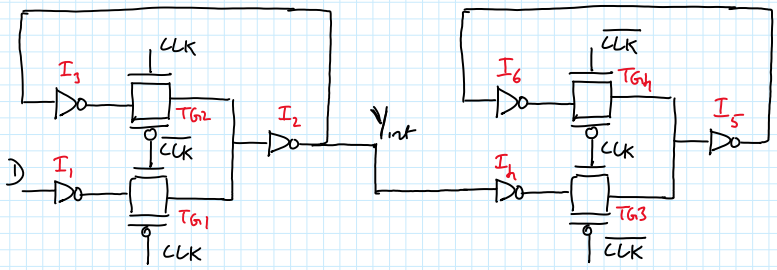
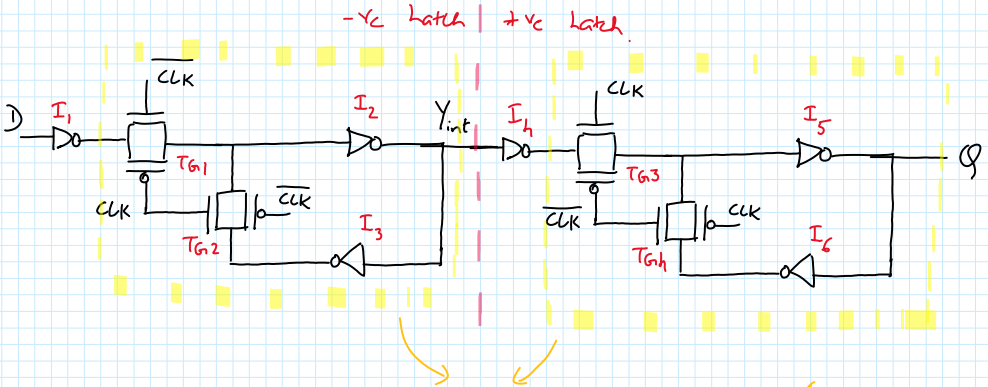
Lecture-11

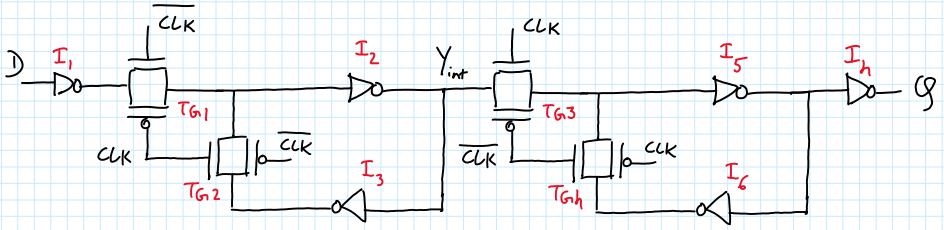


* We will redraw the +ve edge triggered FF as below.



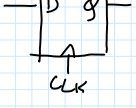
1-bit memory controlled by TG (open or close the feedback in a back-to-back configuration)

* We can push inverter I_4 to the output Q. This will shield the memory element of the second latch from external capacitance. i.e,



* Functionally, there is no change in the operation. (You can verify by passing input @ D and checking Q)

* Note that the T_s (setup time) is still the same, $T_s = t_{PI_1} + t_{PTG_1} + t_{PI2} + t_{PI3}$.
  But the T_{cq} has increased, $T_{cq} = t_{PTG3} + t_{PI5} + t_{PI_4}$

* The above DFF that we implemented has a symbol → 

* This one of the basic cell in a standard cell library.

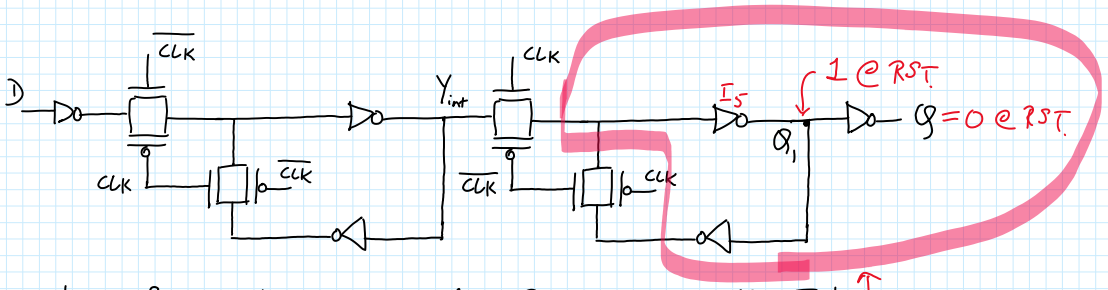
* A Verilog code like below, will be synthesized by the tool to this DFF that we have designed.

```
always @ (posedge CLK)
begin
    Q <= D;
end
```

