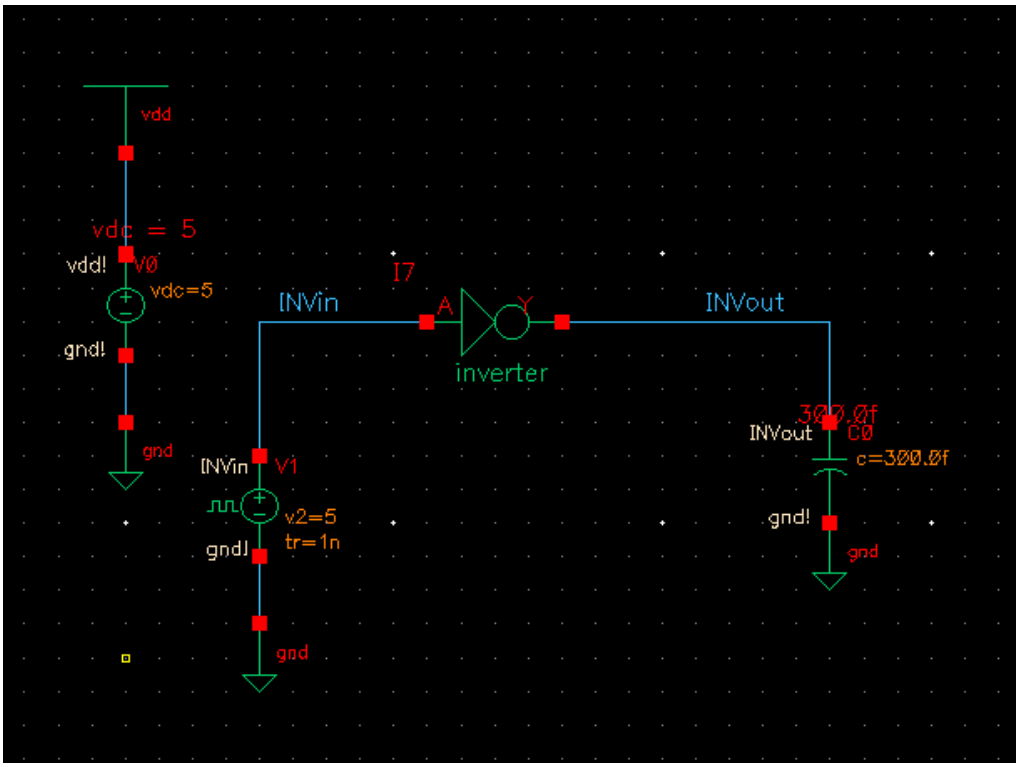
x

ECE520: Lab4 Report – Parameterized Driver, Propagation Delays

Name: Arijit Sengupta, ID: 001441742

1) Hierarchical editing the cells

The **TBdriver** cell is created with the original inverter cell from Lab 3 in it.



