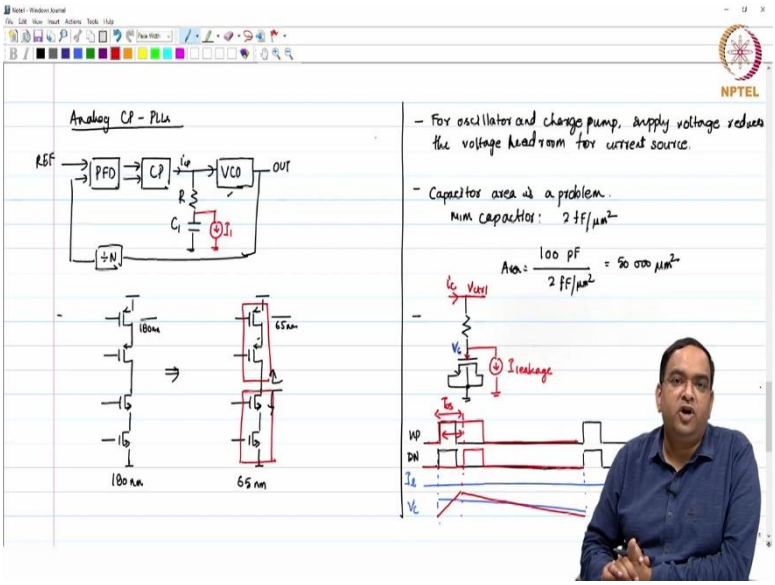
Phase-Locked Loops
Dr. Saurabh Saxena
Department of Electrical Engineering
Indian Institute of Technology Madras

Lecture – 59
Introduction to Digital PLL

(Refer Slide Time: 0:16)



Hello, welcome to this session. In this session, we will start looking at the drawbacks of analog charge-pump PLL and as a solution to the problems in the analog charge-pump PLL, we will introduce digital PLL and look at how we design digital PLLs. So, the analog charge-pump PLL which we are so used to now is something like this. You have PFD, charge-pump followed by loop filter. For easier analysis, we will use only Type-II loop filter here and this is VCO, divider and feedback like this and you have your reference here and this is your output.

So, the problems with the analog charge-pump PLL are as follows. Here when you design a PLL in a given process technology, let us say 180 nm and when you would like to port, when let us say same frequency is needed in 65 nm technology, this kind of PLL is difficult to port from one technology to the other technology. So, let us look at the blocks which you can port. So, the output frequency is the same but what happens is your technology changes. So, when your technology changes, your PFD which consists of NAND gates and other combinational logic, that is easy to port.

Your charge-pump consists of your current sources. Now, this charge-pump, I will just show you the example. So, what happens is when you have a charge-pump in a given process which

is like a switched current source and when you change this from 180 nm process, this is not 180 Volts, so let me just write it, this 180 nm to, when you port this charge-pump to 65 nm for example, then here what happens is your minimum length which you are using here as 180 nm, this particular length actually changes and can be 65 nm. You can still retain at 180 nm, there is no problem in retaining 180 nm but the benefit of going to a lower process is the smaller area for the same block. So, this problem will come.

So, when you change it to 65 nm, the charge-pump relies on the fact that the top current source and the bottom current source, they are exactly matched and it is a known fact that as you go down the channel length or as the channel length reduces, the matching which you will find between these devices, the up and bottom current sources, that will become more and more difficult if you are just reducing the length. This is like the short-channel effects.

So, the idea is that the charge-pump operation relies on the output current or an analog value and as you scale down the process, what you will find is it is difficult to maintain the matching between the top and bottom current sources for the reduced channel length. And if I say even in the reduced technology or process node, if I am using the same length, then what is the point of scaling the charge-pump down to the lower process? That is something which you can always give as a counter-intuitive but a counter to this.

Now, the thing is that this charge-pump has the issue. The other thing is the VCO, if you have an oscillator ring VCO which is operating at 1.2 GHz in case of your 180 nm and you want to implement the same oscillator in your 65 nm process, the oscillator will also scale. But in both the cases, for the oscillator and the charge-pump, the supply voltage reduces the voltage headroom available for current sources in these blocks.