× × ×

* Another basic cell in any standard cell library is a DFF with reset
  - ↳ Reset can be asynchronous (i.e $\varphi=0$ when RST = 1)
  - ↳ Reset can be synchronous (i.e $\varphi=0$ when RST = 1 and there is a CLK edge)

* Let's look at DFF with asynchronous RST. Consider our previous DFF circuit.



* A new logic has to be inverted for RST after all the TG's since asynchronous should be independent of CLK. Since $\varphi=0$, Node Q_1 must immediately go to 1 when RST arrives.

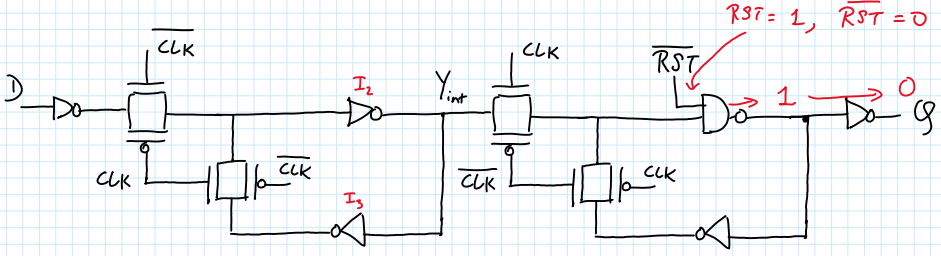
* A logic gate that provides an o/p of 1 when one of the input is fixed is : NAND

i.e

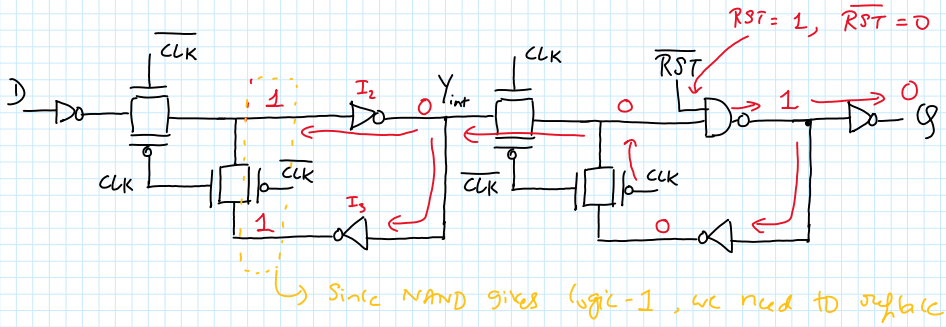
A	B	Y	
0	0	1	} If A=0, Y = 1 (LOGIC-1)
0	1	1	
1	0	1	} If A=1, Y = $\bar{B}$ (INVERTER)
1	1	0	

* A 2 i/p NAND can act as an inverter or fixed logic-1 o/p based on input.

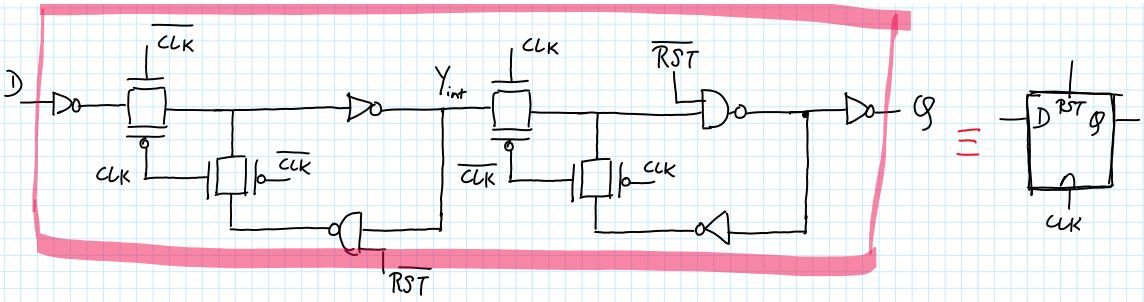
∴ Let's replace I_5 inverter with a NAND gate.



