Static Timing Analysis

Part 1

Amr Adel Mohammady

in/amradelm

This Document Is Dedicated to Thousands of Palestinian Children Who

Were Killed

Lost Their Limps

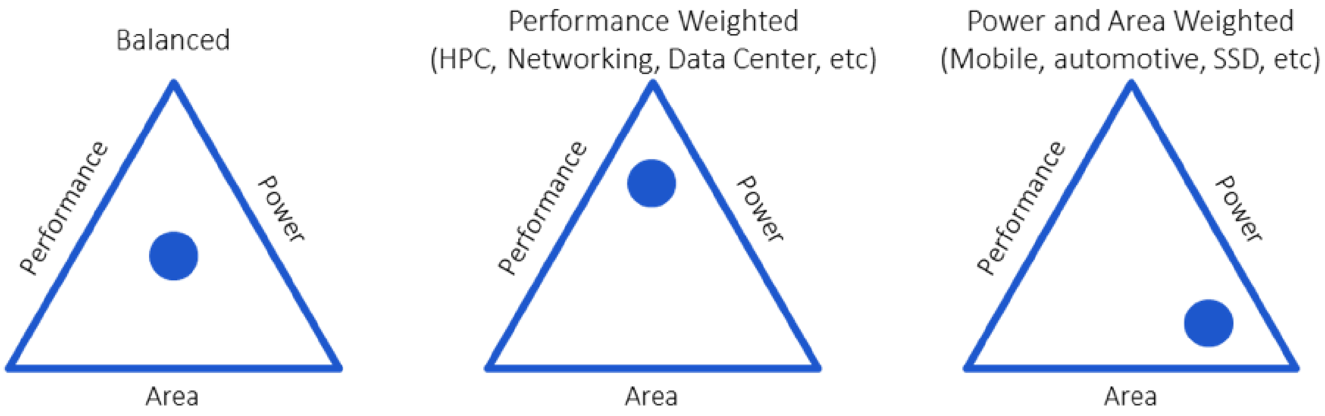
Became Orphans

Are Starved

At The Hands of These War Criminals

Introduction - PPA

- Digital VLSI chip design has mainly 3 targets
  - **Performance** (timing)
  - **Power** reduction
  - **Area** reduction
- Failing to meet the area or power requirements will lead to higher fabrication cost, higher packing cost, short battery life, etc . However, the chip will still operate correctly
- Failing to meet the timing/performance requirements will lead to a chip that doesn't work and will require redesign to fix[1]
- Because of this, timing analysis remains the main and first priority of all design checks



Balanced

Performance Weighted (HPC, Networking, Data Center, etc)

Power and Area Weighted (Mobile, automotive, SSD, etc)

[1] The fix can be simple like reducing the clock frequency or complex like changing the architecture

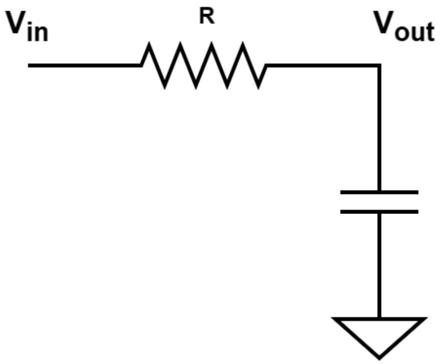
Introduction - PPA

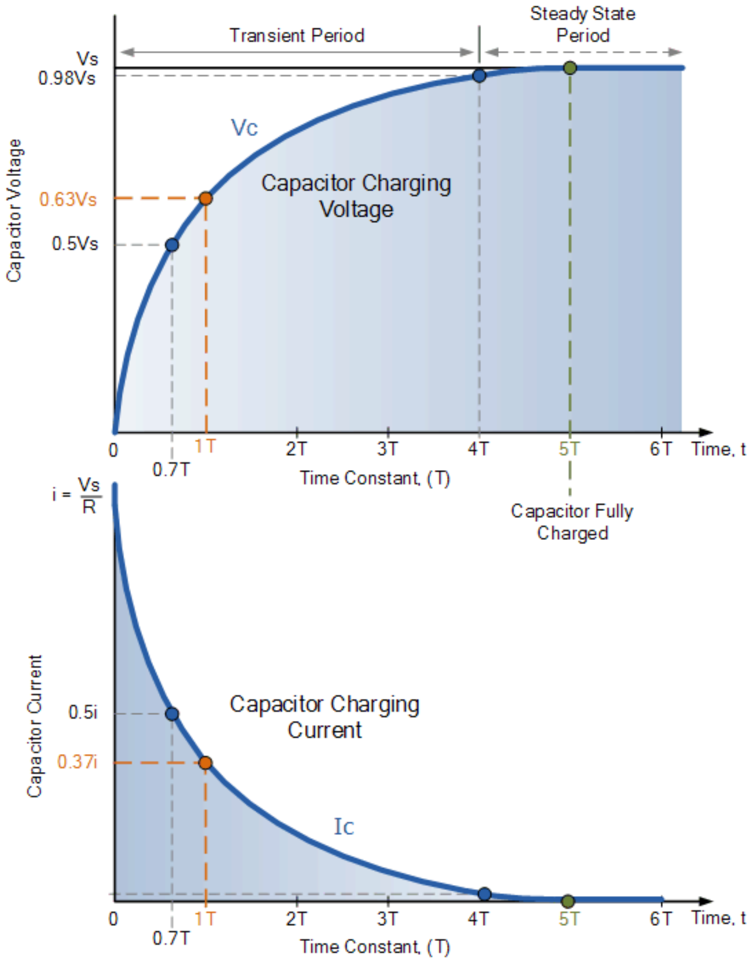
- There are many timing checks that the designer need to make sure are passing to guarantee the chip will work after fabrication.
- **They are:**
  - Setup timing
  - Hold timing
  - Max timing transition
  - Max load capacitance
  - Min pulse width
  - Max delay
  - Min delay
  - Skew
  - Recovery timing
  - Removal timing
- In this part we will go through the basic principles that are needed to understand all these checks. In the next part we will go through each check in details

Interconnect and Cell Delays

RC Delay

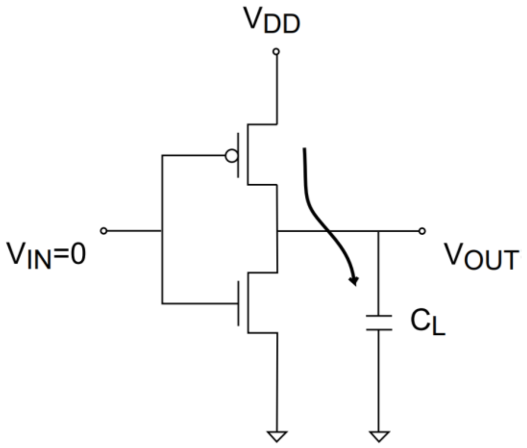
- Any electrical signal propagating through an RC circuit will take some time to charge or discharge the capacitor
- The voltage over the capacitor is governed by the following equation
  - ○ $V_{out} = V_{in}(1 - e^{\frac{-t}{RC}})$
- We say the signal has propagated through the circuit if the capacitor voltage reached 50% of the supply voltage
  - ○ If we substitute $V_{out} = \frac{V_{in}}{2}$ in the above equation we get $t = 0.69RC$
  - ○ From the time equation we can see that the propagation delay is proportional to the resistance and capacitance.

