

Design, Analysis, and Simulation of A 10 Gbps SerDes I/O Link

Bruce Xi

I. SYSTEM OVERVIEW

This project presents the design, analysis, and simulation of a 10 Gbps SerDes I/O link. Shown in Figure 1, the link consists of a serializing transmitter (TX), a differential channel modeled as a transmission line, and a deserializing receiver (RX).

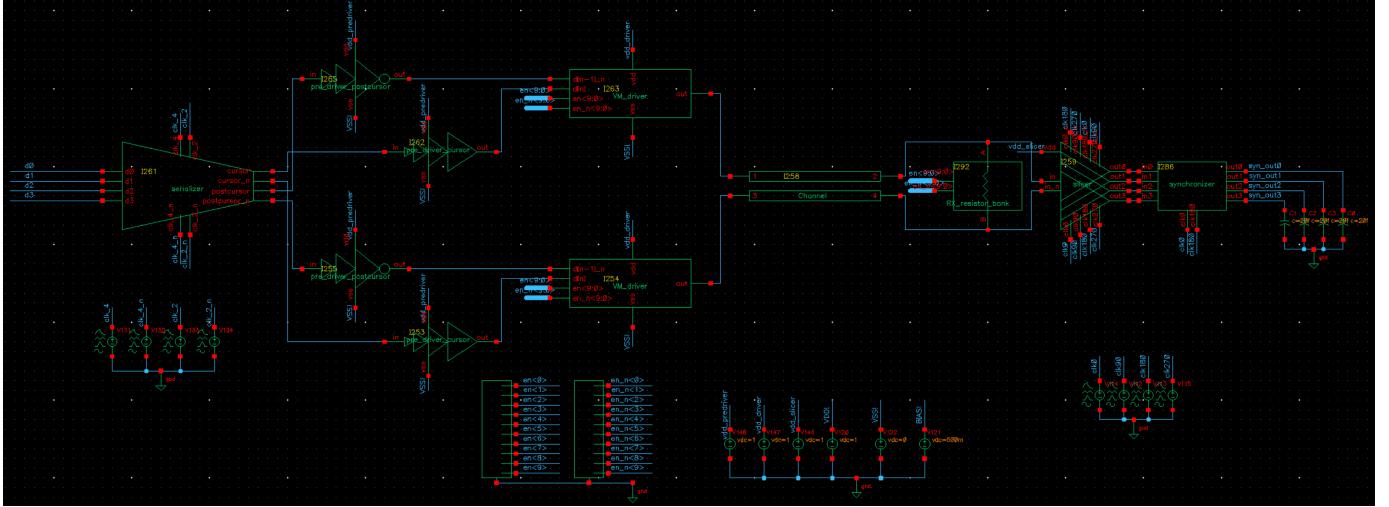


Fig. 1: System schematic of the SerDes I/O Link.

II. CHANNEL CHARACTERIZATION

A. Impulse Response

Given the channel's S-parameters provided by the course instructor, the frequency response of the channel, $H(\omega)$, can be directly obtained as it is equivalent to the channel's S_{21} profile. The impulse response of the channel, $h(t)$, was obtained by performing a fast inverse Fourier transform on $H(\omega)$ using a provided MATLAB script, the result is shown in Figure 2.

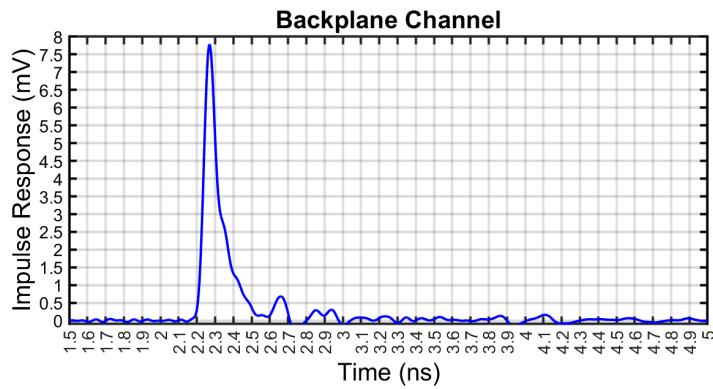


Fig. 2: Channel's impulse response.

B. Pulse Response

The pulse response of the channel $p(t)$ was obtained by convolving a pulse signal $\phi(t)$ with the impulse response $h(t)$, i.e.,

$$p(t) = \phi(t) * h(t)$$

Such procedure was carried out in a provided MATLAB script, the result is shown in Figure 3. Here, we can see that the input is a perfect pulse, that is, being a perfect 1 (1000 mV) for 1 UI, and 0 otherwise. However, due to the nonideality of the channel, at a data rate of 10 Gbps, the output has finite edges, attenuated cursor magnitude (less than 600 mV), and non-zero post-cursors that will cause inter-symbol-interference (ISI) if more symbols were provided in the input.

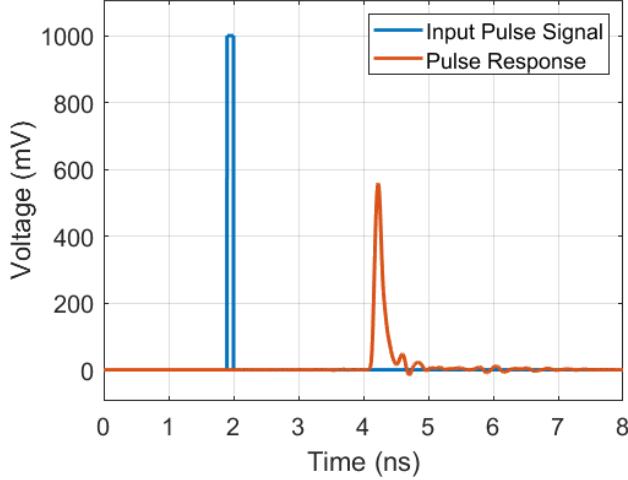


Fig. 3: Channel's pulse response.

Figure 4 shows the pulse response sampled at individual UIs (100 ps intervals), from which the 5 most significant cursors were recognized and summarized in Table I.

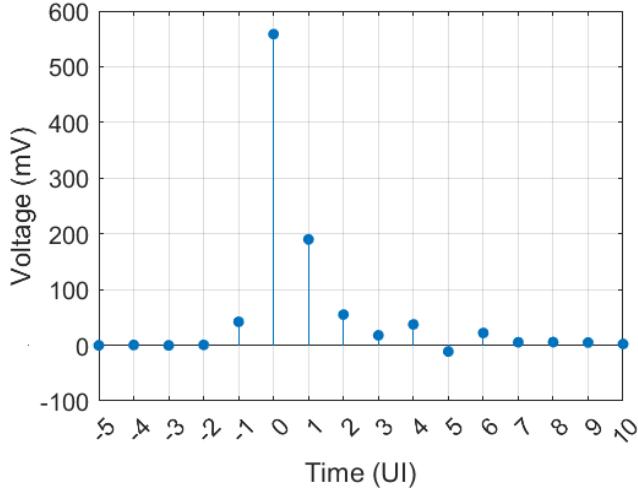


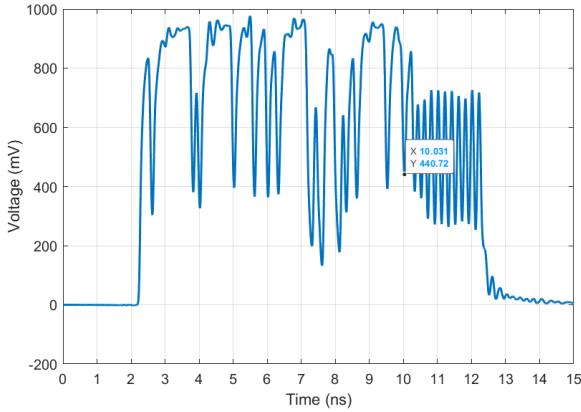
Fig. 4: Channel's pulse response sampled at individual UIs.