* Since there are two memory elements (i.e 2 latches) in this FF, both memories need to be reset. ∴ I_2 or I_3 needs to be replaced by a NAND gate. To find out which, trace back from the output side as follows.



↳ Since NAND gives logic-1, we need to replace I_3 !!

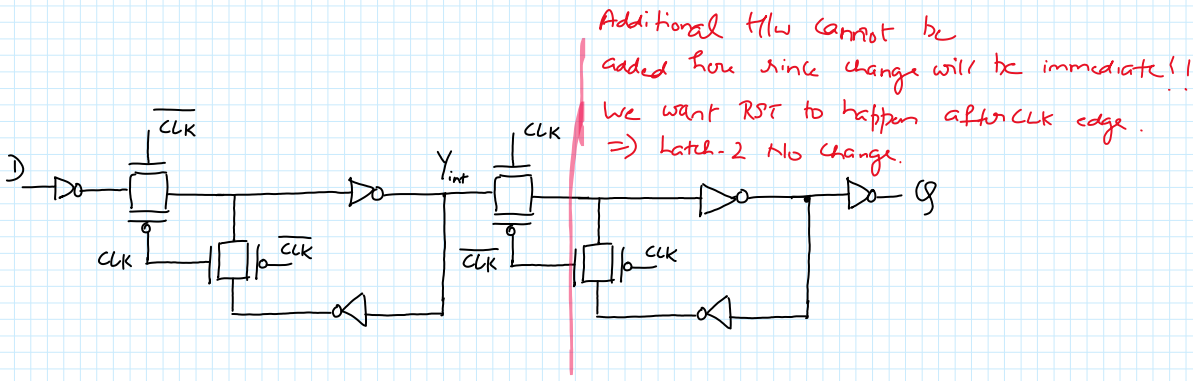


* A Verilog Code like below, will be synthesized by the tool to this DFF that we have designed.

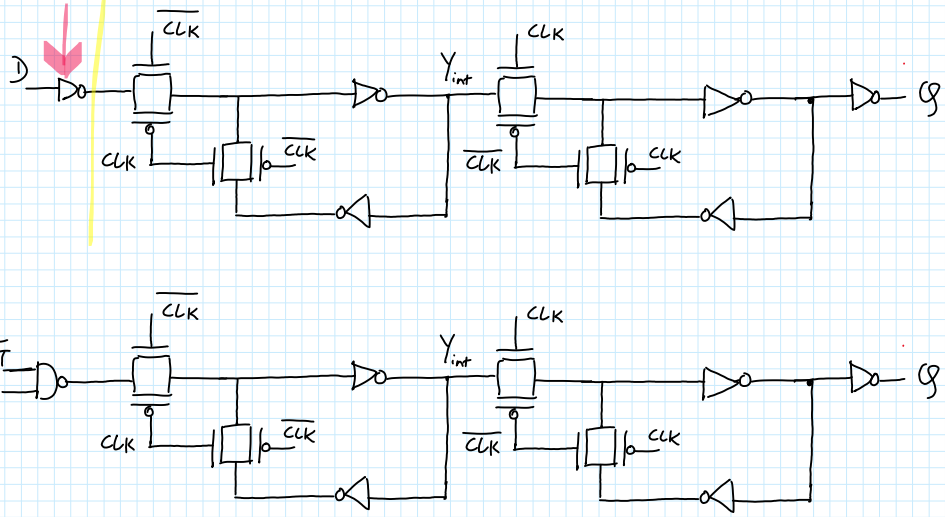
```
always @ (posedge CLK or RST)
begin
    if (RST)
        Q <= 0;
    else
        Q <= D;
end
```

---

* Now, lets switch focus to a DFF with synchronous RST.



* Since we cannot change latch-2, latch-1 cannot be changed as well. (since DFF is built using two similar latches) ∴ Additional circuit must be added before latch-1.