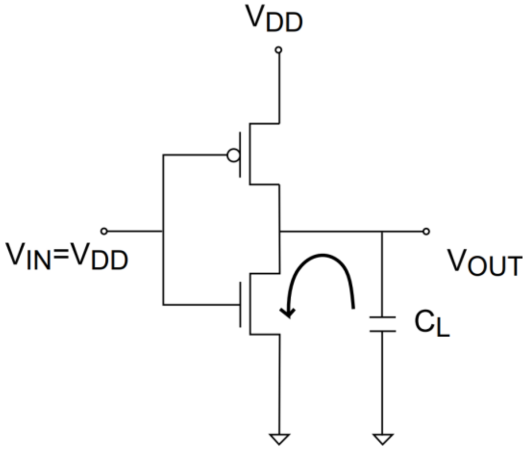




RC Delay – Cell Delay

- To calculate the propagation delay of a logic gate we can approximate[1] it as a simple RC circuit. We will consider a simple inverter.
- **When the input $\boldsymbol{V_{in} = 0}$** :
  - The upper PMOS is ON and the lower NMOS is OFF. Current will flow from the supply to charge the C_L capacitor from low to high.
  - $R_{pmos} = \frac{\Delta V}{I_{pmos}} = \frac{VDD}{\frac{W_p}{2L}\mu_p C_{ox}(VDD - V_{th})^2}$
  - $t_{LH} \ = \ 0.69 R_{pmos} C_L = \frac{0.69\, VDD \,.\, C_L}{\frac{W_p}{2L}\mu_p C_{ox}(VDD - V_{th})^2}$
- **When the input $\boldsymbol{V_{in} = VDD}$** :
  - The upper PMOS is OFF and the lower NMOS is ON. Current will flow from the capacitor to ground to discharge the C_L capacitor from high to low.
  - $R_{nmos} = \frac{\Delta V}{I_{nmos}} = \frac{VDD}{\frac{W_n}{2L}\mu_n C_{ox}(VDD - V_{th})^2}$
  - $t_{HL} \ = \ 0.69 R_{nmos} C_L = \frac{0.69\, VDD \,.\, C_L}{\frac{W_n}{2L}\mu_n C_{ox}(VDD - V_{th})^2}$





[1] For an accurate calculations : https://classes.engineering.wustl.edu/cse463/Chapter_6_CSE463.pdf

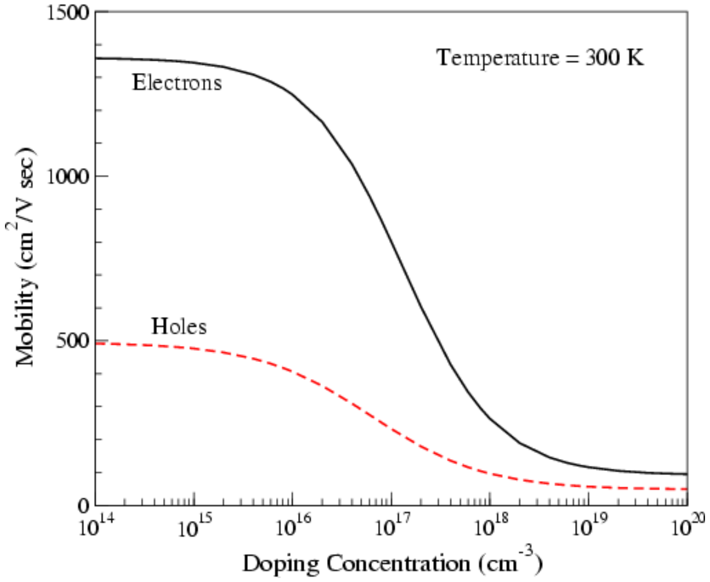
RC Delay – Cell Delay

- **From the equations we observe how to decrease the delay**
  - Increase the supply voltage ($\boldsymbol{VDD}$)
  - Decrease the threshold voltage ($\boldsymbol{V_{th}}$)
  - Decrease the load capacitance ($\boldsymbol{C_L}$)
  - Increase the transistor size ($\boldsymbol{W}$)
  - Decrease the transistor length ($\boldsymbol{L}$)

$$t_{LH} = 0.69R_{pmos}C_L = \frac{0.69\,VDD\,.\,C_L}{\frac{Wp}{2L}\mu_p C_{ox}(VDD-V_{th})^2}$$

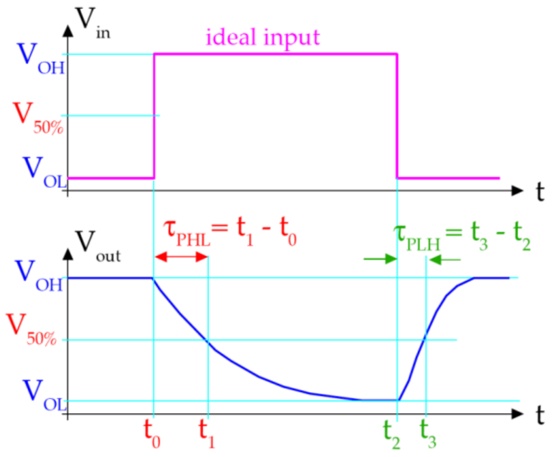
$$t_{HL} = 0.69R_{nmos}C_L = \frac{0.69\,VDD\,.\,C_L}{\frac{Wn}{2L}\mu_n C_{ox}(VDD-V_{th})^2}$$

- **The mobility (μ) of the PMOS holes is lower than the NMOS electrons.**
  - This means that if the PMOS and NMOS have the same size, t_{HL} will be different than t_{LH}
  - We can make the difference small by making the PMOS network size larger than the NMOS network

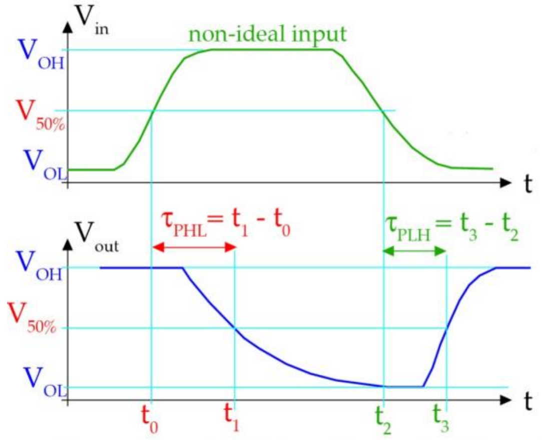


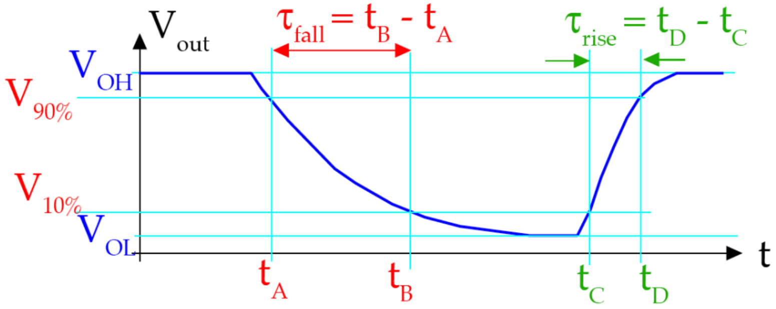
Transition Time

- When we calculated the propagation delay we assumed the input was ideal. But in reality, the input will take time to rise or fall
- For non ideal input, the propagation delay is defined to be the difference between the time when V_{in} reaches 50% and when V_{out} reaches 50%
- The slower the input transition the slower the propagation delay



