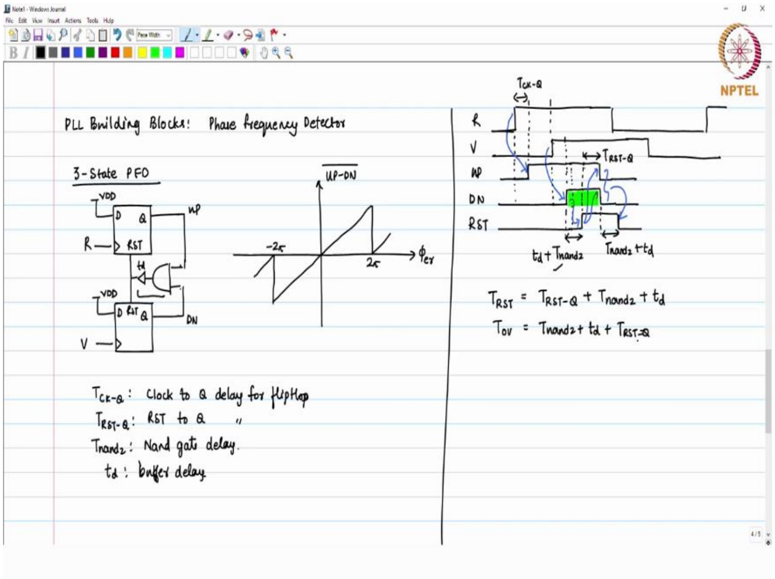
Phase-Locked Loops
Dr. Saurabh Saxena
Department of Electrical Engineering
Indian Institute of Technology Madras

Lecture – 48
Circuit-level Design of PFD: Part I

Hello, welcome to this session. So, by now, we have completed the system-level PLL design. We have also looked at the oscillator design at transistor level, and we are continuing our PLL design with the design of each block at circuit level.

(Refer Slide Time: 0:36)



PLL Building Blocks: Phase frequency Detector

3-State PFD

T_{CK-Q} : Clock to Q delay for flipflop
T_{RST-Q} : RST to Q "
T_{nand2} : Nand gate delay.
t_d : buffer delay

$T_{RST} = T_{RST-Q} + T_{nand2} + t_d$
$T_{ov} = T_{nand2} + t_d + T_{RST-Q}$

So, the next building block which we would like to study now is phase frequency detector. PLL building blocks, the journey continues. It is phase frequency detector. So, earlier we have seen the phase frequency detectors but treating them as ideal. Now, we will look at what happens in the real case. So, the 3-state PFD or phase frequency detector which we have seen is, I am drawing it again. You have a D flip-flop whose input D is always connected to VDD, and which has a clock and output Q.

These two flip-flops, this particular flip-flop is used. So, here you have a clock R going in, V here, the output of these two flip-flops, they go to AND gate and by the way, if they go to AND gate, and if we recall the sources of non-linearities in PLLs, there is a delay also added by adding a buffer to the PFD, and you have a reset signal here, this is UP and this is DN signal.

The transfer function or the input-to-output characteristics of this PFD is $\overline{UP - DN}$ with respect to phase error, we have seen, it is something like this. So, this is what we have, this is 2π and this is -2π. Now, whatever we are seeing here, this is an ideal case. What happens or what is the thing which we neglected earlier but now we are going to consider is the delay of each block. So, there is a T_{CK-Q} delay, and you can say whenever your R or V changes, after how much time your UP or DN signal will change?

The Clock-to-Q delay for flip-flop, this is some finite delay, by the way. Then you have T_{RST-Q} delay. So, it is reset to Q delay for flip-flop, you have NAND delay, T_{nand2}. So, this is NAND gate delay, when the input changes for the NAND gate, after how much time your output will change? And then you have the extra delay which you added for the overlap.

So, this is like just the buffer delay. Now, in the presence of all these delays, let us look at how the output signals will change. So, I will still use a simple R and V like this. Let us say the frequencies are same to keep life easy here and V signal is this. So, this is V. Now, in response to R signal, our UP signal will go high after some time, it does not go high immediately. It goes high after some time. This is UP, and what is that some time?

Well, this time is T_{CK-Q} delay for the flip-flop. Then when you get V signal, the DN signal will also go high after the same amount of time. So, it goes high here. So, both these UP and DN signals are high at this time. Then, when you get UP and DN signals both high, then after the delay of these two blocks, your reset will go high. So, the reset goes high after the delay of $T_{nand2} + t_d$. So, I will just make it let us say like this.

