registers. So, TH 2 TL 2. So, T2CON TH 2 and TL 2. So, they are for 8 0 5 2, but normally we will be talking about 8051. So, we will restrict ourself to 2 timers timer 1 and timer 0 timer 1, and accordingly we will see the setting of this TMOD register TCON register and all that.

(Refer Slide Time: 11:31)

So, basic registers for any timer is like this. So, both timer 0 and timer 1 the are 16 bit wide. So, they are 16-bit timers. So, these registers they have stored the time the time delay as a timer. So, like how much time. So, you are initializing it to some value, and from that point onwards. So, it will start going up. So, it will overflow when that 16-bit value that 65535 is crossed.

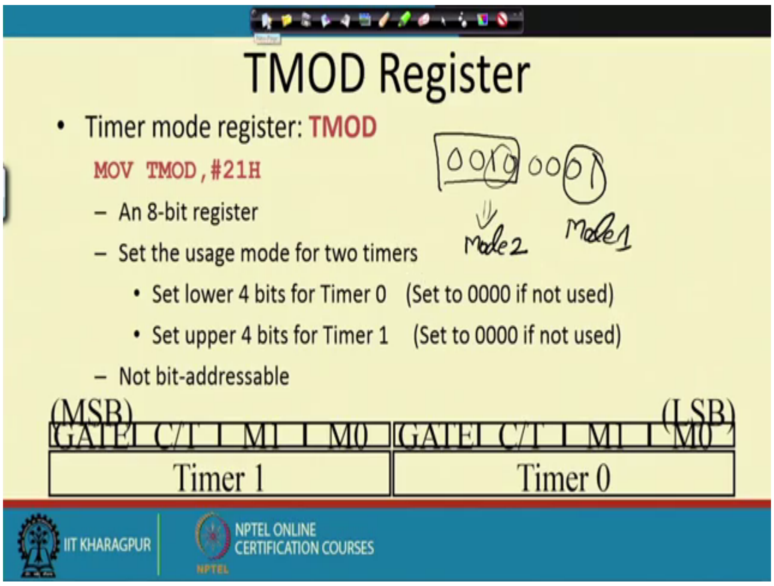
So, in some sense, we can say that it holds the time delay that is remaining in the timer, or if it is counter then it will hold the number of events as a counter. So, this t a TH 0 TL 0. So, timer 0 has got this 2 registers TH 0 TL 0, timer this TH 0 is the higher ordered byte, and TL 0 is the lower ordered byte similarly timer 1 has got TH 1 and TL 1 registers, out of that TH 1 is the higher byte and TL 1 is the lower byte. So, this 16 bit register it can be accessed as 2 separate registers TH and TL. So, we will be seeing that while initializing these registers to some values. So, we have to do it in terms of 8-bit registers.

(Refer Slide Time: 12:52)



So, this is the structure. So, this total I said that it is a 16 bit. So, D 0 to D 15 for timer 0, out of that D 0 to D 7 is in TL 0 and D 8 to D 15 is in TH 0 though it is written like that, but we have to put the values in separately in TL 0 TH 0, and similarly we have got timer 1 that is divided into 2 8-bit registers TL 1 and TH one. So, they these 2 resistors are for holding the time value or the count value.

(Refer Slide Time: 13:23)



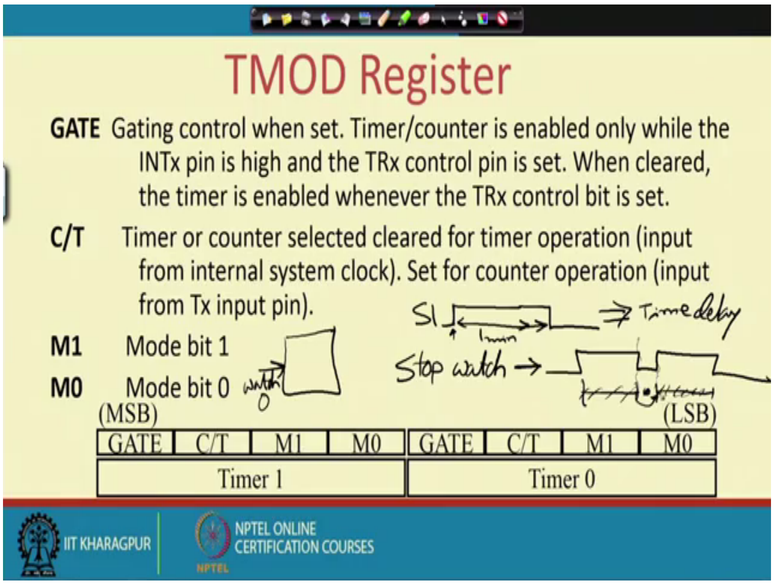
Next we will look into the TMOD register or the timer mode register. So, this is another 8-bit register. So, in this 8 bit register 4 bits are dedicated for timer 1, lower 4 bits are dedicated for timer 0, and among that 4 bits the first most significant bit is the structure is similar for both the nibbles are structure are similar, the most significant bit is the is called gate the next bit is called C by T, that will indicate whether the operation is a counter operation or a timer operation, and then the next 2 bits m 0 and m one. So, they will identify the mode of the mood of the timer that we want to operate.

