A 10 Gb/s High-Speed Serial Data Link Design in 45nm CMOS

Songzhu Zhang 10616167

Abstract

An on-chip *voltage mode transmitter* (VM TX) with equalization allows high-speed data to be transferred reliably from one chip to another a chip via a non-ideal channel. An implementation with 2-tap equalization of such voltage-mode transmitter is provided in this report. To rerceive the data from the channel output, a *clocked comparator* (also called regenerative amplifier) was implemented. The entire link system including transmitter, channel, receiver and other peripherals (serializer, synchronizer, clock buffers, etc) is designed to work with 10Gb/s data rate at 70 °C.

Index Terms

high-speed link, 2-tap equalization, voltage-mode transmitter, low-power link, clocked comparator, regenerative amplifier.

I. Introduction

THIS report presents a design of a high-speed link system. Important justifications and simulation results of my design are provided in the following sections. The performance and specifications for the transmitter (TX) and the receiver are listed in Table I and II. The following assumptions are made in the design of the link:

- The link system has to be a source synchronous in which the TX and RX are clocked with the same high-frequency reference PLL.
- The reference PLL output has no duty cycle distortion (DCD).
- The link is designed to work with at least 2:1 serialization. On top of that, if 2:1 serialization is used, it can be assumed that $D_0, D_1, \overline{D}_0, and\ \overline{D}_1$ are provided and perfectly synchronized (i.e., ideal invereters with zero delay are used to provide $\overline{D}_0, and\ \overline{D}_1$).

II. TX Simulation Results

Note that some simulation results regarding the RX are also presented in this section for ease of comparison and will not be presented again in other sections.

TABLE I: VM TX Performance and Specifications

Name	Value
Eye Height Captured at TX Output (Input of the Channel) over 10,000 UIs	350.0 mV
Eye Width Captured at TX Output (Input of the Channel) over 10,000 UIs	174.03315 ps
Tuning Range of the TX Ouput Impedance	$[31.3281\Omega, 1.27898k\Omega]$
Power Supply V_{DD}	1 V
Data Rate	10 Gb/s
Average Power of Serializer	126.7 μW
Average Power of Pre-Driver	253.8 μW
Average Power of TX Driver	8.749 mW
Average Power of Clock Buffer for TX CLK	40.53 μW
Total Average Power Consumption at TX side	9.17003 mW
Energy/bit	0.917003 pJ/bit

$$Energy/bit\ for\ TX = \frac{\sum Average\ Total\ Power\ Consumption\ at\ TXside}{10G} = \frac{9.17003m}{10G}\ \frac{J}{bit} = 0.917003\ pJ/bit$$

A. *VM TX output impedance plot*

The VM TX is designed to have an ideal output impedance of 50Ω because the characteristic impedance of the channel is 50Ω and the TX uses source-series termination. Due to PVT variations, after the chip is fabricated, its output impedance may not be exactly what we expected. Thus, this VM TX has a tuning impedance range from 31.3281Ω to $1.27898k\Omega$.

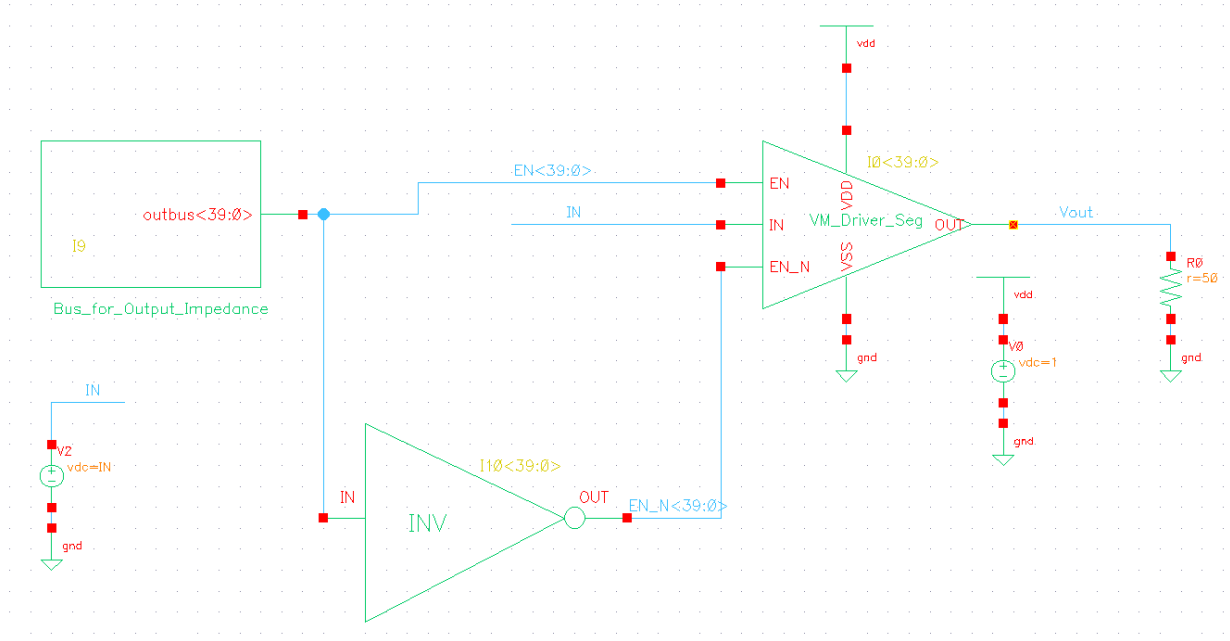


Fig. 1: Circuit Setup for Output Impedance Simulation

Let N be the total number of segments enabled (i.e., $EN = 1V$). To generate the output impedance plot, the waveform of V_o (see Fig. 2) is used in the following formula:

$$V_o = \frac{1}{R_o + 50} \cdot 50 \Rightarrow R_o = 50\left(\frac{1}{V_o} - 1\right)$$

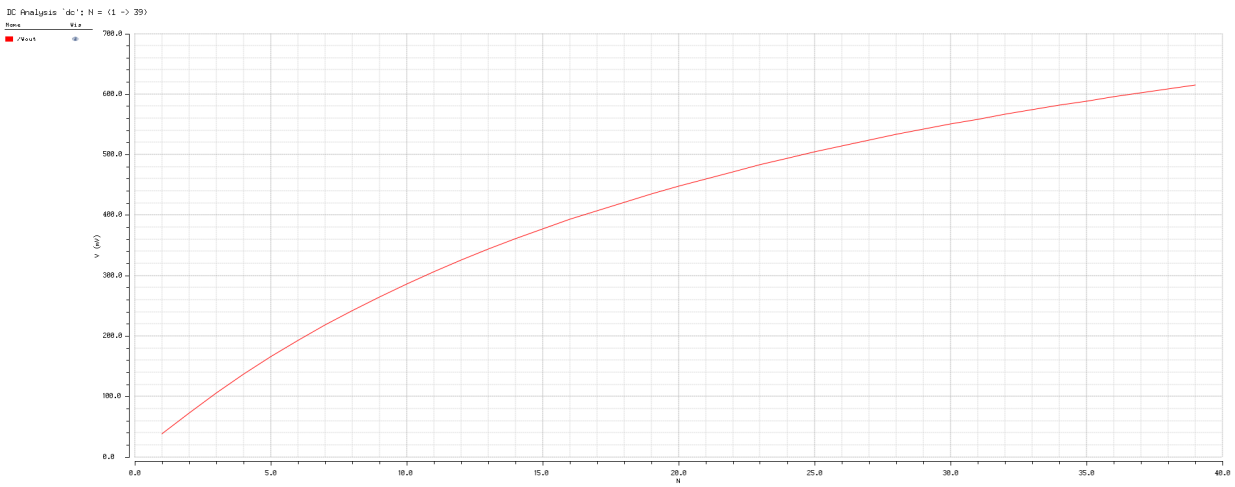


Fig. 2: Output Voltage of the $TXvsN$

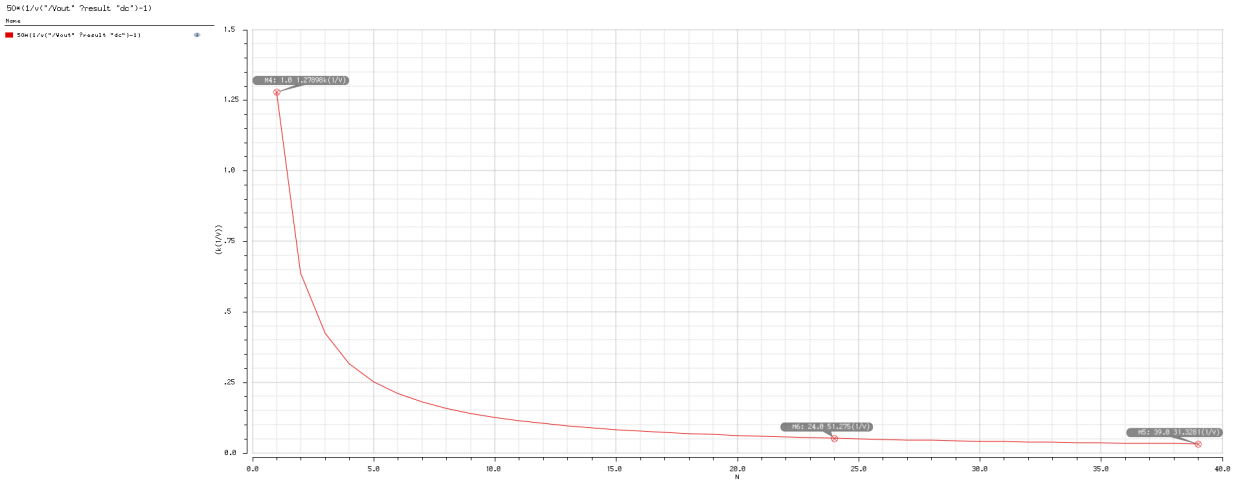


Fig. 3: Output Impedance of the VM TX with N=0,1,2,...,39

Note that when $N = K = 24$, the output impedance of the TX is 51.275Ω, which is close to the desired value of 50Ω.

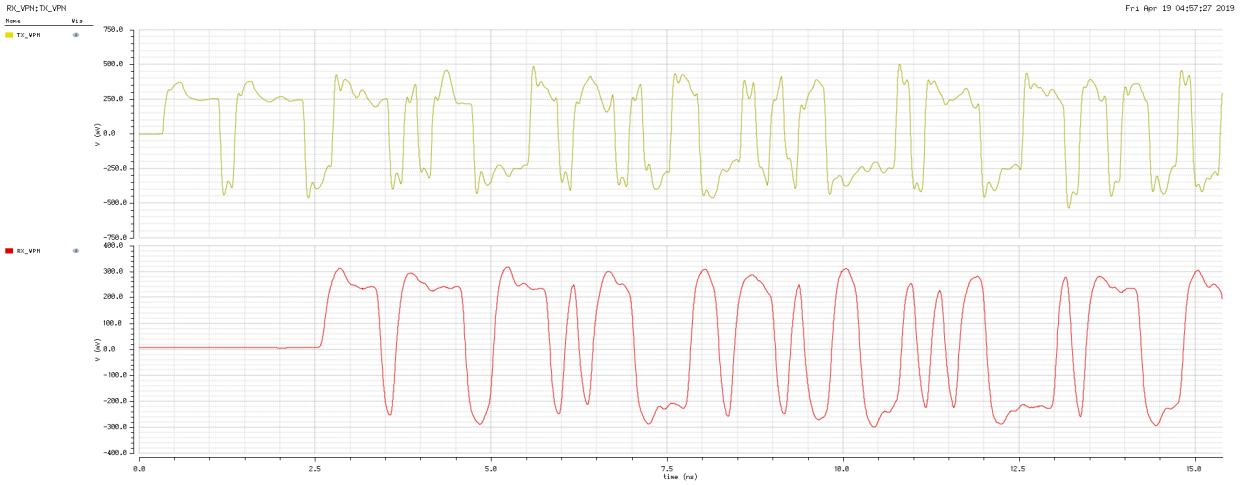


Fig. 4: Transient Response of the TX Output Voltage and RX Input Voltage

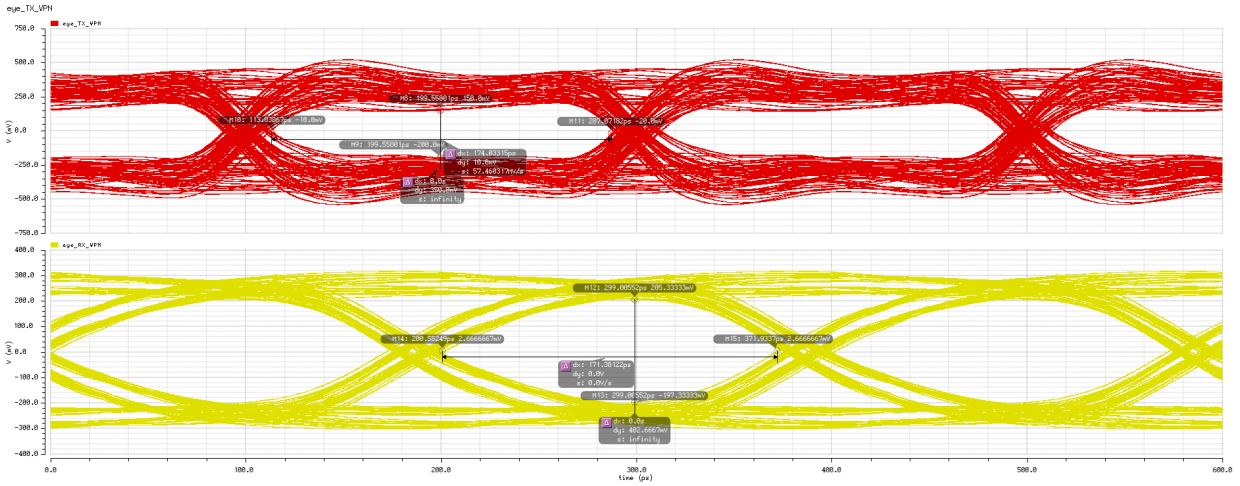


Fig. 5: Eye diagram Captured over 10,000 UIs at TX Output (Input Voltage of the Channel) and RX Input (Output Voltage of the Channel)

