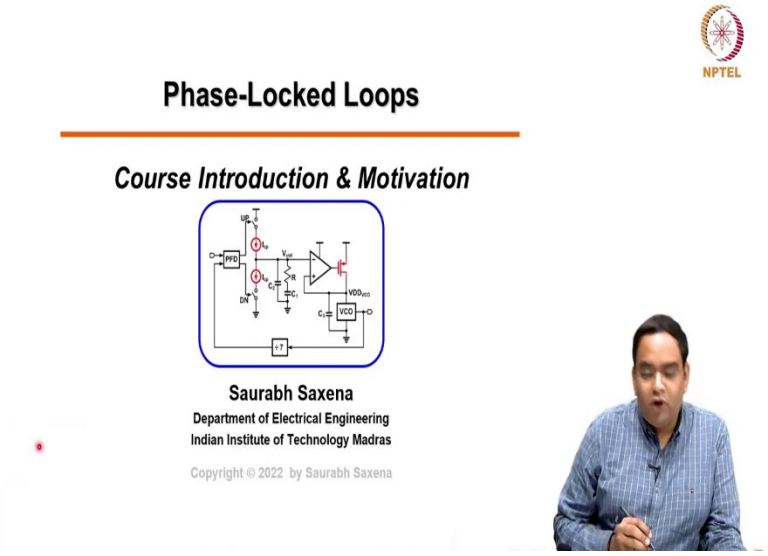
Phase-Locked Loops
Dr. Saurabh Saxena
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 1
Course Introduction and Motivation: Part I

(Refer Slide Time: 0:15)



Welcome everyone to this course on Phase-Locked Loops. I am Saurabh, a faculty at the Department of Electrical Engineering in IIT Madras. The phase-locked loops course is about clock generation for various applications which actually hawk your daily life these days. Now, before we go into the course itself, just think about the things which are periodic in your daily life. The instant I use the word daily, there is a periodicity associated with it.

And as you see the sun rises in the morning, sets in the evening and you have a 24-hour day cycle. This is one periodicity which we do not set but it is set from somewhere else. Then you look at the periodicity of your year, every year is 365 days, 6 and a half hours and then over a period of time you have 366 days. You count 365 days for three years and 366 days for the fourth year. There is again some periodicity associated with it.

Now, both these periodicities, whether you have 24-hour day cycle, or you have 365 or 366 day year cycle, they are very large in terms of absolute time. Now, suddenly you change this particular periodicity which you have in your daily life to the time when you are downloading any

information whether through your wireless network or through your Ethernet network, there you look for something which is like Megabits, Gigabits per second or so on.

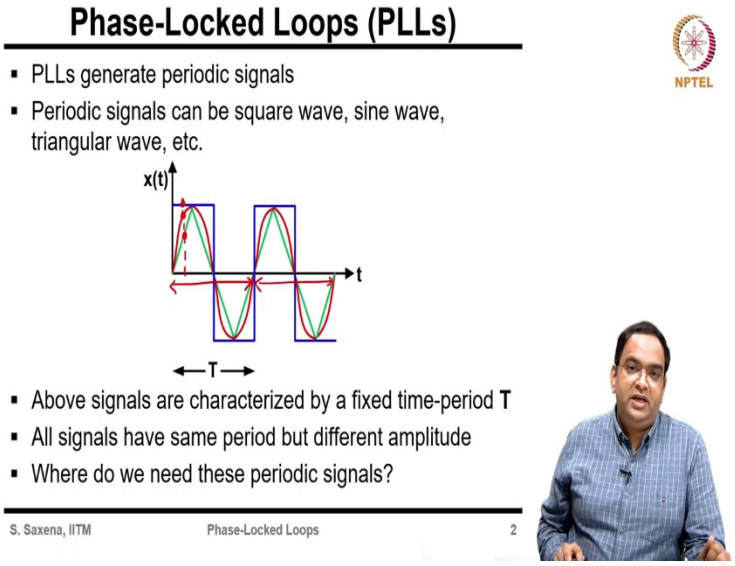
Now, when we say Megabits per second or Gigabits per second, it is like you are receiving one Gigabit every second. And ideally, you would like that all the information which you have received, that should be correct. Now, there will be many things which ensure that this information is correct, but you would never like to download the data and then it is told to you that the data is corrupt because in place of downloading at one Gigabit per second, somewhere, you downloaded one Gigabit in a second minus 100 bits.