So, this is one 8 bit register and it can be used a lower 4 bits for timer 0, and upper 4 bits for timer 1. So, if you are not using say lower 4 bits. So, you have to. So, you should set these values to all 0, and if you using not using the upper 4 bits then you should set all these upper 4 bits as 0, but it is not bit addressable.

So, unlike other registers where I can set this set reset the this bits individually, here you cannot do that. So, we have to prepare the 8-bit pattern in some accumulator, and then we have to move that pattern to this TMOD register, or we can use this type of absolute addressing. So, hash 21 h. So, hash 21 h means it will be; do, 0 0 1 0 and 0 0 0 1.

So, for the upper timer; so, it is 0 0 1 0. So, it is gate is 0 count timer CT BT is 0 m 1 m 0. So, this says that it is 1 0. So, that is mode 2. So, it says that this upper one the timer 2 timer 1 will be operated in mode 2, and this is 0 1 means this will be the lower the timer 0 will be operating in mode one. So, we will see what are these modes slowly. So, let us proceed.

(Refer Slide Time: 15:38)



So, next looking it looking into more detail the TMOD register; now, you see that for many of cases. So, what you want that these timers, they may be operated on 2 different situations, or 2 different type of controls, in the sense that maybe I have an application where, if some event has occurred externally then only the timer should be turned on, and when it is running.

So, it is using internal clock, but as long as some activity is going on in the environment, we should count that. So, this is one type of situation, another type of situation is just to produce pure delay. So, I have to I have activated one signal and maybe it has to be deactivated after some time. So, one type of application is like this. So, I have activated a signal says signal s 1 is activated and I want that for example, in a traffic light controller system we want that red light should be on for say 1 minute.

So, this; so, once this red light is turned on; so, if s 1 corresponds to this red light; so, s 1 is. So, red light is turned on at this point. So, we want that for 1 minute this red light should be on. So, this is 1 minute. So, this is not guided by any external phenomena. So, once the light is turned on. So, microcontroller after 1 minute it should turn it off. So, this is one type of situation. So, pure time delay; so, I can say that these are basically I want to produce a precise time delay, and for that purpose I am using it, in other cases there may be some external event and for which we want to measure the time that is needed.

For example, say we want to we want to make a stopwatch. So, if you want to make a stopwatch, then I have to start the timer, and then this I have 2 when some external event occurs, I have to I press this stopwatch button.

So, in my system if this is the system, there is a there is a watch button and when this watch input comes, then only then only this timer should be activated, and if this watch input is 0 then the timer should not work. So, maybe my watch goes like this. So, watch is on for this much time then it is off like this. So, I my timer should run in this period and again it should run at this period, and when this watch signal is low in this region, it should not run, fine the timer should be stopped in in that time.

So, at this point of time it holds the starting from the watch buttons. So, how much time has passed, or if it is starting at this point. So, it can after that again the when the watch is activated. So, these values remember there. So, this value is carried over to this point, and it will continue like this. So, this way we can have 2 different type of controls in the operation of timer, in one case the timer is controlled totally internally by the microcontroller, and in the second case the timer is controlled both by the external event and the internal controller. So, accordingly so, this class differentiation is made by these gate. So, this gate bit. So, this tells when the whether counter has to the timer has to be enabled, only while the INT x pin is high.

So, you remember that there are 2 INT pins INT 0 and INT one. So, then the INT x, so for timer 1. So, if you set this gate bit to one, then this timer will operate only if this INT 1 pin is high and there is another control bit TR one. So, we will see this TR 1 bit later to this TR 1 control bit is high. So, normally if you are running the timer without any external interference, like say gate.

