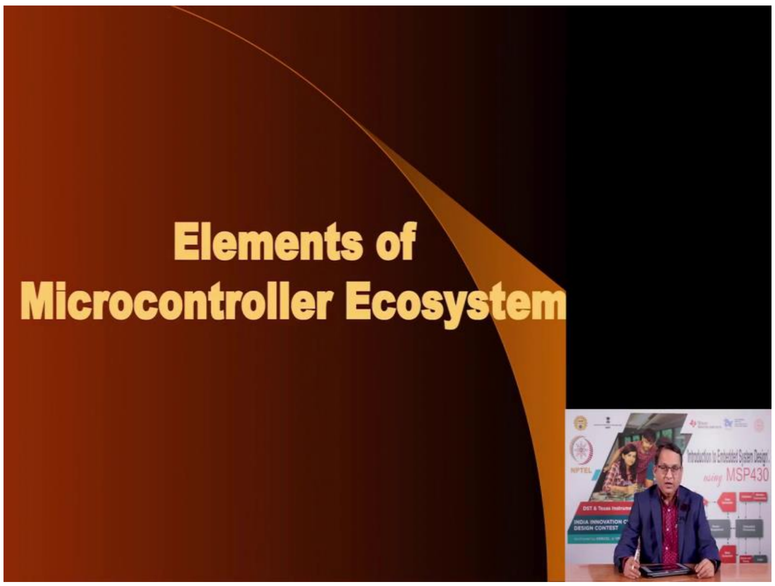
Introduction to Embedded System Design
Professor Dhananjay V. Gadre
Netaji Subhas University of Technology, New Delhi
Lecture 7
Elements of Microcontroller Ecosystem

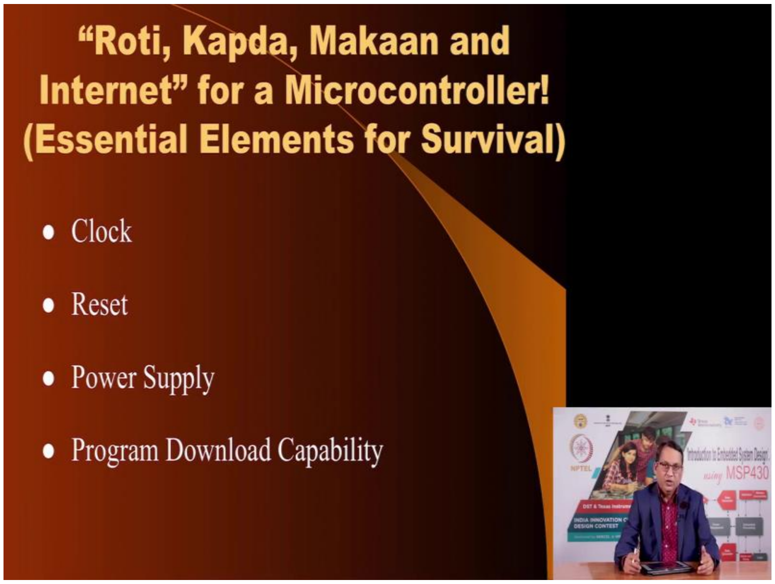
Hello, welcome to a new session on this course on introduction to embedded system design, in this lecture we are going to consider the ecosystem that a microcontroller requires for its operation and once we complete that we will look at the power supply building block as part of the six block representation.

(Refer Slide Time: 00:53)



So, let us start, what are the elements of a microcontroller ecosystem? Meaning what does a microcontroller require to function independently as you know now that a microcontroller is a complete computer on a single chip it is able to perform whatever it is programmed to do just if few elements of its required ecosystem are provided to it., What are these elements?

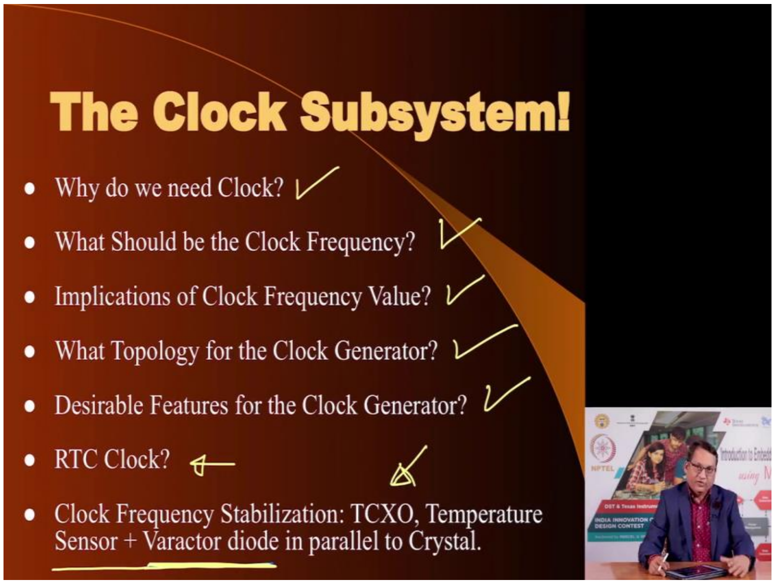
(Refer Slide Time: 01:19)



