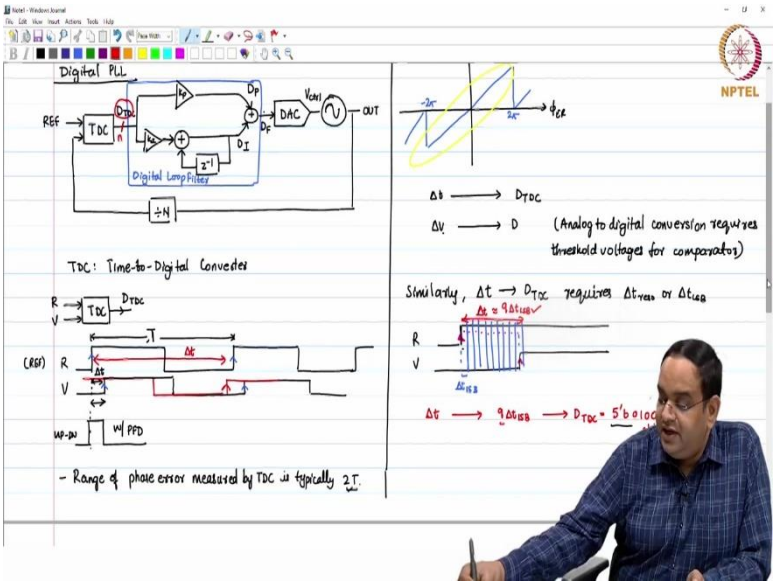
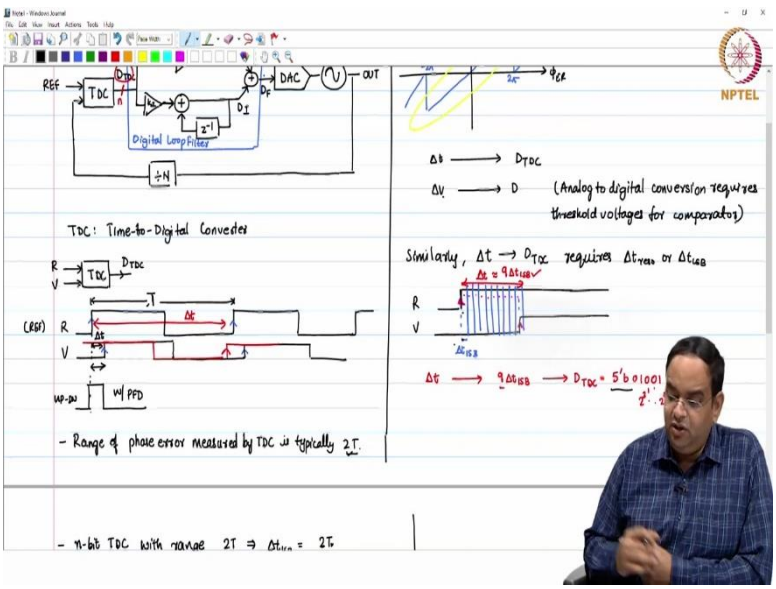
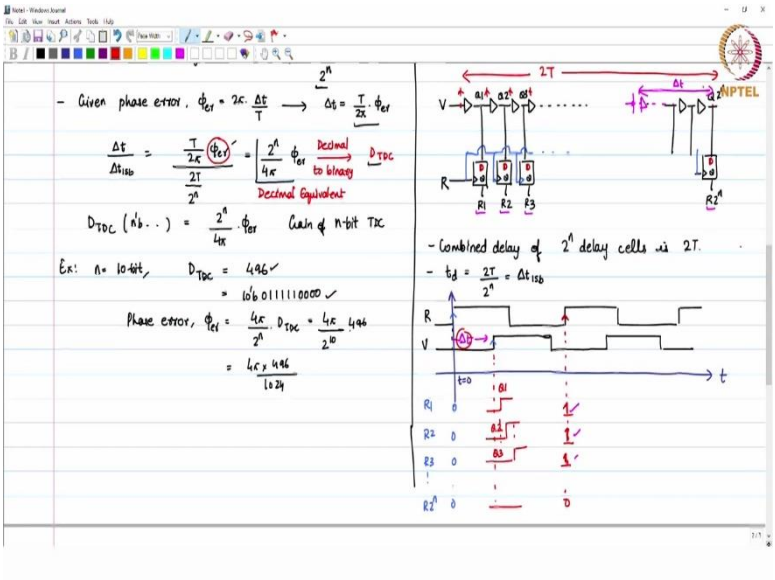
Phase-Locked Loops
Dr. Saurabh Saxena
Department of Electrical Engineering
Indian Institute of Technology Madras