If this happens, then your data file will be corrupted. Now, who ensures that you receive your bit every second at the rate of one Gigabit. There should be some clock which is behind the scene which is making sure that, for one GHz example, one Gigabit per second example, every nanosecond your clock ticks and you receive a bit.

So, there should be something which is making sure that the clock which you are using to send or receive the data is fixed. It ticks every nanosecond, and these days as you go on, we will go through the motivation for this particular course, you see as you go on there are numerous applications where you require clock which is highly accurate and cheaply available at the consumer side.

Otherwise, if the clock is not cheaper, then you will not be able to use it. You cannot think of using an atomic clock in every device which you are using. So, phase-locked loops course is about the generation of such clock signals which are needed in different applications and these signals should be generated accurately as desired by those applications. And these are periodic signals. So, to begin with, if you just want to say that what do the phase-locked loops do.

(Refer Slide Time: 5:01)



Well, phase-locked loops generate periodic signals. That is the bottom line. How do they generate, what all things will be used in generating those signals, that will be in the course. So, these periodic signals, if you look at the diagram which is drawn here, these periodic signals can be square waves, sine waves or triangular waves or any other signals. The thing which is associated with periodic signals is they are periodic in nature.

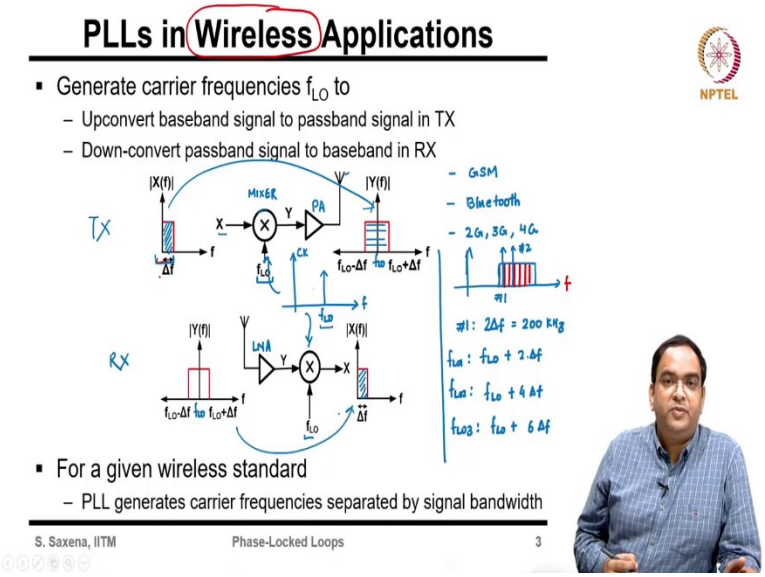
They repeat after every delta time. So, in this particular case, if you see, the signal x(t) is plotted versus time. You see a square wave which is drawn in blue here, it has a time period, T. Time period, T, means that this signal as such repeats every time period T. Now, this is a square wave signal, it is periodic. Then, you see the green one which is a triangular wave which is going like this. This is also periodic in time.

On the other side, you see a sinusoidal kind of signal which is also periodic in T. At any given instant of time, the amplitude of all these signals may or may not be the same. Triangular wave has this amplitude, sine wave has this amplitude, and square wave has this amplitude. So, all these signals have different amplitudes at a given instant of time. But, all these signals are periodic in nature and they are periodic in time period T. They repeat every time period T.

So, these are the signals which will be generated by the phase-locked loops. From where? That is what we will see. Now, the question arises, where do we need these periodic signals? Before we go and say that we can generate such and such periodic signal, we should also know that where

these signals will be needed. Unless we know the application, we will not know the requirement of these periodic signals, the properties of the periodic signals which you need for a given application.