Now, for oscillators it is not a problem because the device length reduces, so the maximum frequency typically increases. So, achieving the same frequency is not a problem in the lower process. But the problem is with the charge-pump and the problem is one thing is you say I am not getting the matching and if you are not getting the matching with a single transistor, you would like to increase the output impedance looking in by having a cascode. You cannot have a cascode because your supply voltage is limited, so the charge-pump faces a real issue. So, this is the problem with the analog charge-pump based PLL as you port the process.

The other problem is that this capacitor which you have, this capacitor occupies a lot of area for a good output jitter. So, the capacitor area is I will say it is a problem. Why is it a problem?

Because as you go down to a lower process, we all know that lower technology nodes, they are much costlier and you are implementing the same kind of MIM capacitor, metal insulator metal capacitor whether it is 180 nm or it is 65 nm or maybe 28 nm or something.

So, the capacitor density generally does not increase as you go down to a lower process, the capacitor area is large, and when the capacitor area is large, actually it is going to cost more as you go down to a lower process. So, the area of the capacitor remains fixed from 180 nm to 65 nm but at the same time, the cost increases. So, this is not a good idea as such.

Now, this capacitor, the metal insulator metal capacitor, MIM capacitor which we normally use in a CMOS process, it has a typical density of you can say 2fF/μm^2. And you think about it, if you are implementing a 100 pF of capacitor, what problems will you face? The area is going to be large.

So, just a typical example, I want to implement 100 pF by using this 2fF/μm^2, so the area is going to be 50,000 μm^2. So, 50,000 μm^2, you can say roughly it is 200-220 micron area. 100 pF is not that much. Many a times you would like to have a much larger capacitor for better phase noise at the output.

Now, the substitute for this capacitor is using a MOSCAP. So, in place of using MIM capacitor, metal insulator metal capacitor, someone can say if the density is a problem, I am going to use a MOSCAP which also scales down with the process. This particular MOSCAP has a real problem that this MOSCAP comes with a current leakage and what is this $I_{leakage}$? This is the gate current leakage. So, I am just modeling the MOSFET used as a capacitor with a leakage current, this is your loop filter which you have.

Now, if you have this current leakage, well, what is the problem? Well, the problem is if I say that I have a constant gate leakage current here I_l, writing leakage is too much. So, I will write this I_l. So, now you know that during your UP and DN pulses in the charge-pump, this I_l is present everywhere. So, if this I_l is present everywhere, you can think about it that this control voltage will always keep on reducing.

So, if I think that I am going to have these overlap pulses as overlap between UP and DN such that it locks to a zero phase offset, ideally this should happen, but what happens is this I_l current comes in and your V_c keeps on dropping. If V_c keeps on changing, V_{ctrl} also keeps on changing. So, I am just showing you that there is a constant current I_l which is present all the time. It may

be very less but the thing is as your capacitor area increases or the capacitor is large, your leakage current is also large. So, this is I_l current, it is always there.