- We have a clear definition for the propagation delay. We need to define the **transition time**
    - For **fall** transition time: it's the time for the signal to go from 90% of supply voltage to 10% of supply voltage
    - For **rise** transition time: it's the time for the signal to go from 10% of supply voltage to 90% of supply voltage





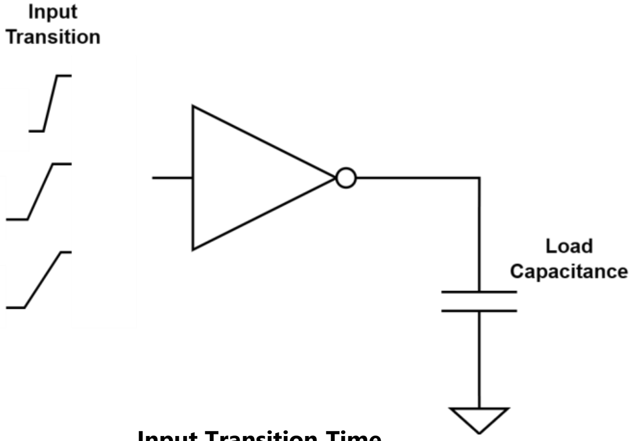
Standard Cell Libraries

Timing Tables – Sizing – MTCMOS

Timing Tables

- We discussed the propagation and transition times and have shown the parameters that controls them
- To calculate the delay of a chain of logic gates we need to substitute the parameters of each gate in the equation to get the delay of each gate then sum them together.
- **However this approach has many issues:**
  - The timing equation is an approximation. To have an accurate calculations we need to run SPICE simulation for each logic gate to get the delay.
  - There are thousands and sometimes millions of gates within a digital VLSI chip, running SPICE simulation or even calculating the approximate equation for each gate would require very huge amount of time.
- To overcome this issue, standard cell designers run simulations for each logic gate separately. The propagation and transition times are calculated at different values for input transition time and load capacitance. The values are then stored in timing tables
- In the next slide we will see how these tables are used to calculate the delay of a timing path

$$t_{prop} = \frac{0.69\, VDD\, .\, C_L}{\frac{W}{2L}\mu C_{ox}(VDD - V_{th})^2}$$



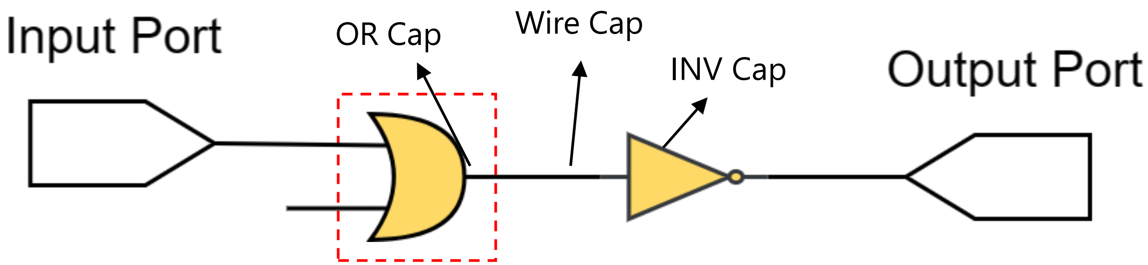
Input Transition Time
t

Load Capacitance
C_L

	1.1	1.2	1.3	1.4
10	2.10	2.20	2.27	3.00
20	2.50	3.00	3.45	3.96
30	2.90	3.40	3.80	4.15

Example Propagation Delay Timing Table

Timing Tables – Steps of Calculations



1 To calculate the delay of the OR gate, we need to know the **input transition time** and the **load capacitance.**
The input transition time from the input port needs to be manually defined by the designer. If not, the tool will assume ideal transition (0)
The load cap is calculated as the sum of the parasitics from the OR itself + the cap of the wire connected to the output of the AND + the gate capacitance of the inverter MOSFETS

2 Once the values are obtained, the STA tool will open the timing table of the OR gate to calculate the propagation delay (in case of rise and fall input transition) and the output transition times (rise and fall). Assume $t = 1.2, C_L = 20$

t →, C_L ↓

	1.1	**1.2**	**1.3**	**1.4**
10	1.83	1.91	1.97	2.61
20	2.18	2.61	3	3.45
30	2.52	2.96	3.3	3.61

OR Rise Propagation Delay Table

t →, C_L ↓

	1.1	**1.2**	**1.3**	**1.4**
10	2.10	2.20	2.27	3.00
20	2.50	3.00	3.45	3.96
30	2.90	3.40	3.80	4.15

OR Fall Propagation Delay Table

t →, C_L ↓

	1.1	**1.2**	**1.3**	**1.4**
10	1.10	1.15	1.20	1.23
20	1.20	1.25	1.28	1.30
30	1.31	1.34	1.40	1.46

OR Rise Transition Time Table

t →, C_L ↓

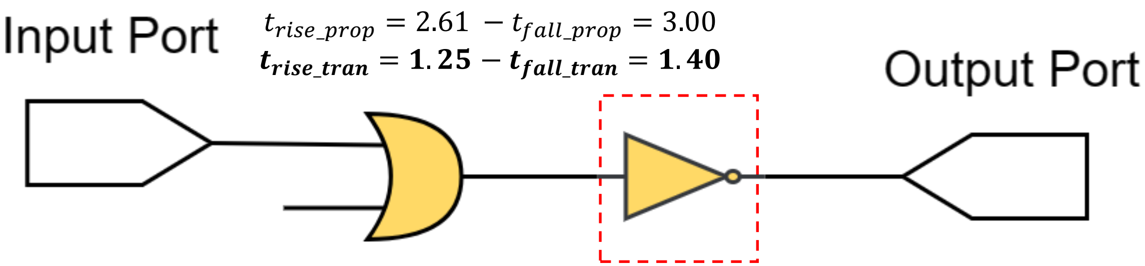
	1.1	**1.2**	**1.3**	**1.4**
10	1.30	1.35	1.42	1.55
20	1.38	1.40	1.53	1.64
30	1.45	1.51	1.60	1.70

OR Fall Transition Time Table

Timing Tables – Steps of Calculations

$t_{rise\_prop} = 2.61\ - t_{fall\_prop} = 3.00$

$\boldsymbol{t_{rise\_tran} = 1.25 - t_{fall\_tran} = 1.40}$



3 Now we calculate the delay of the inverter based on the values we obtained. The input transition time of the inverter is the output transition time of the OR gate. The only thing missing is the load capacitance seen from the output ports. The designer needs to manually define this value or the tool will assume zero cap

4 When the OR gate rises, the inverter will fall and vice verse. Hence, to calculate the fall times of the inverter we will use the rise times of the OR and vice verse. Assume $C_L = 20$