So, I titled it as "Roti, Kapda, Makaan and internet for a microcontroller this is crazy sounding title has been derived from a previous slogan which said for a human survival you need three things roti that is food, kapda clothing and housing. In recent times, even the lowest level of society requires internet for survival and therefore, a human being requires four things, food, clothing, housing and internet and we found interesting parallel that even a microcontroller requires four such things and what are these four things for it survival and active working.

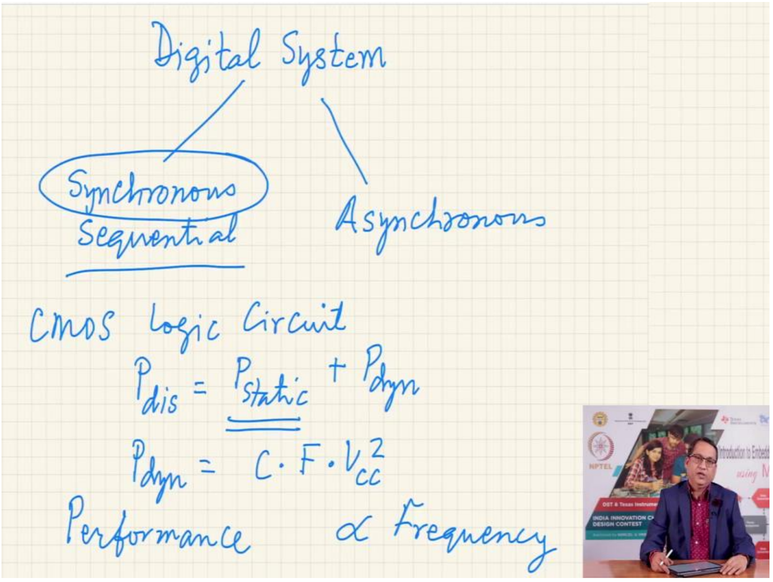
It requires a clock we will see why it requires a clock? It requires a reset circuitry usually that reset circuitry is integrated in the microcontroller itself. It also requires a power supply different from perhaps different from the power supply requirement of the rest of the embedded system application and then it requires a mechanism to download code from the development platform which often is your desktop or laptop computer into the memory of the microcontroller. So, we have to ensure that microcontroller is provided for with these elements then only it can function properly and if the microcontroller function properly then the rest of the embedded system can function properly.

(Refer Slide Time: 03:00)



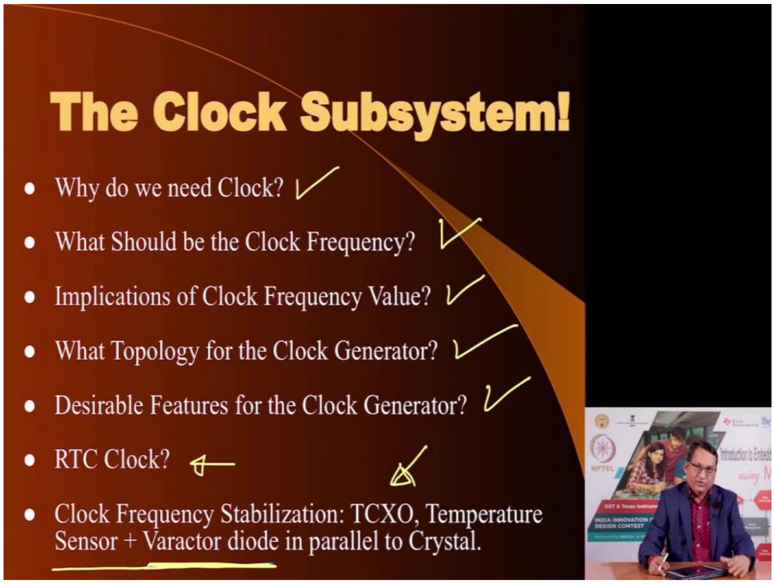
So let's start with the first element of this ecosystem, the clock subsystem. Why do we need a clock? Can we design microcontroller systems without a clock? Most of the time you would have seen that an analogue circuit does not require a clock but a microcontroller is different because it is an example of a digital circuit now digital circuits can be design in two ways what is called as asynchronous sequential circuits and the other type is synchronous sequential circuits.