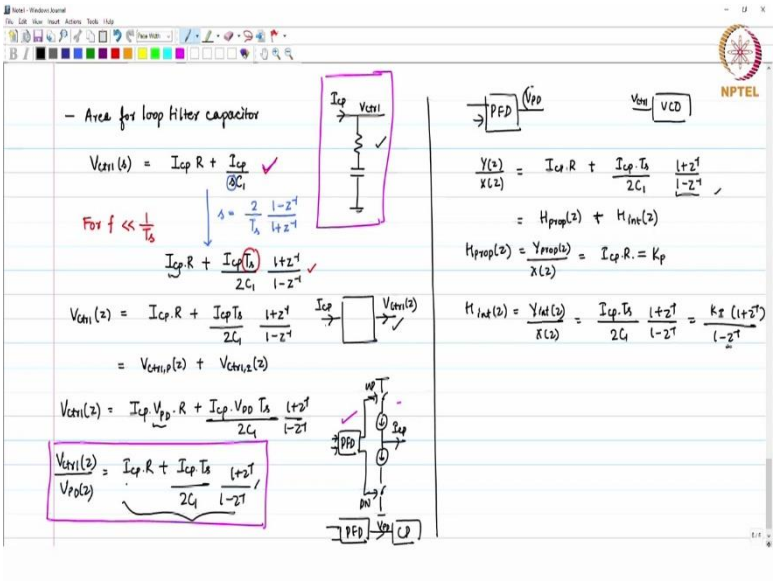
So, your V_{ctrl} voltage will always keep on decreasing or increasing, that depends on the leakage current, it is positive or negative. So, just an example, I am plotting that your voltage always keeps on decreasing and if this thing happens, then you know that it is not possible for the PLL to be locked in this state where your control voltage is always varying. So, this will not happen and if this does not happen, what will happen is that you will have a phase offset in a real case. So, if I am going to have a phase offset, my UP and DN pulses will be like this and you are going to compensate your gate leakage current with phase offset here.

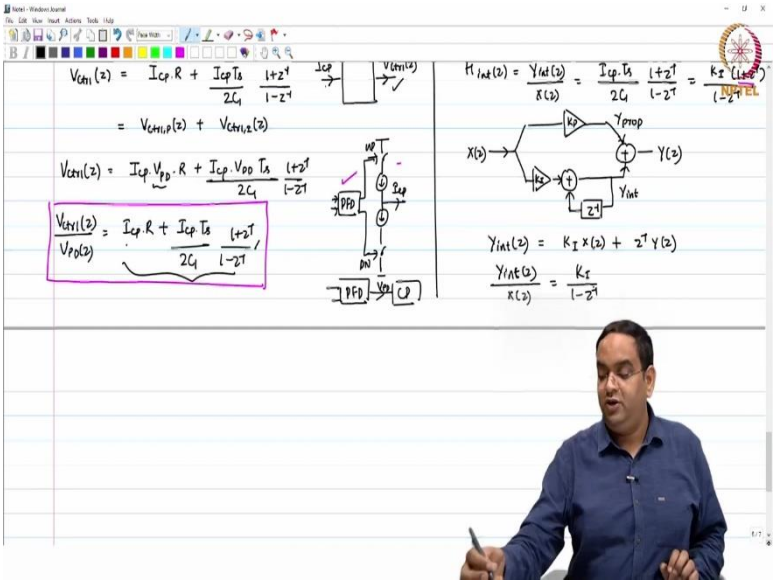
What will happen then? Just a change here, during this time you will increase this potential every time and during the other time, you are going to reduce the potential. Your I_l remains fixed. The only thing is you get extra current from the charge-pump during the time when UP is high and DN is low. So, this thing is going to happen and this kind of control voltage V_c and similarly your V_{ctrl}, this kind of control voltage will increase the reference spur value at the output.

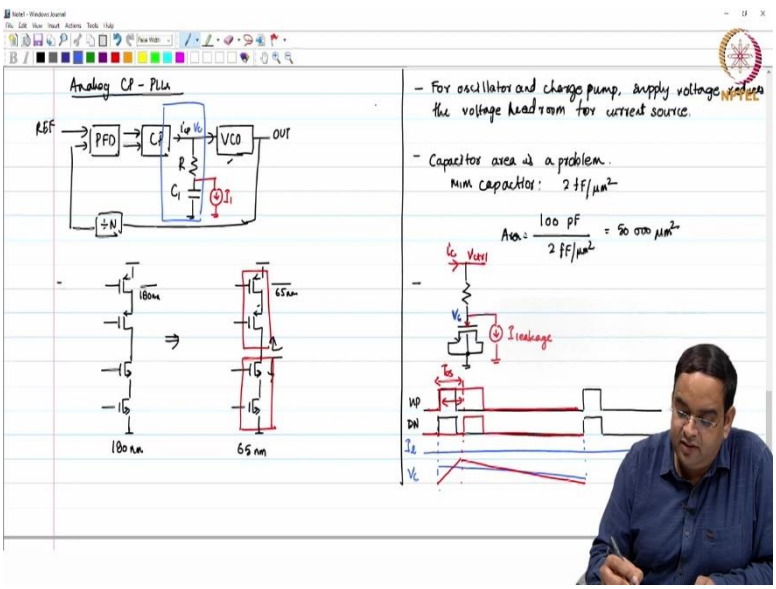
So, using a MOS capacitor in place of a MIM capacitor is surely not an option because that gives a lot of offset at the input of the PFD which is going to increase the spur value at the output. So, this is an issue. So, what we can say is, in summary, the analog charge-pump PLLs are difficult to port from one technology to the other technology for multiple reasons. One, your charge-pump, the matching between the UP and DN current sources, that matching goes bad as you go down to a lower technology. The other thing is your capacitor area which you are having, that particular capacitor area is a lot, it costs more in lower technology nodes.