5 The rise transition time 1.25 is not in the table index, so the tool will do linear interpolation to get the required value

t

C_L

	1.1	1.2	1.3	1.4
10	1.96	2.04	2.10	2.78
20	2.33	2.78	3.20	3.68
30	2.69	3.16	3.53	3.85

INV Rise Propagation Delay Table

$$\frac{x - 2.67}{\mathbf{1.25} - 1.2} = \frac{3.07 - 2.67}{1.3 - 1.2}\ , \qquad x = 2.87$$

t

C_L

	1.1	1.2	1.3	1.4
10	1.87	1.96	2.02	2.67
20	2.23	2.67	3.07	3.52
30	2.58	3.03	3.38	3.69

INV Fall Propagation Delay Table

t

C_L

	1.1	1.2	1.3	1.4
10	0.98	1.02	1.07	1.09
20	1.07	1.11	1.14	1.16
30	1.17	1.19	1.25	1.30

INV Rise Transition Time Table

$$\frac{x - 1.29}{\mathbf{1.25} - 1.2} = \frac{1.36 - 1.29}{1.3 - 1.2}\ , \qquad x = 1.325$$

t

C_L

	1.1	1.2	1.3	1.4
10	1.16	1.20	1.26	1.38
20	1.23	1.29	1.36	1.46
30	1.29	1.34	1.42	1.51

INV Fall Transition Time Table

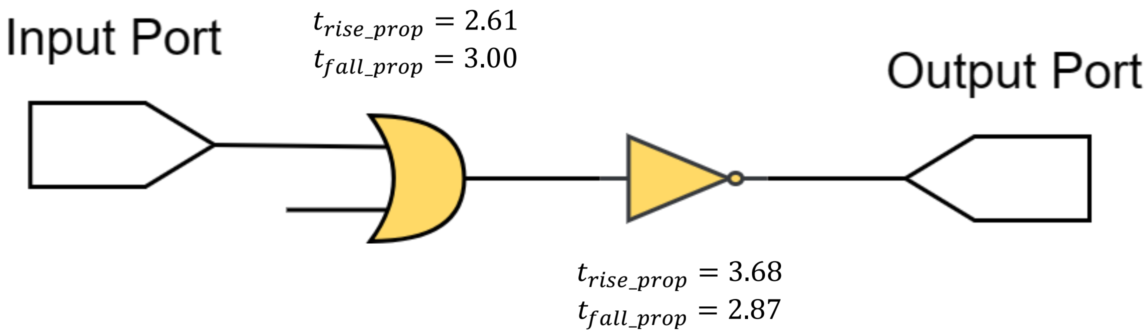
Timing Tables – Steps of Calculations

6 We now have all the propagation delay values we need. As we showed earlier, If the OR gate rises the INV will fall and vice versa. So to calculate the total delay of this timing path:

$$total\ delay = OR_{rise} + INV_{fall} = 2.61 + 2.87 = 5.48$$

Or

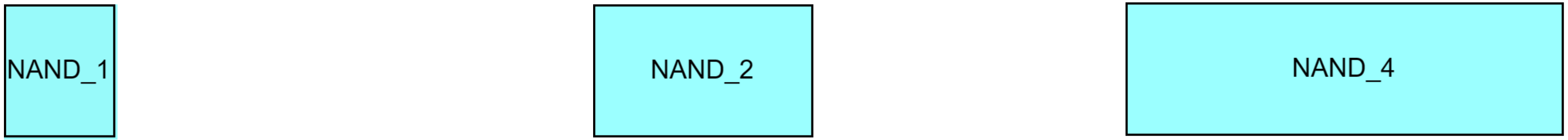
$$total\ delay = OR_{fall} + INV_{rise} = 3.00 + 3.68 = 6.68$$



Sizing

- We can see from the delay equation how the transistor size $(\frac{W}{L})$ affects the delay
- Standard cell designers create multiple cells of the same function with different sizes to get different delays

$$t_{prop} = \frac{0.69\, VDD\ .\ C_L}{\frac{W}{2L}\mu C_{ox}(VDD - V_{th})^2}$$



t

C_L

	1.1	1.2	1.3	1.4
10	2.10	2.20	2.27	3.00
20	2.50	3.00	3.45	3.96
30	2.90	3.40	3.80	4.15

NAND_1 Prop Delay Table

t

C_L

	1.1	1.2	1.3	1.4
10	1.60	1.67	1.73	2.28
20	1.90	2.28	2.62	3.01
30	2.20	2.58	2.89	3.15

NAND_2 Prop Delay Table

t

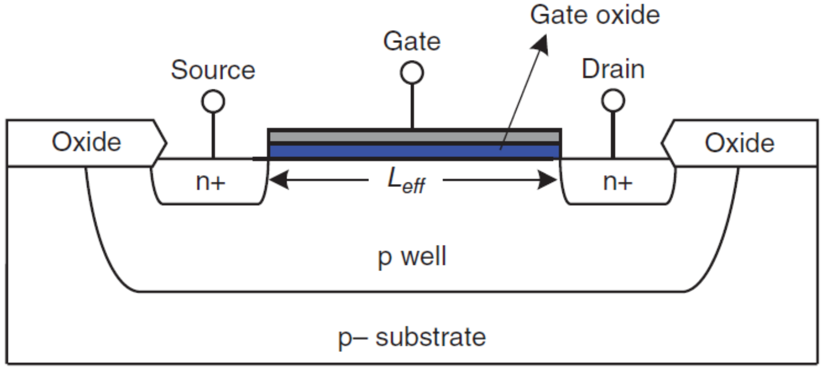
C_L

	1.1	1.2	1.3	1.4
10	1.05	1.10	1.14	1.50
20	1.25	1.50	1.73	1.98
30	1.45	1.70	1.90	2.08

NAND_4 Prop Delay Table

MTCMOS

- Similarly, the threshold voltage V_{th} affects the delay.
- The V_{th} can be controlled during fabrication by controlling the oxide thickness of the MOSFET. A thin oxide will have lower V_{th} and hence smaller delay but higher leakage power consumption
- Another way to control V_{th} is by controlling the doping in the channel, as the doping increases, the threshold voltage increases
- Standard cell designers create multiple cells of the same function with different oxide thickness to get different delays. The different versions are called **HVT** (High Voltage threshold), **SVT** (Standard), **LVT** (Low), **ULT** (Ultra Low), **ELT** (Extreme Low)
- This technology is called Multi Threshold CMOS (MTCMOS)



t

C_L

	1.1	1.2	1.3	1.4
10	2.10	2.20	2.27	3.00
20	2.50	3.00	3.45	3.96
30	2.90	3.40	3.80	4.15

HVT NAND_1 Prop Delay Table

t

C_L

	1.1	1.2	1.3	1.4
10	1.79	1.87	1.93	2.55
20	2.13	2.55	2.93	3.37
30	2.47	2.89	3.23	3.53

SVT NAND_1 Prop Delay Table

t

C_L

	1.1	1.2	1.3	1.4
10	1.58	1.65	1.70	2.25
20	1.88	2.25	2.59	2.97
30	2.18	2.55	2.85	3.11