Lecture – 60
Design of Time-to-Digital Converter

(Refer Slide Time: 0:16)







Hello everyone. Welcome to this session. In the previous session, we arrived at the simplified block diagram of a digital PLL. So, let me just draw that. So, in the digital PLL, we have a TDC which converts the phase error to digital bits. We will look at how it works and the output of the TDC goes to a digital loop filter which has the gain of proportional path K_P and the integral path gain K_I and here you feed it to an accumulator. I will write that with its small signal diagram you can say or the block diagram.

So, you have an accumulator like this where this is z^{-1} which models the delay and this is the accumulated value. The output of the proportional path and the integral path, they add together to give you the final control word which is going to control your oscillator and then you have digital-to-analog converter which converts the digital word to the control voltage. This controls the oscillator and then you can have the feedback divider as required for frequency multiplication.

This is your reference frequency and this is your output frequency. The output of the TDC, I call this as D_{TDC}. The output of the proportional path, this is D_P, and the output of the integral path is D_I. When you combine them, you get the digital control word D_F and D_F gets converted to the voltage V_{ctrl} and that is your output frequency. So, the output of the oscillator is still a voltage waveform as it used to be, we cannot change that. Similarly, the input of the TDC which is reference clock and the TDC clock, they are also voltage waveforms.

So, here if you look at it, by the way I will just put everything here in one single block and call it as it is normally called a digital loop filter. So, here TDC is time-to-digital converter and how it works, let us look at it. So, the block TDC has two inputs, the reference and the feedback and

the output is a control word or a digital word which is D_{TDC}. When we say it is a digital word, it is like a x-bit word.

So, if we have these two signals R and V like this, R is the reference signal which is like a reference coming from a crystal or some other source, you have a reference clock. V is the feedback clock, it can be the same frequency. Of course, different frequency also if it is there and what we are going to do is we are going to calculate this error between these two rising edges as was done in the case of PFD.

So, if you remember, in the case of PFD, you had $UP - DN$ and depending on your $UP - DN$, you get this error whether it is like this or in other case $UP - DN$ can be negative also, that was the case with PFD. Now, we do not have the PFD. What we want to do is that we want to use this Δt time and convert the Δt time into its digital equivalent. That is what the TDC is doing.

So, here you have a reference clock period which is T. So, the clock which acts as a reference in the measurement of time is your reference clock itself. So, this Δt error is now digitized in some digital word. So, the first thing which I will tell here is that in the case of PFD, if you remember, you measured the phase error and it was like -2π to 2π. It was a linear transfer function.

So, similarly, here if one case you are seeing the error is this. In the other case, I can have the rising edge of the clock maybe coming like this, so late, just an example. So, the rising edge is delayed and it is delayed by Δt which is closer to you can say one reference period. Similarly, you can have Δt where the rising edge on R comes after the rising edge on V and that also period can range from small Δt to a large value of Δt.

So, here the range of phase error measured by the TDC is typically two reference periods. So, you can say $-T_{REF}$ to $+T_{REF}$, that gives you $\pm 2\pi$. That is the maximum which you can have or which you will use in general.