C. Worst-case Data Patterns

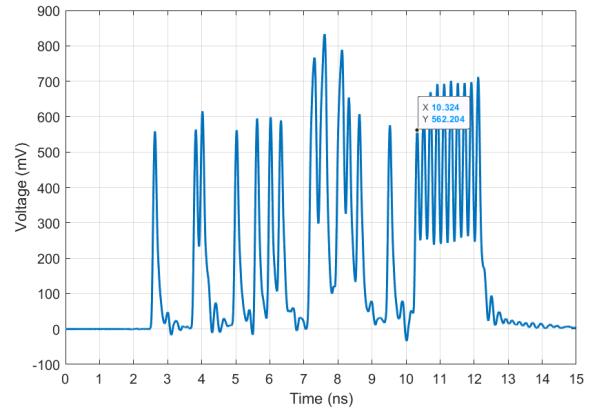
To obtain the worst-case-1 and worst-case-0 patterns, the sampled pulse response $p[n]$ was flipped, yielding $p[k] = p[-n]$. The signs of cursors in $p[k]$ are checked to obtain the worst-case data patterns. A positive cursor produces a 1 in the worst-case 0 pattern and a 0 in the worst-case 1 pattern, and vice versa for negative cursors. The obtained worst-case sequences were fed into the channel, and the outputs were present in Figure 5.

TABLE I: Top 5 most significant cursors

UI	Magnitude (mV)
-1	42
0	559
1	190
2	55
3	19



(a) Worst-case 0 channel output

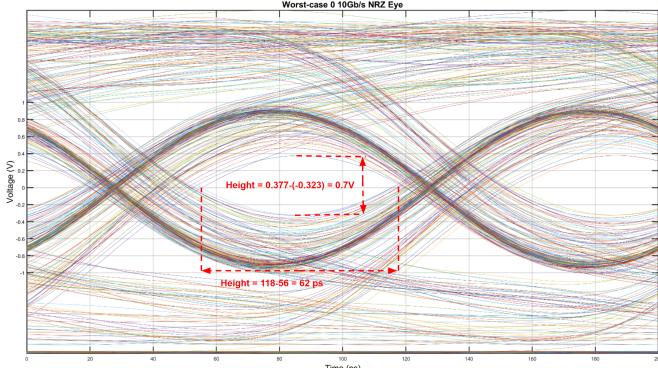


(b) Worst-case 1 channel output

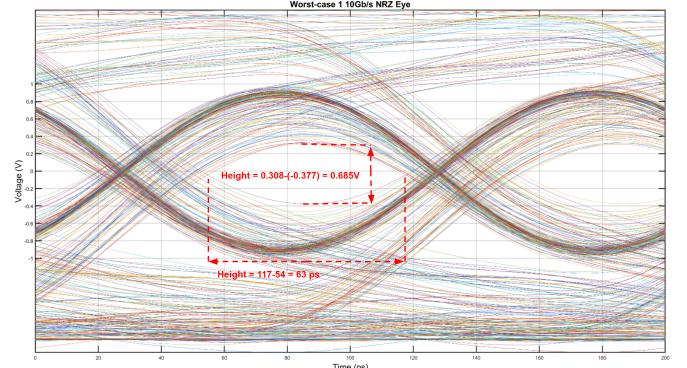
Fig. 5: Worst-case channel outputs

III. EYE DIAGRAMS

Shown in Figure 6. The worst-case 0 pattern produces the lowest 1 of 0.377 V and the highest 0 of -0.323 V. The worst-case 1 pattern produces the lowest 1 of 0.308 V and the highest 0 of -0.377 V.



(a) Worst-case 0 eye diagram.



(b) Worst-case 1 eye diagram.

Fig. 6: Worst-case eye diagrams at 10 Gbps.

IV. TRANSMITTER

The TX architecture is illustrated in Figure 7, comprising a 4:1 serializer, a pre-driver, and a voltage mode (VM) driver. Table II provides a summary of the specifications for the implemented TX. Detailed discussions on the components of the TX will follow, along with sections on simulation results and power consumption analyses.

A. Serializer

The schematic of the 4:1 serializer is depicted in Figure 8. At the input side, a 2:1 serializer incorporating two flip-flops (FFs), one latch, and one multiplexer (MUX) serializes data inputs d0 and d2, while an identical 2:1 serializer processes inputs d1 and d3. All FFs and MUXes are synchronized to a 2.5 GHz clock with a 0-degree phase, whereas the latches operate on a 2.5 GHz clock with a 180-degree phase shift. The outputs from these 2:1 serializers are then combined by another 2:1 serializer,

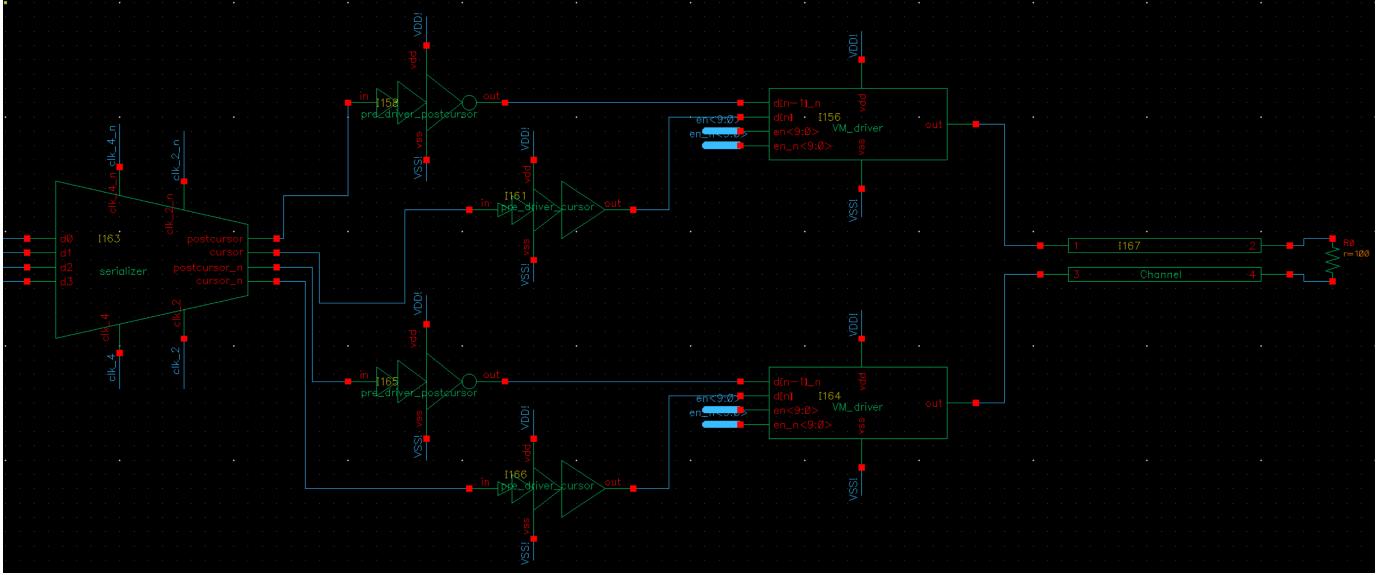


Fig. 7: Schematic representation of the TX highlighting its main components: the serializer, pre-driver, and VM driver.

TABLE II: Summary of the Transmitter Specifications

Input data rate (Gbps)	2.5
Output data rate (Gbps)	10
Average power consumption (mW)	8.63
Equalization method	2-tap TXLE FIR de-emphasis
Termination impedance (Ω)	$50 \pm 30\%$
VDD (V)	1

this time clocked by a pair of 5 GHz clocks phased 180 degrees apart. Employing half-rate clocking, this 4:1 serializer achieves an output data rate of 10 Gbps. The generation of post-cursors involves two additional latches and one MUX, all sourced from the provided libraries. A transient simulation of the serializer with four inputs was conducted as follows:

$$\begin{aligned} d0 &= 1V \\ d1 &= 1V \\ d2 &= 1V \\ d3 &= 0V \end{aligned}$$

Figure 9 illustrates the simulation results of the serialized output at 10 Gbps, displaying the post-cursors (bottom) delayed by 1 Unit Interval (UI) relative to the cursors (top).