(Refer Slide Time: 03:33)



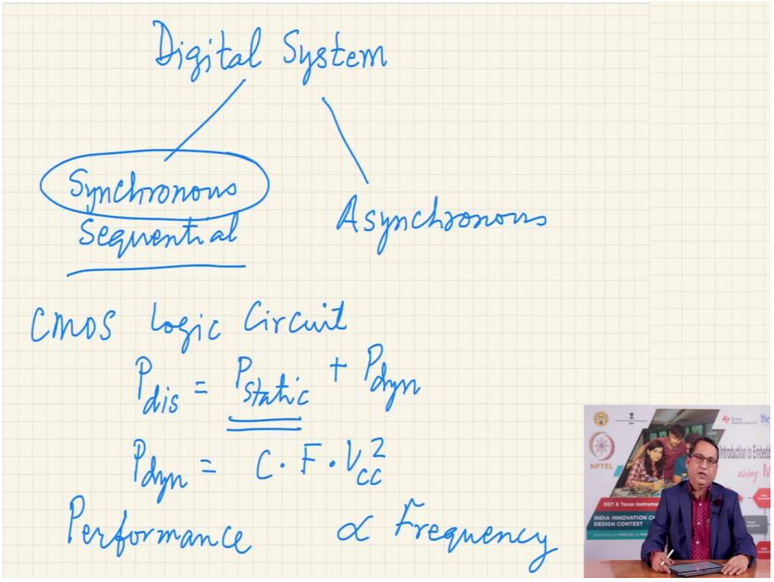
So, any digital circuit or system could be design in two ways synchronous sequential mode and asynchronous. Now, asynchronous sequential circuits are very efficient they consume little power but they are very difficult to design and even more difficult to debug and verify and therefore, almost 99 percent of all logic circuit implementations are of these variety synchronous sequential. Now, the term here synchronous means that it uses a clock circuit for its operation and therefore you need a clock. So, this defines or this justifies the need of a clock because a microcontroller is an example of the synchronous sequential circuit.

(Refer Slide Time: 04:49)



Then the next question arises so we have answered this the next question arises as to what should be the clock frequency? Now, there are two contrasting two diverse diverging arguments that the clock frequency should be as low as possible. Why?

(Refer Slide Time: 05:10)



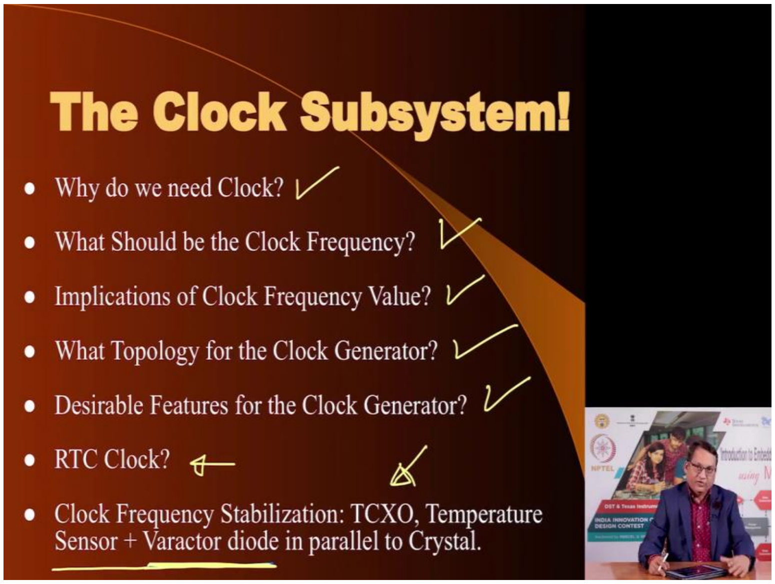
Because in a CMOS logic circuit, the power dissipation consists of two parts the static power dissipation plus the dynamic power dissipation. The static power dissipation is a function of the technology the way the system was built but the dynamic power dissipation in a CMOS circuit is equal to the capacitance that the circuit sees at its load into the frequency of operation into the VCC square which means the dynamic participation is directly proportional to the frequency operation.