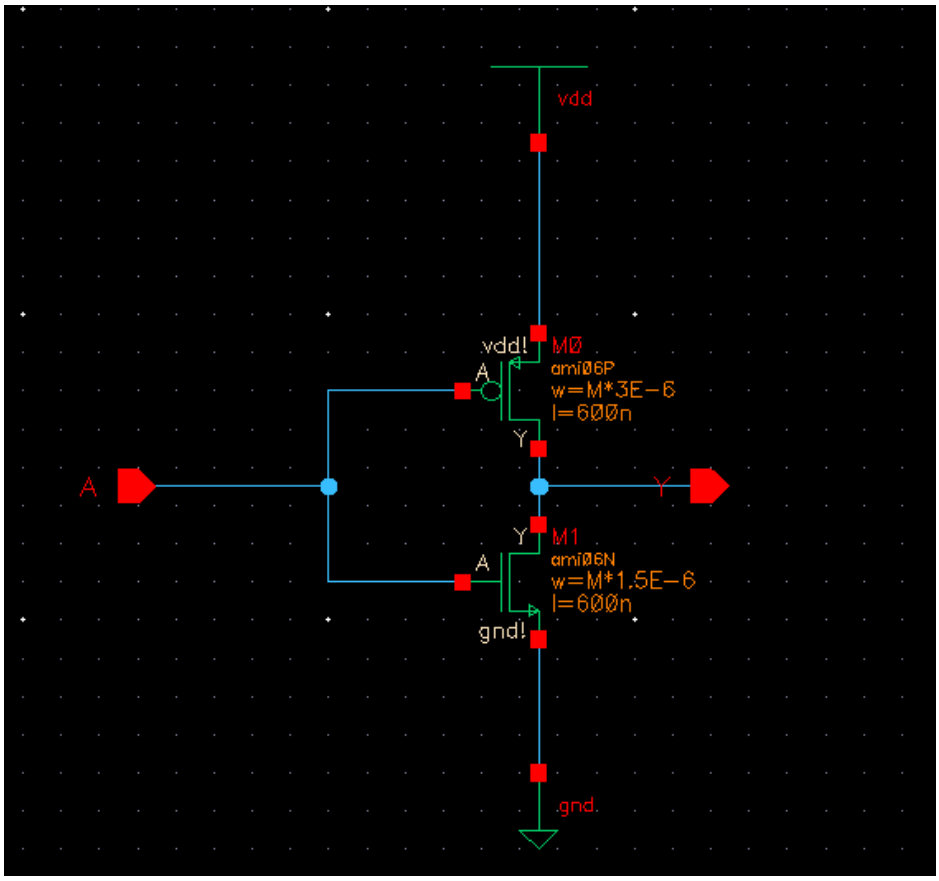
There are Cadence commands for this up and down tree traversal:

- **SHIFT+X** allows to Descend to see the lower levels of the tree.
- **SHIFT+B** allows to Ascend to see the upper levels of the tree.

This circuit will be edited in the next step.

2) Parameterizing the Symbol and Schematics

A copy of the inverter cell is saved as **pinv** where pinv stems for *parametric inverter*.
The width (W) of pmos is changed to **M*3E-6** and the width (W) of nmos is changed to **M*1.5E-6.**

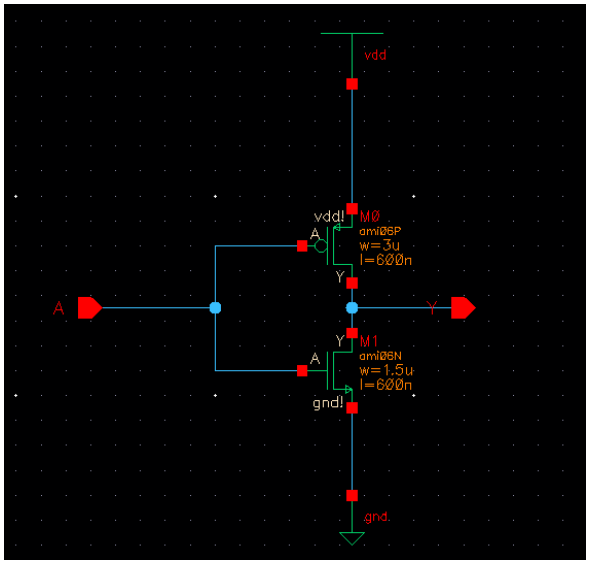


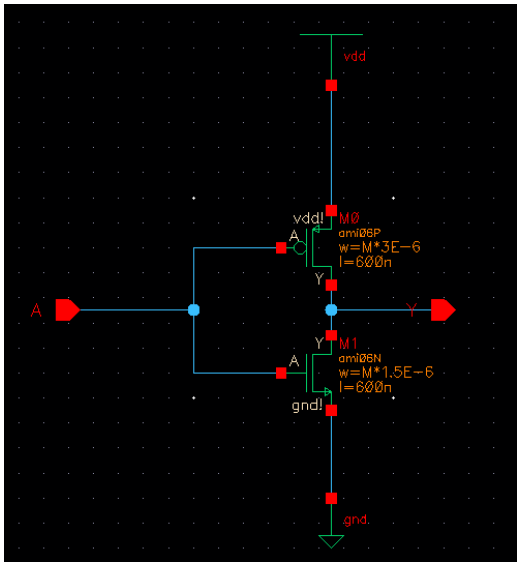
3) Creating an exponentially-scaled driver circuit

4 different copies of pinv are created called pinv1, pinvM, pinvMM, pinvMMM.