LVT NAND_1 Prop Delay Table

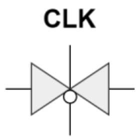
Flip Flop Times

Setup – Hold – Tcq

Flip Flop Internal Operation

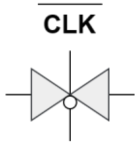
- To understand setup and hold timing we need to look into the internal workings of a flip flop
- The diagram below shows one way to implement D flip flops using inverters and transmission gates.
- The transmission gates acts as a switch that opens or closes depending on a control signal
- The inverter loops are the storage elements that store the data

Transmission Gates



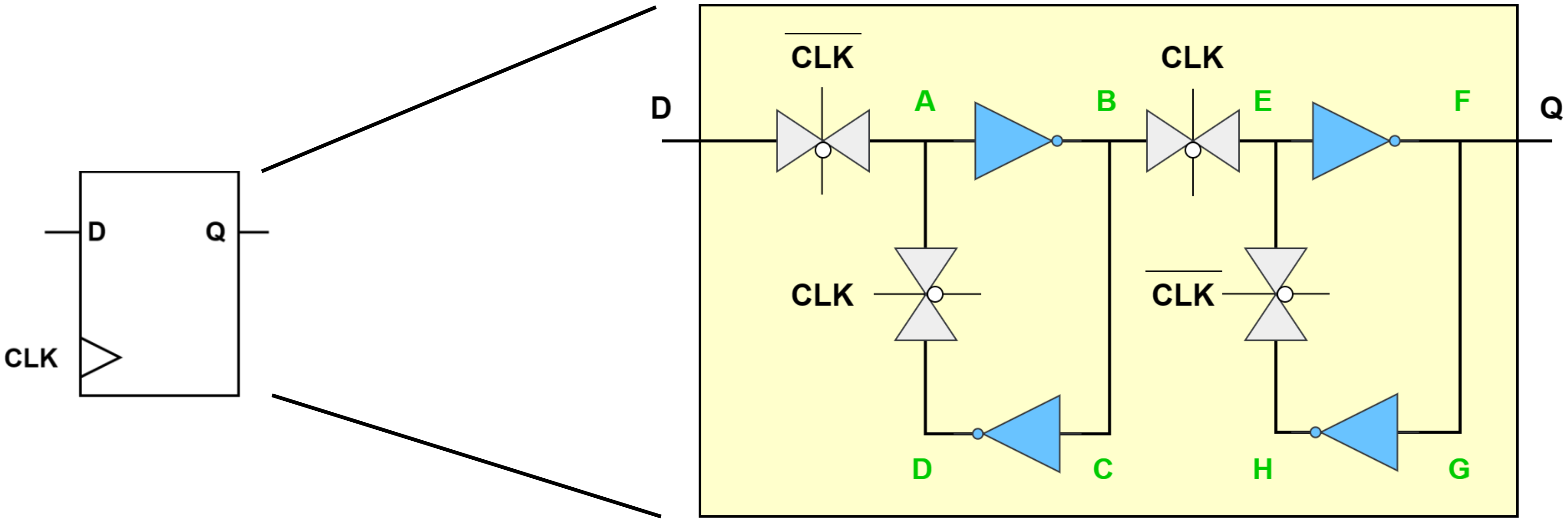
Short circuit when CLK=1

Open circuit when CLK=0



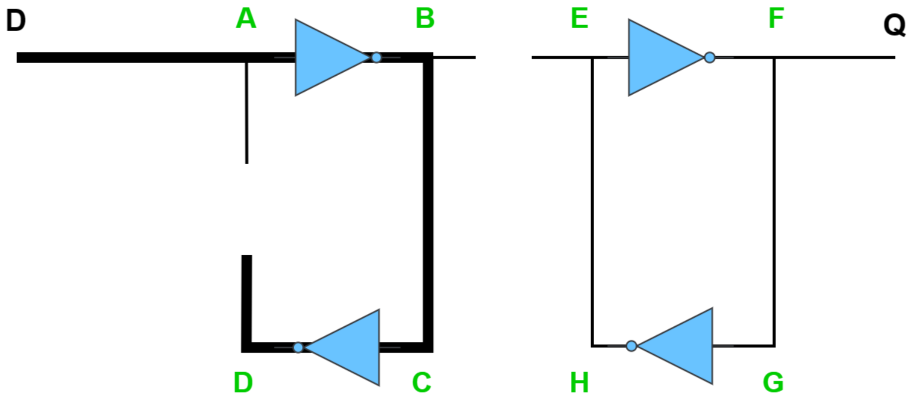
Short circuit when CLK=0

Open circuit when CLK=1

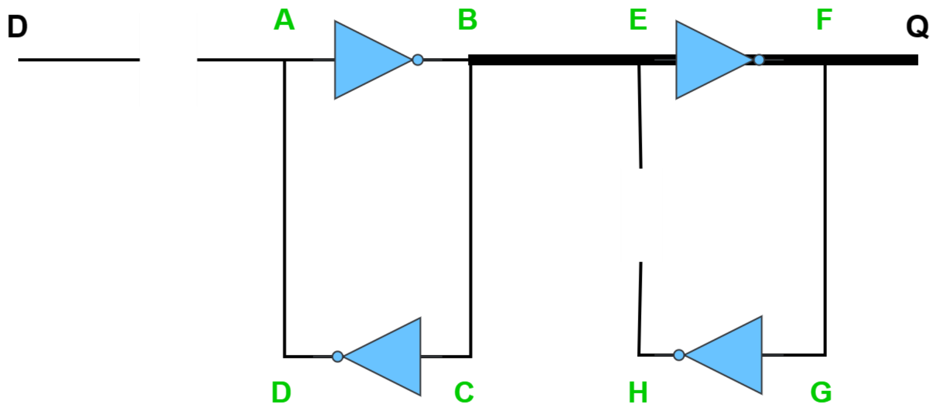


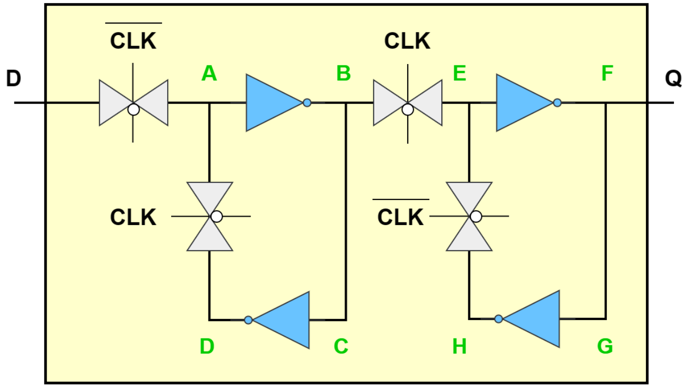
Flip Flop Internal Operation

1 Before the clock edge arrives (CLK=0), the input goes from the input pin D through **A-B-C-D** and waits for the clock edge.



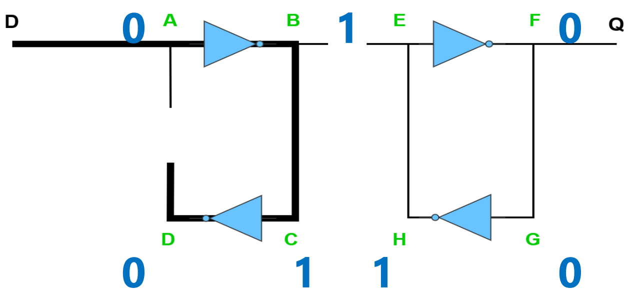
2 After the clock edge arrives (CLK=1), The data flow through **B-E-F** to the output pin Q.