B. Worst Case Eye diagram Captured over 1000 UIs at TX Output V_{PN}

By observing the channel pulse response, the worst-case data pattern can be identified. From Fig. 6, the worst-case input data pattern can be generated by the following MATLAB code:

```
WC0=[1,1,0,1,0,1,0,0,1,0,1,1]; % the worst case 0 pattern
WC1=[0,0,1,0,1,0,1,1,0,1,0,0]; % the worst case 1 pattern
ZeroPad=[0,0,0,0,0,0,0,0,0,0,0,0];
OnePad=[1,1,1,1,1,1,1,1,1,1,1,1];
WC_Stream=[ZeroPad, WC1, ZeroPad, OnePad, WC0, OnePad, ZeroPad, WC1, ZeroPad,
           OnePad, WC0, OnePad, ZeroPad, WC1, ZeroPad, OnePad,
           WC0, OnePad, ZeroPad, WC1];
b = sprintf('%d', WC_Stream)

b =

    '000000000000001010110100000000000001111111111111010100101111111111111
    000000000000001010110100000000000001111111111111010100101111111111111
    000000000000001010110100000000000001111111111111010100
    111111111111100000000000001010110100'
```

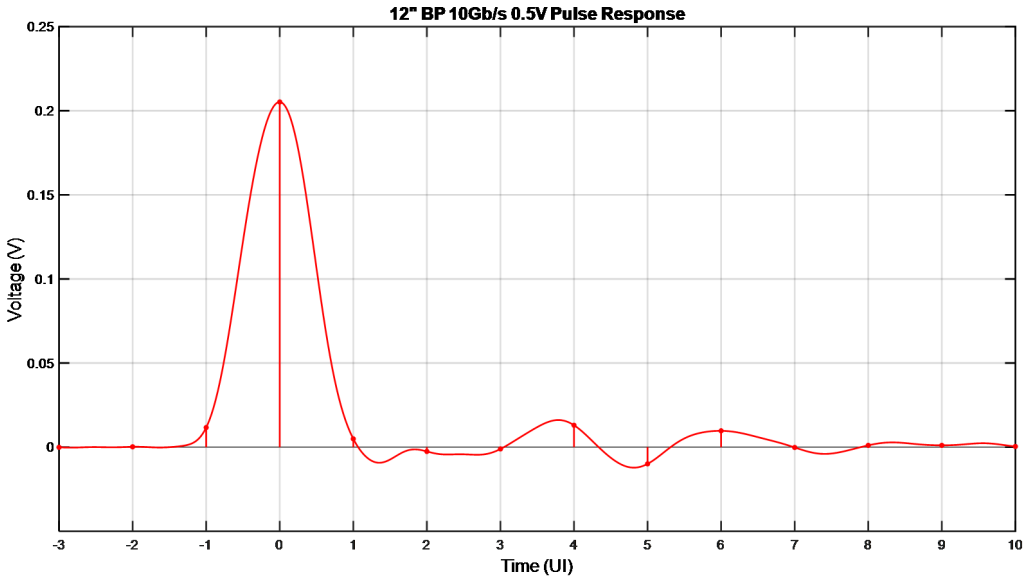


Fig. 6: Channel Pulse Response

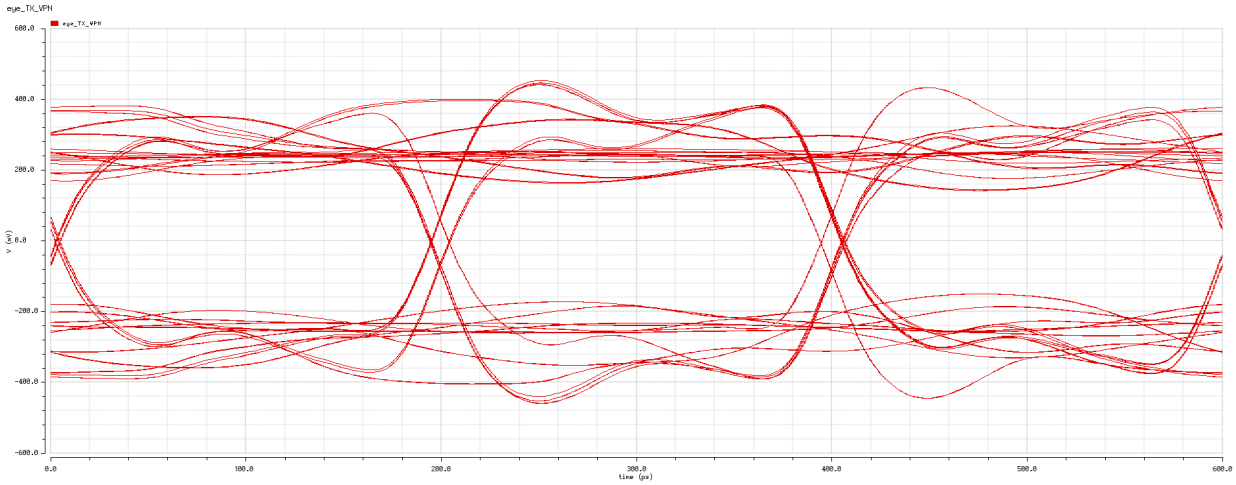


Fig. 7: Worst-Case Eye Diagram Captured at TX Output (Input of the Channel)

III. RX Simulation Results

TABLE II: RX Performance and Specifications

Name	Value
Eye Height Captured at RX Input (Output of the Channel) over 10,000 UIs	402.6667 mV
Eye Width Captured at RX Input (Output of the Channel) over 10,000 UIs	171.38122 ps
Power Supply V_{DD} Power Supply for V_{cm}^{RX}	1 V 924 mV
Data Rate	10 Gb/s
Average Power of Track and Hold Switch	2.852 μW
Average Power of Clocked Comparator	10.21 μW
Average Power of Synchronizer	45.80 μW
Total Average Power Consumption at RX side	71.924 μW
Energy/bit	0.0071924 pJ/bit

$$Energy/bit\ for\ RX = \frac{\sum Total\ Average\ Power\ Consumption\ at\ RX\ side}{10G} = \frac{71.924\mu}{10G}\ \frac{J}{bit} = 0.0071924\ pJ/bit$$

Fig. 8: Waveforms Captured over 3000 UIs Showing Data Source D0, Output of the Clocked Comparator that Samples D0, and Checker Output for Checking D0

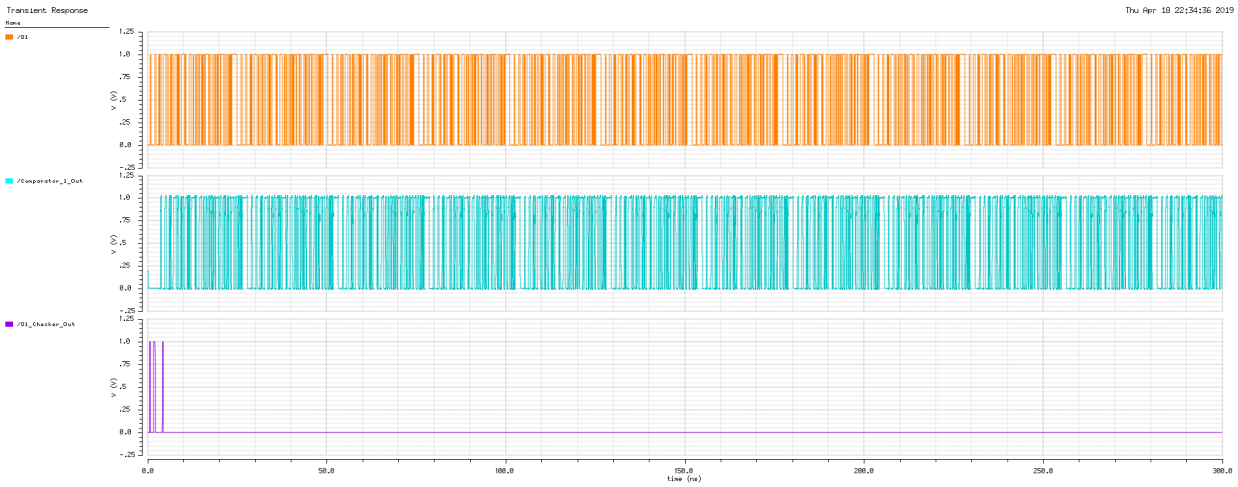


Fig. 9: Waveforms Captured over 3000 UIs Showing Data Source D1, Output of the Clocked Comparator that Samples D1, and Checker Output for Checkng D1

From Fig. 8, it can be observed that the checker outputs several pulses up to about 5ns (see Fig. 10). This may be interpreted as the initialization of the checker rather than bit errors.

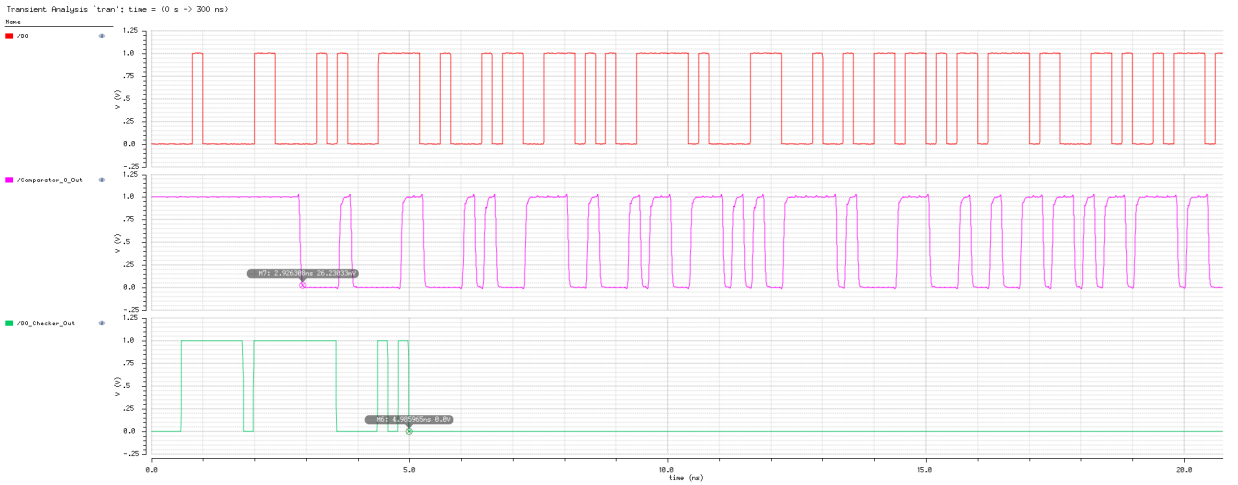


Fig. 10: Partial Enlargement of Fig. 8

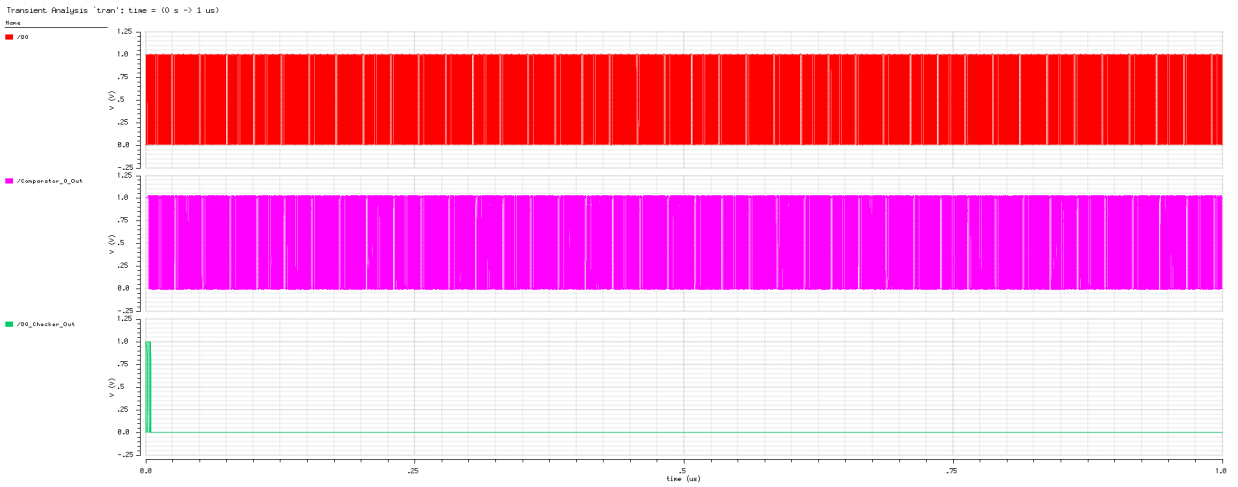


Fig. 11: Waveforms Showing Data Source D0, Output of the Clocked Comparator that Samples D0, and Checker Output for Checkng D0 Captured over 10,000 UIs