(Refer Slide Time: 7:46)



So, next, we will actually see what kinds of applications use such clocks. So, first one is your wireless applications. Now, this wireless application, I brought it in the first place because this is something without which we cannot imagine our lives these days. So, in wireless applications, if you look at it, you have a baseband signal which is shown here with Δf bandwidth. So, this is your baseband signal with Δf bandwidth. This signal on the transmitter side, this is your transmitter side, on the transmitter side, this baseband signal is multiplied by a clock signal.

And what is this clock signal? This clock signal, in the frequency domain, has a single tone, ideally at f_{LO}. If you look at the frequency content of this f_{LO} signal, it is this. So, you multiply the baseband signal with the clock signal on the transmitter side through a mixer and then you have a power amplifier which is PA and you transmit the signal. So, whether you use your wireless standard which is GSM or you use Bluetooth, all these wireless standards work on the similar principle.

Bluetooth, 2G, 3G, 4G and upcoming 5G, all these standards have the information in the baseband. This is where your data bits will be packed using some modulation scheme. And this is going to be converted to the passband, whatever is your LO frequency you have. And your baseband signal will be around the LO frequency and it becomes passband.

So, this is how you convert your baseband signal to your passband signal and you transmit using an antenna. And to do all that, you need a clock which provides you a single frequency f_{LO} to convert to a particular band in the frequency spectrum and transmit it. So, this is where you need your clock. On the opposite side, on the receiver side, this is your receiver, on the receiver side, you will be receiving a similar signal at your LNA which is your low noise amplifier. You receive the signal, you multiply with the similar LO frequency. So, these two things are same. You multiply it and you down convert, and here is your baseband signal which you recover. So, on the transmitter side, you converted your baseband signal through a mixer to your bandpass signal or passband signal, and on the receiver side, you down converted this signal. And this happens in all kinds of wireless receivers.

Now, an important part here is that if you look at any standard, for example, you look at GSM standard. In GSM standard, you are going to have a certain frequency band which is allocated for the particular standard. Within this band which is allocated for GSM, you will have many sub bands. When I say sub bands, you will have many such bands like this. This is in frequency. This is a spectrum which is allocated for a particular wireless standard.

Now, within this band, what will happen is, each user using GSM in a given area will be allocated the frequency band. So, you will have user number 1 here, then you can have maybe user number 2, and so on. So, user number 1, in case of GSM, will have $2\Delta f$ of only 200 kHz. So, you have 200 kHz bandwidth.

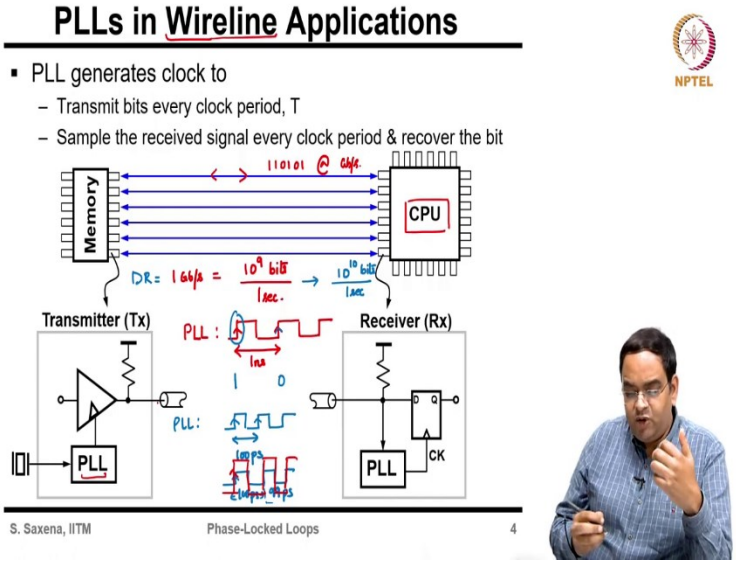
And there will be multiple users which are sitting close by. And each user is given a certain bandwidth in a given area at a given instant of time. So, everyone uses the same phone, it is the same device, whichever company you get, it is not that when you buy any phone, your frequency spectrum is decided. This does not happen. It is allocated dynamically.