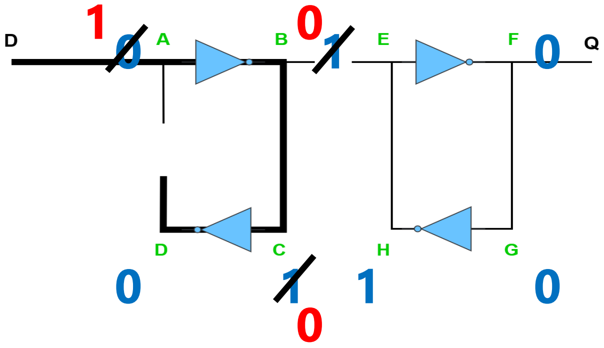


Setup Time

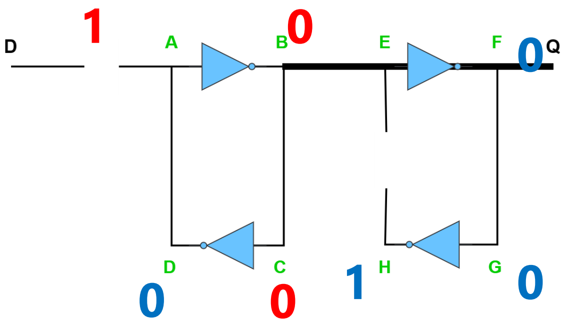
1 To understand how a setup violation happens lets go through this scenario:
Lets assume the FF was storing a logic zero (0)



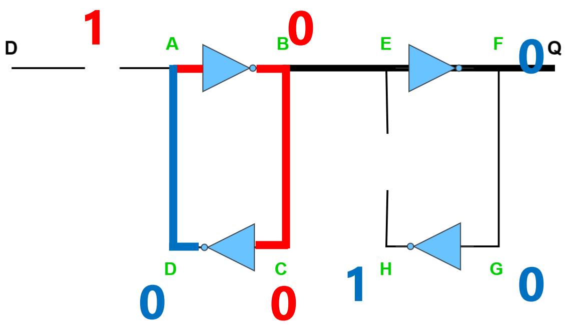
2 Now a **new data** arrives at the D input ,that is logic one (1), and starts overwriting the previous stored value



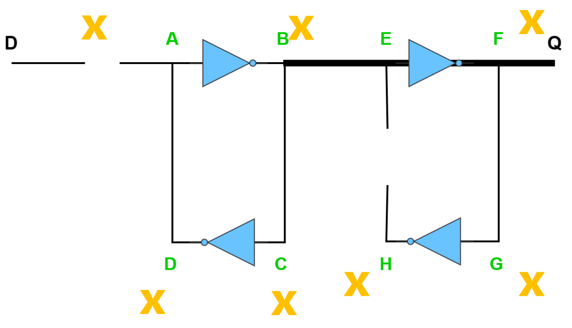
3 The clock edge arrives before the new data have time to overwrite node **D**. The transmission gates switch



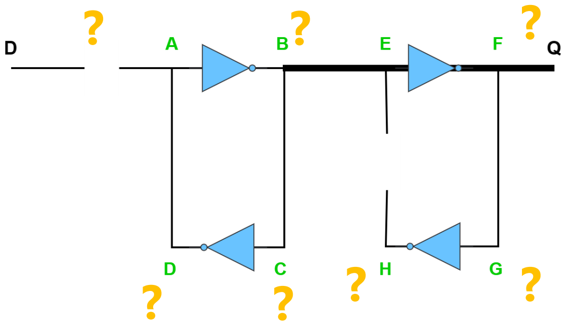
4 The transmission gate between **D** & **A** is now a short circuit, so **D** is trying to force the inverter loop to store the **old data** while **A-B-C** is trying to force the loop to store the **new data**



5 The conflict between the two electrical values will propagate to all the nodes in the FF and the output won't be a 0 or 1. The FF is said to be in a metastable state

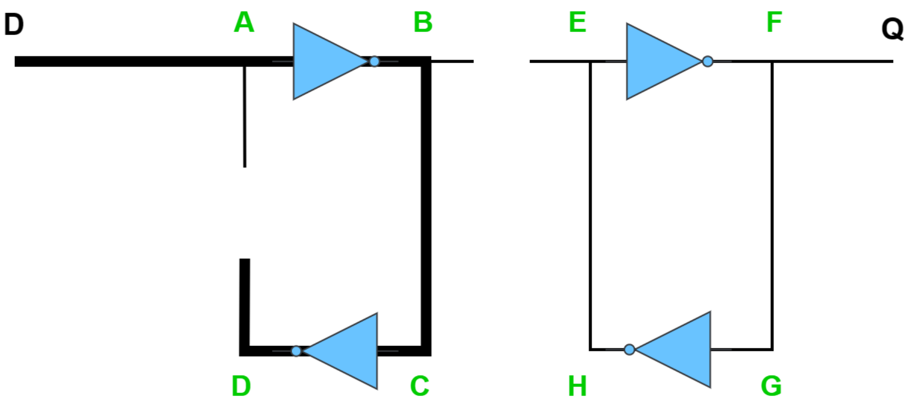


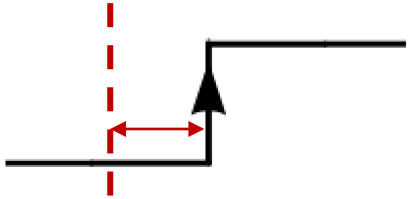
6 After some time one of the 2 values will overcome the other and the FF will leave the metastable state. The final state could be the **old data** or the **new data**



Setup Time

- To avoid the metastability issue we need to make sure the new data propagate through and overwrite all the nodes **A-B-C-D** before the clock edge arrives
- The setup time is then the delay from D pin to **A-B-C-D** nodes

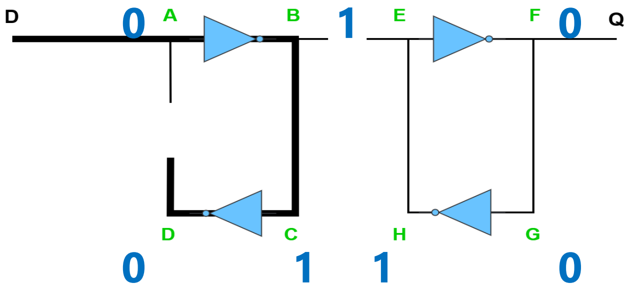




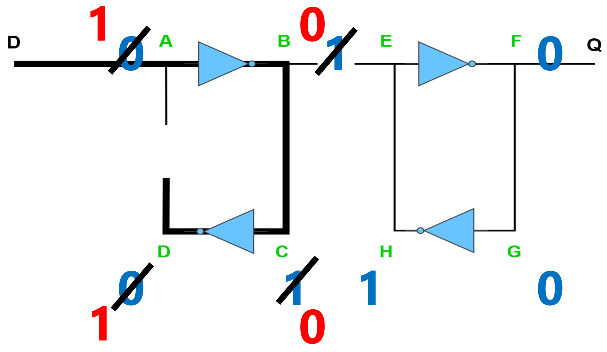
Setup time
The data must arrive at the D pin some time before the clock edge arrives, to give time for the internal nodes A-B-C-D to be overwritten