Now, this Δt is digitized to the control word D_{TDC}, this Δt timing. So, in order to digitize this Δt timing, you need reference or some resolution. So, this Δt, I need to digitize. So, when you have a voltage for example, ΔV and you want to convert it into discrete value D, what you need is a reference voltage or whatever is your comparator threshold. This is like analog-to-digital conversion. Analog-to-digital conversion requires threshold voltages for comparators.

Similarly, Δt to D_{TDC} requires your resolution Δt_{LSB} or Δt threshold, Δt resolution or Δt_{LSB} to convert from or you can say this Δt to D_{TDC} requires some kind of Δt resolution or Δt_{LSB}. What it means is the following. If the phase error is something like this which is just an example, this is R and let us say this is V and this separation between R and V has multiple or you can say multiple slots. Each slot is like, each slot is Δt_{LSB}. If such is the case, then you can count how many slots do we have between these two rising edges. 1, 2, 3, 4, 5, 6, 7, 8, 9, so, 9 complete slots and the 10^{th} slot is half.

So, depending on how you are giving the compared value, you can say the separation between these two rising edges is or this particular value, this $\Delta t \approx 9\Delta t_{LSB}$. So, this is something which you need. Based on this, you can convert this 9 to a digital word. So, if I have 9 as the Δt_{LSB}, my D_{TDC}, I can very well write for above case Δt is actually when I found the separation between these two rising edges, it gave me this is approximately equal to $9\Delta t_{LSB}$.

So, if there is some information about Δt_{LSB}, then taking that into account, I can say my D_{TDC} is nothing but here let us say I am having a 5-bit word. So, I will write 5'b, this is just a nomenclature we normally use and this is 9 here. So, 01001. So, this digital word, the placement values are 2^0, 2^1, 2^2, 2^3. So, you have the decimal equivalent as 9. So, this is how you are going to convert.

So, if you look at it, what we need from the TDC block is the information about what is the Δt_{LSB} and the range which we are going to measure with the TDC normally, max that will be -1 reference period to +1 reference period like the one which we had here. -1 reference period is -2π, +1 reference period is $+2\pi$. So, the range of the TDC is $2T$ and what you have here is the Δt_{LSB} information which you need.

So, if an n-bit TDC with range $2T$ implies that the resolution Δt_{LSB} is given by,

$$\Delta t_{LSB} = \frac{2T}{2^n}$$

So, you know n-bit TDC, how much is the maximum value? So, you can say this is approximately 2^n values you will have. So, this is your Δt_{LSB} and the digital word which you are going to get, that digital word will be having an equivalent binary value.

So, what I am going to get? How am I going to write the performance of the TDC? So, given phase error, now you have to because you are doing it digitally, so you have to specify whether

you are going to have positive or negative phase error, so here let us say we take that into account later. The whole $2T$ range which is from $-T$ to $+T$ is given by n bits.

So, given phase error as,

$$\varphi_{er} = 2\pi.\frac{\Delta t}{T}$$

This phase error is converted by the TDC to you can say the digital equivalent of, so, you can always write,

$$\Delta t = \frac{T}{2\pi}.\varphi_{er}$$

So, the decimal equivalent of this value is given by,

$$\frac{\Delta t}{\Delta t_{LSB}} = \frac{\frac{T}{2\pi}.\varphi_{er}}{\frac{2T}{2^n}}$$

$$\frac{\Delta t}{\Delta t_{LSB}} = \frac{2^n}{4\pi}\varphi_{er}$$

So, here the thing is that I have taken the range 0 to $2T$. You can take the range $-T$ to $+T$ also. The only thing is you have to then take into account the positive and the negative, we can take that into account in digital case also. So, this TDC is if you have a phase error φ_{er}, this phase error φ_{er} is converted to its digital equivalent as you see here, by the way this is decimal equivalent. Whatever value you get here, the phase error is actually that value into Δt_{LSB} but this is what you are going to get from the TDC. This is decimal equivalent and this you do decimal to binary so that you get D_{TDC}.

