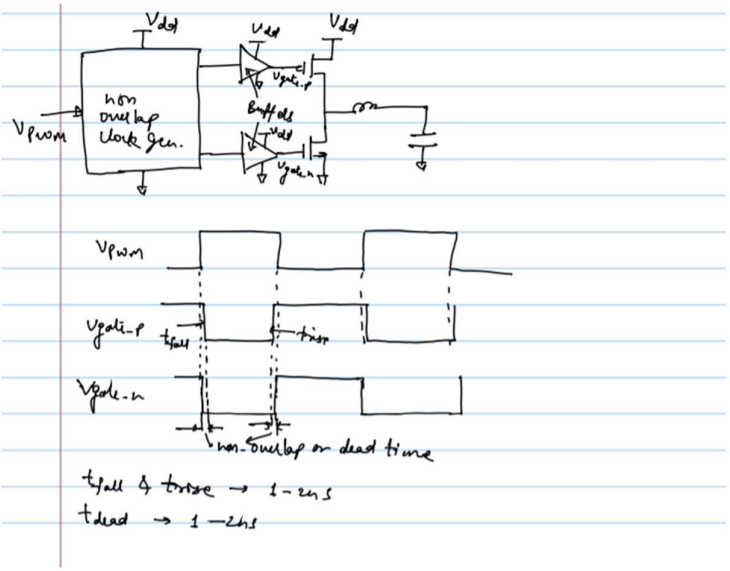
Power Management Integrated Circuits
Dr. Qadeer Ahmad Khan
Department of Electrical Engineering
Indian Institute of Technology, Madras

Lecture – 67
Gate Buffer and Non-Overlap Clock Generator in Gate-Driver Circuit

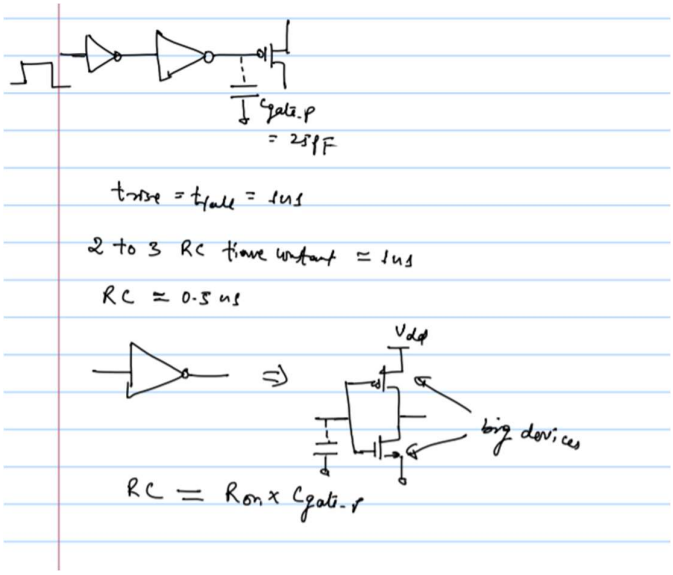
Gate driver circuit: Non-Overlap Clock Generator + Gate Buffer

We need gate buffer to drive large cap because your FETs are very large, and those gate caps will be like 10 to 20 pF or maybe even more. In order to drive such a large cap, we need a buffer. A smaller inverter can't drive that because the rise and fall time we are looking at the gate of powerFET is order of 1 to 2 nanoseconds for 10 MHz kind of frequency. You can't afford very large because your efficiency will degrade. The block level diagram is shown in below.



We need non-overlap time or dead time on both the sides to make sure that only one switch is turning on at a time. Dead time (t_{dead}) is also order of 1 to 2 nanoseconds and you can't afford very large because your efficiency will degrade.

Gate buffer: Buffer means you have progressively scaled inverters to drive large cap. Let's assume C_{gate-p} is 25 pF.