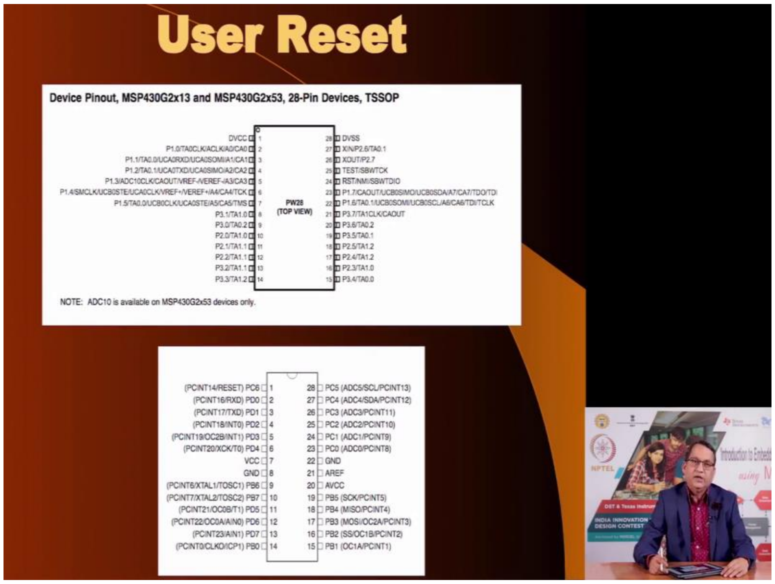
So, if you increase the clock frequency that would consume the battery faster. If you double the clock frequency it will double the power dissipation which means it will empty the storage that is your battery in half the time. Now, this is one argument why you should keep the clock frequency low. What is the argument towards keeping the clock frequency high? Well if the clock frequency performance of a computer is directly proportional to frequency.

For a given a computer if you double the clock frequency then it will execute the same program in half the time and so these are two arguments which are pulling in opposite direction and therefore in as a efficient embedded system design you must strike a balance, one which optimizes the power dissipation and the other side where the frequency is high enough for the system to work properly.

So, we have seen why we need a clock? We have seen what should be the clock frequency we have also considered the implications of this clock frequency that if we keep the clock frequency very low it will certainly provide a longer operation on a given battery but its implication could be that it may not be able to fulfill the requirements, the timeliness of a program and therefore you may want be increase it and as a designer as a engineer you would have to strike the right balance trade of between low enough frequency to keep the power down and high enough for it to perform its designated task.

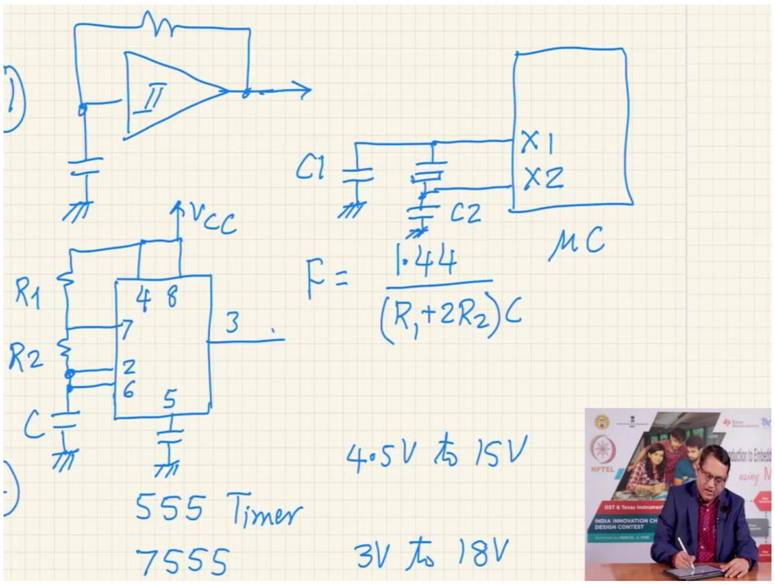
(Refer Slide Time: 07:59)





What are the various topologies for generating this, the clock for microcontrollers? Now if we look at the pin out of typical microcontrollers, you will see that it offers you a couple of pins on which you can connect an external component such as a quartz crystal and often times it also has an internal RC oscillator.

(Refer Slide Time: 08:34)



Let us see what this circuit looks like. Or in general how do we create a clock signal? So, one simple method we already discussed in a previous lecture was to use ashbin trigger with a capacitor and resistor to generate the frequency this is this could be used. Another one you could use very popular IC called the 555 timer IC.