Hold Time

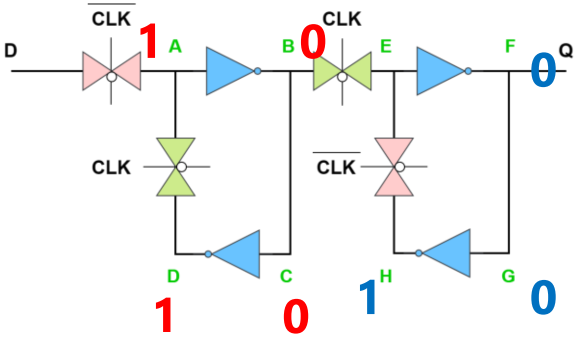
1 To understand how a hold violation happens lets go through this scenario:
Lets assume the FF was storing a logic zero (0)



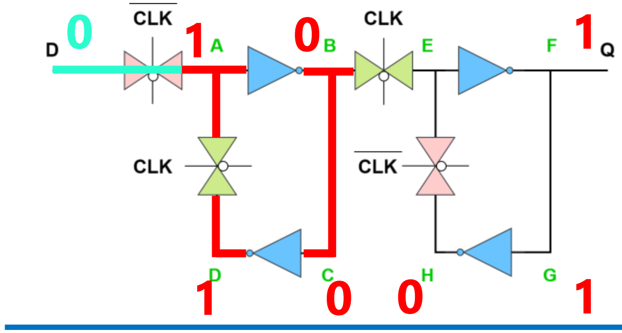
2 Now a **new data** arrives at the D input ,that is logic one (1), and starts overwriting the previous stored value



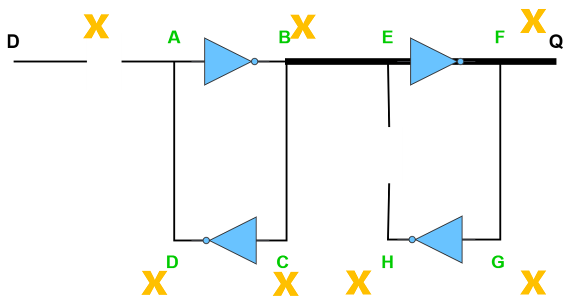
3 The clock edge arrives, the red transmission gates starts to open while the green ones starts to close (short)



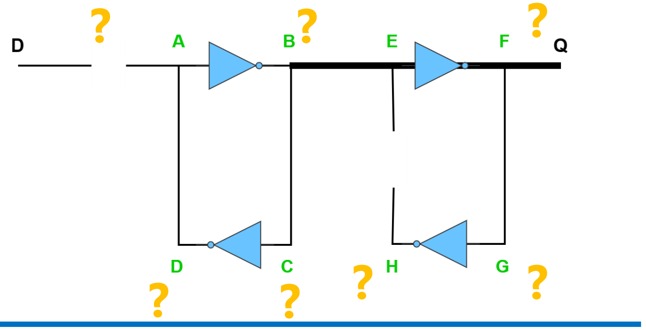
4 Before the red gates completely open, a **newer data** arrives at the D pin, The signal at the D pin is trying to force the inverter loops to store the **newer data**, the nodes **A-B-C-D** are trying to force it to store the **new data**



5 The conflict between the two electrical values will propagate to all the nodes in the FF and the output won't be a 0 or 1. The FF is said to be in a metastable state

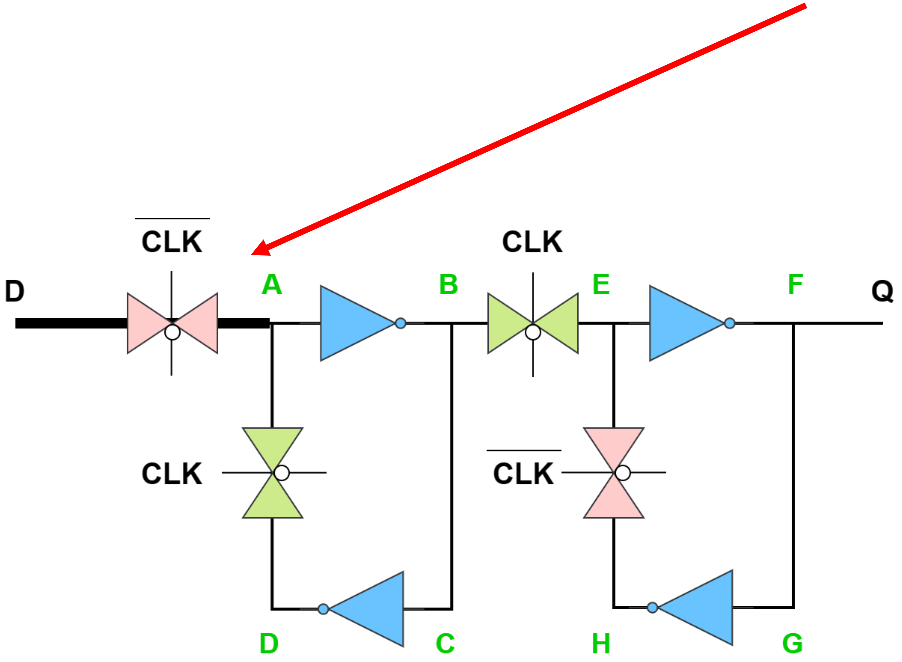


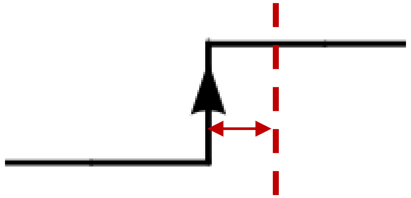
6 After some time one of the 2 values will overcome the other and the FF will leave the metastable state.



Hold Time

- The metastability happened because the transmission gates take some time to fully open or close.
- We need to make sure no new data arrives at the D pin until the red transmission gate is fully open circuit
- The hold time is then the transition time of the red transmission gate from the D pin to **A** node