B. Pre-drivers

Two pre-drivers, consisting of inverter chains, are specifically designed for cursors and post-cursors. The cursor pre-driver features an even number of inverters to maintain the signal phase, whereas the post-cursor pre-driver utilizes an odd number of inverters to invert the signal phase. The selection of the number of inverters and their sizing was meticulously optimized for delay using logical effort techniques. Figure 10 and Figure 11 illustrate the topologies of these two pre-drivers, detailing their inverter counts and sizing.

C. Equalization

A 2-tap Finite Impulse Response (FIR) equalizer was implemented in the TX due to its straightforward design and integration. The taps, each with a 3-bit resolution, comprise a cursor and a post-cursor. The values of these taps were determined using a provided MATLAB script and are displayed in Table III. Representing the taps as fractions, the cursor tap is $\frac{5}{7}$ and the post-cursor tap is $-\frac{2}{7}$. The equalizer integrates seamlessly into the VM driver, enhancing the signal by applying de-emphasis to the transmitted data. Equation 1 illustrates the relationship between the transmitted data ($y[n]$), the cursor data ($d[n]$), the post-cursor data ($d[n - 1]$), and the equalization taps ($1 - \alpha$ and $-\alpha$). Given the tap values, it is evident that $\alpha = \frac{2}{7}$ and $1 - \alpha = \frac{5}{7}$.

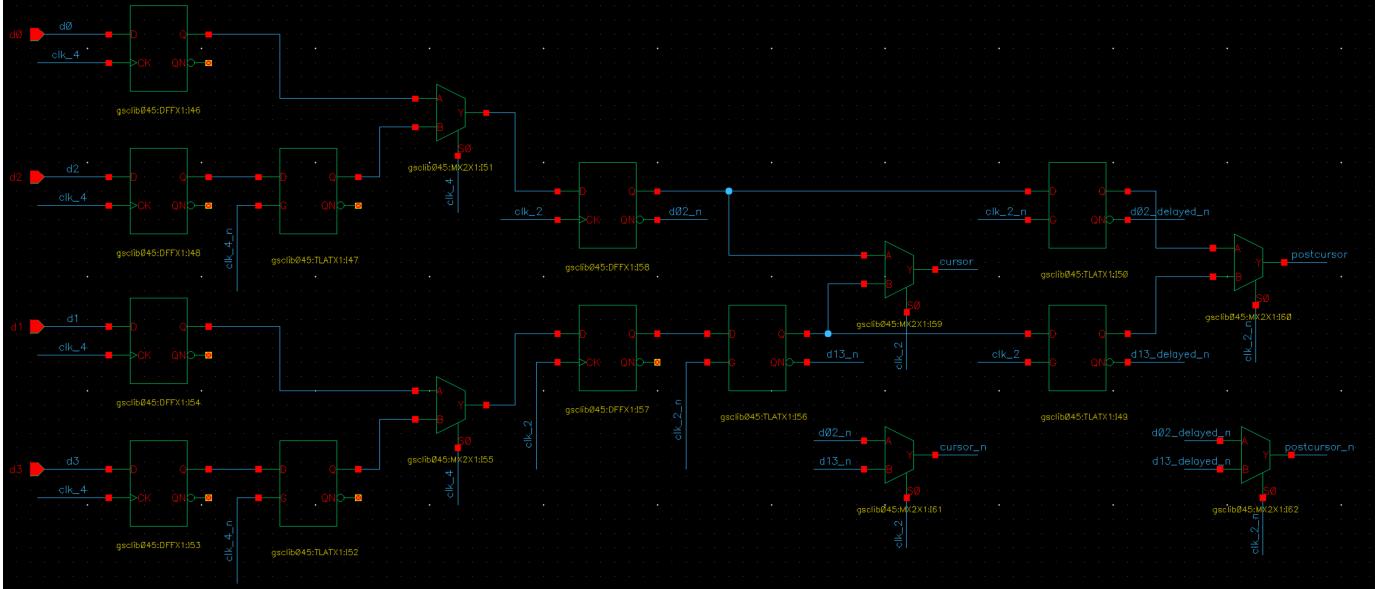


Fig. 8: Schematic diagram of the 4:1 serializer, showing the configuration of flip-flops, latches, and multiplexers used to achieve data serialization.

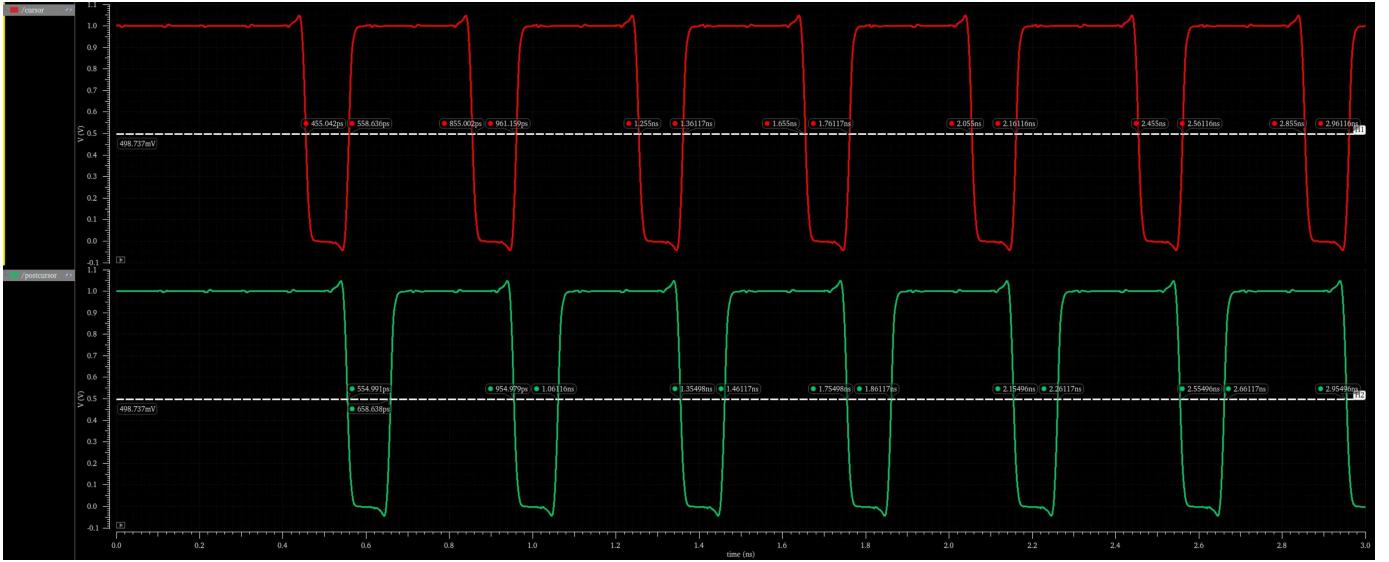


Fig. 9: Transient simulation results for the 4:1 serializer, highlighting the serialized output waveform at 10 Gbps and the timing of post-cursors.

$$y[n] = (1 - \alpha)d[n] - \alpha d[n - 1] \quad (1)$$

TABLE III: Values of the TX-FIR Equalizer Taps

Cursor	Post-cursor
0.7143	-0.2857

D. VM driver

1) *Tunable termination impedance:* Figure 12 illustrates the basic configuration of a high-swing VM driver, which is essentially an inverter with additional resistors in series with the transistors (not depicted in the figure). The combined resistance of the pull-up path and the pull-down path, R_U and R_D , must match the channel's characteristic impedance Z_0 , which is also

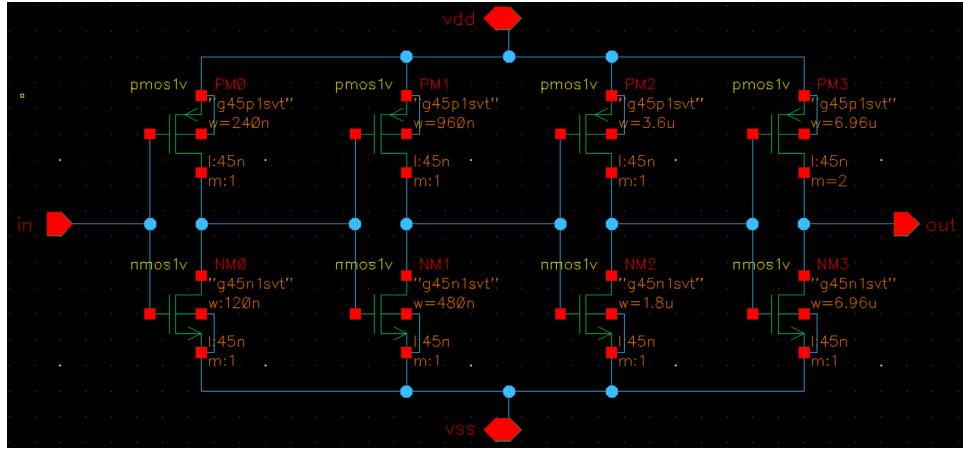


Fig. 10: Schematic of the cursor pre-driver, depicting the even-numbered inverter chain configured to maintain signal phase.