So, reset goes high after this time, this is your $T_{nand2} + t_d$, the delay of the buffer. Once the reset goes high, then UP signal and DN signal will go low. So, UP signal and DN signal will go low after T_{RST-Q} delay. So, this delay is T_{RST-Q}. Then, when both the signals are down, then the reset signal will also go down after another $T_{nand2} + t_d$ delay.

This is same, $T_{nand2} + t_d$. So, let us just look at the transitions. R triggers this after delay T_{CK-Q}, V triggers DN, both UP and DN, then they trigger reset, reset is going to trigger both the signals UP and DN to go low but that is T_{RST-Q}, and then your reset, when UP and DN both go high, then these two signals trigger reset to go low.

So, this is how the actual transition happens. So, now, the time for which the reset signal is high is given by,

$$T_{RST} = T_{RST-Q} + T_{nand2} + t_d$$

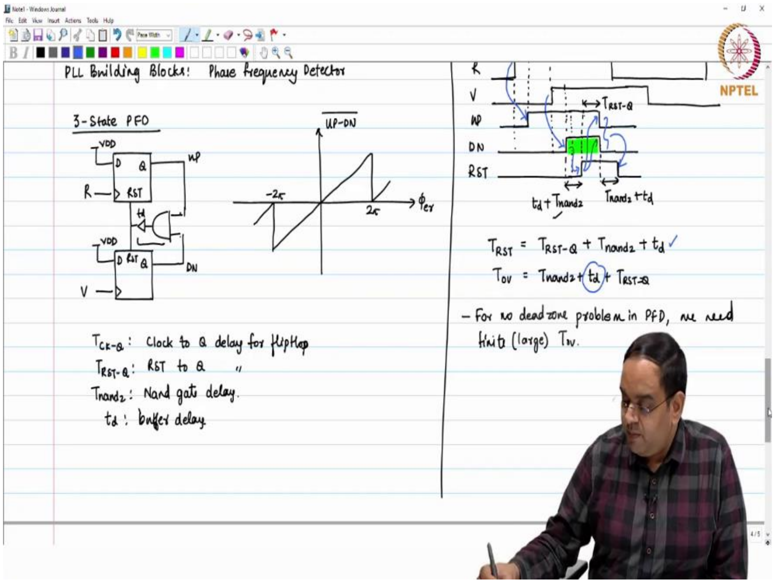
This is the time for which the reset signal is high. The other important thing which we need to know what is the time duration during which both UP and DN signals are high, that time duration is this.

And what is that value? I call that as T_{ov}. T_{ov} is given by,

$$T_{ov} = T_{nand2} + t_d + T_{RST-Q}$$

So, now you see in this particular case that your T_{RST}, the time during which the reset is active, that time is the same as the time during which overlap is there between UP and DN pulses.

(Refer Slide Time: 9:18)



PLL Building Blocks: Phase Frequency Detector

3-State PFD

T_{CK-Q} : Clock to Q delay for flipflop

T_{RST-Q} : RST to Q "

T_{nand2} : Nand gate delay.