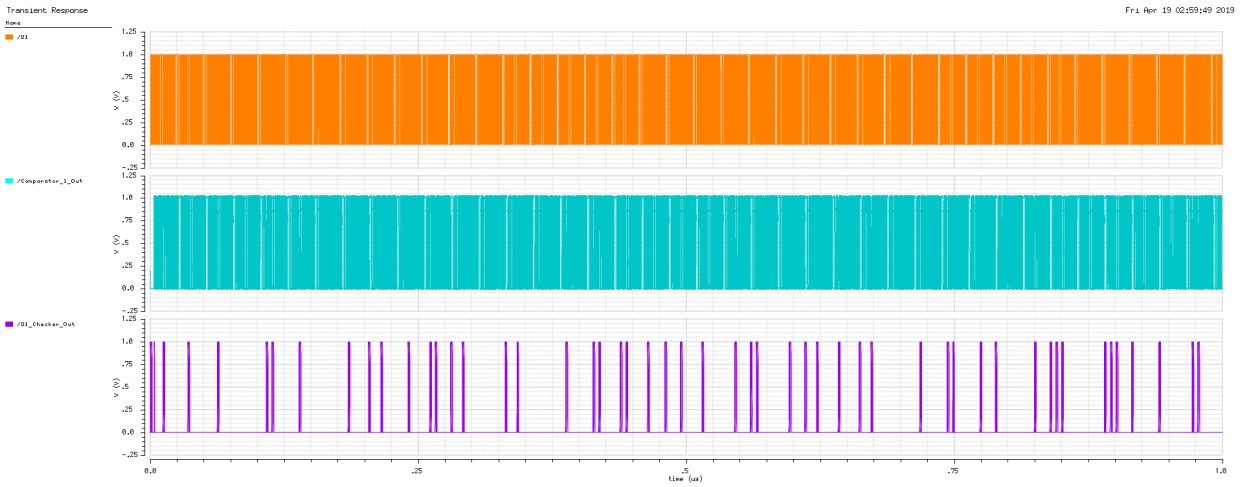


Fig. 12: Waveforms Showing Data Source D1, Output of the Clocked Comparator that Samples D1, and Checker Output for Checkng D1 Captured over 10,000 UIs

To capture the transient response over 10,000 UIs of the output of the two PRBS checkers without waiting for a very long time, I increased the rise/fall time for clock source and data sources. Because the components at the RX side have not been fully optimized yet, it is extremely sensitive to timing. As a result, when I increase the rise/delay time, the checker for D1 outputs pulses periodically.

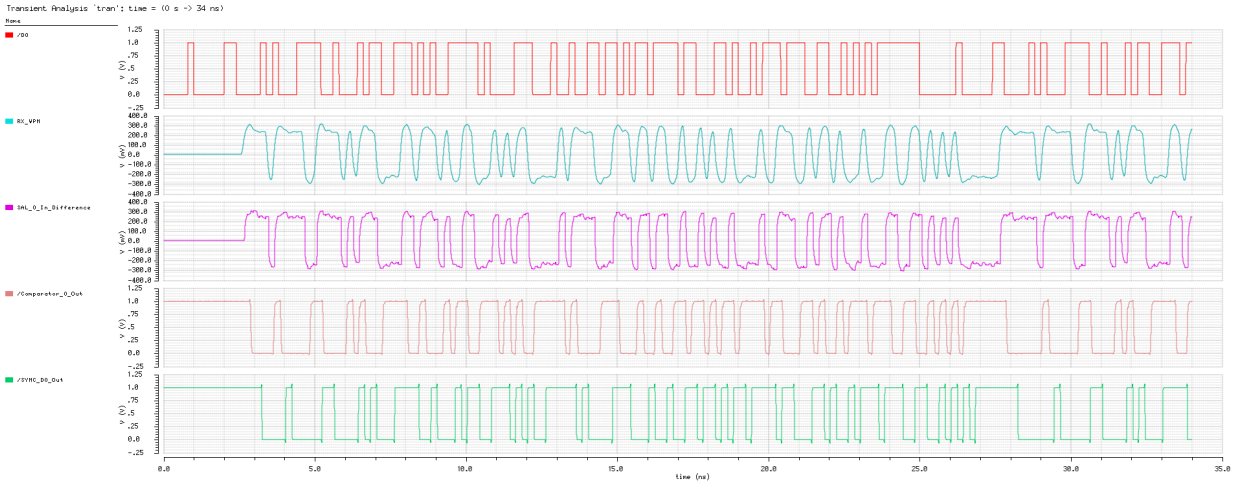


Fig. 13: Waveforms Showing the Flow of D0 Data Traveling from the Source to the Input of the Checker

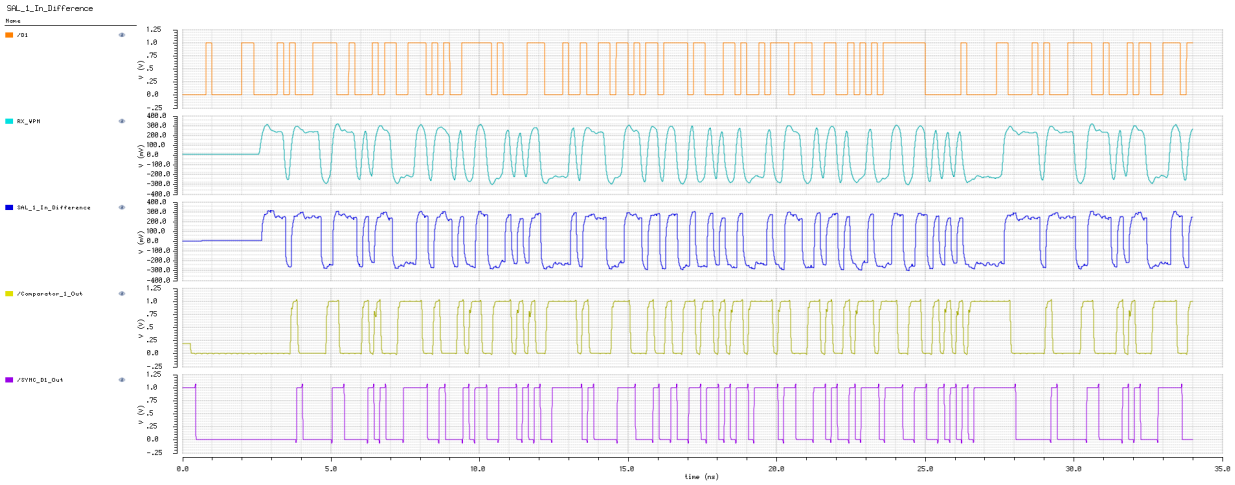


Fig. 14: Waveforms Showing the Flow of D1 Data Traveling from the Source to the Input of the Checker

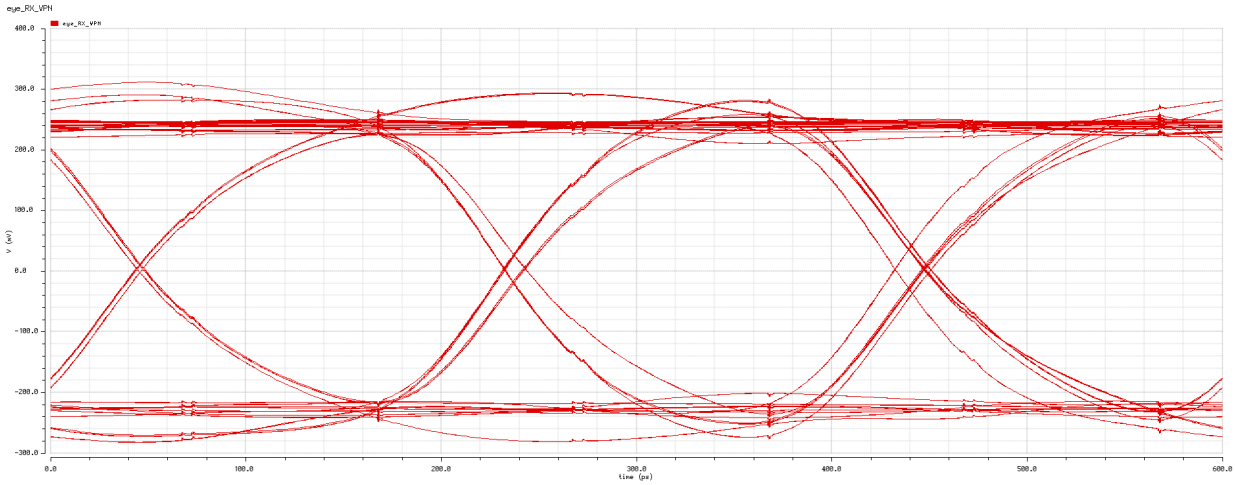


Fig. 15: Worst-Case Eye Diagram Captured at RX Input (Output of the Channel)

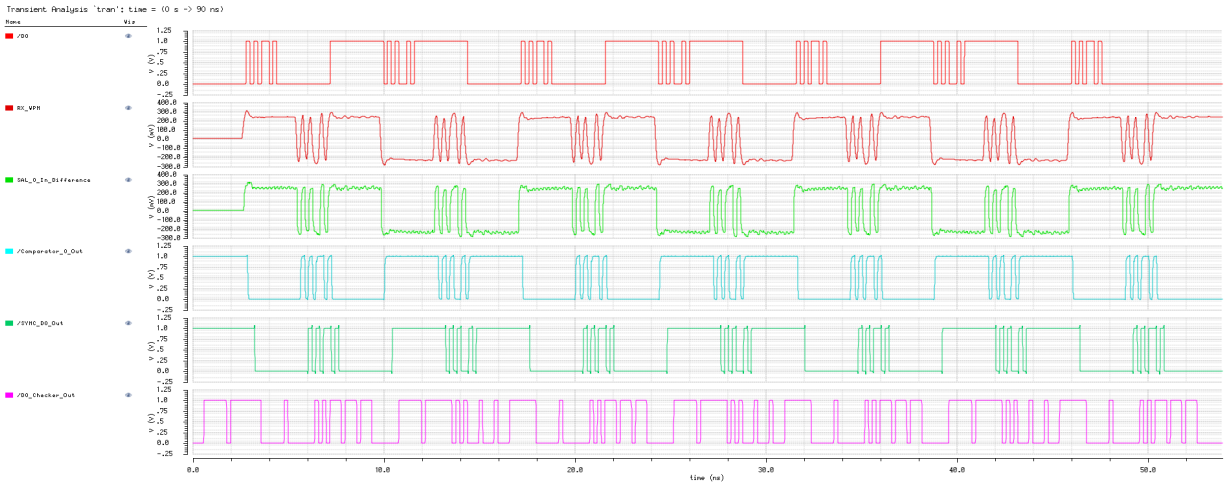


Fig. 16: Worst-Case D0 Data Flow Traveling from the Source to the Checker

Fig. 17: Worst-Case D1 Data Flow Traveling from the Source to the Checker

IV. Design Overview

The top level schematic for the entire link system is shown in Fig. 18. All the design parameters are presented in Fig. 19.

A. *Top-Level Schematic of the Link System*

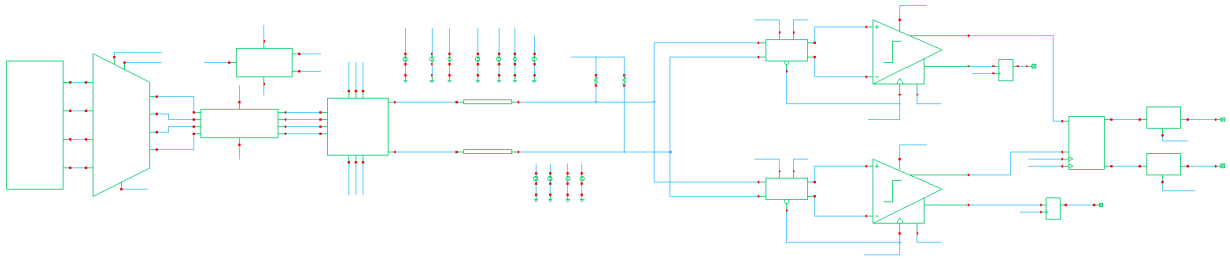
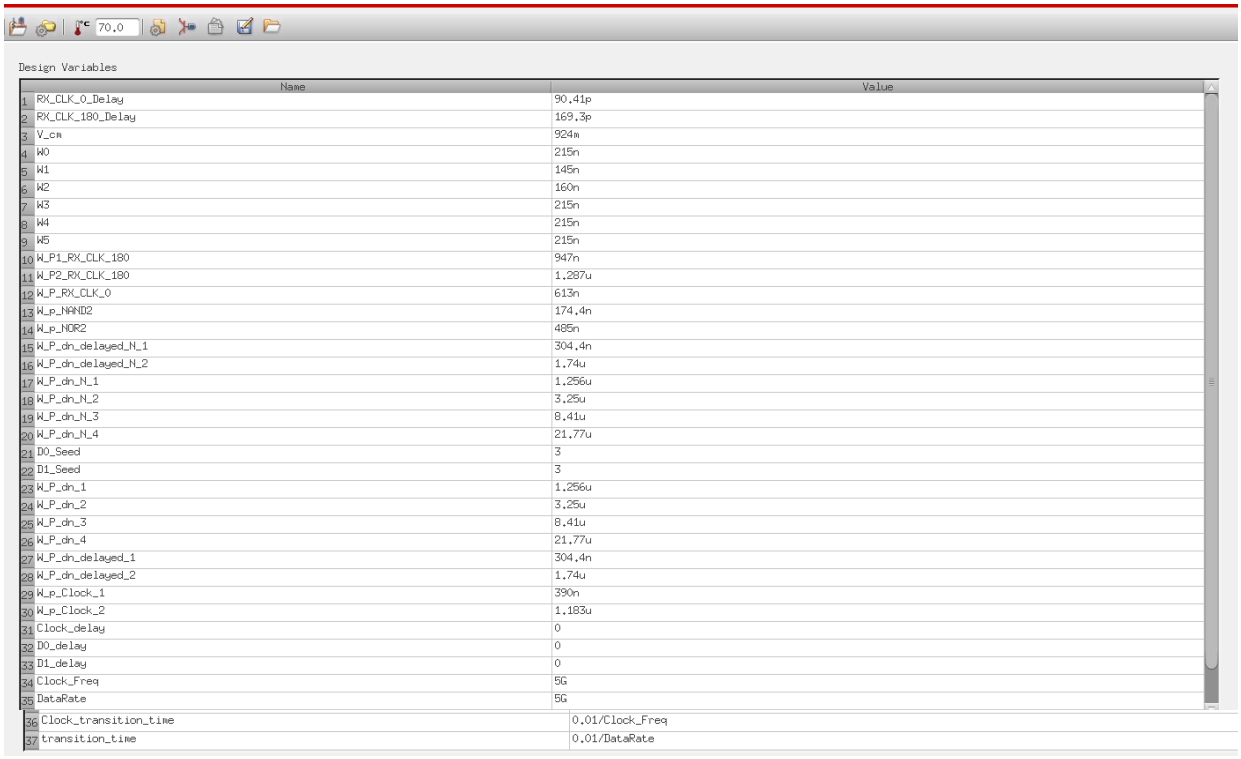


Fig. 18: TOP Module Schematic for the Entire Link System

It should be noted that, instead of using differential termination to absorb signal reflections on the RX side, single-ended termination was adopted because the strong arm latch inside the clocked comparator only works when the peak value of the two input signals V_P^{RX} and V_N^{RX} is around $1V$. By adding a common-mode voltage offset for the differential voltage at the input of RX, V_{PN}^{RX}, the peak value of the two input signals V_P^{RX} and V_N^{RX} can reach $1V$.