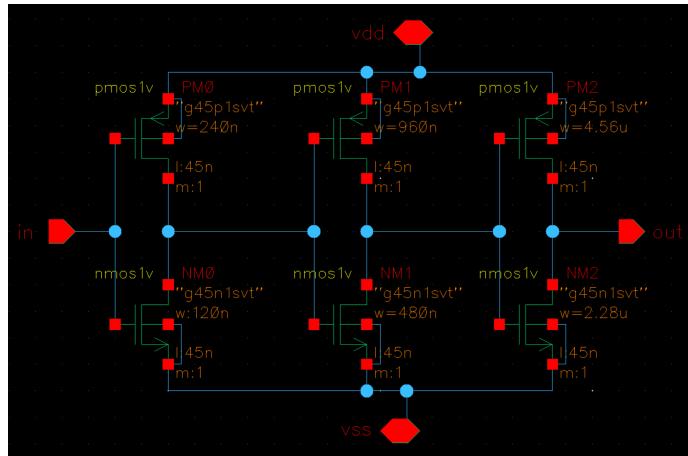


Fig. 11: Schematic of the post-cursor pre-driver, showing the odd-numbered inverter chain designed to invert the signal phase.

the termination resistance at the receiver, R_T . For this project, it is specified that $R_T = 50\Omega$ with a tuning range of $\pm 30\%$, equating to a range of $35\Omega \sim 65\Omega$ for both R_U and R_D . To facilitate tuning, the driver is segmented into 10 parallel-connected

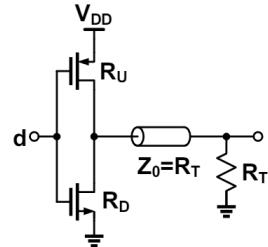


Fig. 12: Illustration of a basic high-swing VM driver.

units, each with an impedance of 350Ω .

Figure 13 shows the configuration of the segmented VM driver. Here, $N = 10$, and impedance tuning is achieved by varying the number of active segments through the *en* signal, as demonstrated in Table IV.

TABLE IV: Ideal VM driver impedance tuning

Enable segments	1	2	3	4	5	6	7	8	9	10
Impedance (Ω)	350	175	116.7	87.5	70	58.3	50	43.8	38.9	35

Figure 14 displays the design schematic for the segmented VM driver, each segment designed to have an output impedance

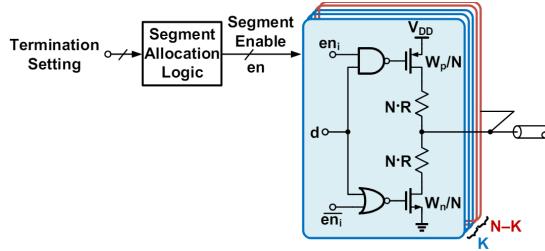


Fig. 13: Illustration of a segmented VM driver designed to enable impedance tuning.

of 350Ω . By controlling the enable signals at a Nyquist frequency of $5GHz$, the resulting impedance tuning is shown in Figure 15. The nonlinearity of the transistors results in a non-linear relationship between the number of enabled segments and the impedance. At $5GHz$, enabling 7 segments yields an impedance of approximately 48.5Ω , which is close to the desired 50Ω , maintaining the required $\pm 30\%$ tuning range. Table V summarizes these results.

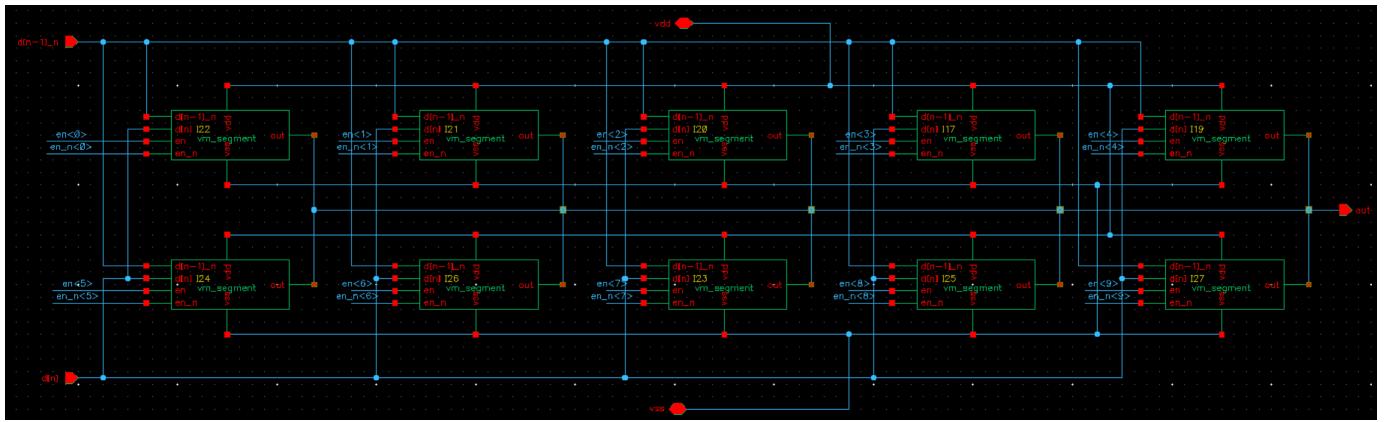


Fig. 14: Schematic of the segmented VM driver.

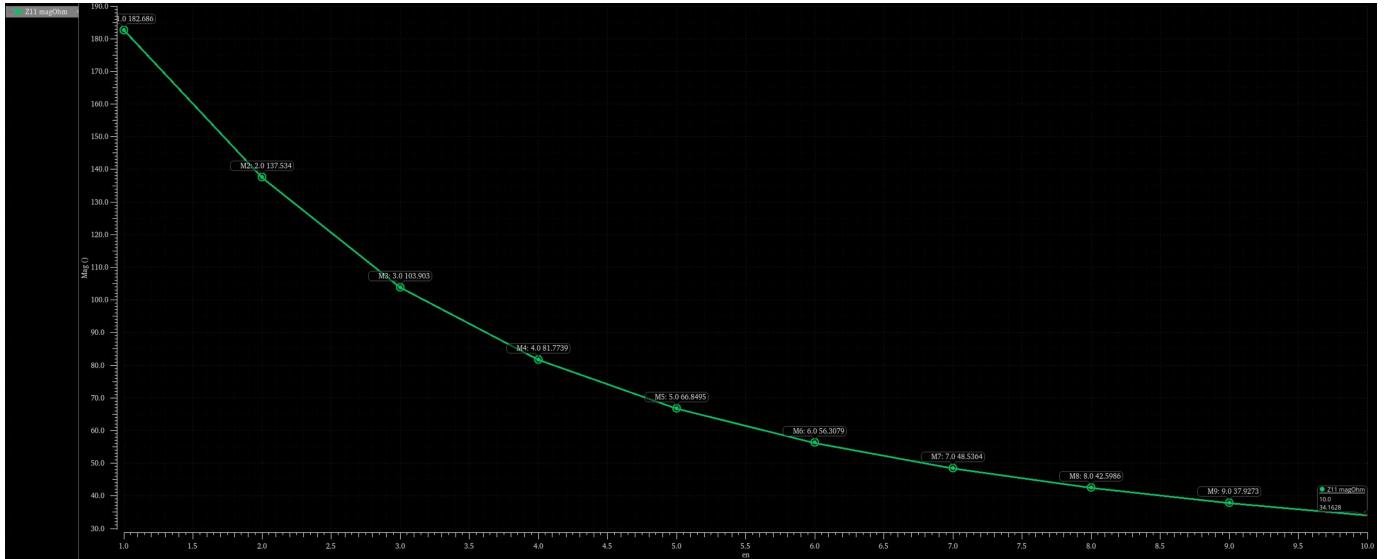


Fig. 15: Tuning of the VM driver impedance by activating different segment counts.