So, your same phone at some point of time will be generating the frequency $f_{LO}+2\Delta f$ because the phone has to up convert the baseband signal of each user to different LO frequencies. So, you may have f_{LO1} as this, $f_{LO2} = f_{LO}+4\Delta f$, and $f_{LO3} = f_{LO}+6\Delta f$. So, each user at a given instant of time in a given area will be using different frequencies but it is the same device, purchased from the same company, from the same vendor also, and for each device in the same area, they can have different frequencies.

So, the clock generator which is inside each phone is generating different frequencies as desired. Now, you just think about it, we initially told that you need to generate a periodic signal. Now it is not like you have to generate only one periodic signal. You have to generate periodic signal with different periods as desired. And whichever block is there which is doing this job, that block should be capable enough to do that.

So, this is what happens in wireless applications and clock generation is one of the integral parts of your wireless applications. If your clock is not good, it will be a real challenge to transfer the data from the baseband to the passband and then receive the signal back. It is not possible.

(Refer Slide Time: 15:34)



Next, you have wireline applications. Now, what is this wireline when the whole world is talking about wireless? Well, without the wireline you cannot think of downloading or uploading the things from your computer in Gigabits per second. Wireline application is about transferring data over a wire, but here the data rates are pretty huge. So, you have a CPU, which every smartphone also has, and you have a memory. You have to transfer data from CPU to memory both the ways. And this communication will be happening, the bits may be 1001, any kind of bits, any kind of sequence you have, but these bits are transferred between the CPU and the memory at very high rate running in Gigabits per second also. One Gigabit per second refers to 10^9 bits transferred in 1 second. So, you are transferring at that rate. Now, who is making sure that you are receiving 10^9 bits per second? There is a clock generator on the transmitter side and there is a clock generator

on the receiver side also. This particular phase-locked loop is generating a clock, or you can say a periodic signal every nanosecond.

When I say a periodic signal every nanosecond, it is, in a simple manner you can think of a square wave whose rising edge comes every nanosecond. So, what happens here? At every nanosecond, whenever you have the rising edge of the clock, you say, okay, pass the bit, whether it is 1 or 0. So, you say pass the bit, so, you pass 1 here. Now, if the next bit which needs to be transferred is 0, then at the next rising edge you pass 0, you transfer the data.

So, if the same data rate which we are looking at 10^9 bits/s, if I change this to let us say, 10^{10} bits/s which you like to have. These days we are looking for as high data rate as possible. Data rate is number of bits transferred per second. As high the data rate you want, you need to generate a clock at that high data rate. So, from 10^9 bits/s, when you are transferring 10^{10} bits/s, then you need a clock whose period is 100 ps. This should be your new PLL which generates at that data rate. And just think about it, the PLL is generating a rising edge at every 100 ps. And if this particular PLL misses 1 ps in every alternate clock period, for example, misses 1 ps means that you have one cycle as 100 ps and other cycle as 99 ps.

If it happens that your clock generator gets corrupted, anything, any device can behave badly. If that happens, then you lose the information, whether you lose the information while transmitting or you lose the information while receiving. If you lose the information, what happens? Your data gets corrupted. So, you have to maintain some kind of, we will see what exactly in terms of clock properties during the course, but you cannot have such kind of clock period which keeps on changing all the time. Because it is not only about the clock period which keeps on changing, because a lot of things depend in your wireline applications, all the applications, a lot of things depend on the clock period. I do not care in this case whether I get the rising edge like this or whether my amplitude is this or, at least for now, my amplitude is this. Till the time I am getting the rising edge at every 100 ps, it is good.