Design Variables

	Name	Value
1	RX_CLK_0_Delay	90.41p
2	RX_CLK_180_Delay	169.3p
3	V_cm	924m
4	W0	215n
5	W1	145n
6	W2	160n
7	W3	215n
8	W4	215n
9	W5	215n
10	W_P1_RX_CLK_180	947n
11	W_P2_RX_CLK_180	1.287u
12	W_P_RX_CLK_0	613n
13	W_p_NAND2	174.4n
14	W_p_NOR2	485n
15	W_P_dn_delayed_N_1	304.4n
16	W_P_dn_delayed_N_2	1.74u
17	W_P_dn_N_1	1.256u
18	W_P_dn_N_2	3.25u
19	W_P_dn_N_3	8.41u
20	W_P_dn_N_4	21.77u
21	D0_Seed	3
22	D1_Seed	3
23	W_P_dn_1	1.256u
24	W_P_dn_2	3.25u
25	W_P_dn_3	8.41u
26	W_P_dn_4	21.77u
27	W_P_dn_delayed_1	304.4n
28	W_P_dn_delayed_2	1.74u
29	W_p_Clock_1	390n
30	W_p_Clock_2	1.183u
31	Clock_delay	0
32	D0_delay	0
33	D1_delay	0
34	Clock_Freq	5G
35	DataRate	5G
36	Clock_transition_time	0.01/Clock_Freq
37	transition_time	0.01/DataRate

Fig. 19: Design Parameters for the Entire Link System

B. TX Implementation

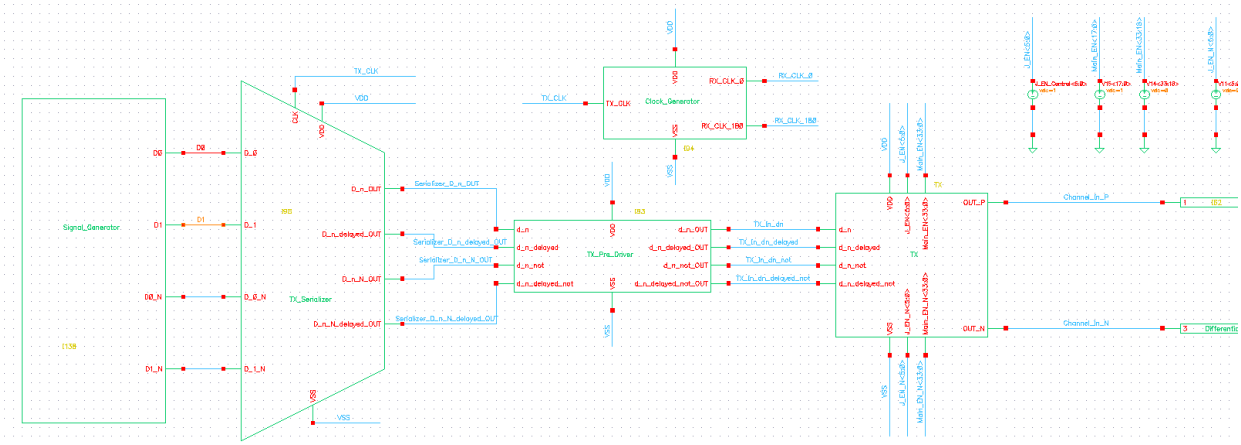


Fig. 20: Local Top-Level Schematic for the TX End

1) Justification for Some Important Design Parameters: To design a 2-tap FIR TX, the first thing is to determine the value of α, based on which other design parameters can be determined. MATLAB results show that the tap coefficients $w[0] = 0.751$, and $w[1] = -0.249$.

$$\left.\begin{aligned} y[n] &= (1-\alpha)\,d[n] - \alpha d[n-1] \\ y[n] = w[n] * d[n] &= w[0]\,d[n] + w[1]\,d[n-1] \\ w[0] &= 0.751 \\ w[1] &= -0.249 \end{aligned}\right\} \Rightarrow \left\{\begin{aligned} w[0] &= 1-\alpha = 0.751 \\ w[1] &= -\alpha = -0.249 \Rightarrow \alpha = 0.249 \approx 0.25 \end{aligned}\right.$$

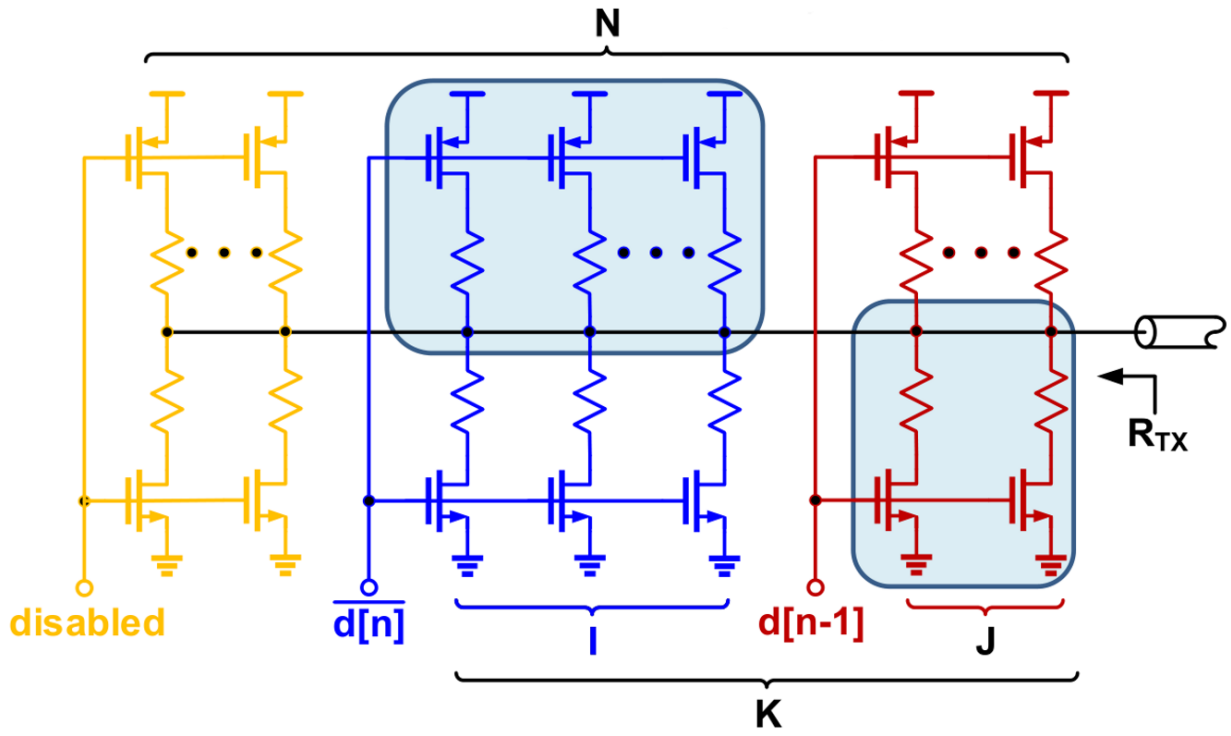


Fig. 21: Two-Tap De-Emphasis Driver [1]

$$\left.\begin{array}{c} R_U = \frac{NR}{I} \\ R_D = \frac{NR}{J} \\ K = I + J \\ \frac{R_D - R_U}{R_D + R_U} = 1 - 2\alpha \end{array}\right\} \Rightarrow \left\{\begin{array}{c} I = K - K\alpha = 24 - 24 \cdot \frac{1}{4} = 18 \\ J = K\alpha = 24 \cdot \frac{1}{4} = 6 \end{array}\right.$$

Since $\alpha \approx 0.25 = 1/4$, K should be an integer multiple of 4 given that $J = K\alpha$ and J is an integer. Thereby, a reasonable guess of K value would be 24. To find the optimum values of N, K, I, J, and R in Fig. 21 such that the VM TX has a tunable range that contains the tuning range of $\pm 30\%$ from the optimum value 50Ω, the following set of inequality equations needs to be solved.

$$\left.\begin{array}{c} \frac{NR}{K} = 50 \\ 0 < R \leqslant 35 \\ NR \geqslant 65 \\ N > 0 \\ K = 24 \\ NR \geqslant 600 \ \ (an\ initial\ guess) \end{array}\right\} \Rightarrow \left\{\begin{array}{l} K_{\min} = 24 \\ N_{\min} = 40 \\ R_{\max} = 30 \end{array}\right.$$