MOS capacitor is not an option because of the gate leakage current and your charge-pump also has a problem with the reduced voltage headroom to operate while still maintaining the matching between the top and bottom current sources. So, this is the problem we have.

(Refer Slide Time: 14:38)







So, we are going to address these problems one by one. The first one, let us look at it, the area for the capacitor. So, this is a problem which we need to address, the area for the loop filter capacitor. The other one which we are looking at is the charge-pump issue. We will find that while solving the problem for the area for the loop filter capacitor, that problem will also get solved with some additional inputs.

So, here the idea is that this particular block, the charge-pump plus your loop filter capacitor is implementing at V_{ctrl}. The transfer function which it is implementing is given by,

$$V_{ctrl}(s) = I_{CP}R + \frac{I_{CP}}{sC_1}$$

Now, we are going to replace this s here with its digital equivalent using bilinear transform, and what is this bilinear transform by the way? Bilinear transform is given as,

$$s = \frac{2}{T_s}\frac{1 - z^{-1}}{1 + z^{-1}}$$

If I do this, then we get,

$$V_{ctrl}(z) = I_{CP}R + \frac{I_{CP}T_s}{2C_1}\frac{1 + z^{-1}}{1 - z^{-1}}$$

So, ideally speaking what we have done is, we have just converted this analog transfer function to its digital equivalent. For frequencies much lesser than the sampling rate which I am using to implement this digital transfer function, for $f \ll \frac{1}{T_s}$, these two transfer functions will have the same input-to-output response.

So, now I will just write this in terms of V_{ctrl}, from Laplace domain I will write in terms of z-domain. I have the transfer function as,

$$V_{ctrl}(z) = I_{CP}R + \frac{I_{CP}T_s}{2C_1}\frac{1 + z^{-1}}{1 - z^{-1}}$$

This particular transfer function can be implemented using two paths. So, we get,

$$V_{ctrl}(z) = V_{ctrl,P}(z) + V_{ctrl,I}(z)$$

So, ideally speaking, this $V_{ctrl,P}(z)$, if you think about it, what is this transfer function doing, $V_{ctrl}(s)$? This is I_{CP} times R, so whatever your phase error you have at the output of the PFD,

whatever your PFD output is, that PFD output, the phase error, PFD detects the phase error and your PFD output is getting multiplied.

So, this transfer function $V_{ctrl}(s)$ can be written as, let me just come to the block because this is a standalone value where I am just considering this loop. What I am doing here is, I am saying this is I_{CP} and this is V_{ctrl}. So, this particular analog transfer function is now represented by a digital transfer function where your I_{CP} is the input and your $V_{ctrl}(z)$ is the output. That is what you see here.

Now, to simplify this a little further, I know that this charge-pump which I am having is controlled by, this is I_{CP}, this is controlled by phase your PFD output whose gain, so this PFD output, you have a phase error at the input of the PFD, you get UP and DN pulses. So, from PFD block to your charge-pump block. So, you have a phase error at the input and this is like the voltage which you have.