So, if I write D_{TDC} as, though it is a binary value, you can say this is like n-bit whatever number you have. This binary value is $\frac{2^n}{4\pi}\varphi_{er}$. So, if your phase error is 4π, 4π is your $2T$, 2 times the time period, your decimal equivalent is 2^n. If your phase error is 0, decimal equivalent is 0 or binary equivalent is 0. So, this is effectively the gain of n-bit TDC in general.

So, when you use the TDC here in this PLL, you have a phase error and this is a n-bit TDC and the n-bit spans the range $-T$ to $+T$. In that case, whatever the D_{TDC} value you are getting, I told that here I have been using this dashed line here, this dashed line just means that there are

n, there are multiple signals. So, if I want to write this as an n-bit TDC, I will say n here. So, this n-bit TDC gives you this D_{TDC}.

So, if you just look at independently the value of n-bit TDC and n-bit TDC equal to some value in terms of 1000 or other things. So, for example, if n is 10-bit for example, n=10 and $D_{TDC} = 496$. This is the numerical value, the decimal value. In binary case, you can say this n-bit TDC, the decimal value is equal to this is going to be 0, then you have 256. So, 10 bits will be there, you will have 256, you are going to have 128, then you will have 64 and you will also have 32 and rest all are going to be 0.

So, what you see here is that the digital equivalent is going to be this whereas the decimal equivalent is this. Here 496 does not mean the error is 496 times the reference period. Well, this 496 means phase error is actually, so phase error is given by,

$$\varphi_{er} = \frac{4\pi}{2^n} . D_{TDC} = \frac{4\pi}{2^n} . 496$$

$$\varphi_{er} = \frac{4\pi \times 496}{1024}$$

This is the phase error value you will see.

So, what is the meaning of the D_{TDC} output that is clear by now that it measures the timing difference between the reference and the feedback clock. What you need is Δt_{LSB}. Somehow you have to get it and once you get that, you have to create this Δt_{LSB} in a similar way as you generate the comparator threshold. Once you get that D_{TDC}, this is what it will mean.

Now, a typical implementation, we will not go that much in detail but a typical implementation of TDC is as follows. There are multiple ways in which you can implement TDC. So, one way is that let us say I have delay line. So, this is the delay line which consists of 2^n delay cells. So, I will name this node as Q1, Q2, Q3 and so on and it goes up to $Q2^n$.

So, all these are delay cells. These outputs are fed to registers. So, if I write it like this, let me just make it a single and then copy paste. So, these are fed as inputs and you have a clock here and this is your output Q. So, this is your register, by the way. So, you have these registers. Normally what you will see is just a single line saying that these are registers, they are not connected by the way. I will just disconnect them in a moment and similarly, you have on all the cases which you have.

So, these are registers which are clocked. So, I am having a clock which connects all of these registers. So, let me just connect this. So, a clock is there which connects all of these blocks. Finally, it comes here. So, I am going to call this as let me call this as R1, R2, R3, R2^n. And this clock what you are seeing here is another signal which is your reference signal and this is your feedback signal.

Now, if I make sure that the delay from here to here, combined delay of 2^n delay cells is 2 reference periods. This is made sure independently. Then delay of each cell is given by,

$$t_d = \frac{2T}{2^n} = \Delta t_{LSB}$$

So, now just think about it, let us say when you are having the reference clock coming like this, this is your reference clock and your feedback clock is coming somewhere here, this is your feedback clock.

So, you start looking, you feed these two signals to your circuit above with respect to time. So, when let us say all the signals start at time instant t=0. Before that, all the signals were actually 0. So, in that particular case, if I look at R1, R2, R3 and so on, I am looking at all these values. So, I will look at R1, R2, R3 and so on, you can say R2^n. So, as you see from here, so these are the registers which sample the value at the rising edge.