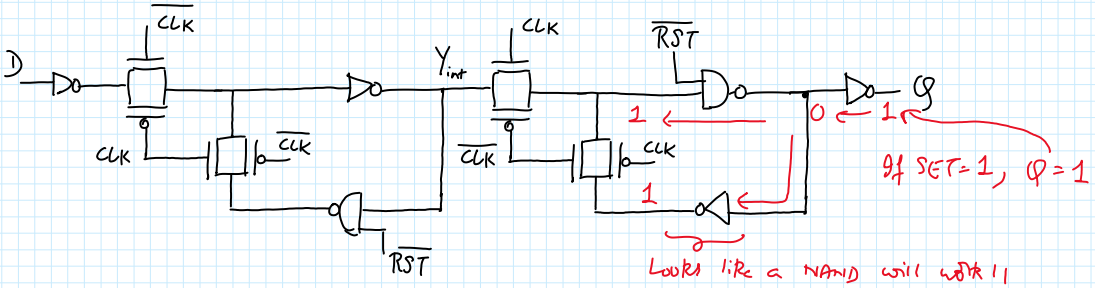
i.e if RST=1, the DFF latches 0 @ the edge !! (Synchronous RST)
if RST=0, NAND gate acts like an inverter, which is our DFF

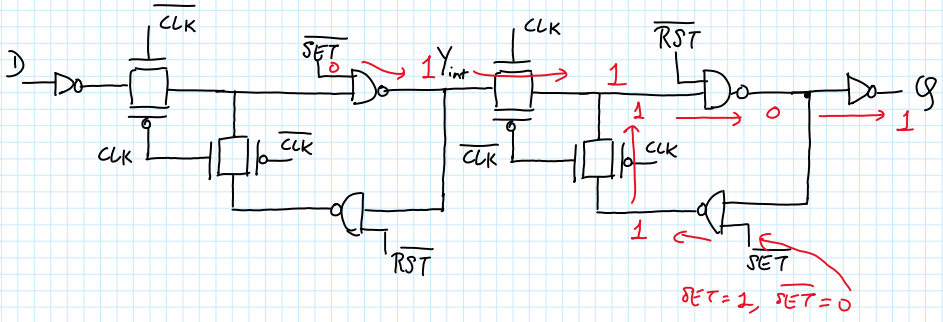
* A Verilog code like below, will be synthesized by the tool to this DFF that we have designed.

```
always @ (posedge CLK)
begin
    if (RST)
        Q <= 0;
    else
        Q <= D;
end
```

---

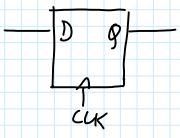
* Another important cell to have is a DFF with asynchronous SET and RST.
* If your Verilog code has $Q \Leftarrow 1$, we need to pick a DFF with SET



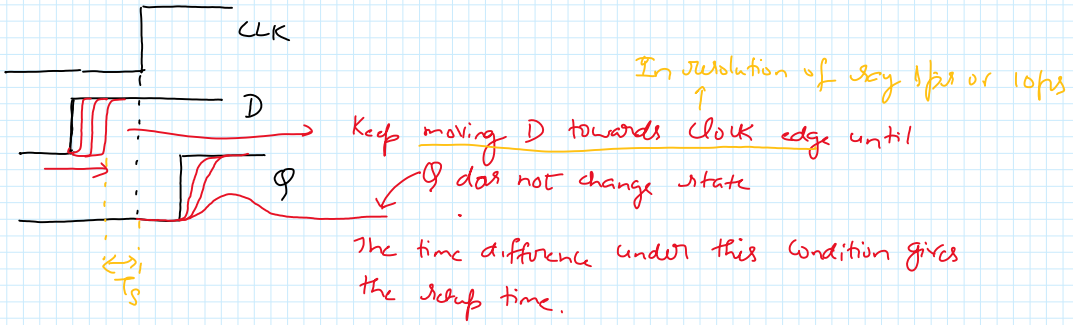


---

* Obtaining T_S, T_H and T_{CQ} from simulations.