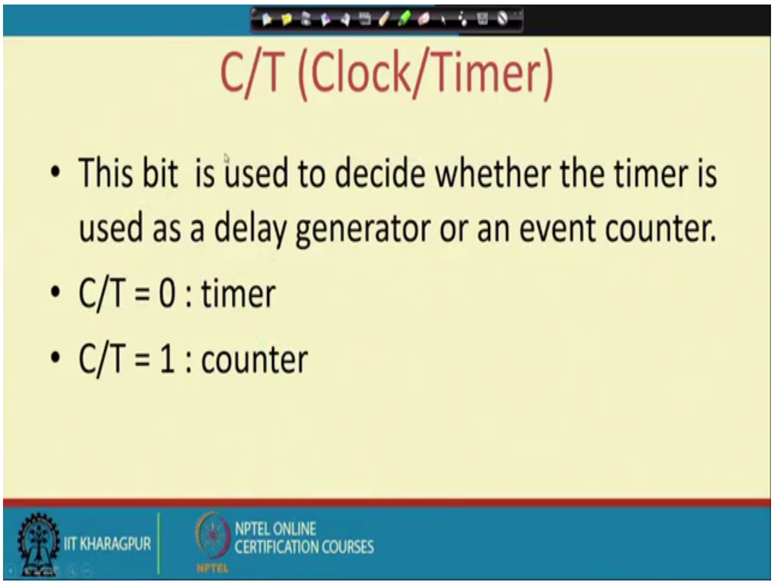
So, in that case the gate bit should be set to 0, and to start the timer we have to set the corresponding TR bit to one. So, start timer 1. So, you have to set TR 1 to one, to start timer 0 you have to start set TR 0 to 0. So, to run a timer what we have to do is first set, the timer value and then we have to set this TR bit the TR x bit. So, that is the operation. So, we will see this location of these TR x bits slightly later. So, we have this gate bit is like that, when cleared the timer will the timer is enabled whenever the TR x control bit is set.

So, if this gate bit is 0, then this whenever TR x is one, the timer will operate if the gate is one. So, it is controlled by both the INT x pin and the TR x bit. So, both hardware and software will control the timer if gate is one, if gate is 0 only software will control the timer, the next bit is C by T timer or counter select. So, it is it is select like cleared for timer operation. So, if this bit is 0.

So, we are willing to operate timer 1 in the timer mode, and if this bit is 1 we want to operate the timer in the counter mode. So, this T x pin should be high then and this C y T pin should be high, and then for INT as I have already said that for internal operation when it is operating in the timer mode. So, microcontrollers crystal frequency will be taken as the clock and in case of counter operation this T 0 and T 1 pins.

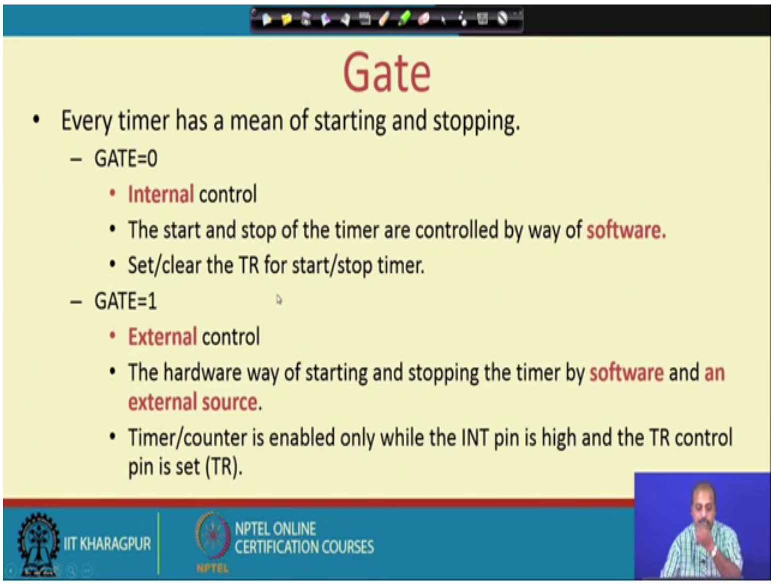
So, they will be used as input for the event. So, this t for timer 1 it is T 1, for timer 0 it is T 0, then we have got 2 more bits m 1 and m 0. So, they are called mode bit. So, depending upon the mode in which we want to operate this timer. So, we have to set this m 1 and m 0 bits.

(Refer Slide Time: 21:58)



So, this C by T bit. So, it is used to decide whether the timer is used as a delay generator or an event counter. So, if it is C by T is 0. So, this will be operating as a timer. So, that is a delay generator and if C by T is 1 then it is a counter. So, this is for counting operation gate part.