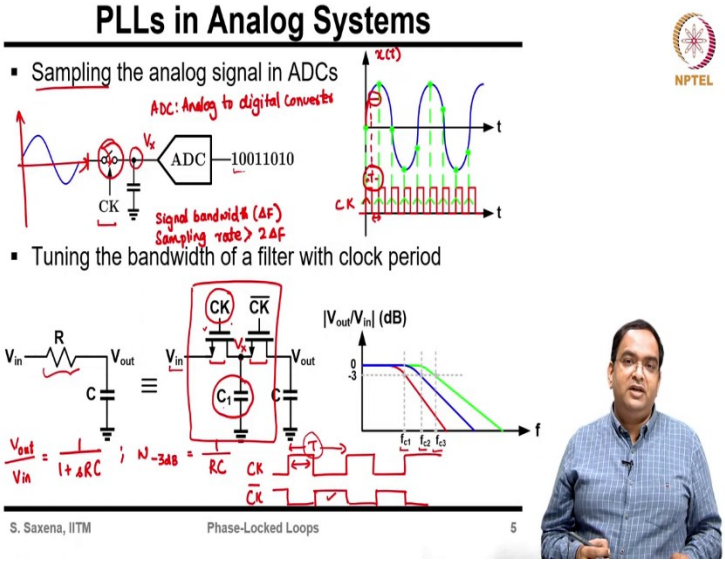
Because I can say that I have the rising edge of the clock and you can transfer the bit or you can receive the bit. So, this is the place where in wireline applications, when you want to do the communication at a very high speed, it is not only that you have to use a clock at very high speed. It is like, when you are using at a very high speed, challenges become a lot more to generate the clock, to generate a periodic signal at that high data rate.

So, you can think about it now, that the mobile phone which you are using or any of the wireless device which you are using, which is communicating with the outside world, either by Bluetooth, WiFi, ZigBee or 2G, 3G, 4G, they are all using a clock, which is designed by one of you guys who are working in the industry. And similarly, on the same mobile phone, you are communicating with the outside world through wireless standard.

But within the phone, you are communicating through wireline method of communication, where your processor communicates with the memory, you process anything, you want to do any calculation, all these things are happening in the processor, in the software. When they are happening in the software, they fetch the data from the memory to the CPU. And that communication is happening with the wireline and you want it fast, well, it will be done fast if your processor is running at a very high speed.

You may sometimes see that when you give some kind of a more cumbersome program to your processor, things heat up. Why do the things heat up? When it tries to process the stuff at a very fast rate, when we say fast, it refers to the clock at which you are doing it. So, in literal terms, you can say it is fast, in technical terms, you can say the clock is at a very high rate, clock has a very high frequency, or the period of the clock is very low. So, these are the two applications at a very broad level where you are going to use a clock.

(Refer Slide Time: 23:17)



Next, it is not only this, well, there are many such applications if you look at very basic analog systems. You know that all the signals which you have in this place around you, they are analog in nature. Is it not? So, when you speak into your phone, your voice is an analog signal that gets converted to a digital signal, like bits, like 1010 and that is facilitated by a block called ADC which is analog to digital converter.

So, ADC here stands for analog to digital converter. Now, this particular analog to digital converter, how does it convert the signal from analog to digital? The first thing which comes is your analog signal is sampled using a clock. It is later getting converted to digital signal. So, for example, you have this sine wave, this sine wave is going to be sampled as shown on the right.

So, here I am having this sine wave, incoming analog signal, x(t) with respect to time. I have a clock which is shown here, CK. At every rising edge of the clock, the switch closes. And when the switch closes, whatever value of the analog signal you have at the input, the same signal comes here. And when, so, what is going to happen here is, in this particular case, if you look at it, for the example which you have, for the rising edge of the clock the switch closes.

When the switch closes, you have your output here, which is voltage V_x, this voltage is going to be same. And then, when the switch opens when the clock is low, at that time whatever value you have, that value is held. A capacitor, you have some voltage on the capacitor, you leave the capacitor open, ideally the capacitor is not going to lose any charge. So, that voltage is held. Now, similarly, you are doing this, so, when you have this held voltage, you go and characterize this held voltage.

And you say, okay, this voltage is equal to 110 something, you digitize the signal and when you digitize the signal, you can represent in terms of bits. So, the fundamental operation which lies behind the working of ADC is this sampling, sample the analog signal. And this analog signal is sampled at a fixed rate, it depends on what rate you want to sample it, it should be just greater than the Nyquist rate.

