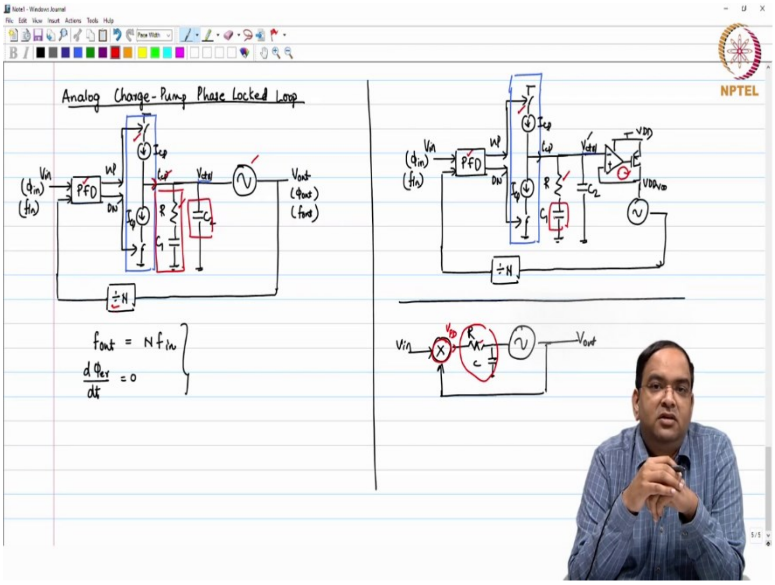
Phase-Locked Loops
Dr. Saurabh Saxena
Department of Electrical Engineering
Indian Institute of Technology Madras

Lecture – 65
Course Summary

(Refer Slide Time: 0:15)



Hello everyone. Now, welcome to the last session of this course on PLL. So, I would like to just summarize what all important things we have learnt during this whole course. So, our final goal was to learn how to design a phase-locked loop and what we learnt after a lot of analysis is the analog charge-pump PLL, charge-pump phase-locked loop. It was discussed in great detail so that whenever you would like to design a PLL, given the resources, you can design the analog charge-pump PLL.

In the block diagram of this analog charge-pump PLL, you had phase frequency detector and the output of the phase frequency detector goes to a charge-pump. We have seen many methods on how to implement the charge-pump but the basic block is only this that you have current sources, you have switches and that is the only thing which you need to control.

The output of the charge-pump goes to a loop filter, a passive loop filter with R C_1. Then you had added additional capacitor which is C_2. The output of that goes to an oscillator which generates the frequency with respect to the control voltage. The oscillator output is divided with a frequency divider where the divider operates on the frequency of the signal and not on the amplitude.

So, actual signal here is V_{in}, this is V_{out}. In terms of phase and frequency, you have φ_{in} or f_{in} which is the reference frequency in general and then you have, here you can look it at as φ_{out} or f_{out}. So, this is like UP and DN, the current source I_{CP} here, this is i_{cp} which I am writing, this is V_{ctrl}. So, this is the basic charge-pump PLL which we discussed in this course. We arrived at this architecture after understanding a lot of requirements from the PLL.

The PLL is considered locked if you have $f_{out} = Nf_{in}$ and the rate of change of phase error is zero which implies that $\frac{d\varphi_{er}}{dt} = 0$. It should not be increasing all the time. So, to begin with, this is our main requirement and we implemented this. The PFD measures the phase error between the input and the reference signals. It gives you pulse width modulated output in terms of UP and DN.

The charge-pump which is this block, the charge-pump converts this pulse width modulated phase error to pulse width modulated current i_{cp}. i_{cp} is actually integrated by the capacitor C_1 and C_2 also here. It is integrated by the capacitors which gives you the control voltage you desire for the given output frequency. So, we added this resistor here for stabilizing the loop otherwise you will have only two integrators in the loop which will be unstable.

Then we looked at the detailed design of PFD, charge-pump, how to have this loop filter and we dealt with the oscillators also. We talked about the low-swing oscillators, the full-swing oscillators, large-swing oscillators and we went ahead with the supply regulated VCO also. We went ahead and designed. For example, this oscillator was not directly controlled in this manner, it was controlled using a voltage regulated design.

So, we had a voltage regulated loop which controls the current through the oscillator and this is VDD, this is VDD_{VCO} in our case. So, what happens here that when you have a supply regulated oscillator, you will have to stabilize this loop also internally and the overall loop should also be stabilized. So, mostly what we do is we design in such a way that from V_{ctrl} to VDD_{VCO}, you have a bandwidth which is much larger than the PLL bandwidth, 4 to 5 times bandwidth of this regulator loop is quite convenient to design.

So, we talked about different methods in which we can design, we can have this supply regulation for the oscillator. We looked at single-ended oscillator, differential, pseudo-differential oscillators with cross-coupled inverters between the oscillators and with feed forward resistors from the power consumption point of view. So, you can say that this is a very basic charge-pump PLL which you may need in many different applications in which you can design.

Then before this particular block, this was our main focus of the course to design but before this, we also tried to understand how your phase and frequency variables change in large signal analysis. So, there we looked at the pull-in range, hold-in range and lock-in range for a simple PLL. The simple PLL which we studied was just a mixer based phase error detector which was like this and then you had a very simple loop filter or you can have the loop filter which you want, loop filter and oscillator.

So, our starting point was this. Now, you just think about it, starting point was this at the block level. You can say it is just a model, you have a mixer, you have R C filter and you have an oscillator with the desired characteristics. We looked at the frequency acquisition in such kinds of PLLs when as soon as you start the PLL what will happen and then from here slowly we build up from mixer.