(Refer Slide Time: 22:18)

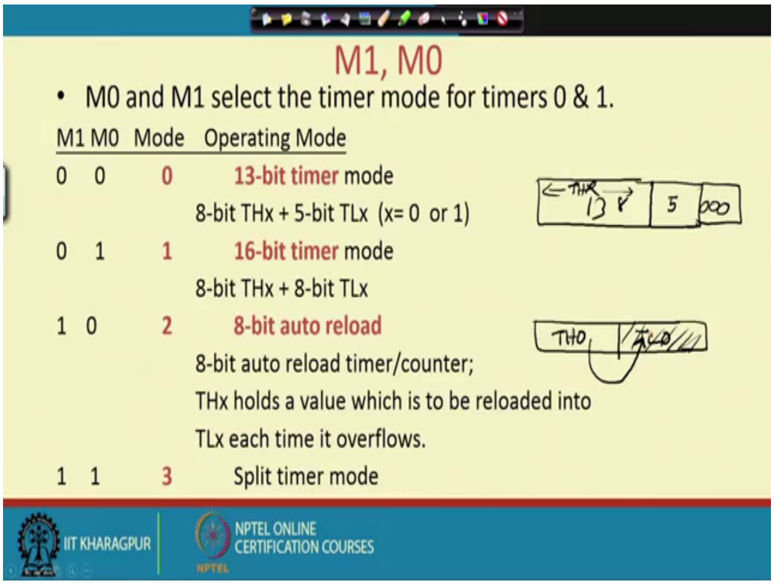


So, gate part every timer has a mean of starting and stopping. So, gate equal to 0 and gate equal to 1. So, gate equal to 0 means as I said that it is totally internal controlled. So, it is soft software-controlled operation, the start and stop of the timer will be controlled by the software by setting the TR bit. So, we will see that TR bit. So, you can once you set that TR bit the timer will start, when you want to stop the timer we have to clear the TR bit.

On the other hand, this gate equal to 1. So, this is external control. So, you have got. So, it is both by software and external source. So, both software and hardware will control the timer. So, this timer in this case will be enabled when INT pin is high, and the TR controller pin is control b t set, that is TR. So, this is. So, this will be controlled bit it is not a pin, it is a bit in the some register.

So, this bit should be set, and then the timer will start and when you were to stop it, either you can reset this TR bit or you can disable this INT. So, either of them can be done for stopping the timer, and in this case to stop the timer we have to disable the TR bit.

(Refer Slide Time: 23:40)



So, these are the modes of the timer operation. So, we can have 4 different modes of operation with the timers mode 0 mode 1 mode 2 and mode 3, mode 0 is a 13-bit timer mode. So, this 8 bits will be stored in a THx register and 5 bits will be in the TL x register. So, x is 0 or once if you are; so, for timer 0 x equal to 0. So, 13-bit timer; so if the your time value that we have is 13 bit. So, this is the 13-bit value, but this, but I am using this TH, TL pair for holding the value fine. So, how to do it; so, this THx will hold the higher order 8 bits and this TL x will have the lower order 5 bits.

So, this is that 8 bit higher ordered 8 bit and lower ordered 5 bits it will be clubbed with 3 more 0s, that will be holding the lower ordered of a lowered ordered lower ordered 5 bits will be there in the TL x register, on the other hand mode 1 is a 16-bit timer mode. So, we can get hired higher delay. So, it is 8 bit THx plus 8-bit TL x; so, total is 16-bit timer value, and there is another very interesting time of type of timer structure that we have is called, 8 bit auto reload; that means, So, here. So, it after the timer has expired this timeout bit has been set the timer will restart. So, it will restart with some initially set time value it will be restarting.

So, in this case. So, 8 bit auto reload timer counter. So, here this TH x will hold a value which is to be reloaded into TL x each time it overflows. So, it is. So, it is interesting like this. So, we have got this TH x and TL x registers registered pair set TH 0 and TL 0 now this TL 0 is actually operating as the timer. So, TH 0 just holds the initial value. So, your timer is an 8-bit timer. So, when this timer overflows then again, this TH 0 value will be copied into TL 0 and it will restart. So, if you are trying to generate a periodic operation. So, this is very much useful. So, your periodic delay can be generated in in this way. So, you can have this TH 0, TL 0 pair and this TH 0 will be operating like that.

So, this is very useful and there is another timer mode which is known as a split timer mode. So, we will see this when we go into that corresponding mode to it is slightly complex. So, we will go into that.