2) *De-emphasis:* For de-emphasis FIR equalization, which is decoupled from impedance tuning, each segment in Figure 17 is divided into m slices, each with distinct transistor sizing and controlled by either the cursor or the inverse post-cursor signal, as shown in Figure 16. During de-emphasis equalization, certain slices activate their pull-up paths while others their pull-down

TABLE V: Real VM driver impedance tuning

Enable segments	1	2	3	4	5	6	7	8	9	10
Impedance (Ω)	182.7	137.5	103.9	81.8	66.8	56.3	48.5	42.6	37.9	34.1

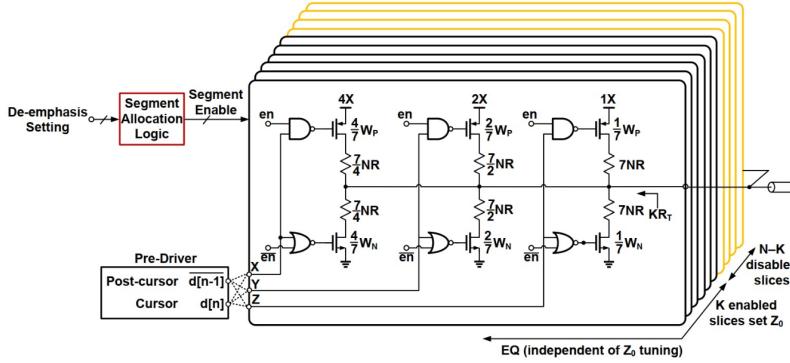


Fig. 16: Segmented VM driver with de-emphasis slices, independent of impedance tuning.

paths. The resulting impedances, R_U and R_D , are aligned with the equalization tap values according to Equation 2.

$$\frac{R_D - R_U}{R_D + R_U} = 1 - 2\alpha \quad (2)$$

Despite the configuration for equalization, the shunt impedance comprising R_U and R_D must still sum to the combined termination impedance of $KR_T = 7 \times 50 = 350\Omega$.

$$\frac{1}{R_D} + \frac{1}{R_U} = \frac{1}{KR_T} \quad (3)$$

Combining Equation 2 with Equation 3, we derive that $R_U = \frac{KR_T}{1-\alpha}$ and $R_D = \frac{KR_T}{\alpha}$. Given $\alpha = \frac{2}{7}$, it follows that $\frac{R_U}{R_D} = \frac{2}{5}$. In the implemented VM driver, R_U is governed by the inverse of the post-cursor signal, and R_D by the cursor signal. The sizing of the transistors and resistor values for the slices in each segment are accordingly determined, as depicted in Figure 17. Figure 18 displays the transient simulation waveform of the VM driver, showcasing the achieved de-emphasized voltage levels.

E. TX Simulation Results

Figure 19 displays the TX output for 6 UIs, using alternating data patterns. Figure 20 presents the eye diagram at the TX output over 10,000 UIs, generated by a PRBS7. The dimensions of the eye diagram are summarized in Table VI. Figure 21 illustrates the eye diagram at the TX output under the worst-case data pattern scenario. The measurements of the eye's height and width for this case are detailed in Table VII.

TABLE VI: Eye Diagram Dimensions for PRBS7 over 10,000 UIs

Height (mv)	60.6
Width (ps)	54.0

TABLE VII: Eye Diagram Dimensions for Worst-Case Scenario

Height (mv)	46.9
Width (ps)	59.4

F. TX Power and Energy Analysis

Power consumption is determined by measuring the average current drawn from the V_{DD} , which is 1 Volt, for each of the three components over a period of 100 UIs. These components include the serializer, the pre-driver, and the driver. The average current for each component is then multiplied by the V_{DD} to calculate the power. The energy per bit is subsequently calculated by dividing the power by the data rate, which is 10 Gbps. Table VIII provides a summary of the power and energy per bit for each component in the TX.

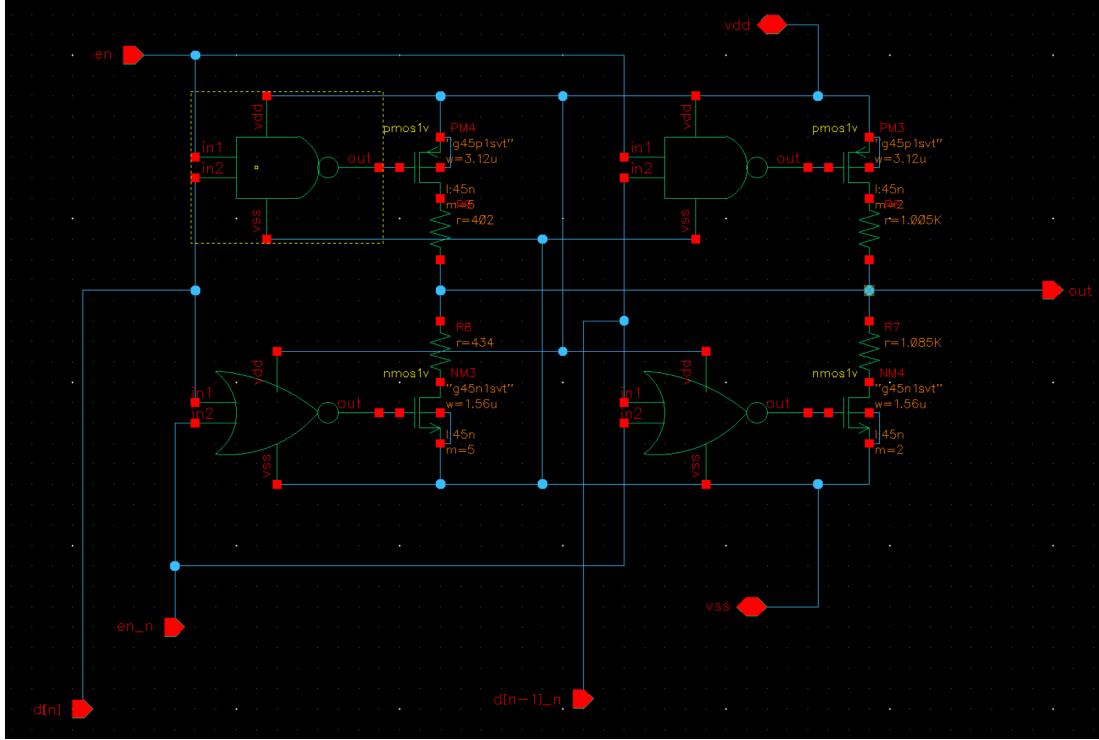


Fig. 17: Schematic of a VM driver segment, featuring two slices for de-emphasis adjustment.