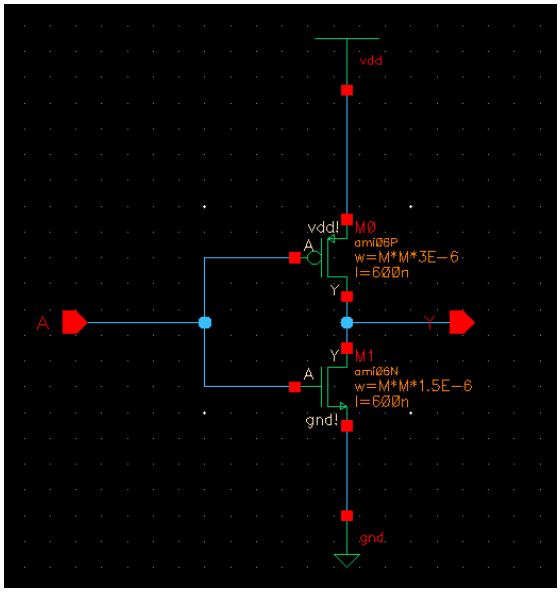
The idea is that these 4 copies will all be M times bigger than the previous one.

- pinv1 will be a size 1 inverter containing a size 1 nmos and size 2 pmos.
- pinvM will be a size M inverter containing a size M nmos and size 2M pmos.
- pinvMM will be a size M^2 inverter containing a size M^2 nmos and size $2M^2$ pmos.
- pinvMMM will be a size M^3 inverter containing a size M^3 nmos and size $2M^3$ pmos.