So here, every time period T, I sample the signal. So, technically, the signal which you are sampling, the sampling rate should be twice the bandwidth of the signal. If the bandwidth of your signal increases, the sampling rate will also increase and if the sampling rate increases, it means your time period T will come down.

So, what do you have? Your signal bandwidth, whatever signal bandwidth you are sampling in your ADC, if signal bandwidth is Δf, your sampling rate should be greater than $2\Delta f$. This is the Nyquist criterion. So, you need a fixed clock to sample the input signal. So, you need a clock there. Then, if you look at another application of the clock, in the other application of the clock comes the digital tunable filter.

So, what is the tunable filter? If you look at the filter on the left-hand side, you have V_{out}/V_{in}. In this case, it is very simple. It is $\frac{1}{1+sRC}$. The filter bandwidth here is $\frac{1}{RC}$. So, when you want to design a filter, a simple low pass filter, you will choose the resistor component, you will choose the capacitor and you will design the filter. And this particular filter is going to have -3 dB bandwidth as $\frac{1}{RC}$. Suppose you want to change the filter bandwidth, what will you do?

You have to change R or C, that is one way to do it. On the other side, this switched capacitor here, this is a capacitor C_1 and there are two switches, 2 MOS switches which are used to replace the resistor in this particular case. When they replace the resistor in this particular case, depending on the clock period or you can say depending on the clock frequency, you will be switching this capacitor C_1 to V_{in} on half the clock cycle. When the clock is high, when CK clock is high, at that time your intermediate node V_x will be connected to V_{in}.

When your clock is low, so, let me just first tell you what this clock high and clock low are. So, you have a clock CK. When clock is high, I am talking about this period. So, if you look at it, if you understand the MOS switches, you will know, when the CK voltage is high, NMOS will be turned on. When CK clock voltage is low, NMOS is turned off. So, the switch on the left turns on when CK is high. $\overline{CK}$ is nothing but just the inverted of this signal.

So, when CK is high, NMOS connects the V_x node to V_{in} and V_{in} comes at V_x node. When CK is low, at that time $\overline{CK}$ is high, when $\overline{CK}$ is high, your V_x node connects to V_{out}. So, whatever charge you have on the capacitor C_1, that charge gets transferred to C. And after doing some maths, you can find out that depending on this time period T, we are not going to do this maths right away, but this is just to tell you that depending on this time period T, your capacitor C_1 is switched back and forth between V_{in} and V_{out}.

So, just think about it, you have a capacitor, you are switching back and forth between 2 supplies. So, it is taking charge from V_{in}, transferring to V_{out}. And if it is doing it more often, that means, it is taking charge from V_{in} more often, more often in terms of time period means more often is time period is less. Every clock cycle it does it, if I say it does more often, that means within the same time, it is doing more number of times, that means that clock period is low.

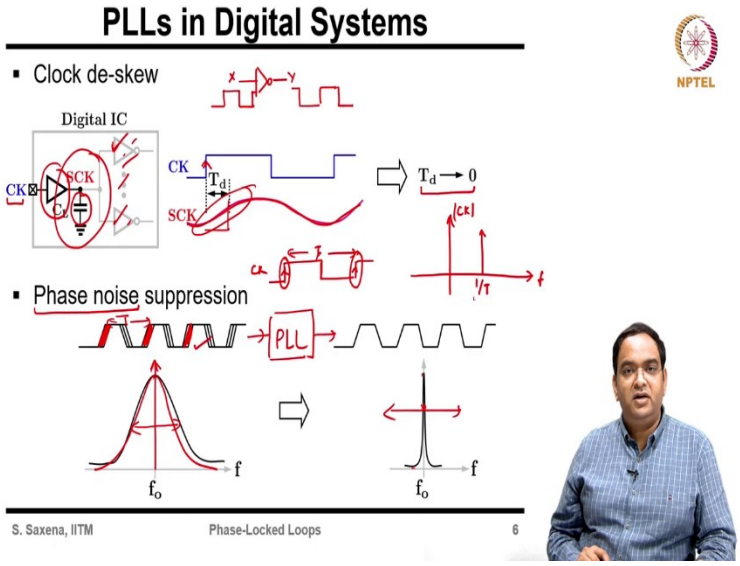
So, it is taking more and more charge from V_{in} and transferring to V_{out}. The average charge which is drawn from the supply and transferred to the capacitor C is going to be more. So, in literal terms you can say, if I am using some component and I am drawing more per unit, more charge per unit time, effectively it is behaving like a low resistor.