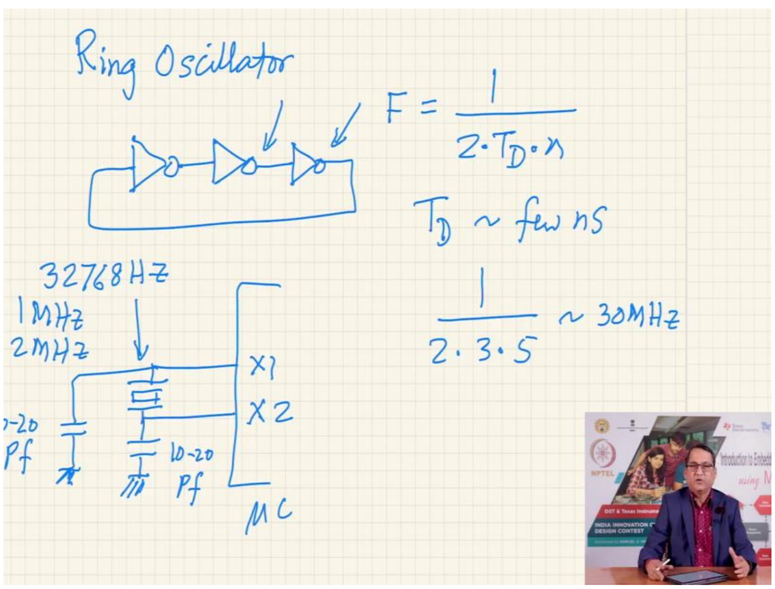
It has 8 its a 8 pin IC. You connect 4 and 8 pins to VCC then you have pin number 7 you connect a resistor R1 and then pin number 2 and 6 you shot them and you have another resistor this is R2 and from here you put a capacitor to ground C. you have pin number 5 which would decouple and put a capacitor and on pin number 3 you get the output voltage. The frequency of such a topology is 1.44 divided by R1 plus 2R2 into C. And you can tally the values of the resistances and the capacitors to get your required frequencies.

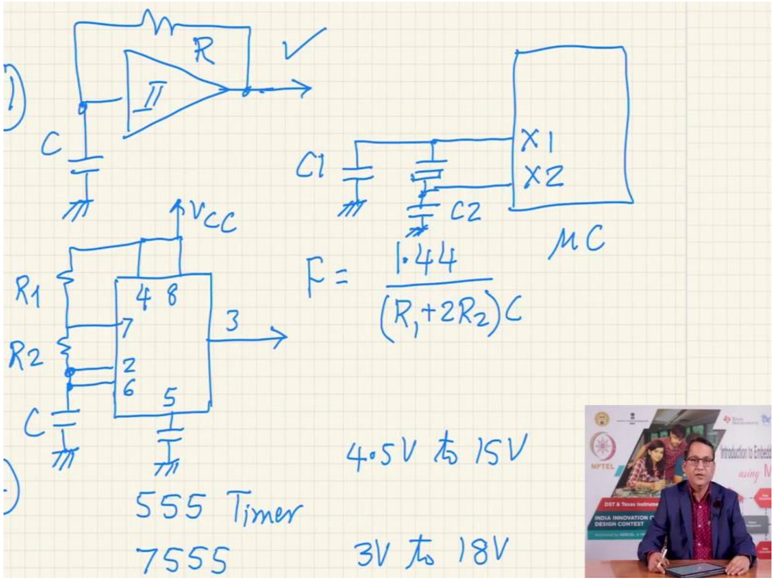
Now, what happens is the typical 555 oscillator timer will require a power supply in the range of 4.5 Volts at least to 15 Volts for operation and if your microcontroller also operates in a similar power supply range you do get microcontrollers which works at 5 Volts then you could use a 555 timer. In case you have a microcontroller which requires a lesser operating voltage say 3 Volts, then instead of regular 555 timer you can use CMOS variant of this timer called 7555 with identical pin configuration, but with the much diverse power supply requirement it can operate from 3 Volts to 18 Volts.

So, you could perhaps use such a external clock generator. The output of this or output of this would be connected to your microcontroller. Here is a microcontroller and you may have pins called X1 and X2 on which if you do not want an external clock generator then you connect an appropriate crystal, a quartz crystal between these two pins. It may also require you to use some small value capacitors these will be mentioned in the datasheet of your microcontroller and you must follow those specifications.

In case you do not want to use this crystal. Then instead of this crystal the output of such a clock generator one or say 555 or 755 this output be could be connected to X1 or X2 input as mentioned in the datasheet. There is another interesting method of generating clock and that is by way of what is called as ring oscillator.

(Refer Slide Time: 11:55)





Ring oscillator as a name suggest is a ring of components in fact it uses a NOT gate and inverter and odd number of inverters. So, if I take 3 inverters and I put them connect them like this power it with appropriate supply voltage then at any point here it will show oscillations and the frequency of those oscillations will be equal to 1 upon 2 times the delay period of this device into n. In this case n is the number of elements in this case 3.

From the datasheet of this inverter you can find out what is the delay time for typical TTL family components TD is of the order of few nanoseconds. Therefore, you might get may be 1 by 2 into 3 into 5 nanoseconds so this is about 1 by 30 nanoseconds that will give you roughly 30 Megahertz of operation if you want a high frequency.