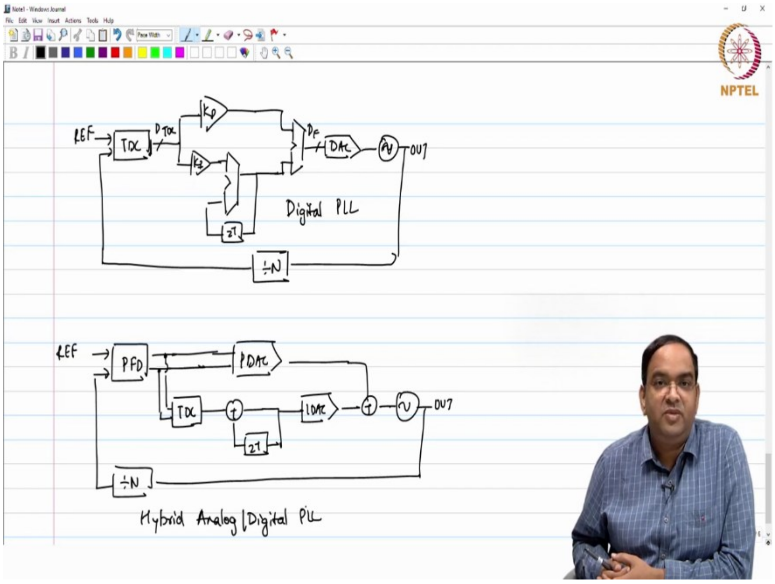
We came to the phase error detector, then phase frequency detector. From the loop filter because of the limited acquisition range, we came to the Type-II PLL which has this R C where R C became must to acquire the frequency which we want. Then because you had this R there, you had ripples, so you added a capacitor C_2 and that is how we came up with this loop filter. Then conversion from the phase error which you see here that this mixer output was only V_{PD}, the average was done using the loop filter.

Now, we need to convert this because in place of using sinusoidal mixers, we are now using square wave clocks. So, we need to convert this pulse width modulated signal to our equivalent current or voltage. The current summation was easier. So, we had added i_{cp} here. So, this is how we convert it and then after this we control the oscillator.

So, that is how we came to know about it. Then for the frequency multiplication, we need to have the frequency divider. We looked at a simple divider using D flip flop or TSPC flip flop where we can easily divide by 2^n numbers. We also looked at how we can get different frequency division by using a cascade of divide by 2/3 circuits.

After doing all this, we actually went ahead and found the problem with this kind of PLL that this capacitor takes a lot of area. So, can we replace this capacitor using a MOSCAP which has a higher density? Well, that led to leakage issues, hence we could not do that. So, we replace this integration which we are doing using a digital accumulator. Because we know that as you go down, the lower the technology, having larger voltages is not possible and process variations for R C are a lot. Implementing a digital accumulation is much easier. We implemented this with a digital accumulator.

(Refer Slide Time: 10:35)



So, then we build up on that and finally, we had this block where we had a TDC which converts the phase error into its digital equivalent. Then the loop filter became digital which is K_P and then you had K_I. This was our digital accumulator. So, we represented this block as an accumulation like this and these two were then added with z^{-1} here. The output of this and this, they go to another accumulator given like this.

Then you had digital values, so, you needed to have a DAC and then DAC controls the oscillator, oscillator output is fed back in this manner. So, in this case, this is the basic digital PLL just replacing all the blocks by their digital equivalent which gives us bits like 1 or 0. So, here we had the reference clock and we had the output clock here.

The output of the TDC is digital bits D_{TDC} and everything is in terms of bits here till this point which is the control word for your DAC. When we did this, we found that the quantization noise of the TDC is one of the biggest problems for this digital stuff and no matter the benefit which you have got with removing the loop filter capacitor and replacing it by the digital accumulator, TDC quantization noise is just spoiling the party.

So, what we did we said that let us separate this now, digital accumulator part or the integral path and the proportional path and then we came up with this hybrid PLL where you give the input to PFD and PFD output is taken by our proportional path DAC which feeds directly to the VCO without any quantization. The output of the PFD goes to a TDC and we could implement a 1-bit TDC without any error in the polarity of the frequency error detected. This goes to your accumulator and then goes to an accumulator, it goes to IDAC and they both control the oscillator then.

So, this is what we did. Then, very briefly, this particular PLL has now become a hybrid PLL. So, this is reference, this is output, this one is digital PLL and this is our hybrid PLL, we call it hybrid analog or digital PLL. Then we looked at how we are going to implement proportional path DAC such that there is no quantization. We realized some problems with IDAC that you have a range versus resolution trade-off. So, we replaced direct control of IDAC with a digital delta sigma and that is how we implemented it.

So, all these techniques which we have learnt, we can surely build up on this while designing ICs, PLL ICs for any given application. The basic concepts remain the same. So, I hope this course has given a good number of basic fundamentals to start designing PLLs. The basic concepts have been covered. Then depending on your requirements, you can choose either of these designs or you can explore more to design the PLL for any given application whether it is wireless, wireline, ADC, power management or anything, wherever you need PLL, these concepts can be used.

An important thing is that dealing with phase or frequency, it is not limited to PLLs. You can explore more that how this concept of phase or frequency can be applied for analysis or controlling other kinds of feedback loops. So, thank you very much and all the best for this course. Thank you.