So, you have V_{PD}. So, from your V_{PD} to your charge-pump output, the gain which we are using is I_{CP}. So, what I am trying to do here is the following. These two are standalone blocks. So, now I will write,

$$V_{ctrl}(z) = I_{CP}V_{PD}R + \frac{I_{CP}V_{PD}T_s}{2C_1}\frac{1+z^{-1}}{1-z^{-1}}$$

I am writing everything in terms of z-domain. So, we get,

$$\frac{V_{ctrl}(z)}{V_{PD}(z)} = I_{CP}R + \frac{I_{CP}T_s}{2C_1}\frac{1+z^{-1}}{1-z^{-1}}$$

So, this is the transfer function which you can say I am implementing from our PD output to our control voltage. So, the previous expression which we wrote here, this one, this one for a standalone, this circuit.

Now, when we bring our PFD and charge-pump together, this is what you get. Now, come back to this and look at it that from PFD output, somehow, we do not know how, but somehow, we are converting this phase detector or the PFD, right now that is a voltage, but somehow we are saying that we are implementing this transfer function to V_{ctrl} which is controlling our VCO.

So, whether this function can do it or not, that is something which still needs to be checked. So, let us now just take this function as such as,

$$\frac{Y(z)}{X(z)} = I_{CP}R + \frac{I_{CP}T_s}{2C_1}\frac{1+z^{-1}}{1-z^{-1}}$$

I am going to divide this transfer function in two parts. So, we get,

$$\frac{Y(z)}{X(z)} = H_{prop}(z) + H_{int}(z)$$

Why the integral part? Because there is $1 - z^{-1}$ which acts like an integrator.

So, if you look at $H_{prop}(z)$. It is telling $\frac{Y(z)}{X(z)}$. So, H_{prop} is given by,

$$H_{prop}(z) = \frac{Y_{prop}(z)}{X(z)} = I_{CP}R = K_P$$

Similarly, $H_{int}(z)$ is given by,

$$H_{int}(z) = \frac{Y_{int}(z)}{X(z)} = \frac{I_{CP}T_s}{2C_1}\frac{1+z^{-1}}{1-z^{-1}} = \frac{K_I(1+z^{-1})}{1-z^{-1}}$$

So, this is the transfer function which you have.

Now, these two things, it is worth as looking here as z which is like discrete in nature. The block diagram for this can be that I give $X(z)$ and from $X(z)$, I have a proportional path with K_P, it gets added up, I will use an adder here right now, and then this particular transfer function which you are seeing here $\frac{K_I(1+z^{-1})}{1-z^{-1}}$.

For simplicity, let us ignore this part. If you want, we can also implement this. Just ignoring the numerator, that is the zero frequency which you have, what we are doing here is we are having a K_I and then on the top of the K_I, we are integrating, this is just a block diagram, by z^{-1} here like this and it gets added up here. So, this is your $Y(z)$ based on $X(z)$.

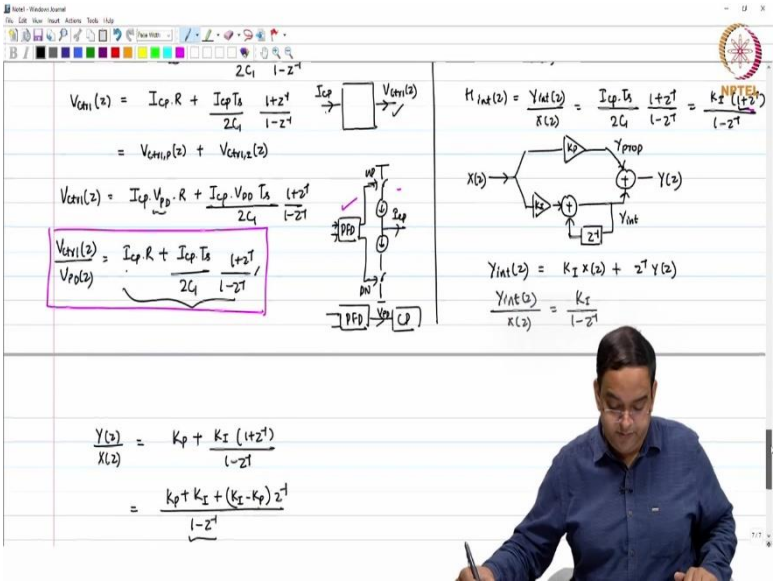
Now, just look at it. This is Y_{prop} and this is Y_{int}. So, what is Y_{int}? $Y_{int}(z)$ is given by,