Please note that 555 or 755 timer cannot provide a very high frequency beyond couple of megahertz and so it may be suitable for some applications. Anyway, these are the method of creating a external clock signals with which you can feed your microcontroller. It turns out that such a need is usually not required such a topology is not required why? Because microcontrollers have built in oscillators, they have built in RC oscillators and you can choose do you want a regular oscillator or do you want to use an internal RC oscillator.

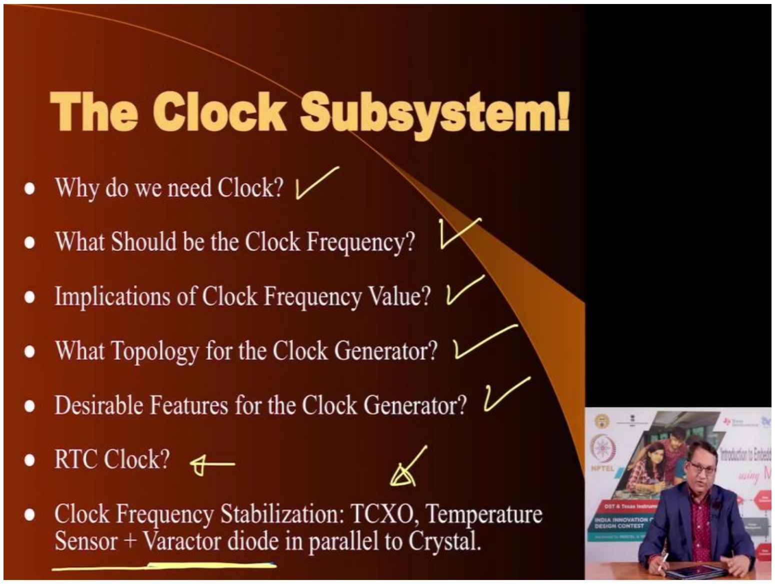
The internal oscillator you can imagine would be of this type and therefore since it uses these R and C to provide the clock frequency. The variation in the resistor and capacitor values would change the frequency and therefore it may not be used for very accurate measurement of time but if your application is ok with the little degraded performance of the clock generator as in its not as accurate as you would want then this could be a good low cost solution.

In case you want higher precision in time measurement then you would use this option where on the designated pins of the microcontroller X1 and X2 you would connect a crystal of appropriate frequency together with some capacitors if the datasheet say so and typically the values of these are of the order of 10 to 20 Pico frats here and the crystals are could be in the range of the lowest crystal that you may get.

The lowest crystal incidentally happens to be 32768 Hertz and I will come to this I have also mentioned it in the past but we will go through this again and going up to 1 Mega Hertz, 2 Mega Hertz and so on going up to few tens of megahertz 20, 30 megahertz crystals may be available and you could use them for determining the clock frequency. The oscillator part the active part of the oscillator is inside the microcontroller. Therefore, all you need is external couple of components and this provides the source of clock.

As I mentioned in the previous lecture, if the microcontroller offers clock scale ability then you should start with the lowest clock frequency crystal because internally you could use a multiplier to increase a clock frequency to high value that you would want at given point of time. And if you so desire that you do not require a high enough clock frequency then you can scale it down to the lowest value which would be the crystal value in this case 32768 hertz. So, that would be a good choice for a crystal to be used in such microcontrollers which have this option of run time clock scale ability.

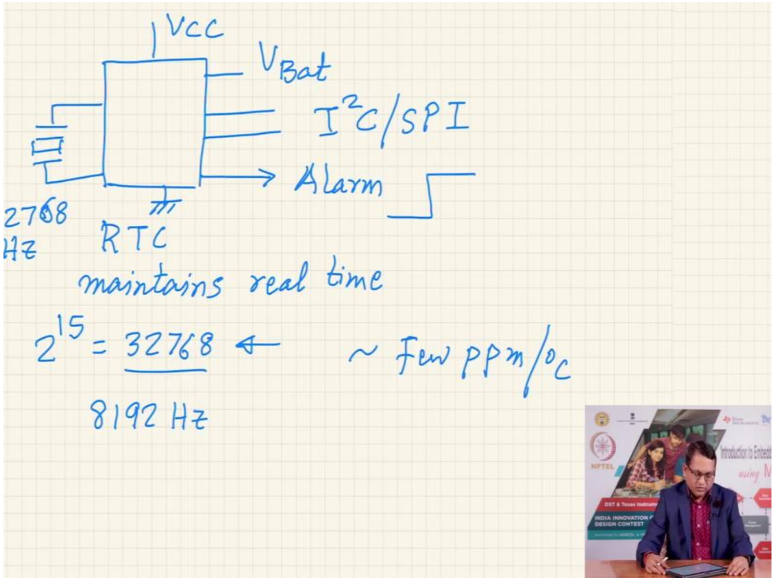
(Refer Slide Time: 16:35)



So, now we have seen what topologies of the clock generators and also what are the desirable features of the clock generator we have seen that we would like to have the clock generator the frequency of which we can change at run time but this is usually a inherent and built in feature of a microcontroller and you must evaluate whether your application requires such a function then you must choose a microcontroller which offers that functionality.