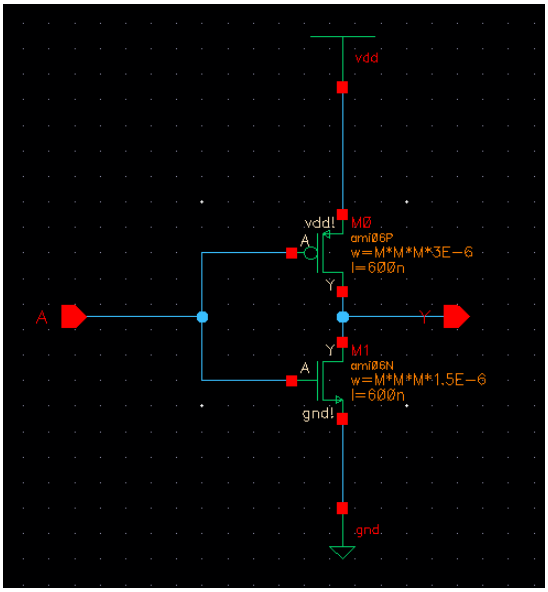




pinv1



pinvM



pinvMM

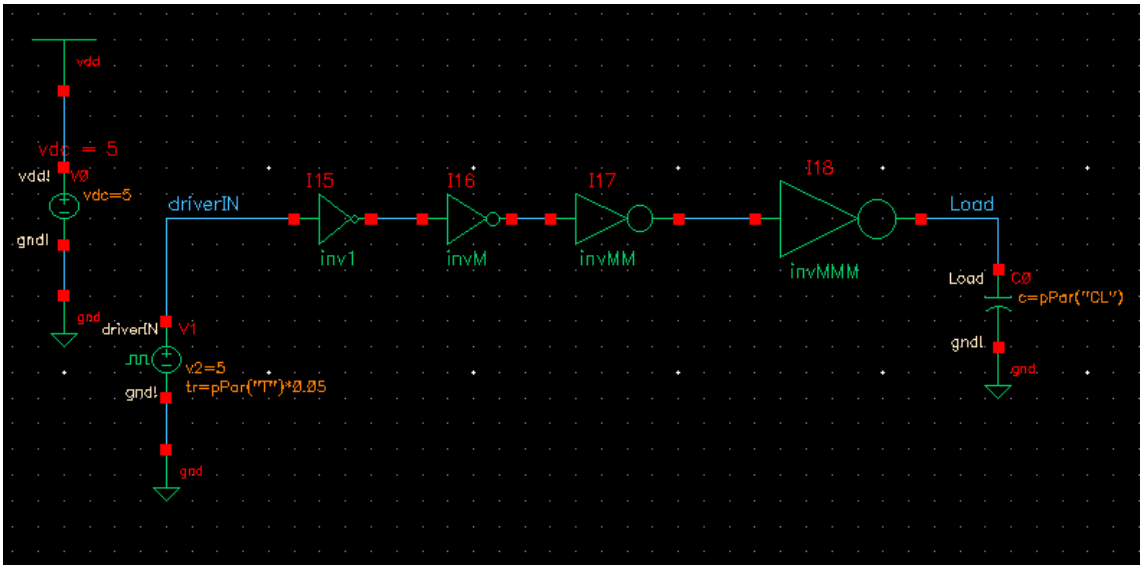
pinvMMM

In the TBdriver circuit, the capacitance is changed to **pPar("CL")**, making it also parametric.

The pulse generator is made parametric too using a single parameter **pPar("T").**

Rise & Fall times = pPar("T")*0.05, Pulse width = pPar("T")*0.45

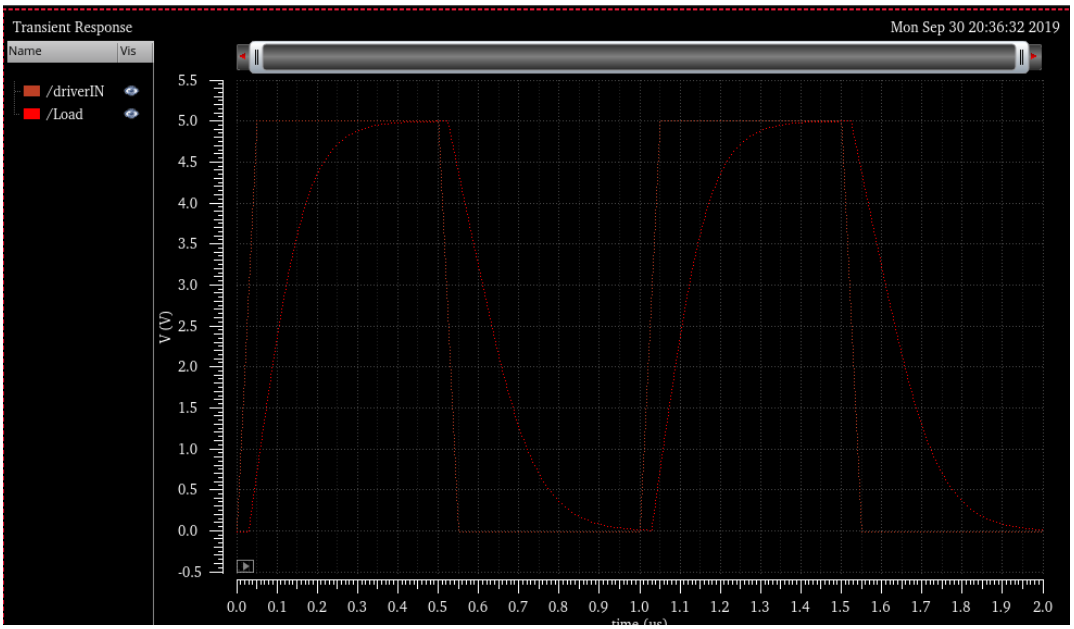
Therefore, the **TBdriver** schematic:




```
INFO (SCH-1170): Extracting "TBdriver schematic"
INFO (SCH-1426): Schematic check completed with no errors.
Getting schematic propert bagGetting schematic propert bagINFO (SCH-1181): "LAB4 TBdriver schematic" saved.
```