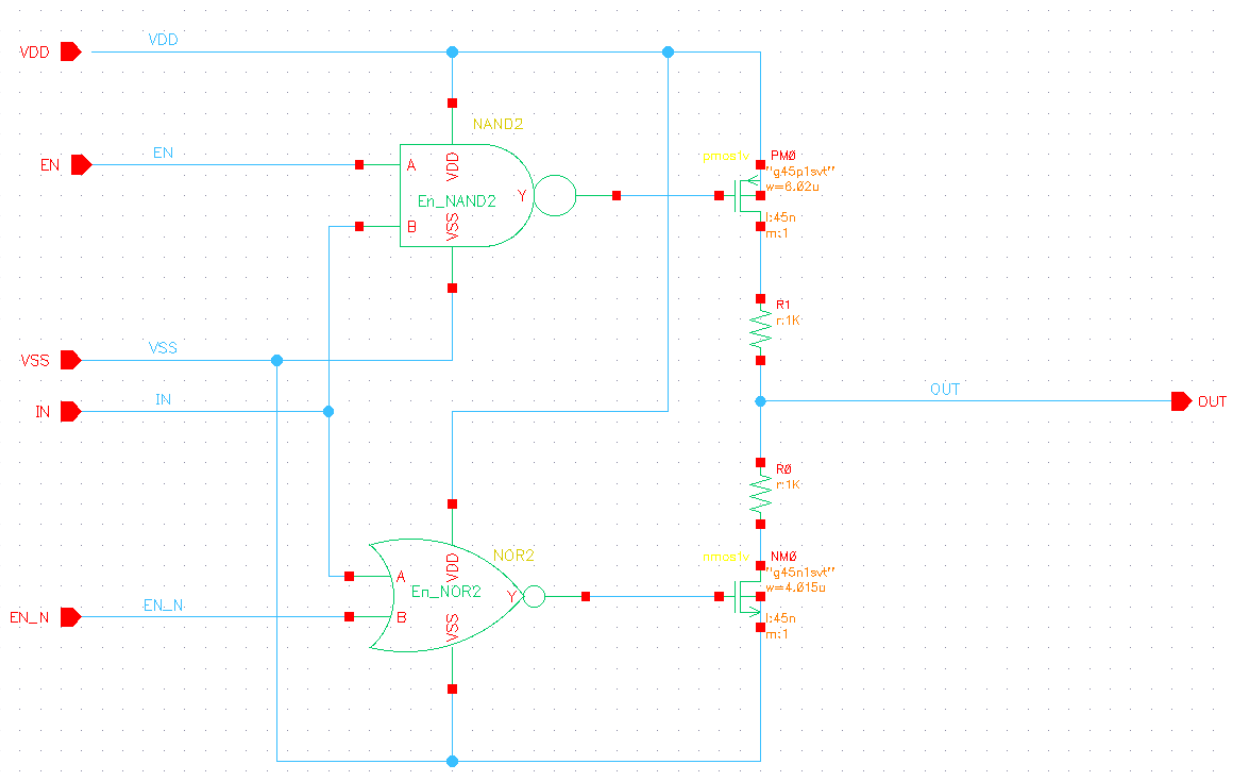


Fig. 22: Schematic of Each Individual Segment of the VM TX

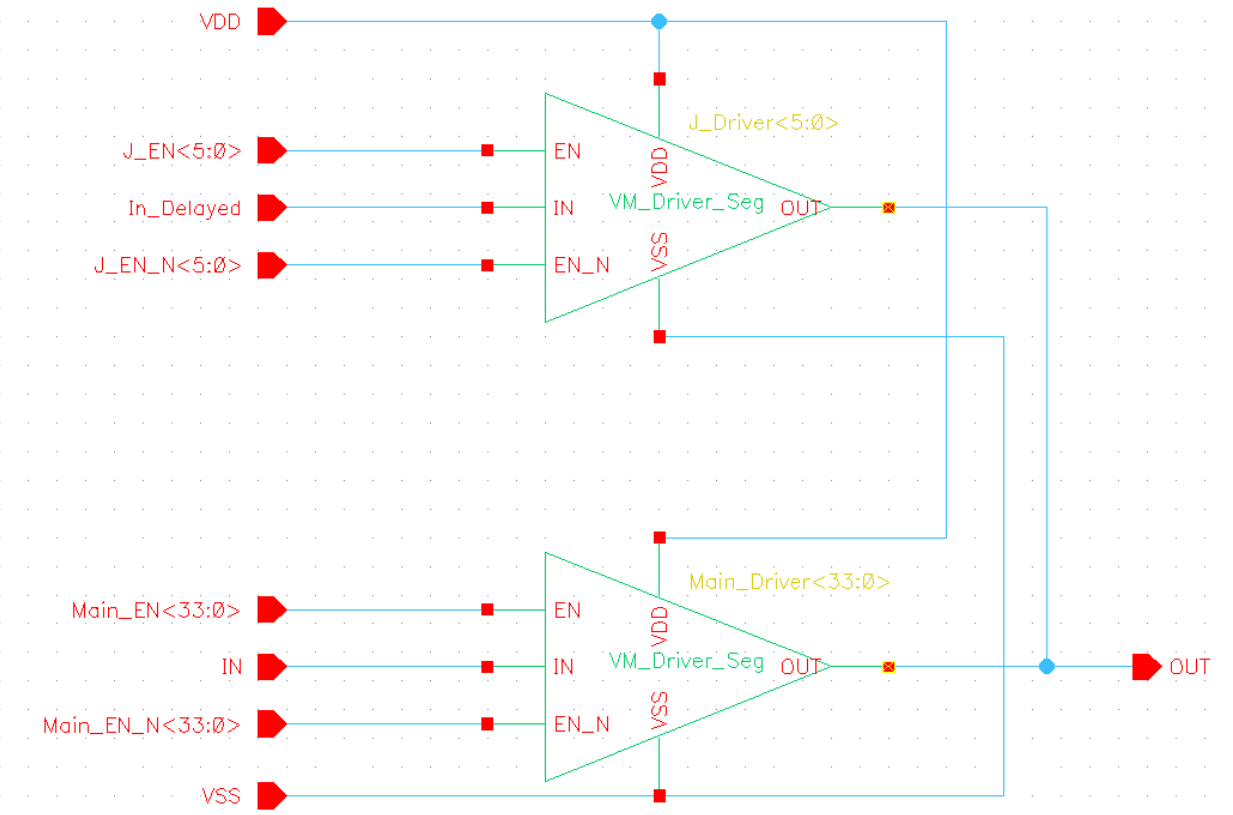


Fig. 23: VM Driver Schematic

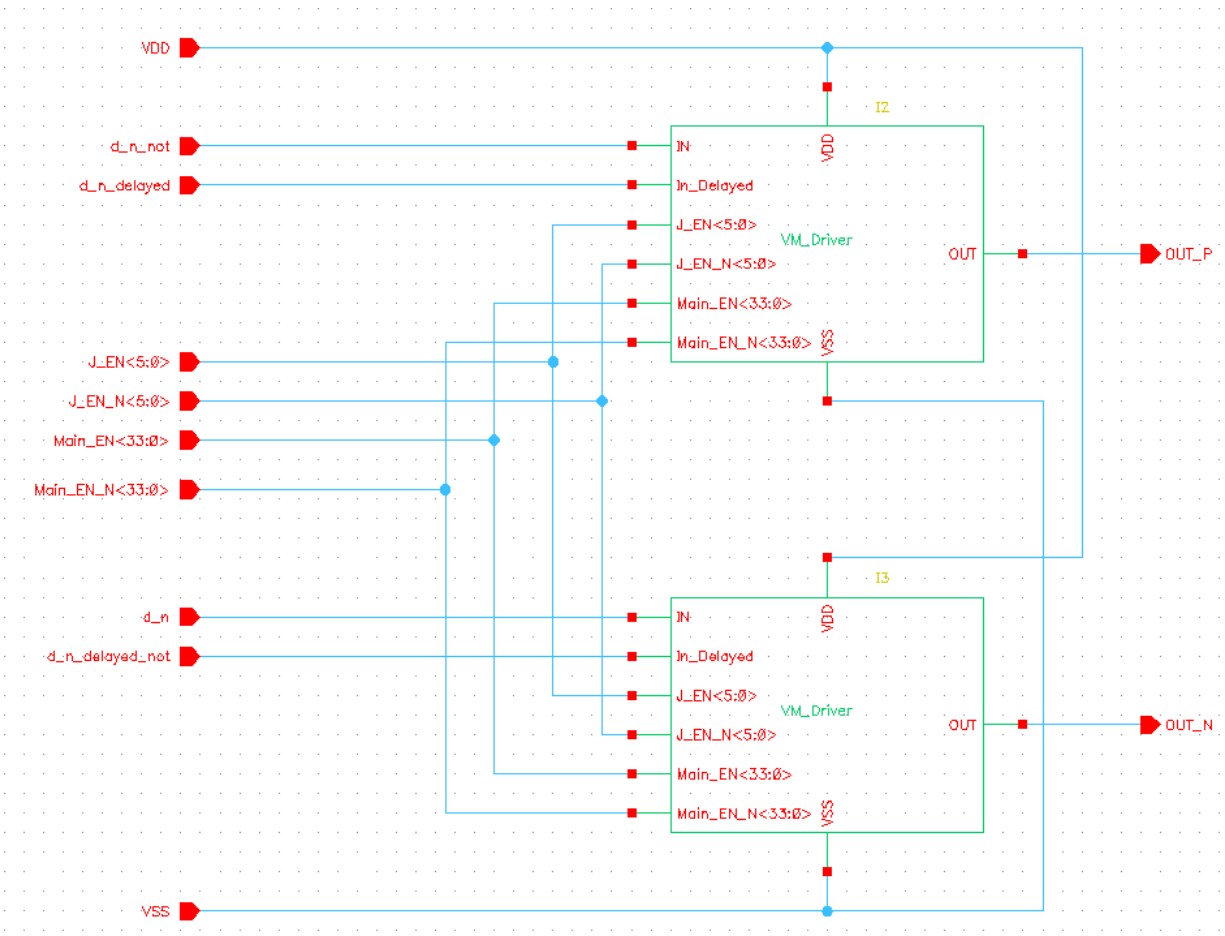


Fig. 24: Transmitter Schematic

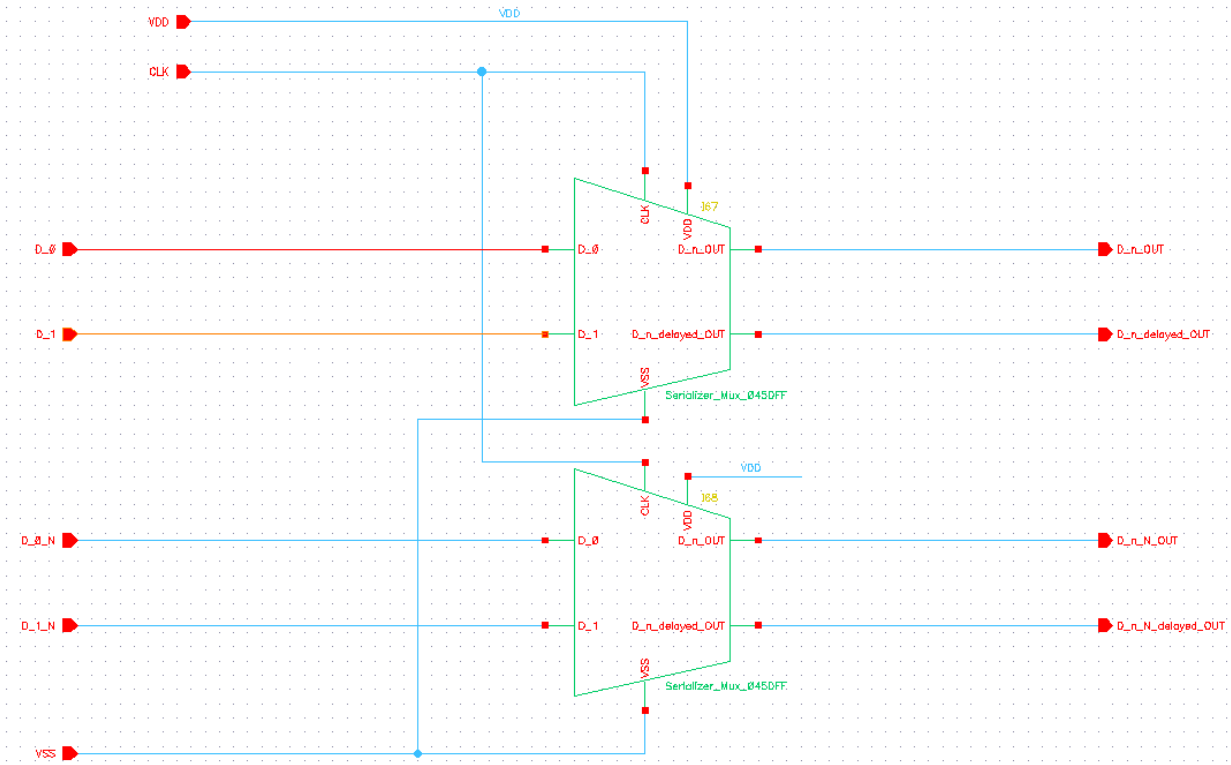


Fig. 25: Serializer Schematic

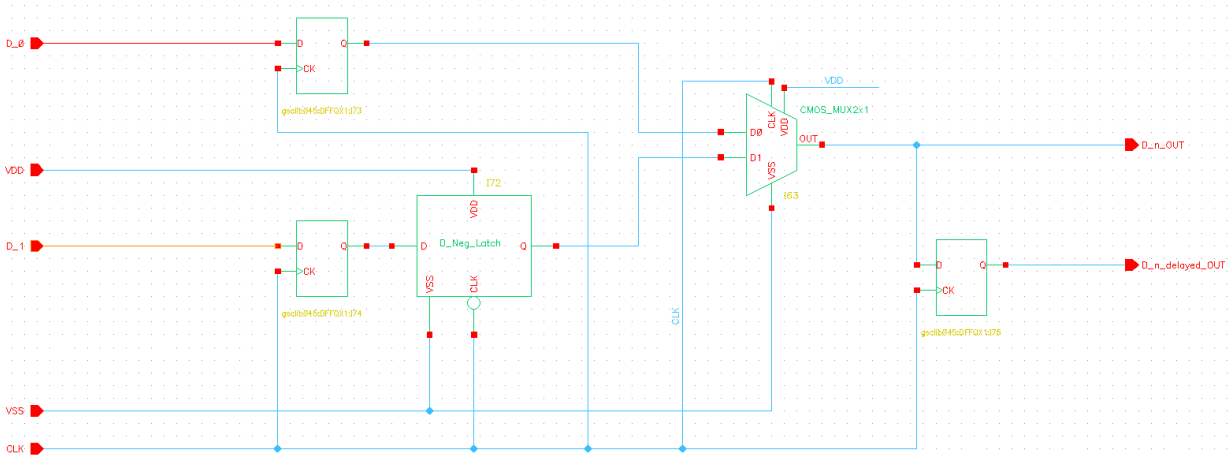


Fig. 26: Serializer Mux 2:1

2) 2:1 Mux with Two D Flip-Flops and One High-Speed Negative D Latch: The two D flip-flops are used to synchronize the two input data sequences because the two data sequences coming from preceding stages may not be perfectly aligned with the transmitter clock. Thus, the two D flip-flops are used to re-time the two input data sequences to prevent output glitches and/or pulse-width distortion. The negative D latch is used as a $\frac{T_{clock}}{2}$ offset latch; it prevents the two input data from transitioning simultaneously.

3) High-Speed NegativeD Latch Implementation: In Fig. 26, there is a negative D latch implemented using domino logic. Fig. 27 shows a popular design of a negative high-speed D latch using domino logic [3].

The reason for using domino logic in designing the latch is that it has smaller power consumption, less number of transistors, and propagation delay compared to static CMOS logic and dual rail domino logic [4]. In addition, since the clock frequency is quite high (5GHz), a high-speed negative D latch is preferable.