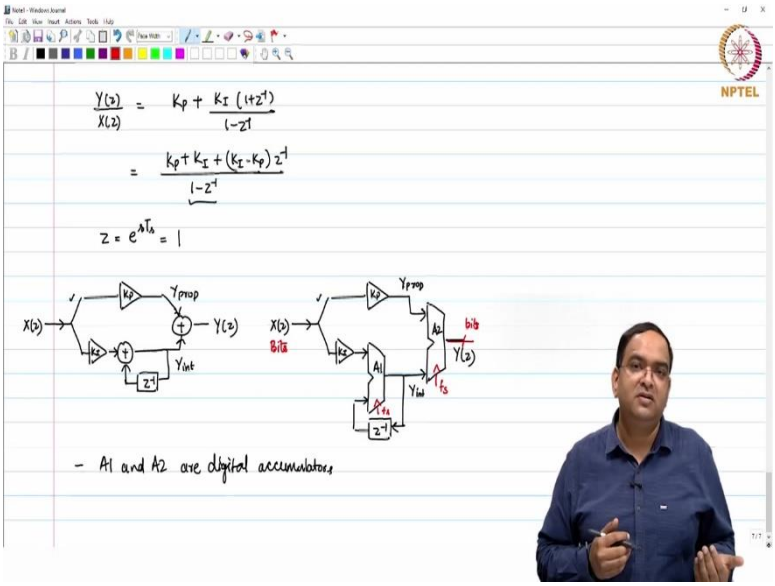
$$Y_{int}(z) = K_I X(z) + z^{-1}Y(z)$$

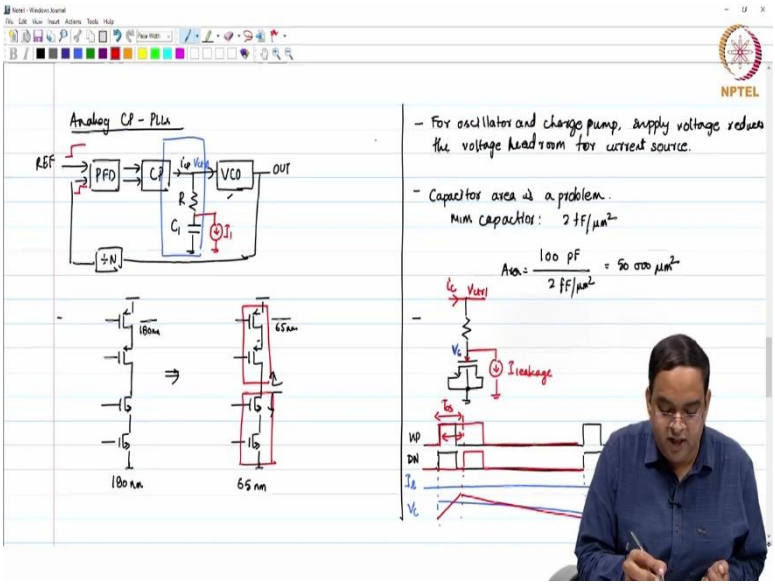
$$\frac{Y_{int}(z)}{X(z)} = \frac{K_I}{1-z^{-1}}$$