4) Measuring the propagation delays

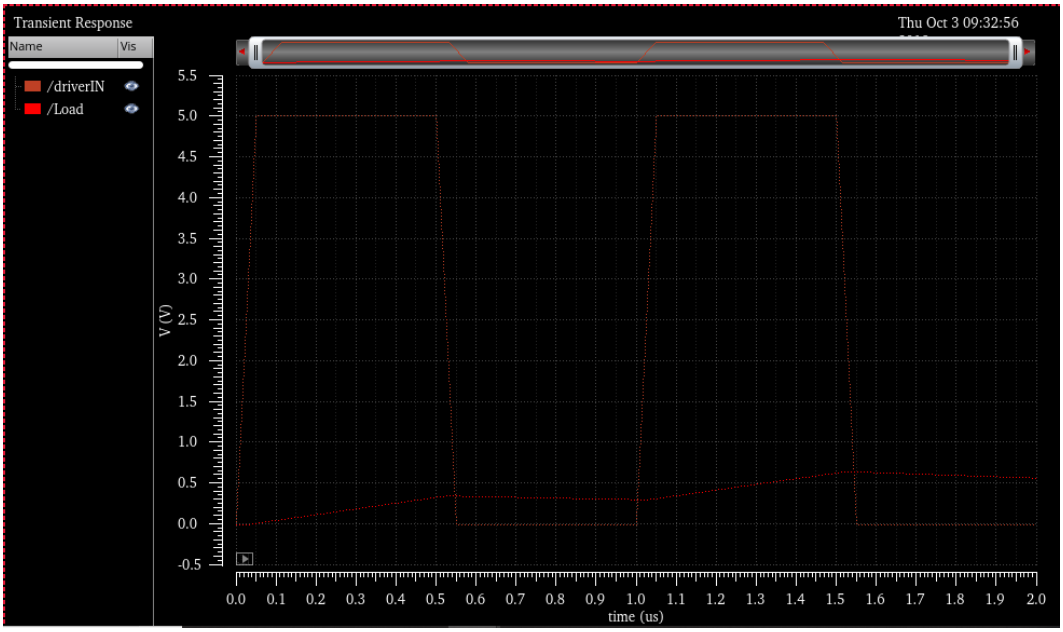
Analysis 1: For the values - **M=1, CL=20p, T=1u**



Therefore, the rising propagation delay $\mathbf{t_{pdr}}$= 103.52ns – 25.50ns = **78.02ns**

The falling propagation delay $\mathbf{t_{pdf}}$ = 631.47ns - 525.77ns = **105.70ns**

Analysis 2: Progressively increase the load (CL) to 1nF, keeping **M** and **T** the same.
The rise/fall propagation delay increases substantially, because the load is 50x higher. The output signal is unable to reach all the way up to 5V, and manages to reach around only 650mV.



Therefore, the rising propagation delay $\mathbf{t_{pdr}}$= 533.98ns – 26.30ns = **507.68ns**

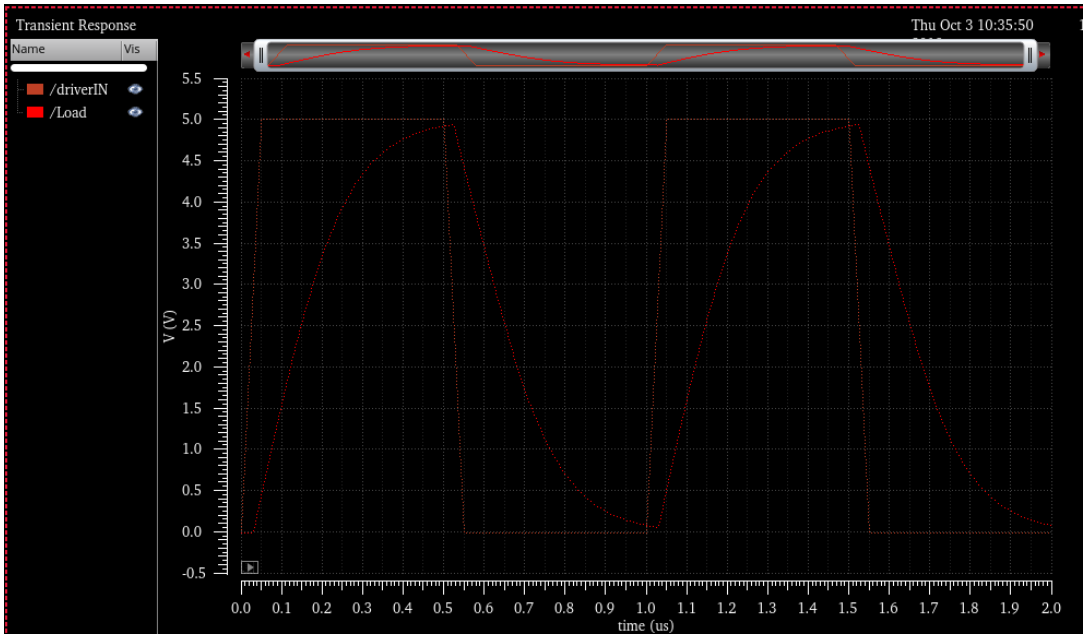
The falling propagation delay $\mathbf{t_{pdf}}$ = 1028.47ns - 525.29ns = **503.18ns**