So, when the first rising edge on R comes, what you sample is 0, 0, 0 and 0. All of them are 0. Then after some time, you see a rising edge on V coming after a certain delay. If it is after a certain delay that you get rising edge on V, then that rising edge on V is going to propagate through these delay cells after the delay. So, if I look at here, my Q1, Q1 will actually become 1, this is not R1 signal but I am just plotting, this is your Q1 after the delay. Q2 signal will actually become 1 after another Δt_{LSB} delay.

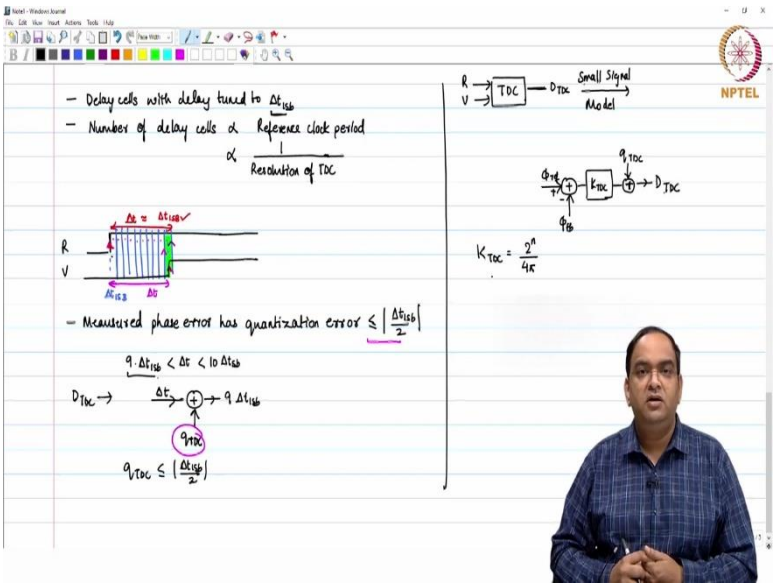
Similarly, Q3 signal will become 1 after, so this is Q2, this is Q3. It will become 1 after some delay. Since the range of this delay line is $2T$, so, till the time you get the rising edge here, you will not be able to get Q2^n as 1. So, then what happens is that a few of them will still be 0 and others will become 1. So, when you get the next rising edge, at the next rising edge, R1 will be sampled as 1, R2 will be sampled as 1, R3 will be sampled as 1 and similarly R4 and other values will be sampled as 1 and some other values will be sampled as 0.

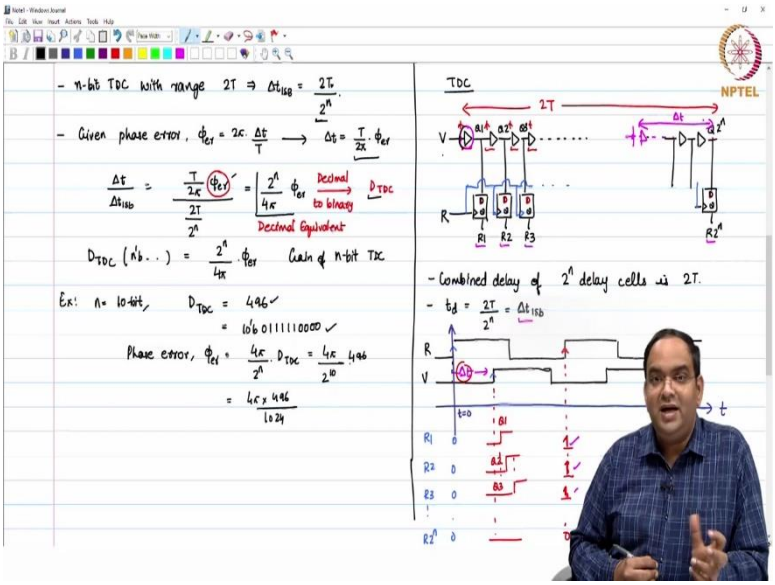
So, you can think about it that the number of flip flops which will register value 1 at the next rising edge, those number of flip flops can, you can say if this whole delay, it will be easy for you to understand like this, if this whole delay is $2T$, depending on how you choose, you can also choose a range of T, this is just one example.