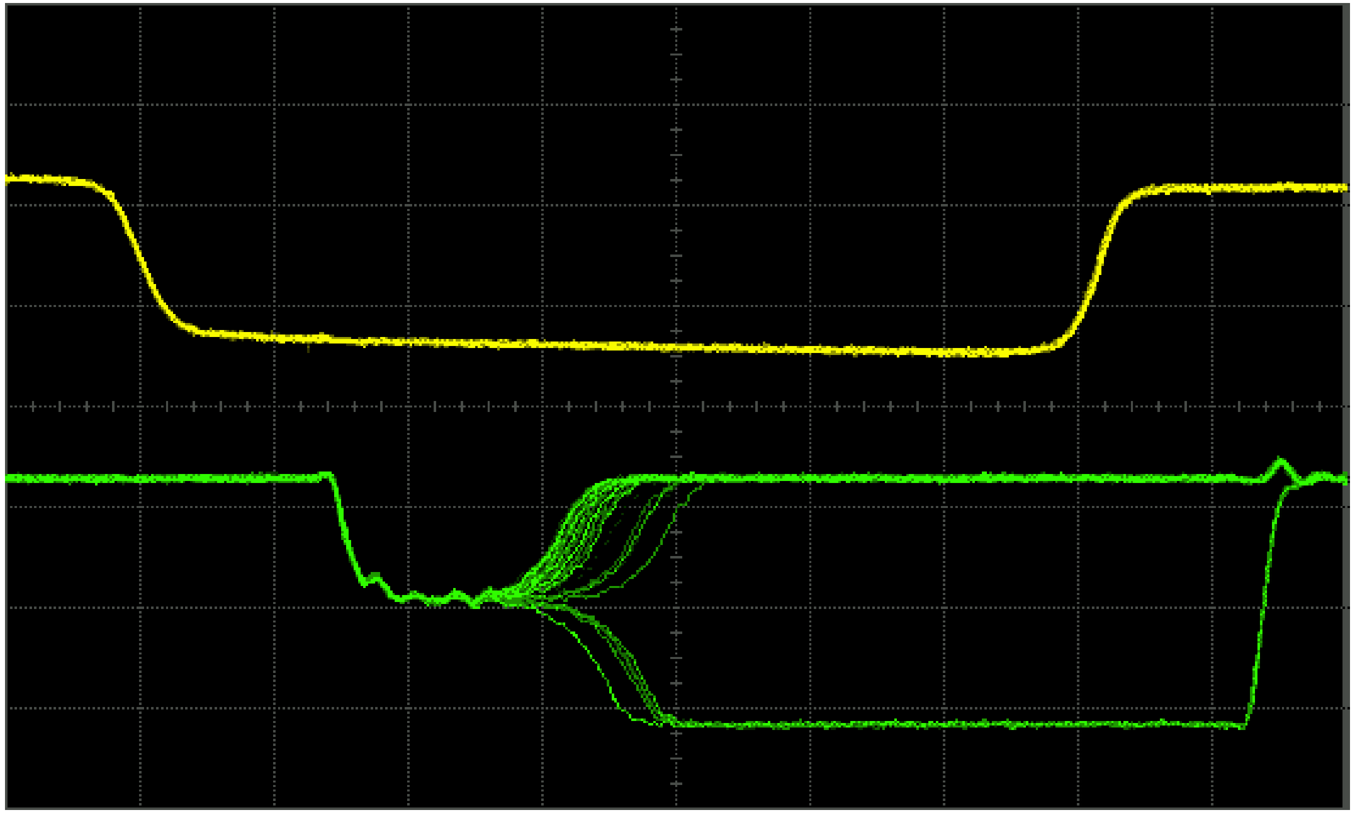




Hold time
The value at the D pin must remain constant for some time after the clock edge arrives, to give time for the transmission gate to become open circuit

Metastability

- The image below shows metastability in a FF. The FF becomes metastable (not 0 or 1) for some time before it settles to 0 or 1

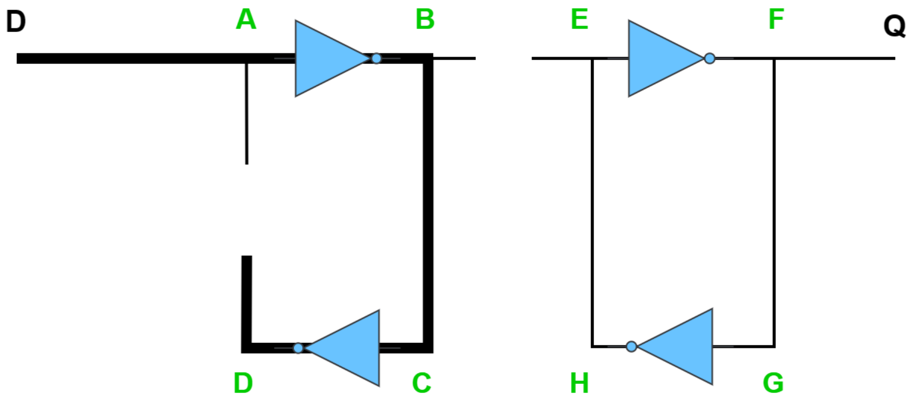


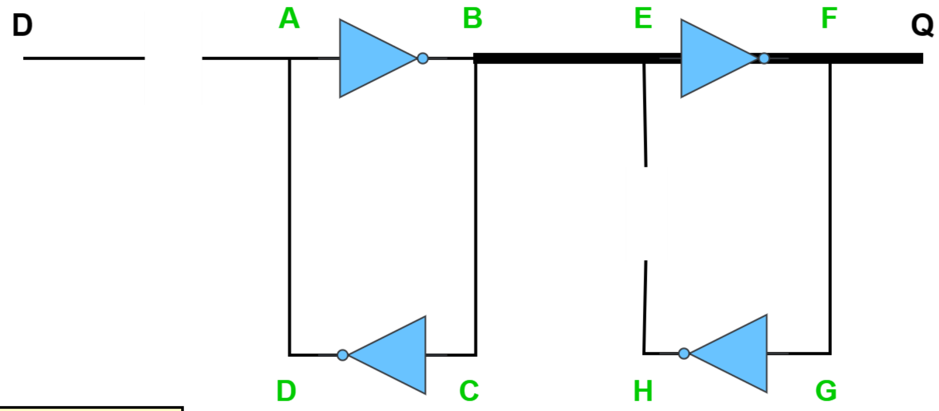
Picture taken from W. J. Dally, Lecture notes for EE108A, Lecture 13: Metastability and Synchronization Failure 11/9/2005.

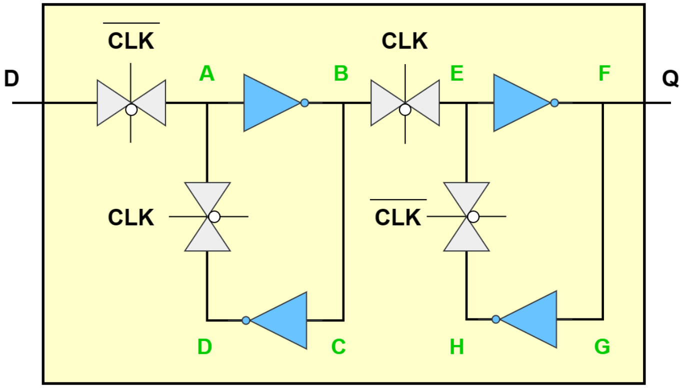
Tcq Time

1 Before the clock edge arrives (CLK=0), the input goes from the input pin D through **A-B-C-D** and waits for the clock edge.

2 After the clock edge arrives (CLK=1), The data flow through **B-E-F** to the output pin Q. This delay through **B-E-F** to the output pin Q is called clock-to-Q time or Tcq







in /amradelm

References

- [1] https://classes.engineering.wustl.edu/cse463/Chapter_6_CSE463.pdf
- [2] https://www.iue.tuwien.ac.at/phd/park/node30.html
- [3] https://www.electronics-tutorials.ws/rc/rc_1.html
- [4] http://web.mit.edu/6.012/www/SP07-L13.pdf
- [5] https://www.linkedin.com/pulse/understanding-power-performance-area-ppa-analysis-vlsi-priya-pandey/
- [6] https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ecc04789069f3e19bebe5814ce3608aa609e4403

Thank You!