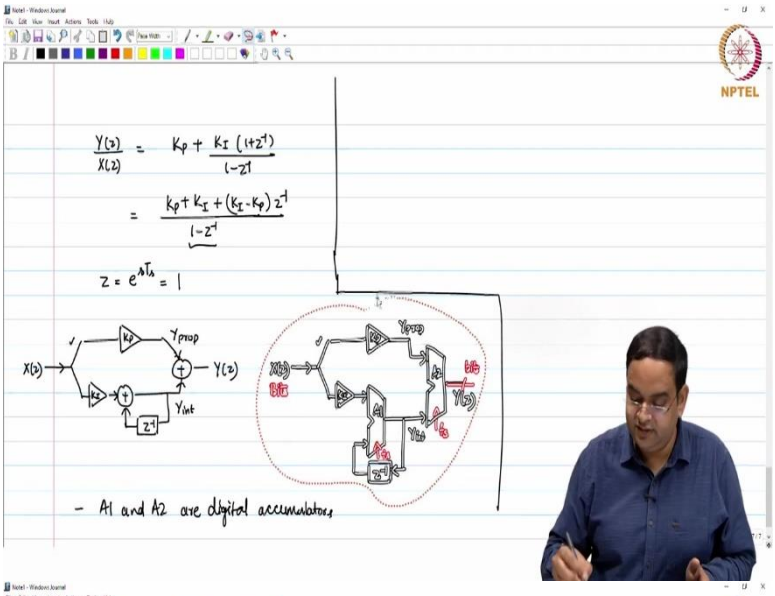
So, I told you, yes, we are ignoring this part. This is just a zero which we have. Later we will find that it can be neglected while you are implementing it and many other things will come which will add to this stuff.

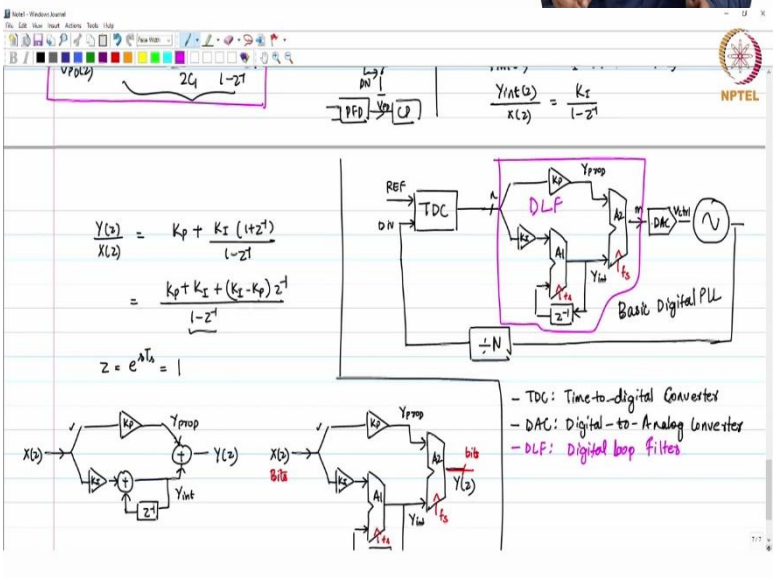
(Refer Slide Time: 26:55)






Analog CP - PLLs
REF
PFD
CP
VCO
OUT
R
C1
I1
÷N
180nm
65nm
180 nm
65 nm
- For oscillator and charge pump, supply voltage reduces the voltage headroom for current source.
- Capacitor area is a problem.
MIM capacitor: 2 fF/μm²
Area = 100 pF / 2 fF/μm² = 50 000 μm²
Vctrl
Ileakage
UP
DN


Y(z)/X(z) = Kp + KI(1+z⁻¹)/(1−z⁻¹)
= (Kp + KI + (KI − Kp)z⁻¹)/(1−z⁻¹)
z = e^{sTs} = 1
Yprop
Yint
X(z)
Y(z)
Bits
A1
A2
- A1 and A2 are digital accumulators


Y(z)/X(z) = Kp + KI(1+z⁻¹)/(1−z⁻¹)
= (Kp + KI + (KI − Kp)z⁻¹)/(1−z⁻¹)
z = e^{sTs} = 1
REF
DN
TDC
DLF
DAC
Vctrl
÷N
Basic Digital PLL
- TDC: Time-to-digital Converter
- DAC: Digital-to-Analog Converter
- DLF: Digital loop filter