If you choose a range of $2T$, then you can say that $2T$ minus whatever this phase error is, just an example, if this, the delay of the elements till here is Δt, then only the registers which are coming before this, they will register 1, other values will register as 0. So, when you sum up all these register values, you will find that you can easily calculate this delay. So, range is $2T$, you have this Δt, so, whatever value you are going to get at that point, you can actually find what is the decimal equivalent. So, what we need is that whatever these register values you get, these register values are saved and these register values are summed up to give you a representation for this Δt error.

So, what you see here is that we need such delay cells, we need the registers to latch these values. So, this is one simple method. The important part here is that you need to get this Δt. Somehow this has to be tuned, your LSB size is what you require and this has to extend over the whole reference clock period. So, this is a simple implementation. There are multiple ways in which you can implement the TDCs. You can always refer to the literature for that.

(Refer Slide Time: 31:57)





So, if I just want to conclude the TDC part, the thing is that you require delay cells with delay tuned to Δt_{LSB}. This is important because if the delay of these elements which you are seeing here is not Δt_{LSB}, is something else or you do not know, then you cannot comment on what the phase error you have. So, somehow you have to make sure that the delay is fixed or within the margin of error what you can tolerate.

And the number of delay cells which you require is proportional to your reference clock period. Sometimes you make sure that the phase error is, by some other means, by dual loop or something, you make sure that the phase error will not increase beyond this range, you can then limit the range of the TDC. But in general, the number of delay cells is proportional to the reference clock period. It is directly proportional to the reference clock period and inversely proportional to the resolution of the TDC which is also called Δt_{LSB}. So, better the resolution, more the number of delay cells.

Now, you may ask that whenever you are designing any such things, a greater number of delay cells will take more area. So, why would we like to have that? Well, the answer lies in the fact that when you are quantizing the phase error, so you had this phase error, if you go back, so, here if you look at it, the region where your rising edge lies, is this. So, based on your logic, you can say either my phase error digital equivalent is this or count till here like $9\Delta t_{LSB}$ or you can say I will count because it is coming after this, but the edge I have not got, so I will count this.

So, your digital equivalent or you can say the measured phase error has quantization error limited by, that is a better word, it is limited by $\left|\frac{\Delta t_{LSB}}{2}\right|$. So, actually if you were doing it in an

analog manner, then whatever this phase error is, it is exactly this Δt. But what you are saying is my phase error is either the value is $9\Delta t_{LSB}$ which is actually less than the Δt error and if you say it is equal to $10\Delta t_{LSB}$ which is more than the phase error.

So, your actual phase error whatever you are writing $9\Delta t_{LSB}$, if this is the value of D_{TDC}, if D_{TDC} says the actual phase error Δt gets added with the quantization noise q_{TDC} to give you $9\Delta t_{LSB}$. So, the TDC comes with an inherent quantization noise and this quantization noise is limited. q_{TDC} is limited between $\pm\frac{\Delta t_{LSB}}{2}$. So, you are not measuring the phase error exactly but what you are measuring is phase error with an addition to the phase error. In addition to the phase error, what you get is a quantization error.

So, the TDC block which is measuring the phase error between R and V signals and gives you D_{TDC}, the small signal model of this can be represented by, you have, you can write I am having phase error as φ_{ref} and φ_{fb}, I subtract this phase error and I multiply by the gain. What is the gain of the TDC or let me just write as K_{TDC} plus I add the quantization noise q_{TDC} and then I get D_{TDC}. That is the small signal model of the TDC.

If you are not doing any noise analysis, then we can ignore q_{TDC} but this is what exactly happens. And K_{TDC} we have seen, it depends on what range you are implementing with how many bits. So, if we are implementing a range of 4π, then in that case, this is $\frac{2^n}{4\pi}$, this is the K. So, this is how we will detect the phase error to give us the digital word which is used by the loop filter. Thank you.