Now, think back that here the average charge which is drawn from V_{in} and transferred to V_{out} depends on how often you are doing it which in turn depends on the time period T. So, at the end, the effective resistance value of this particular network depends on the time period because the resistance can vary with the time period, the filter bandwidth from V_{in} to V_{out} varies with the time period.

So, whenever you would like to use a filter where you do not want to change physical value of the resistor or capacitor, but you want the filter bandwidth to vary, like between f_{c1}, f_{c2} to f_{c3}, you can use such kind of architecture. And your filter bandwidth will vary depending on the time period of the clock which you are using to clock this circuit.

So, in just the last four slides, what I have shown to you is that the clock is of significant use in your wireless applications, in your wireline applications, in the basic analog to digital converter, and by the way for digital to analog converter also, and for tunable filters. The list does not end here.

(Refer Slide Time: 33:57)



Next, you see the use of PLL in digital systems. So here, when you have an input clock which is a periodic signal and you have to drive a lot many gates, see clock will come, clock may feed through multiple blocks, maybe flip flop, maybe inverter, a gate, logic gates or some other thing, it will do that. So, when you are using a single block, no matter how large it is, you are driving, you may have to drive even much larger load as shown by just inverters here.

What is going to happen? You were feeding clock here at the input, because there is a lot of parasitic loading here, the actual clock signal changes. It is simple RC delay, charging discharging delay. When this clock signal changes, what happens is, if you understand the operation of an inverter, what is the operation of an inverter? You feed at the input, signal like this, X, you will get at the Y an inverted signal. That is what we understand from the ideal operation of an inverter.

But what happens here, the signal is so much skewed from 0 to V_{DD} for the inverter that your transition point for the output will vary. So, in this particular case, what the PLL will try to do is, it will somehow, again this is to be seen during the course, somehow, it will try to make, T_d, the delay which you are having between the rising edge of the input clock and the clock which is actually driving your digital logic, 0.

How? You have to add some kind of block here which is associated with the PLL which will enforce this. This is where you get the clock. The other term which you will have to deal a lot during this course is phase noise suppression. And PLL helps in reducing the phase noise. This is

something which needs to be seen and understood which will come during the course that sometimes you have a clock which is really noisy.

So, I will tell you the meaning of the noise. When you have a single clock like this, when I say a clock is clean, you have a simple square wave with time period T. If you look at the frequency spectrum of this clock signal, you will have in this case, you know that this is a periodic signal, if it is an ideal periodic signal with time period T, you can find the frequency content using the Fourier series.

You will have the fundamental at $\frac{1}{T}$, then you will have the other harmonics at $\frac{2}{T}$, $\frac{3}{T}$, and so on. So, that will happen, but nearby $\frac{1}{T}$, you will see a single frequency content. But, if the signal is noisy, then what happens is, each edge of the clock signal does not come at the same time, same absolute time.

So, that is what I have shown you here with a variation in every rising or falling edge of the clock. This is the variation, the variation is not fixed, it is random in nature. And when this variation is random in nature, you look at the frequency spectrum of the signal, you will see that the signal does not have a single tone. There is some average time period associated with it.

It has a dominant tone at f_{LO} but it is surrounded by many other tones. When does this happen? Let us say, when you are transferring your clock or you are sending a clock from one end of the room to the other end of the room through some wire, this can happen. So, there are many places where these things happen. So, the actual signal, the ideal signal gets corrupted.

Here, the corruption of the signal means that the rising edge of the signal is not coming at the same rate, it is getting changed which in frequency domain appears as a spreading of the signal. So, what the PLL block does is, you connect a PLL and you will get rid of this noise. And your signal will become quite clean, the spread of this gets minimized to this. That is what the PLL does.