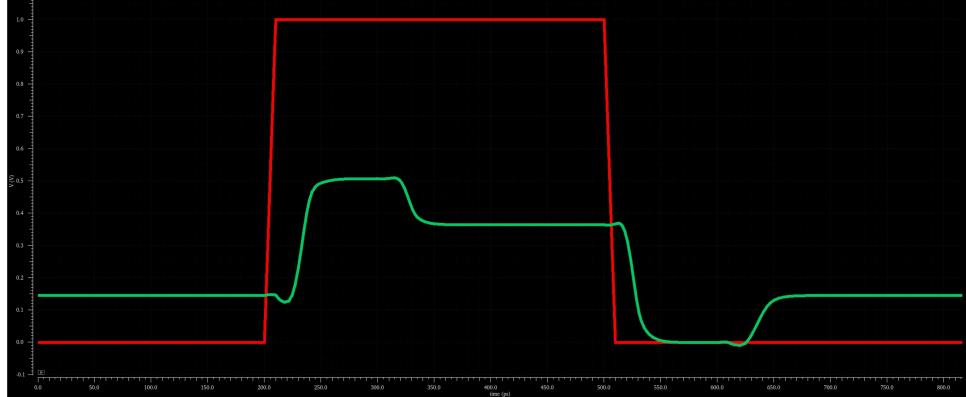


Fig. 18: Transient simulation of de-emphasized voltage levels in the VM driver.

TABLE VIII: Breakdown of Power and Energy per Bit for the TX

Component	Average power (mW)	Energy per bit (pJ)
Serializer	0.155	0.0155
Pre-driver	0.783	0.0783
VM driver	7.69	0.769
Total TX	8.63	0.863

V. RECEIVER

Figure 22 illustrates the architecture of the RX, which functions as a 1:4 deserializer. It includes a tunable resistor bank, four slicers, and four synchronizers. Given that the incoming data rate from the channel is 10 Gbps, the slicers are driven by four clock signals with phases of 0, 90, 180, and 270 degrees, each operating at 2.5 GHz. The synchronizer is clocked by both the 0-degree and 180-degree clocks. Table IX provides a summary of the RX specifications.

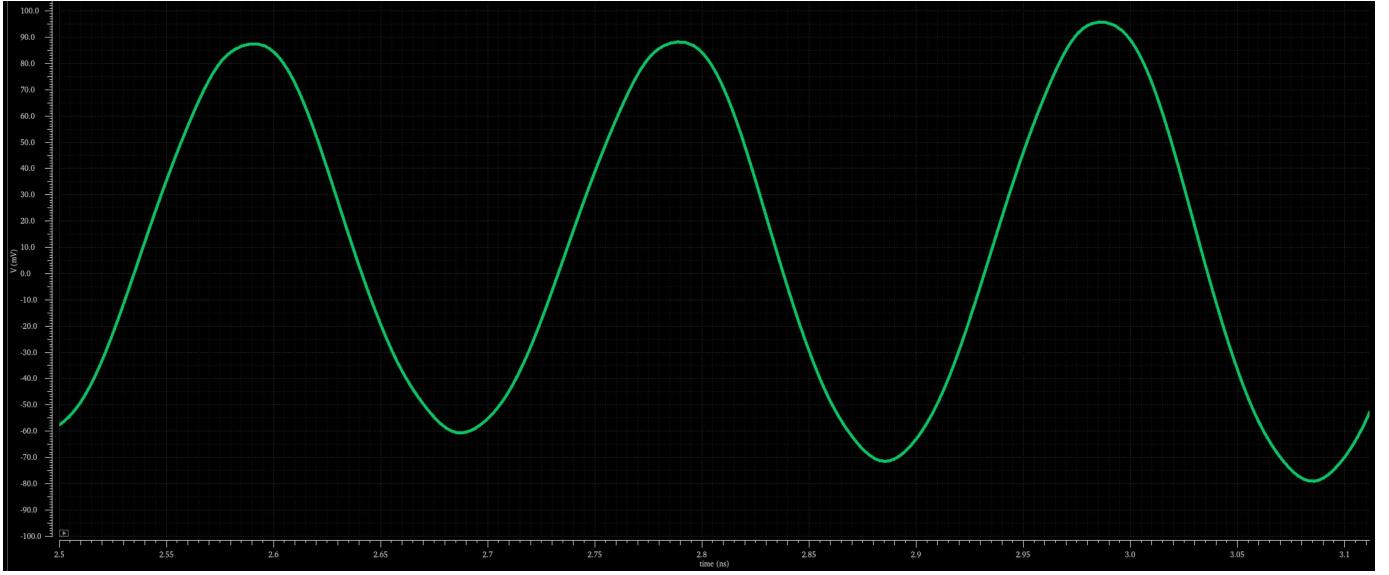


Fig. 19: TX output displaying alternating data transitions over 6 UIs.

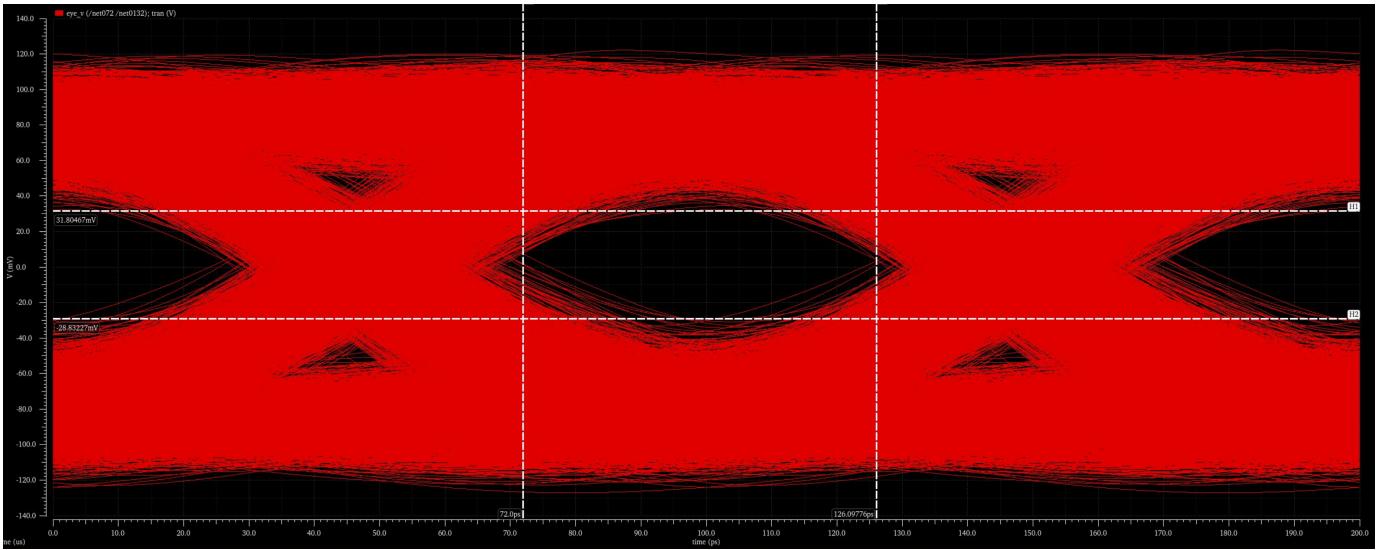


Fig. 20: Eye diagram at TX output for a PRBS7 data pattern, captured over 10,000 UIs.

TABLE IX: Summary of the Receiver Specifications

Input data rate (Gbps)	10
Output data rate (Gbps)	2.5
Average power consumption (mW)	0.07
Impedance (Ω)	$100 \pm 30\%$
VDD (V)	1

A. Impedance Tuning

As depicted in Figure 24, the RX tunable resistor bank is composed of 10 parallel resistor slices. Each slice features a resistor with a resistance of $R = 570\Omega$ and a transmission gate, which is controlled by an enable signal and functions as a switch. The NMOS and PMOS transistors within the transmission gate are each sized at $12\mu m$. The configuration of the slices, transistor sizing, and resistor value are selected to ensure that the RX impedance at 5 GHz is $100\Omega \pm 30\%$, ranging from 70Ω to 130Ω . Figure 24 illustrates how impedance tuning varies with different numbers of slices activated. The findings are summarized in Table X. Specifically, with 5 slices activated, the impedance is 135.7Ω ; with 7 slices, it is 97.8Ω ; and with all 10 slices, it is 67.4Ω .