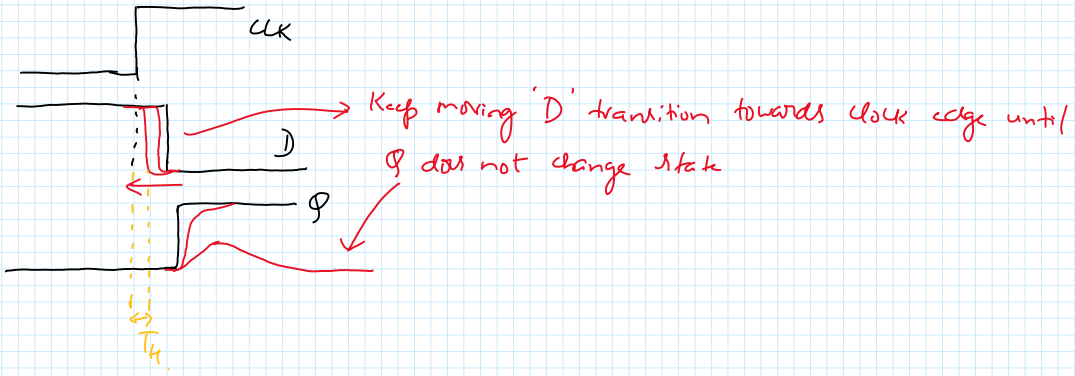
a) T_S :-



b) T_H :-

---

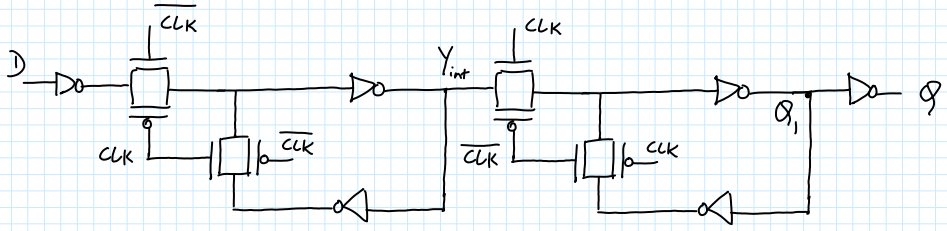
b) T_H :-



c) T_{CQ} :- Having known T_S & T_H Apply D & CLK, without violating setup & hold criteria. Now find out 50% of CLK to 50% of Q delay.

— × — × — × —

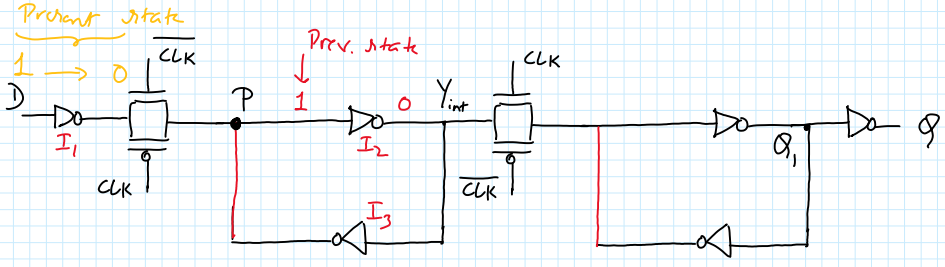
* One final note on DFF



On a chip, it is imperative that you reduce the cap load on the clock line, to reduce power consumption: $P = CV^2f$ → f is already high for clock

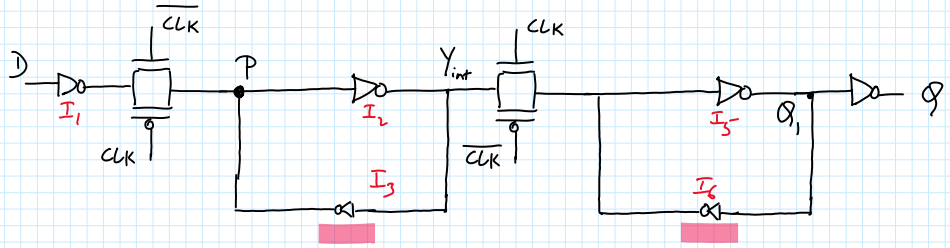
C: depends on how many MOSFETs it is driving.

The DFF above has total of 8 MOS with CLK or $\overline{CLK}$ drive. There are designs to reduce this clock load One of the popular design is to remove two TG's.



* Assume Node P had logic 1 due to the previous state of the latch.
* Now, when CLK = 0, a new D input of logic-1 is applied.
* At node P, I_1 and I_3 are fighting for dominance.
  - ↳ I_1 wants to enforce logic 0 on P
  - ↳ I_3 wants to enforce logic 1 on P
* We want I_1 to win since we want to enforce the new input into the latch

* We want I_1 to win since we want to enforce the new input into the latch.
  ↳ To ensure this I_3 strength must be very small !!



I_3 and I_6 are weak inverters. i.e $W/L < 1$ or L is increased for both PMOS & NMOS!!

* This kind of logic is called "Ratio'd logic" since we rely on ratio of strengths (strength of I_1 vs I_3) to ensure functionality

* Observe that the set-up time calculation is not straightforward since it depends on the sizing of I_1 & I_3 !!! .. Design is not straightforward