t_d : buffer delay

$T_{RST} = T_{RST-Q} + T_{nand2} + t_d$ ✓

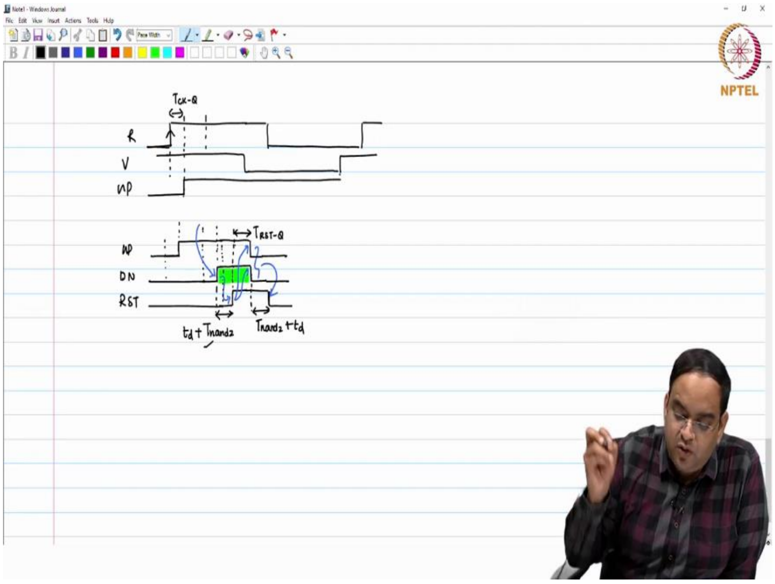
$T_{ov} = T_{nand2} + t_d + T_{RST-Q}$

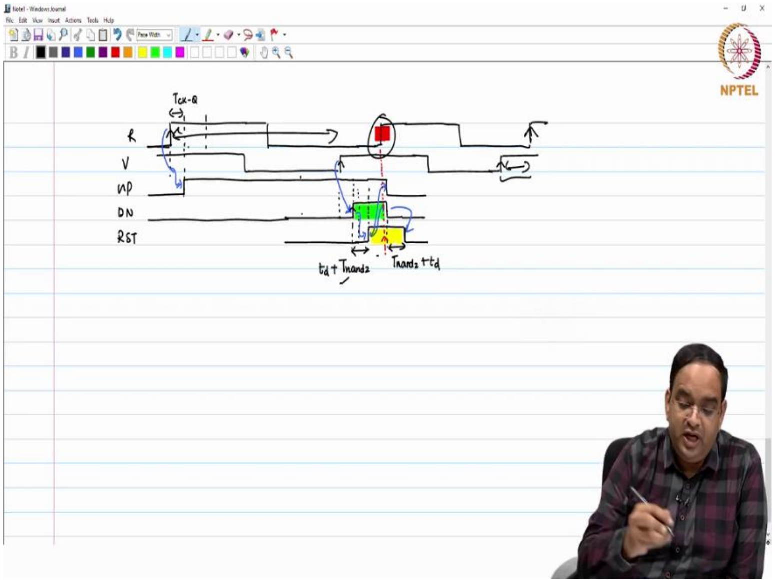
- For no dead zone problem in PFD, we need finite (large) T_{ov}.

So now, for no dead zone problem in PFD, we need finite or you can say depending on the requirement, large T_{ov}. In order to increase T_{ov}, the only options if we are using or the control which we have is go and vary t_d. But when we are going and varying the t_d, what also happens is that your T_{RST} also increases. Well, increasing T_{RST} is a problem or not, right now we do not know, but we will soon know about it.

So, if that is the case, what we know is that in order to have a larger T_{ov}, we have to increase t_d because that is the controllable part we have. In a given technology, you can have a finite NAND gate delay or T_{RST-Q} delay. So, t_d is something which you can reduce, you can remove the buffer, t_d will be 0, you can add more buffers, t_d will increase.

(Refer Slide Time: 10:36)





But what happens is at the same time, your T_{RST} also increases. Now, if T_{RST} increases, then what happens? So, when you have V pulse let us say coming very close to the, when the phase error is quite large. So, when the phase error is quite large, you start with a large phase error.

So, in that case what you do is rising edge goes high here, your UP signal will go high at this point, T_{RST-Q}, it remains high, it waits for the time when V signal goes high. Now, when V signal goes high, at that point, your DN signal goes high. So let us just remove this. So, my DN signal will come here. I am just using the previous drawing so that something is repeated here.

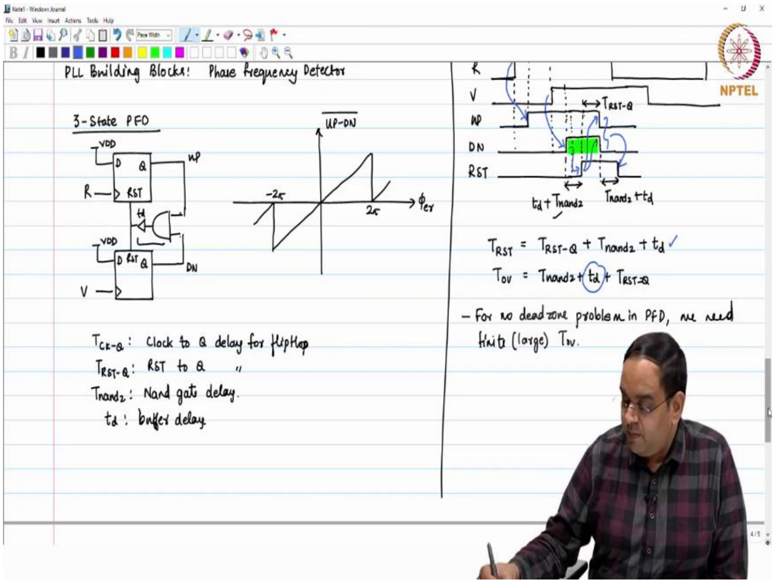
So DN signal remains low for a long time. It goes high after whatever your T_{CK-Q} delay, and then UP signal remains at the same point, then both UP and DN, both the signals are high, after some time your reset signal goes high. When your reset signal goes high, at that time, so let us just do this slightly, not much change. I am just doing it here. So, reset signal goes high, when reset goes high, what happens is your UP and DN signal both go low, and when they both go low, then your reset goes high later.