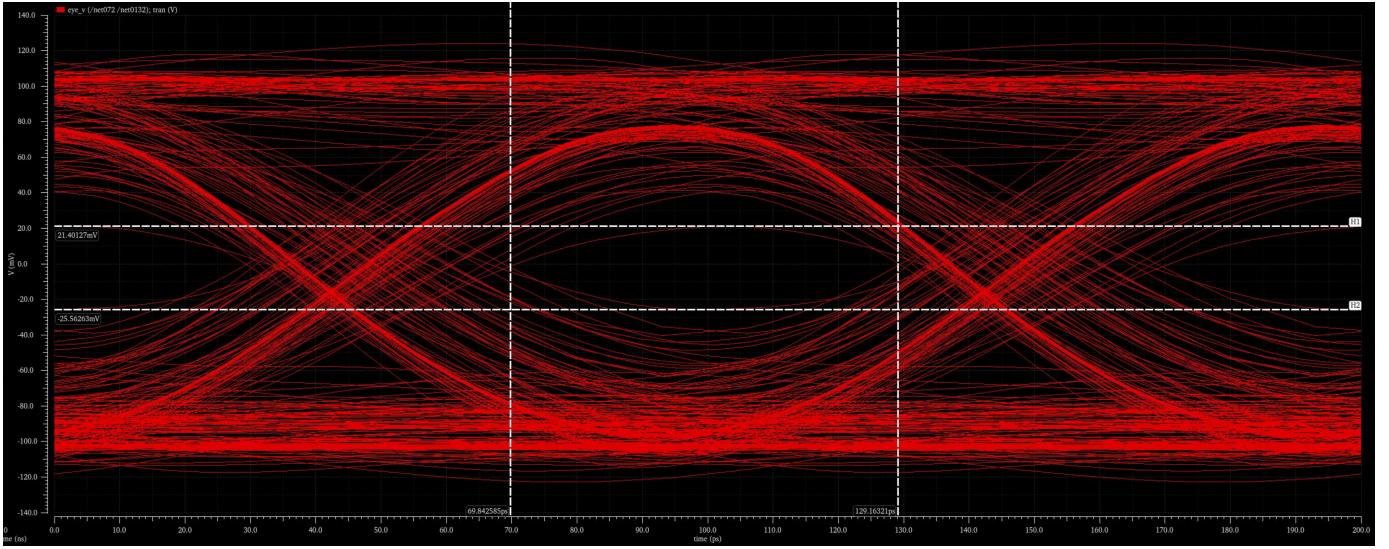


Fig. 21: Eye diagram under worst-case data pattern conditions at the TX output.

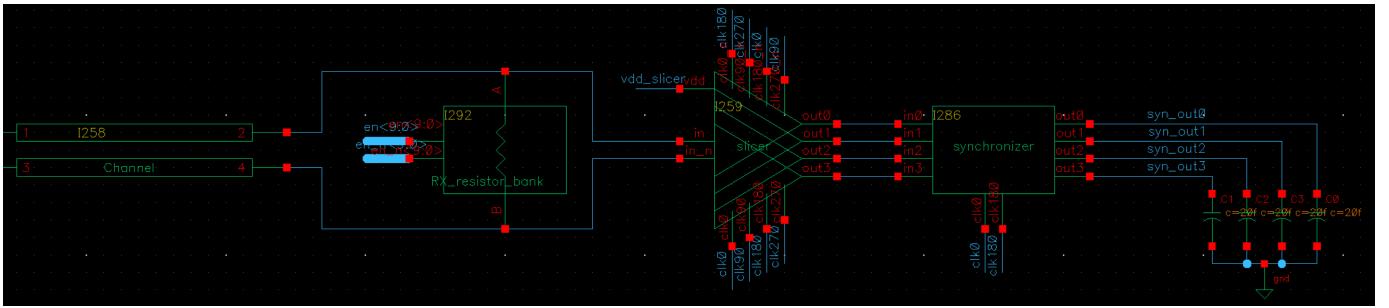


Fig. 22: RX architecture.

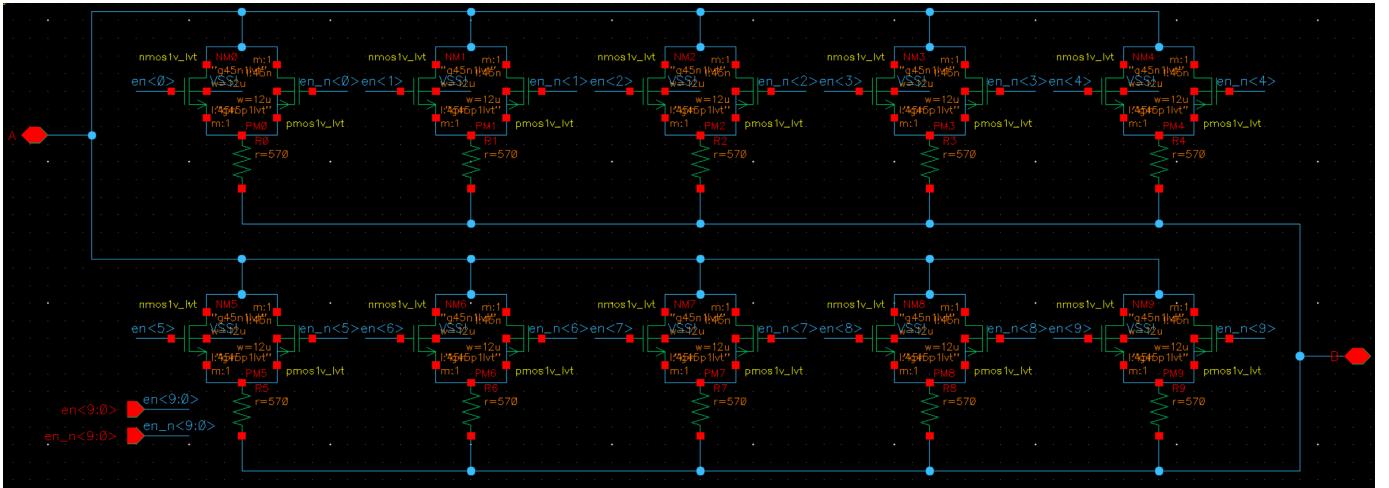


Fig. 23: RX tunable resistor bank.

TABLE X: RX impedance tuning

Enable slices	1	2	3	4	5	6	7	8	9	10
Impedance (Ω)	343.1	265.0	206.3	165.2	135.7	114.1	97.8	85.2	75.3	67.4

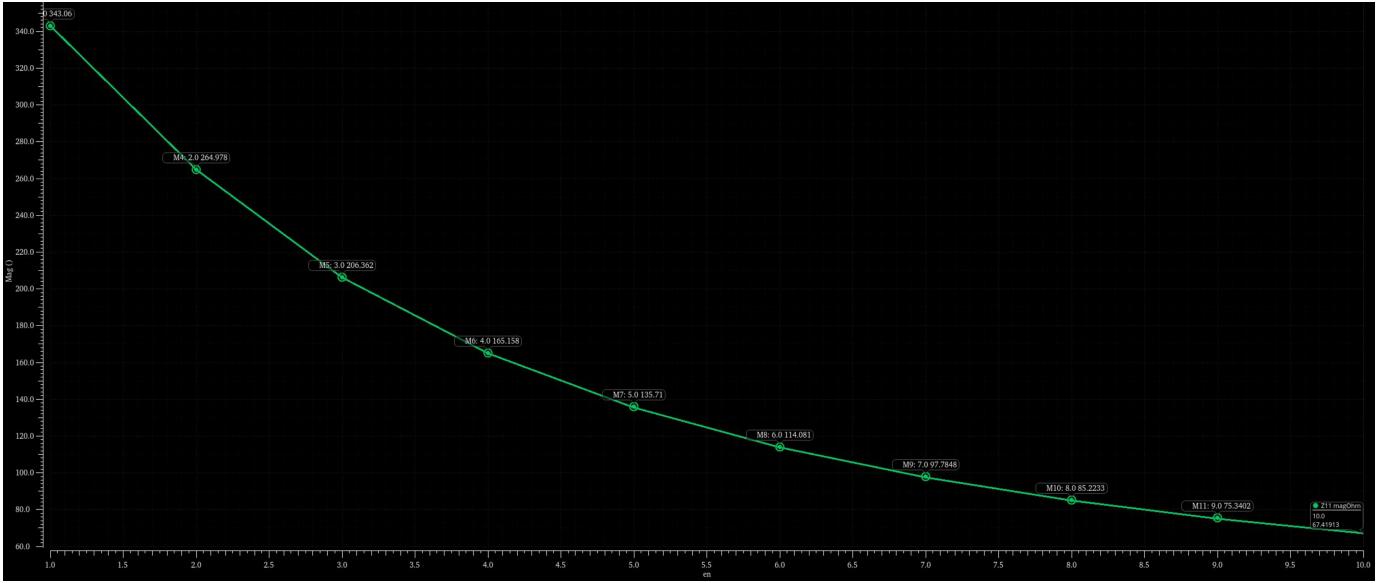


Fig. 24: RX impedance tuning.