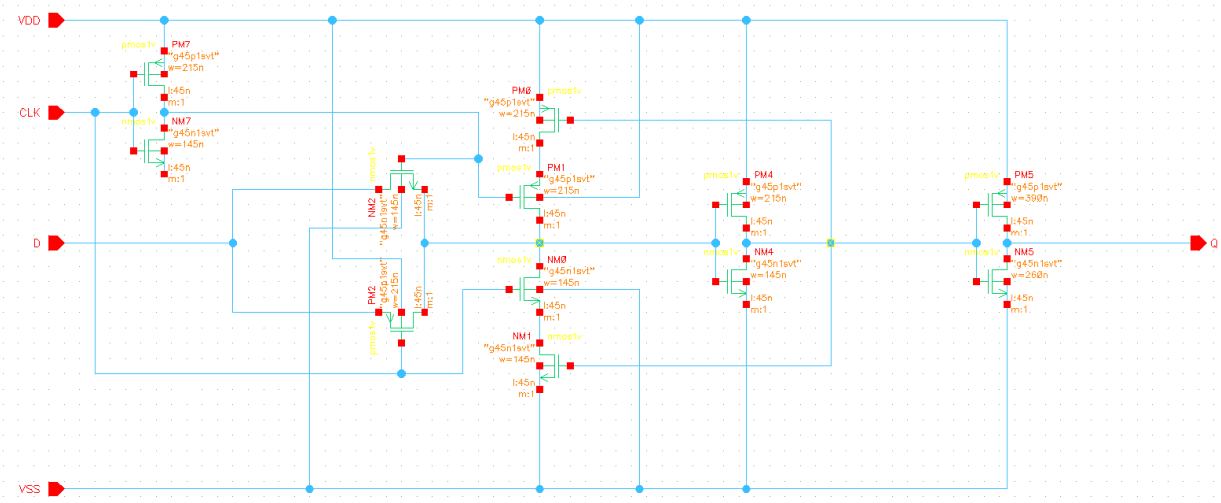


Fig. 27: High-Speed Negative D Latch Implementation

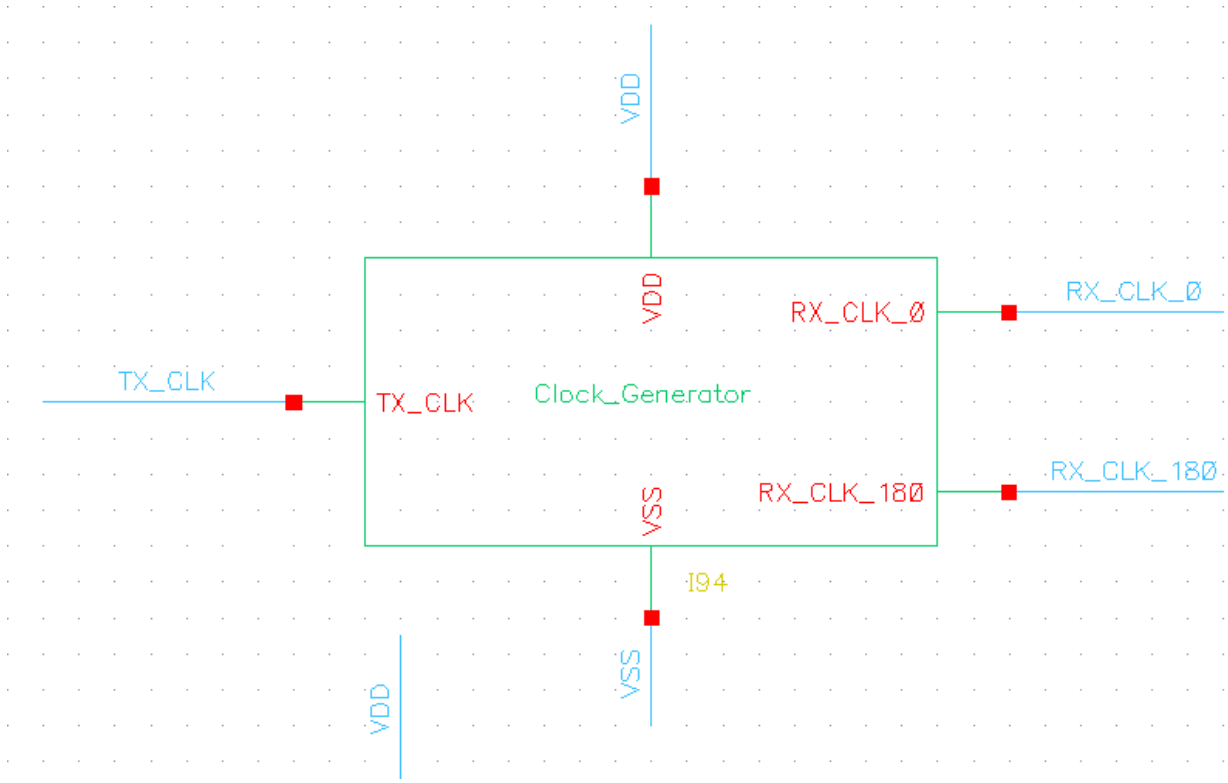


Fig. 28: Clock Generator Symbol

4) Clock Distribution:

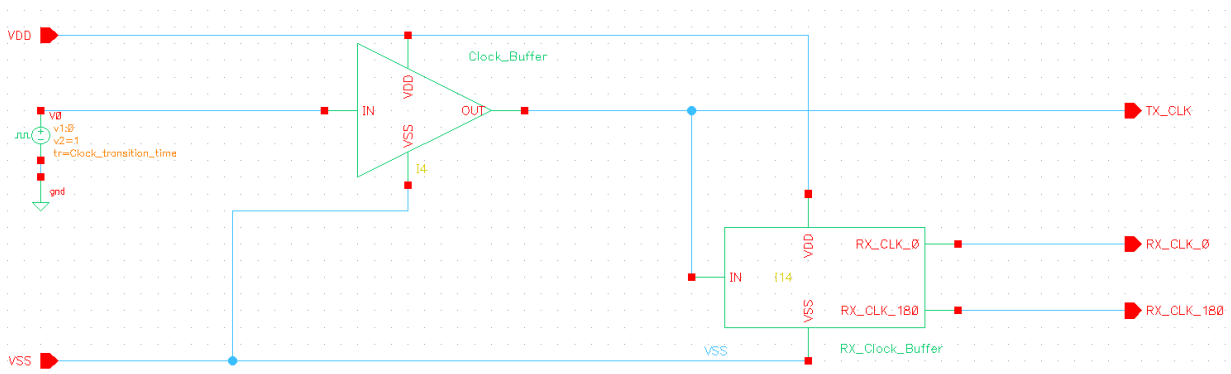


Fig. 29: Clock Generator Schematic

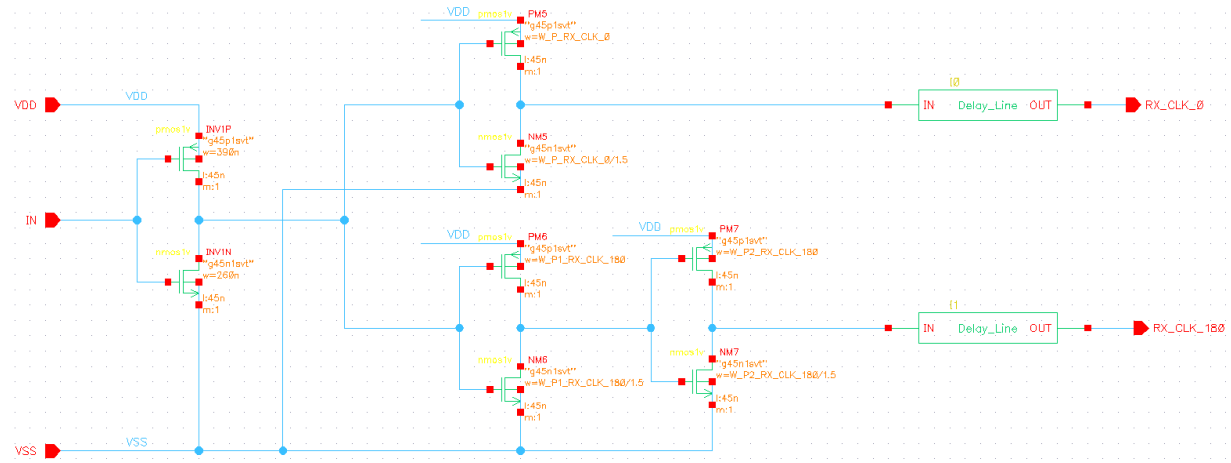


Fig. 30: RX Clock Buffer Schematic

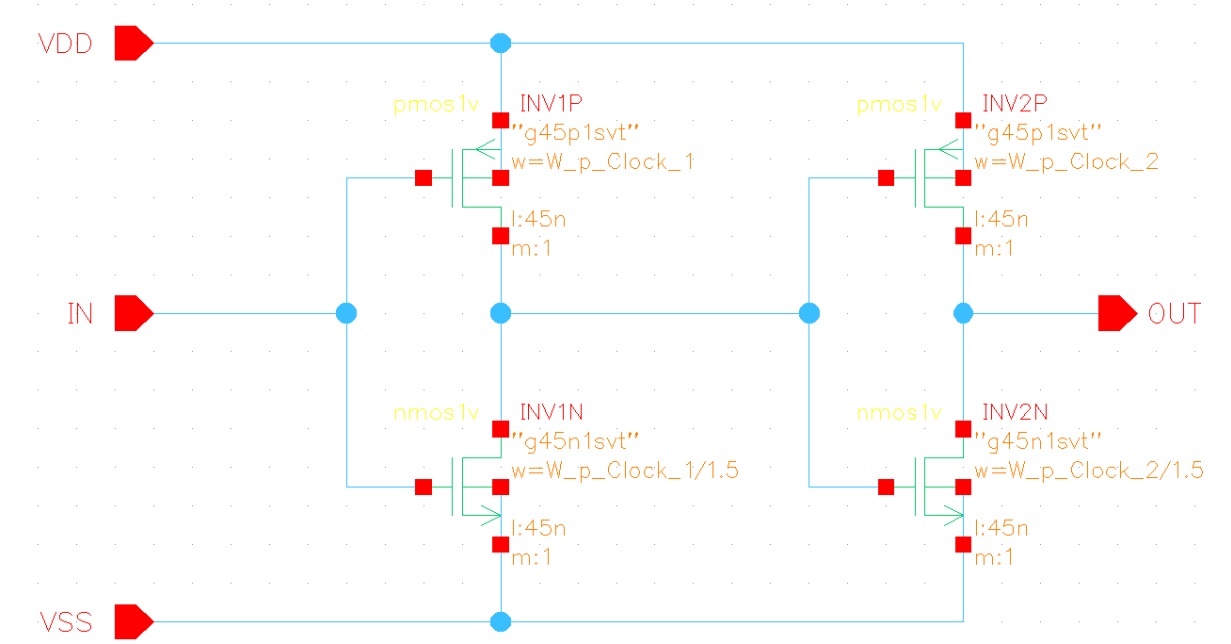


Fig. 31: Clock Buffer for TX CLK

5) Clock Distribution: One of the most critical parts of VLSI systems is clock distribution. There are two clock buffers inside the cell *Clock Generator* (see Fig. 28). The one with name *Clock Buffer* is used to drive all the D flip-flops and D

latches inside the serializer. It is optimized to reduce the delay through the use of logical effort. The other clock buffer with name *RX Clock Buffer* is used to drive the clocked comparator, D flip-flops, the PRBS checkers, etc. Inside the cell *RX Clock Buffer* (see Fig. 30), there is an inverter at the leftmost side, which drives two inverter chains with output name RX CLK 0 and RX CLK 180 respectively. The purpose of the leftmost inverter is to isolate the TX clock buffer chain from the two RX clock buffers so that when the sizes of RX clock buffers change, we do not need to resize the clock buffer for the TX side. For academic integrity, I need to point out that the two Delay Line cells are model written by VerilogA, which means they are ideal delay lines.

The eye diagram captured at Channel Output (RX input) has eye height around 400mV, which is quite small. There are several solutions to receive and recover the data coming from the channel. The first one is to design an excellent clocked comparator that works even the input voltage difference is around 250mV (assuming the sampling clock is not at the position where the eye height is the biggest); designing such an excellent clocked comparator takes time and the final report is due several hours later. The second solution is to use an active CTLE; it can increase the eye height and make the design of the clocked comparator easier, which also takes some time to do. The third option is to design a clock buffer that serves as a delay line to skew the rising edge of the clock signal RX CLK 0 right at the position where the eye height reaches its maximum value; this solution sounds doable and relatively easy, but in practice, it is not. Using logical effort method to size an inverter chain that serves as the delay line is not practical because this method cannot produce very accurate results when it comes to introducing a specific delay rather than minimizing the delay along a path. On top of that, if I use this method, then whenever I make changes to the RX side, I need to resize the inverters in the inverter chain (delay line). Considering that the RX side is very sensitive to the sampling clock, even 3ps off from the best position will lead to the failure of the RX, this method should not be adopted. If the report was due several days later, I would use an active CTLE at the RX side and refine the clocked comparator to avoid the use of the ideal delay line.

6) Pre-Driver: In my design, the following design parameters are used.

$$
\begin{aligned}
&N = 40 \\
&K = 24 \\
&J = K\alpha = 6 \\
&I = K - K\alpha = 18 \; where \; \alpha \approx 0.25
\end{aligned}
$$

Since $(N - J) > J$, the load capacitance of the pre-driver that drives $d\,[n]$ or its complement will not be the same as the other one which drives $d\,[n-1]$ or its complement. To equalize and minimize the path delay of both pre-drivers (i.e., inverter chains), different number of inverters are used in designing the pre-drivers. These two pre-drivers were then used to make a symbol called *TX Pre Driver* (see Fig. 32).