Thus, it shows that the drive capability of the final inverter is horrible.

This can be overcome by optimizing **M**, the ratio of the progressive increase. By setting **CL = 1nF** (or 1000 pF), we made the load 50x higher from the previous analysis where CL = 20pF. So, the

drive capability must be 50x better, meaning 50x higher transistor sizes for the very last inverter. This must be achieved in 3 consecutive exponential steps (i.e., the last inverter is size M^3). Intuitively, we expect $M^3 \approx 50$. So, $M \approx 50^{1/3} \approx 3.68$. So, we will rerun with M=3 and M=4.

Analysis 3: For the values - **M=3, CL=1n, T=1u**

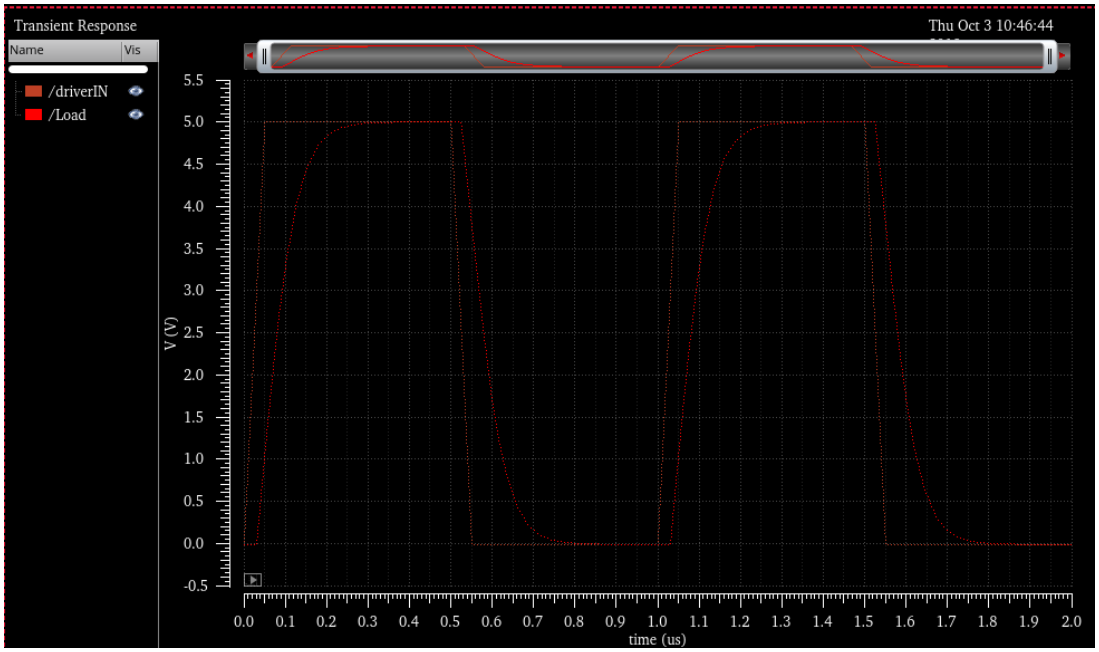


Therefore, the rising propagation delay t_{pdr}= 146.91ns – 25.60ns = **121.31ns**

The falling propagation delay t_{pdf} = 651.49ns - 525.64ns = **125.85ns**

Thus, it is observed that the propagation delay **decreases** when M is increased to 3, but it is significantly higher than the initial value when CL was 20pF.