Apart from the clock frequency that is required by the microcontroller you require an additional clock a complete clock system and that clock system is called a real time clock. A real time clock is a dedicated function dedicated circuit which provides a certain functionality which is to maintain a real time.

(Refer Slide Time: 17:30)



So, RTC maintains real time that is it can be used to know what is a current time of the day at whichever location you may be. Of course for this it requires a source of frequency and usually the source of frequency is a crystal and that crystal is usually 32768 hertz and this frequency this seemingly odd number is not that odd for computer and electronics engineers because this number is exactly equals to 2 raise to power 15 is equal to 32768.

Historically these crystals were used in digital watches when they started appearing in late 60s and early 70s and this low enough frequency was chosen because again this clock signals circuits were CMOS spaced and the power dissipation had to be kept low because they were battery operated and therefore a balance was achieved that frequency should be low enough and the crystal should be not large enough therefore they found that 32768 is good trade of it is small enough in size and would to relatively less per dissipation.

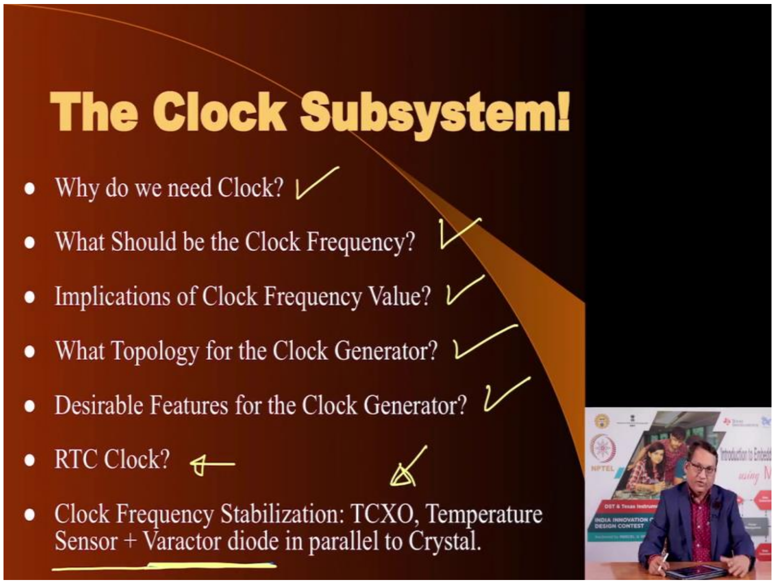
Although for record I may mention here that the very first digital cord crystal based clock that came on the market was the Seiko clock and it actually used 8192 hertz crystal but subsequently the common frequency of use was this and this frequency has since been adapted and adopted for use with RTC chips, usually this is a 8 or 16 pin IC, it has power supply pin it also has a battery backup.

So you would have additional pin cord V bat and it would have a mechanism to communicate with the micro controller, usually this would be the I square C or SPI interface we have discuss that communication protocols are inherently some of the some of the communication protocols are inherently available on certain micro controllers so this micro controller could talk to this real time chip through these communication protocols it may also have alarm out alarm pin so that when you program this real time clock to wake you up on certain alarm when the time is reached this signals the alarm pin would go from 0 to 1 and this could be used to interrupt the micro controller to do whatever was expected to be done.

The frequency as I mentioned again is 32768 hertz, the accuracy of the crystal here would determine how good the real time is being maintained and typically these crystals have temperature coefficient which is of the order of few parts per million per degree centigrade and therefore the traditional real time quartz crystal based real time clocks offer roughly a drift of few seconds per months kind of numbers.