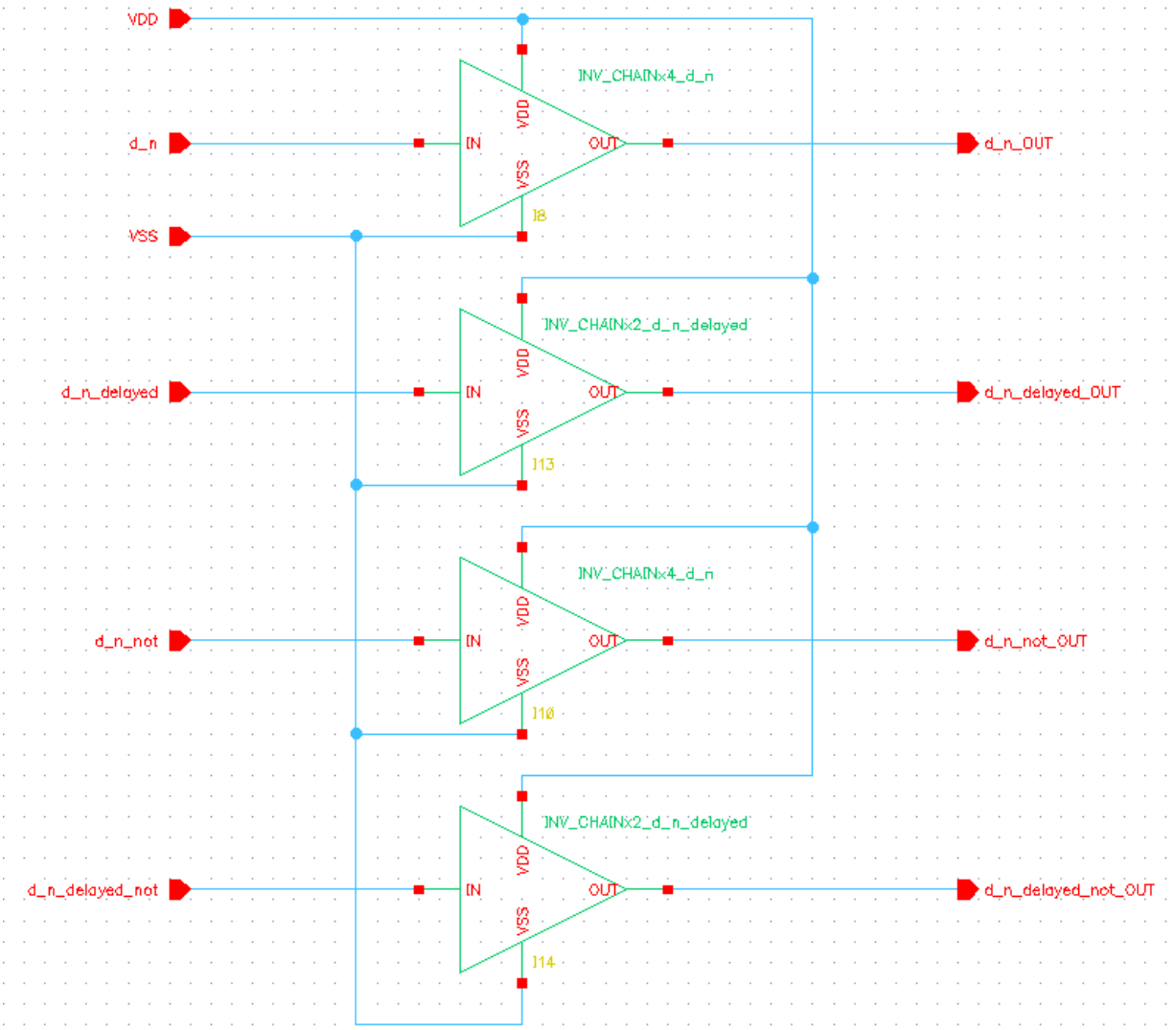


Fig. 32: TX Pre-Driver

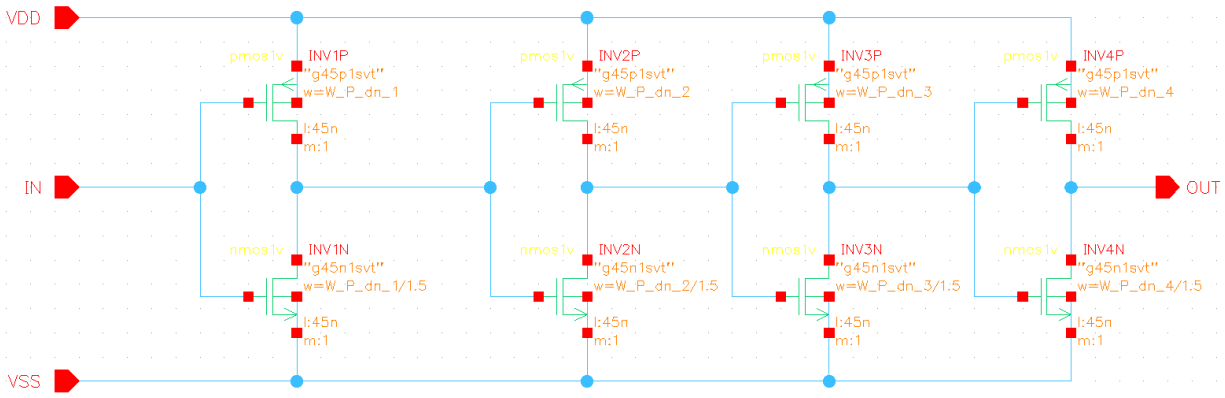


Fig. 33: Buffer for $d\,[n]$

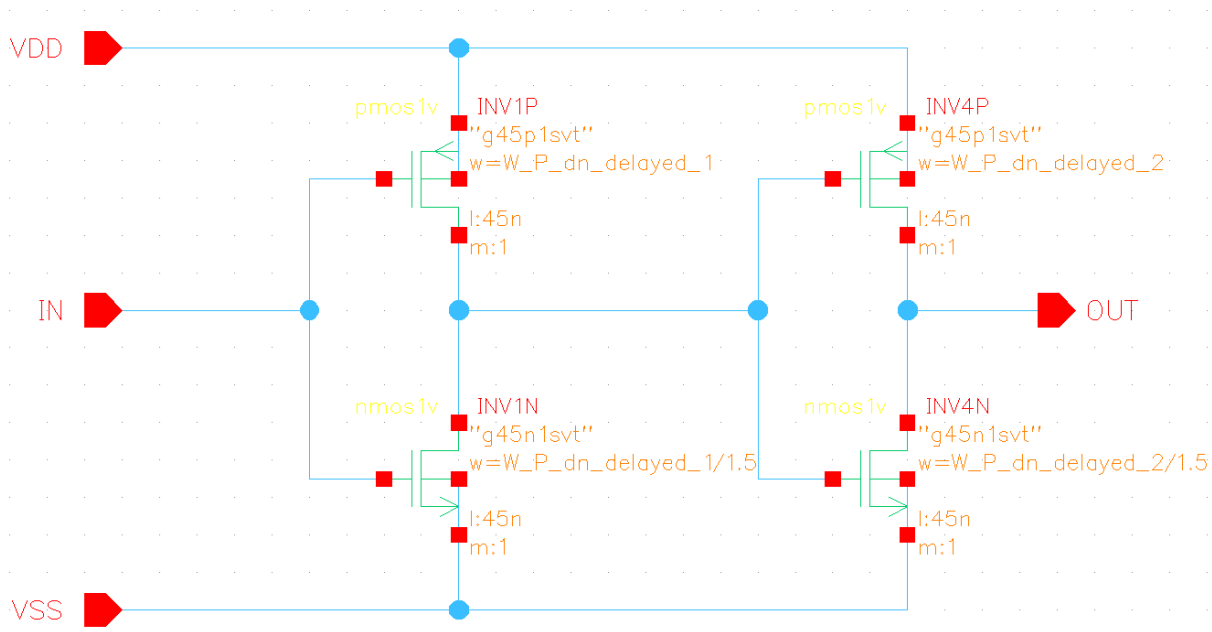


Fig. 34: Buffer for $d\,[n-1]$

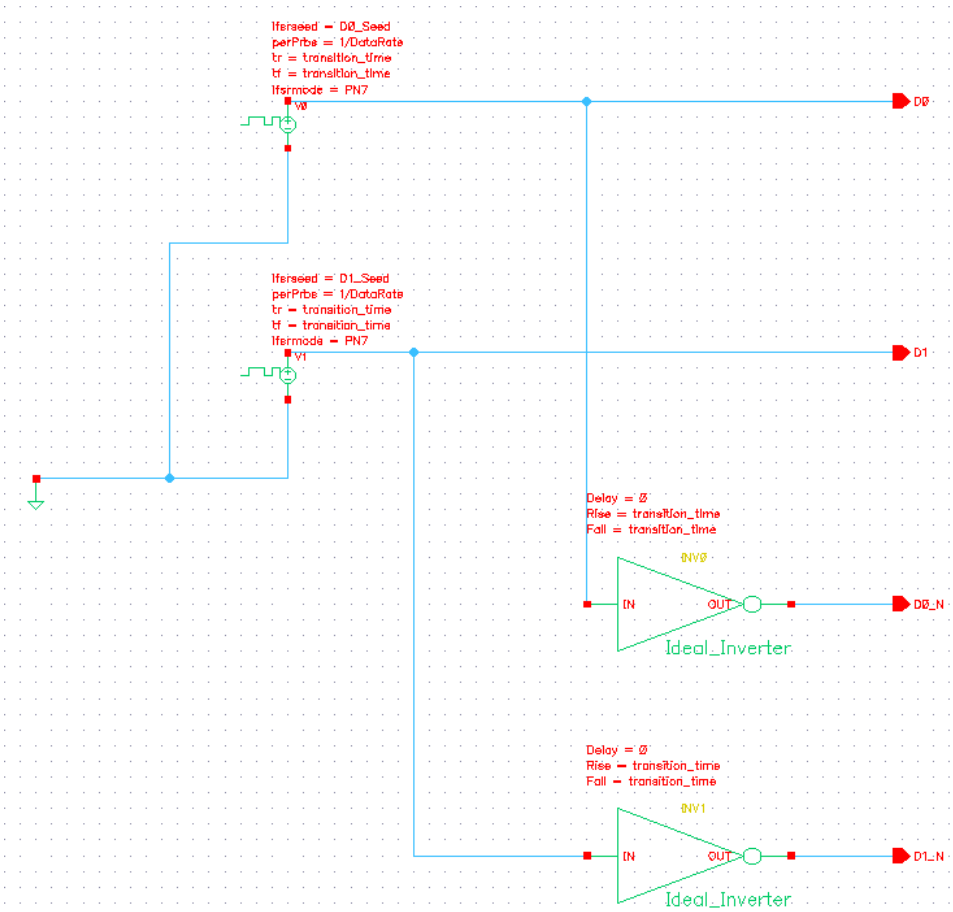


Fig. 35: Signal Generator Schematic

C. RX Implementation

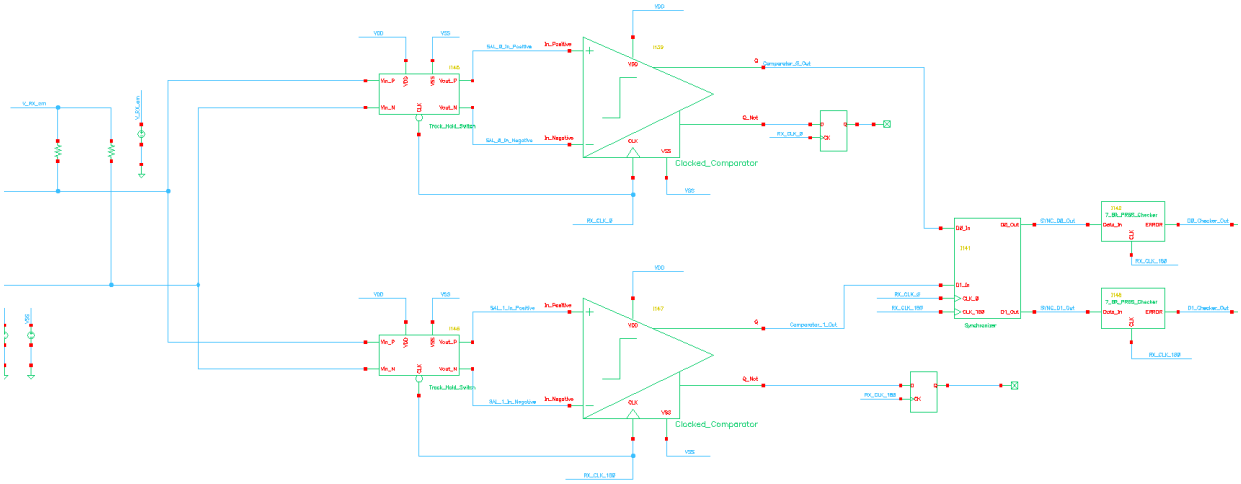


Fig. 36: Local Top-Level Schematic for the RX End

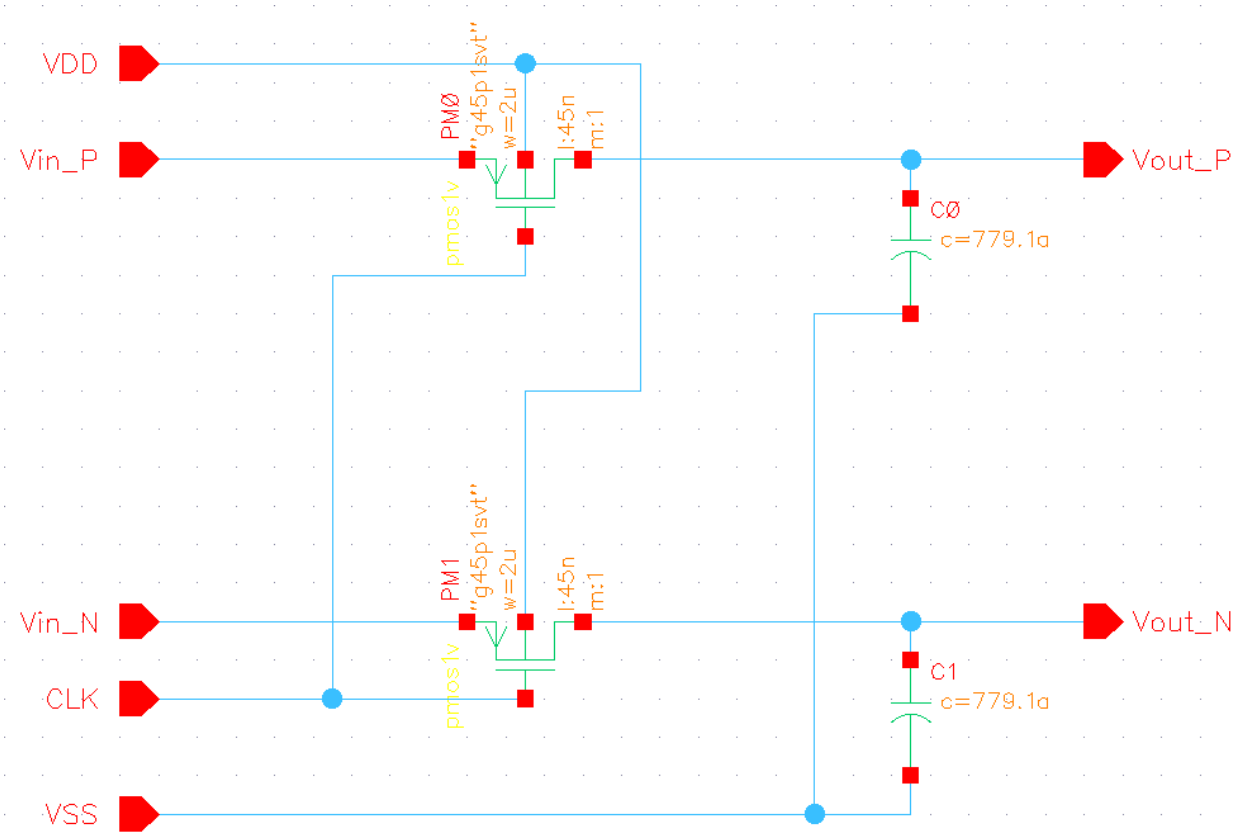


Fig. 37: Track and Hold Switch

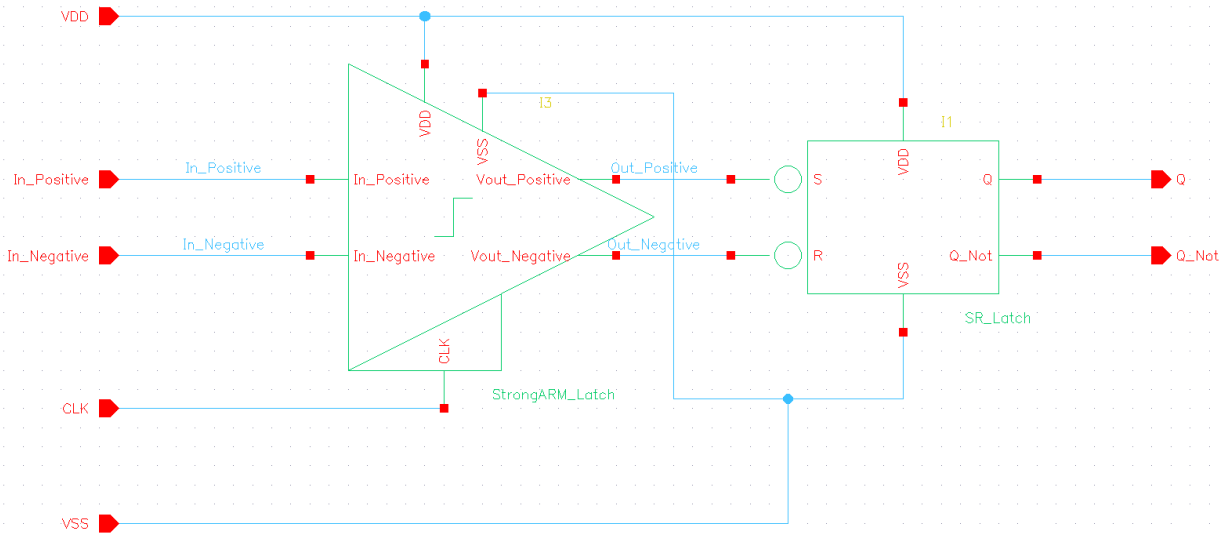


Fig. 38: Clocked Comparator

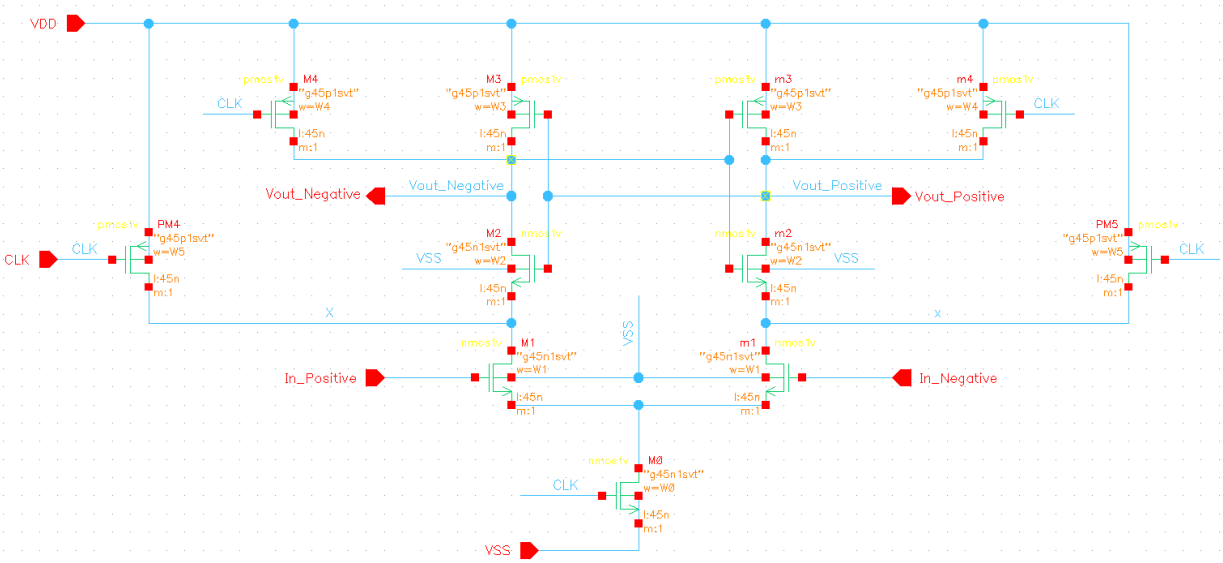


Fig. 39: StrongARM Latch

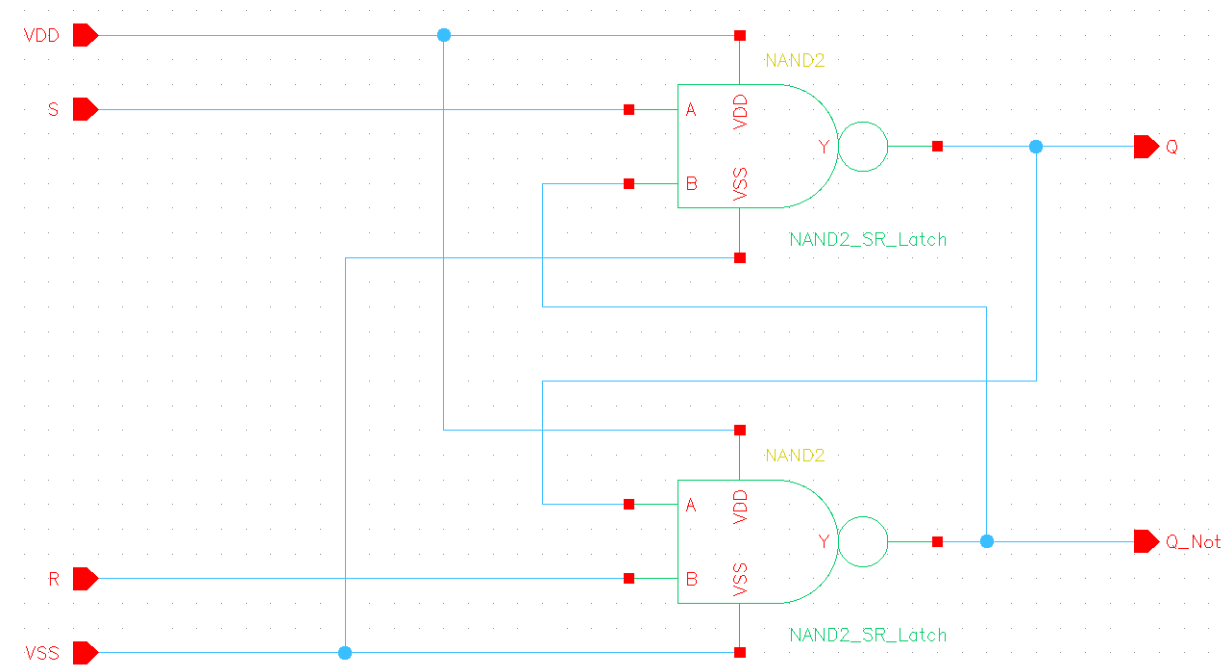


Fig. 40: SR Latch

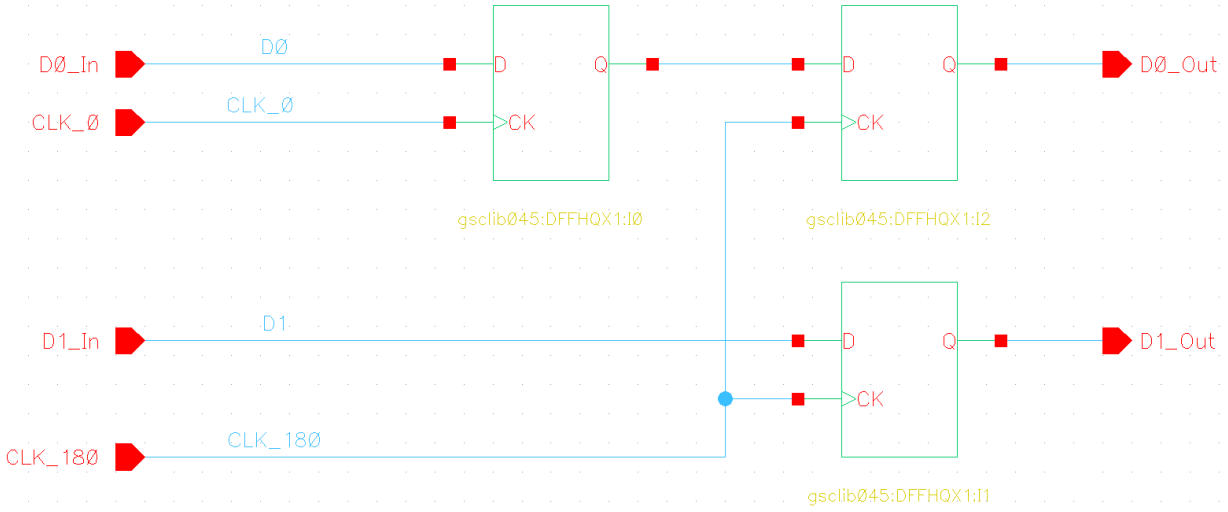


Fig. 41: Synchronizer

V. Justification for the Choice of Clock-Rate at TX and RX

The maximum clock frequency is limited by TX circuits (e.g., D Flip-Flops) as well as TX clocking circuits (e.g., clock buffers). The timing overhead $(t_{clk-to-q\ \max} + t_{setup})$ of a static Flip-flop is only about 3FO4 [**?**], whereas the maximum frequency a CMOS clock buffer (i.e., CMOS inverter chains) can support is 8FO4 [**?**]. This shows that the clocking circuit is a key bottleneck that limits the maximum clock frequency.

The TX is designed to transmit data at 10Gb/s. With full rate clocking, the clock frequency needs to be 10GHz. However, the minimum clock period supported by CMOS clock buffers is 8FO4, which is approximately 120ps for 45nm technology; this clock period (120ps) corresponds to a maximum clock frequency of 8.33GHz, which is far less than 10GHz [2]. Therefore, full-rate clocking does not work well with 10Gb/s data rate because the clock signals will experience *severe* attenuation in the voltage domain as well as the spreading in the time domain when they propagate through clock buffers; these detrimental effects are the result of the limited RC bandwidth of each inverter in the clock buffers [2].