So, the transfer function which you see here now, this is K_P and K_I. So, let us just look at it. The overall transfer function is given as,

$$\frac{Y(z)}{X(z)} = K_P + \frac{K_I(1 + z^{-1})}{1 - z^{-1}}$$

$$\frac{Y(z)}{X(z)} = \frac{K_P + K_I + (K_I - K_P)z^{-1}}{1 - z^{-1}}$$

So, you have a zero and you have a pole. The pole as per the z transfer function is at $z = 1$ and $z = 1$ here is like, by the way z is replaced by e^{sT_S}. So, $z = 1$ is what we are saying is at $s = 0$. So, that is what we have.

Now, given this transfer function, what the exact value we are going to choose for K_P and K_I, that is still something to look at and we will see that but before that, the question is how you are going to implement this and this is a digital block which z^{-1} is like a delay in the value. So, this block is going to be implemented using digital accumulators. So, I will just bring this first and then replace each of these blocks with their corresponding digital counterparts.

So, here just paste this. I think putting it here is a better option. So, first this accumulator which you are seeing, what you will see here is that this goes to an accumulator which is normally represented by this block K_I and then you have this z^{-1}. Similarly, this one you can have a similar block here, it goes here and this goes here. So, this is your $Y(z)$. This is Y_{prop}, this is Y_{int}. These A1 and A2, A1 and A2 are digital accumulators.

Now, if they are digital accumulators, depending on what speed you are clocking them. So, normally all these accumulators are clocked at a certain frequency and in this case, this is your sampling clock period. So, f_s normally here is equal to your reference frequency. These are all digital accumulators.

Now, think about it whether you implement this digital accumulator or the digital loop filter in 180 nm or you are going to implement this in 65 nm, these things will remain the same. There is no problem in porting from one technology to the other technology. Actually, as you go down lower to a lower technology node, in that particular case, you will find that in implementing this, you will actually save a greater amount of power.

So, now the thing is that these particular digital accumulators, they will operate at a clock, that is one thing. The other thing is now they need, you are having a digital accumulator. So, they

need bits at the input and their output is also going to be in bits, 1 0. You are implementing an accumulator and your digital filter, this is like a digital loop filter, you will have to feed in the bits and get the bits out.

What was PFD giving earlier? PFD was giving us analog, the phase error. So, if you have any phase error at the input, PFD was giving the difference between, the timing difference. Now, that timing difference has to be converted into bits. So, now, given this loop filter, you are going to use this particular loop filter which is helping you to replace the loop filter capacitor in the analog PLL.

So, let me just remove this part now, this is the loop filter which we are adding. So, what I want to do is that I want to convert the phase error from PFD in bits form. So, I still have the reference clock and voltage, they remain the same. So, I need a block which converts the timing information between your reference and your divided clock to digital bits. And that particular block is called TDC. It gives you the information about the phase error in bits.

Similarly, the output of the loop filter is digital. This dash just tells you that there are a number of bits. How many? That we still do not know. But this is like n bit or m bit you can have, depending on how many bits information you have and this has to be converted to the oscillator. So, I have an oscillator here which normally works on voltage, you need this V_{ctrl} voltage to change the oscillator frequency. So, I have to convert these bits to V_{ctrl} and if I have to convert these bits to V_{ctrl}, what I need is a DAC, digital-to-analog converter. And then you have this VCO frequency, VCO output. Well, the VCO output remains same. This is $\div N$. This is your basic digital PLL.

Here, TDC is Time-to-Digital Converter. It converts the timing error or the phase error which you have at the input of the PFD earlier, the reference and the feedback divided signal. It converts that error into digital bits. That is what the TDC does and your DAC converts the digital output of the loop filter to analog control voltage which is needed for the oscillator. So, using TDC, DAC and this whole block is normally called Digital Loop Filter, DLF. We need to see how we can use all these blocks actually to make our PLL work better or the way we want. Thank you.