Analysis 4: For the values - **M=4, CL=1n, T=1u**



Therefore, the rising propagation delay t_{pdr}= 81.04ns - 25.73ns = **55.31ns**

The falling propagation delay $\mathbf{t_{pdf}}$ = 580.05ns - 525.64ns = **54.41ns**

Thus, it is observed that the propagation delay **decreases** when M is increased to 4 and it is significantly lower than the initial value when CL was 20pF, which proves that the load is driven better when M is optimized.

Analysis 5: Multiple analysis are done to obtain the optimum values of M when T and CL are changed. The values obtained are highlighted in the table below –

For **T= 1µs:**

CL	M				
	1	**3**	**4**	**5**	**8**
20pF	**78.02ns**	6.44ns	6.28ns	6.14ns	6.74ns
100pF	225.09ns	**16.99ns**	10.27ns	8.14ns	7.18ns
1nF	507.68ns	121.28ns	**55.39ns**	41.15ns	12.78ns
3nF	978.73ns	228.83ns	147.03ns	**82.22ns**	25.34ns
10nF	1150.36ns	296.8ns	226.11ns	201.47ns	**67.98ns**

Thus, the optimised values of M for different CL values are highlighted above (in bold), where the propagation delay is somewhat constant and less than 100ns, which cannot be drastically improved by keeping on increasing the value of M.

The optimized M value for 100pF can be considered as 2 also, since the propagation delay is somewhere between 50ns and 70ns, which are acceptable values in terms of the circuit.

Therefore, the optimum M values are as follows:

20pF – M=1, 100pF – M=2, 1nF – M=3, 3nF – M=4 and 10nF – M=8.

The last inverter for M=8 becomes 8^3=512 times higher compared to that for M=1. This however makes the channel width (W) very high, and makes the surface area of the TBdriver circuit impractically very high. This should be kept in mind during design of the circuit because this will give rise to a significantly larger circuit.

For **T= 500ns:**

CL	M				
	1	**3**	**4**	**5**	**8**
20pF	**74.57ns**	5.35ns	4.29ns	4.09ns	4.45ns
100pF	139.33ns	**14.87ns**	8.19ns	6.07ns	4.96ns
1nF	237.59ns	101.48ns	**52.8ns**	28.38ns	10.51ns
3nF	557.43ns	139.28ns	125.23ns	**76.61ns**	22.82ns
10nF	662.46ns	194.29ns	147.11ns	111.96ns	**63.88ns**

For **T=200ns:**

CL	M				
	1	**3**	**4**	**5**	**8**
20pF	**53.55ns**	4.15ns	2.99ns	2.72ns	2.91ns
100pF	77.67ns	**13.56ns**	7.22ns	4.74ns	3.41ns
1nF	95.11ns	49.49ns	**41.28ns**	27.23ns	8.94ns
3nF	377.22ns	68.89ns	50.98ns	**46.95ns**	21.33ns
10nF	548.17ns	95.83ns	84.22ns	70.31ns	**46.1ns**

Thus, it can be observed that when the switching is faster (i.e. T is smaller), the propagation delay is less comparatively, keeping the other values constant.

The above analysis for 3 T values show that increasing the value of the parameter M decreases the propagation delay subsequently. However, when the value of T is 500ns or 200ns, the Load value fails to reach 5V with M<3 typically. This is due the faster transition from HIGH state to LOW state.

Therefore, even though the propagation delay is less than 100ns for M=1 or M=3 for certain values of CL (20pF, 100pF), the output does not reach 5V. Hence these values of M cannot be actually said to be optimised with respect to the other parameters of the circuit.

With further testing, it is observed that the optimised M value for T=500ns (-1.5x greater) and T=200ns (-2.5x greater) should be greater than that for T=1µs. This proves that the channel length is directly dependent on the frequency of the circuit, because high-frequency operations such as T = 200ns requires faster switching which can be easily overcome by choosing parameters.

Final Inference:

The value of M subsequently controls the propagation delay, when T<=1µs. For faster transitions, the M should be increased accordingly, so that the output reaches 5V before the capacitor starts discharging and the circuit undergoes transition (HIGH to LOW or vice-versa).

The optimum values are obtained in the table above, and it is seen that the propagation delay for faster transitions (like 200ns) is significantly lower than that for slower transitions (like 1µs), even though the output waveform is distorted and do not reach 5V for smaller capacitances (like 20pF).