So, this is the clock which you have. Now, interestingly, you see this. It changes here, this changes, this one, and then DN signal, because DN goes high, that triggers the reset. So, this is what you see. And then what you are going to have is when both UP and DN signal go low, finally your reset signal will also go low. The problem is not this, the problem is the rising edge of the next transition is actually appearing between the reset when the reset is high.

So, when the reset, in a given flip-flop, when the reset signal is high, at that particular instant of time, no changes at the input will be taken into account. So, no change, when the reset signal is high and no changes are taken into account, then what happens is that during this, this transition is missed.

So, if you miss this particular transition, then in the next transition, you will see the rising edge on V appearing first. If you look at this clock cycle, the phase error is a large positive value. So, you will get $\overline{UP - DN}$ accordingly, but what happens because you miss this transition due to the finite reset time, in the next clock cycle, what is going to happen is that you will get the rising edge on V first, and rising edge on R later. So, these two transitions will now change the sign of the feedback together. And then it will continue with the negative only because now it appears that the phase error is $2\pi -$ some value, this is a negative phase error. It is not like the large phase error which you see here anymore.

(Refer Slide Time: 15:19)



PLL Building Blocks: Phase Frequency Detector

T_{CK-Q} : Clock to Q delay for flipflop

T_{RST-Q} : RST to Q "

T_{nand2} : Nand gate delay.

t_d : buffer delay

$T_{RST} = T_{RST-Q} + T_{nand2} + t_d$ ✓

$T_{ov} = T_{nand2} + t_d + T_{RST-Q}$

- For no dead zone problem in PFD, we need finite (large) T_{ov}.

So, in the presence of your finite reset time, what happens is that for small phase error, the input-to-output characteristics remain the same. Then when the phase error increases beyond a certain value, you run into this problem. And when you have this problem, then the $\overline{UP - DN}$ will go negative, and then it will do something like this, and this, by the way this is still 2π.

So, it is going to happen on both the sides, this is still -2π. If you are looking in terms of phase error, then this phase error is, you are drawing with respect to phase error, this is $2\pi\frac{T_{RST}}{T_R}$, where T_R is your reference period. So, what you see the curve which was having the same sign for positive or negative frequency error, suddenly that curve has started having both positive and negative values for the positive phase error itself.

Now, one of the extreme cases will happen when the T_{RST} increases to such a large value that this particular waveform becomes like this where the average of the positive and the average of the negative, or the sum of these two in one 2π phase error is zero. That is the worst case. If that happens, then you cannot use the phase frequency detector for frequency detection at all because this will become similar to your EXOR gate, similar to your Exclusive-OR phase detector, or your SR latch-based phase detector, where the average value of the phase detector output for given frequency error is going to be zero.

So, it cannot detect frequency anymore. So, the usable frequency has a limit and what is the limit? That this particular phase error should be,

$$2\pi\frac{T_{RST}}{T_R} < \pi$$

$$T_{RST} < \frac{T_R}{2}$$

$$T_{RST} < \frac{1}{2f_R}$$

So, I can very well say that,

$$f_R < \frac{1}{2T_{RST}}$$

So, for a given reset time, there is an upper limit on the reference frequency which you can use for the given PFD.

If your T_{RST} is large, your reference frequency which you can use to clock the PFD will also be smaller. So, what happens is that whenever you design a PFD, you should make sure that your reset period is not limiting your reference frequency. Normally, it does not unless your reset period is quite large or you would like to have the reference frequency also quite large.

So, that is the case which you need to take into account and by the way even when you do not have your reset period large enough that you are getting π here, π for the phase error, even when it has a small negative value, it is going to increase your pull-in time for the PLL because even for the positive phase error, there will be time when you will have incorrect feedback.

So, it increases the pull-in time. So, normally you would like to have T_{RST} as small as possible, but at the same time to avoid the dead zone problem in PFD, you would like to have T_{ov} region also as required or as large as required, let me put it that way. So, in order to overcome this problem, we have other kinds of PFDs also which we will see in the next session. Thank you.