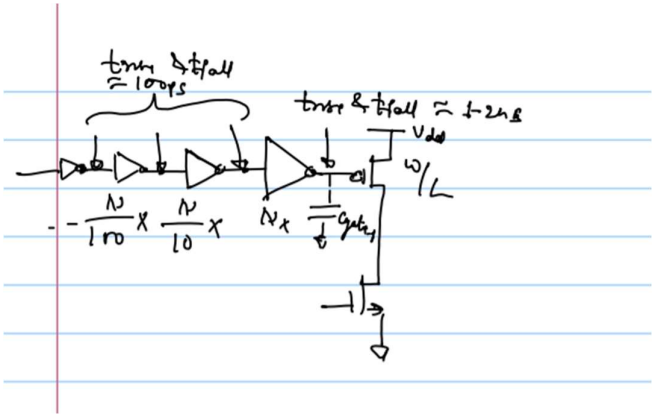
Since we are looking for rise and fall time of 1 nanosecond, so what kind of RC time constant we require? Let's say we are measuring from 10% to 90%. So, we will require roughly 2 to 3 time-constants. So, consider we need 2 RC time constants. So,

$$2\ R_{on}\ C_{gate-p} = 1ns$$

We know C_{gate-p}, so we will get R_{on}. From this you can calculate what kind of size you will require. This is one way and other way you just simulate and find what size of inverter you will require to get 1 nanosecond kind of a rise time. Now, this device is also big, and it will have its own cap again. So, the previous one also should be capable of driving that. In order to drive that, again you have to find what size you will require based on your RC time constant.

Since we are looking at 1 nanosecond kind of rise and fall time at the gate of powerFET, that does not mean at each stage you can have 1 nanosecond because that will add a lot of delay and that will hamper your non-overlap time. So, you have to make sure that the previous stages rise and fall time should look like 100 picosecond or so.

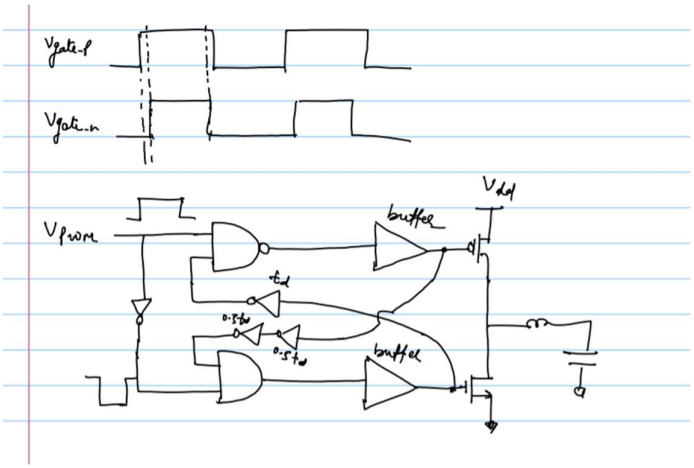
When you are increasing the size, you know that the gate cap is increasing. So, roughly you can estimate the size in order to get a 100 picosecond. In the older technologies one inverter can drive roughly 2 times of its own size without adding too much delays. But these days, these technologies you may find like maybe 5 or more than 5 times or so and you can scale the sizes accordingly as shown in below figure.



Which means the slew rate at the gate of powerFET is mostly driven by the final inverter and before that you are simply making sure that this whole chain should be able to drive the gate cap of each stage and that's why you progress gradually rather than simply putting 2 or 3 inverters. In some cases, you may even require more than 5 inverters or so. It all depends on what kind of rise and fall time you are looking at each stage.

Note that each inverter is driving both PMOS and NMOS, but the final inverter is driving only PMOS or NMOS. So, you cannot say that whatever the size of powerFET you have, you can simply scale down that by 10 to get the size of final inverter.

Non-overlap clock generator: The logic to generate non-overlap clock is shown in below.



Let's say we need to turn on PMOS powerFET. Before we turn on PMOS, we have to make sure that NMOS powerFET has turned off completely and that can be told only by looking at the gate of NMOS. That's why you should always take feedback after the buffer.

But if you look at any digital logic book, they might be taking feedback from anywhere because they are not driving such a huge gate cap. There the rise time, fall time and delays looking at the gate node will be relatively same compared to the previous stages. In that case you can take the feedback before this buffer also. But here you know the rise time itself is 1 nanosecond, so you should always take feedback after this buffer.