With half-rate clocking, the clock frequency only needs to be 5GHz $(i.e., T_{CLK} = 200ps)$ for transmitting data at 10Gb/s. Given that $T_{CLK} = 200ps > 8FO4$, half rate clocking is good enough to tolerate the ISI caused by the CMOS clock buffers. Therefore, the clock frequency in my design is chosen to be 5GHz along with a 2:1 serialization. It is required that the link

must be a source synchronous link, meaning that the TX and RX are clocked with the same high frequency reference PLL. As a result, the clock frequency at RX is also chosen to be 5GHz.

VI. Conclusion

The proposed high-speed link implementation works, but it is very sensitive to timing due to the fact that I do not have enough time to optimize the RX. It is sensitive to timing in the sense that, to ensure the RX works properly, the rise/fall time of the clock signal has to be small and aligned with the maximum eye height position. Further improvements are needed to make the RX and TX more robust.

References

[1] Lecture Notes
[2] Yang, C. K. (1998). DESIGN OF HIGH-SPEED SERIAL LINKS IN CMOS. Retrieved March 31, 2019, from https://pdfs.semanticscholar.org/9899/30658259fa6519a02ccc483904a622f95389.pdf.
[3] Hodges, D. A., Jackson, H. G., & Saleh, R. A. (2005). Analysis and design of digital integrated circuits: In deep submicron technology. New Delhi: Tata McGraw Hill Education.
[4] R. V. Anugraha, D. S. Durga and R. Avudaiammam, 'Design and performance analysis of 2:1 multiplexer using multiple logic families at 180 nm technology,' 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2017, pp. 1849-1853.