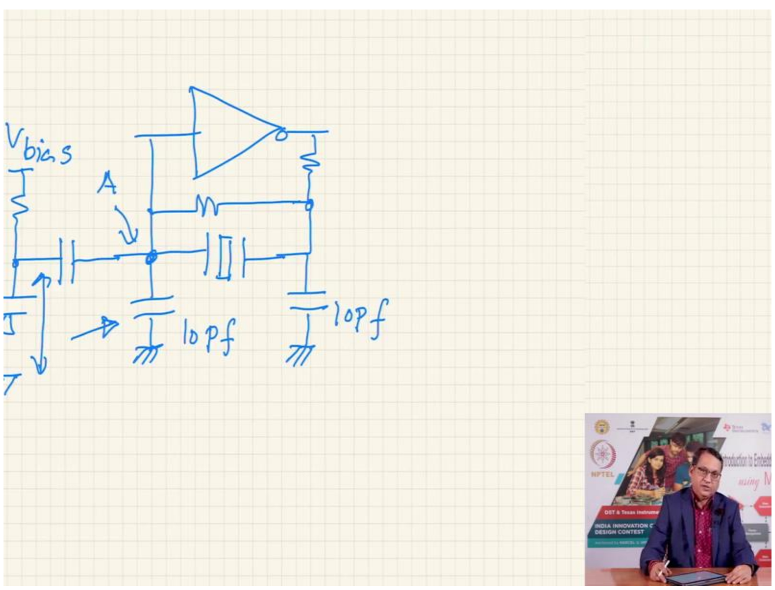
If you want even more accurate maintenance of time then there are two options one is to regularly correct the frequency or the time that this clock is maintaining and one way to do that is to get the actual time from the internet there are services which offer the time and in fact your PC and laptop often fetch the time from the internet sources and correct the clock in case it has drifted, the other is to used more accurate source of clock and one way to do that would be to maintain this crystal and this circuit at a constant temperature.

(Refer Slide Time: 22:11)



Let us go back to presentation to see what we are talking of. You can stabilized the clock using technique cord TCXO meaning temperature compensated crystal oscillator, this is basically a small enclosure where the temperature of that enclosure is maintained to a fixed value irrespective of the outside ambient temperature and this ensures that the crystal does not drift.

(Refer Slide Time: 22:54)

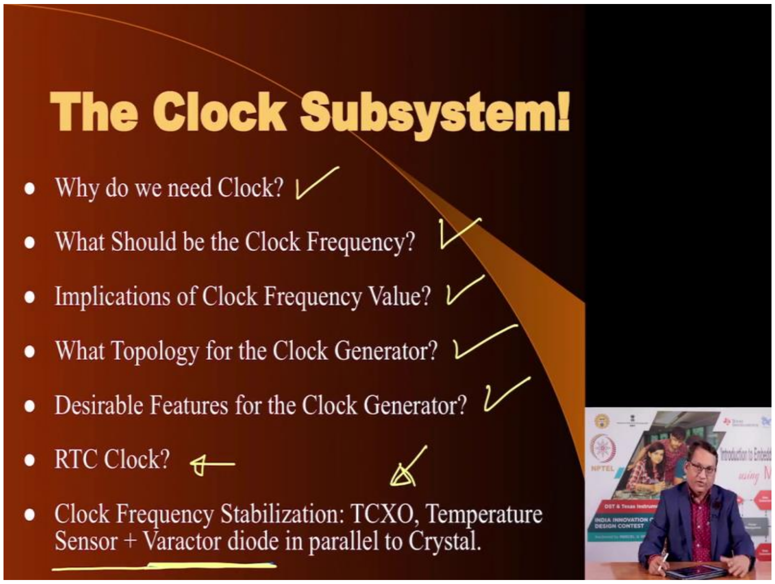


Another method uses a unfortunately this method consumes a lot of power to maintain that temperature, another method uses a temperature sensor and a varactor diode and the way these crystal works the circuit diagram of crystal oscillator is like this that you would have an inverter and you would have big large value of resistor another resistor here and you would have a crystal between these pins and you would have couple of capacitors as I mentioned low value capacitors.

Now, this is typically say 10 Pico farad is also 10 Pico farad and say this could be say 1 mega hertz now if I change this crystal if I change this capacitor I can load the frequency operation and I can change it a little bit, similar way I can add a therefore instead of changing it manually I could add a varactor diode say a varactor diode is nothing but a diode which is operated in reverse bias.

I need to decouple the varactor diode here by using a large capacitor and I apply a bias voltage here say some V bias by changing the bias voltage the capacitance between this two points can be changed and therefore the effective capacitor as seen at this point A would change which would treat the frequency operation and so in many circuits where they want to offer a better temperature stabilization they use a temperature sensor to monitor the temperature and to compensate for any variation in temperature.

(Refer Slide Time: 24:46)



They change the bias voltage on this varactor diode which has been place in parallel to one of the capacitor of the crystals so as to maintain fixed clock frequency. So, these are various methods of having higher accuracy clock frequency signals in application where you require more precise measurement than what would be available with just the cords crystals or with the RC oscillator.