B. Slicer

Figure 25 depicts the architecture of the four slicers, which handle input data at 10 Gbps from the differential channel. Each slicer contains a track-and-hold (TH) switch, a StrongARM (SA) latch, and an SR latch. The slicers are individually clocked by four distinct clock signals with phases of 0, 90, 180, and 270 degrees. Additionally, within each slicer, the TH switch and SA latch are controlled by a pair of clocks that are out of phase with each other. This dual-phase operation divides the slicer's activity into two stages. In the first phase, the TH switch's clock is high, enabling it to track the differential voltages of the input data, while the SA latch's clock is low, allowing it to pre-charge or reset its internal nodes. In the second phase, the TH switch's clock goes low, holding the voltages steady, and the SA latch's clock turns high, enabling it to sample the voltages and regenerate the digital signals. These signals are then captured by the SR latch, where they remain stable until collected by the synchronizer.

1) *Track-and-hold Switch:* A TH switch tracks the input differential voltages while the clock is high and holds the voltages when the clock is low. This behavior yields two advantages: (1) The subsequent SA latch encounters DC voltages when it begins sampling, thus it is unaffected by input jitters. (2) At any given time, only two of the four clocks are high, resulting in only two TH switches being active. Consequently, the load capacitance seen by the upstream circuit is reduced by half.

Figure 26 illustrates the design of the TH switch, comprising two transmission gates that separately handle the positive and negative inputs. Both the NMOS and PMOS transistors are sized at 960 nm. Figure 27 displays the transient simulation waveform of the TH switch, clearly showing that when the clock is high, the output tracks the input, and when the clock is low, the output remains relatively stable despite the spikes from the clock transitions.

2) *StrongARM Latch:* The SA latch regenerates the digital signal based on the differential voltages it has sampled, restoring the signal swing to its full range ($0 \sim 1$ V) and mitigating the effects of noise and losses caused by the channel. As depicted in Figure 28, the SA latch operates in four phases. During the Reset/Pre-charge phase when the clock is low, both differential output signals (out^+ and out^-) are high. Upon the clock transitioning high, it enters the Sampling phase, where the voltage difference in the differential inputs causes slight variations in the differential outputs. In the Regeneration phase, the small disparity between out^+ and out^- during the Sampling phase is amplified, driving one output high and the other low. Finally, in the Decision phase, a digital signal with a significantly larger voltage swing is regenerated from the differential inputs and captured by the subsequent SR latch.

The design objective for the SA latch is to ensure sufficient gain in the Regeneration phase to obtain valid and correct digital signals. Figure 29 displays the circuitry of the SA latch. To achieve this goal, the transistors marked in the figure are carefully sized to produce the correct transconductances and node capacitances, which determine the gain. Table XI summarizes the transistor sizing. Figure 30 presents the transient simulation waveform of the SA latch, demonstrating its correct operation across the four phases.

3) *SR Latch:* The SR latch follows the SA latch to capture the regenerated data, ensuring they remain stable until collected by the synchronizer. Its truth table is provided in Table XII. Figure 31 illustrates the circuitry of the SR latch, where all PMOS transistors are sized at 360 nm and forward-biased with 0.5 V to enhance speed. All NMOS transistors are at the minimum size

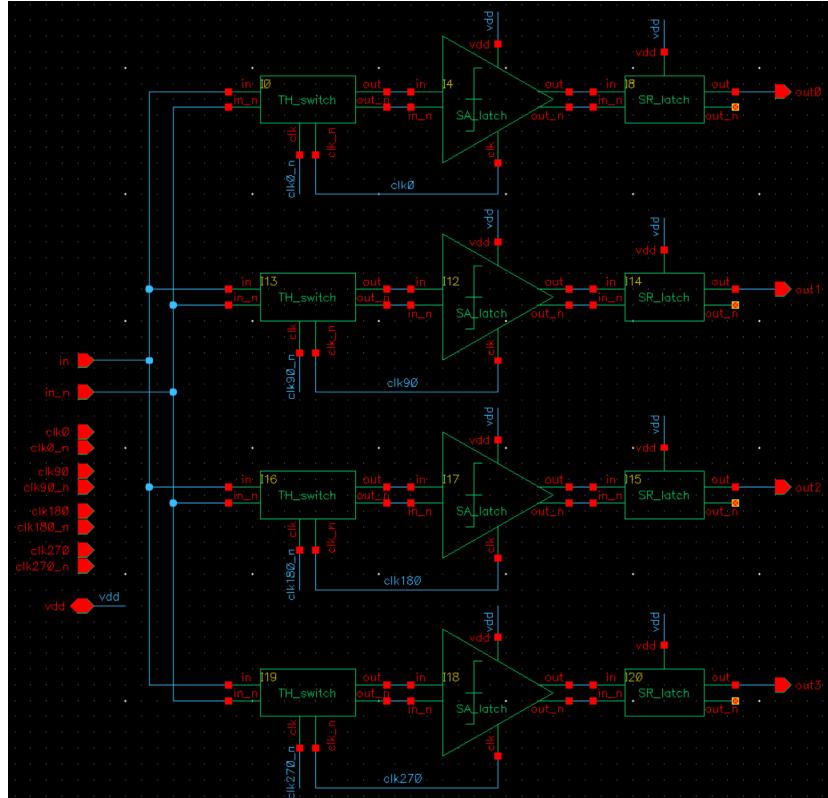


Fig. 25: Slicer architecture.

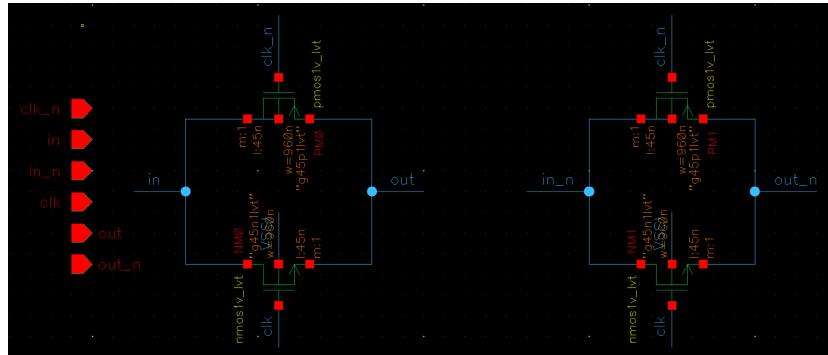


Fig. 26: Circuitry of the TH switch.

TABLE XI: StrongARM latch transistor sizing

Transistors	Sizing (nm)
M0	1200
M1, M1'	600
M2, M2', M3, M3', M4, M4', M5, M5'	120

of 120 nm. Figure 32 displays the transient simulation waveform of the SR latch, confirming its correct operation as specified in Table XII.

TABLE XII: SR latch truth table

in	in_n	Action
1	1	No change
1	0	out = 1
0	1	out = 0
0	0	Not allowed

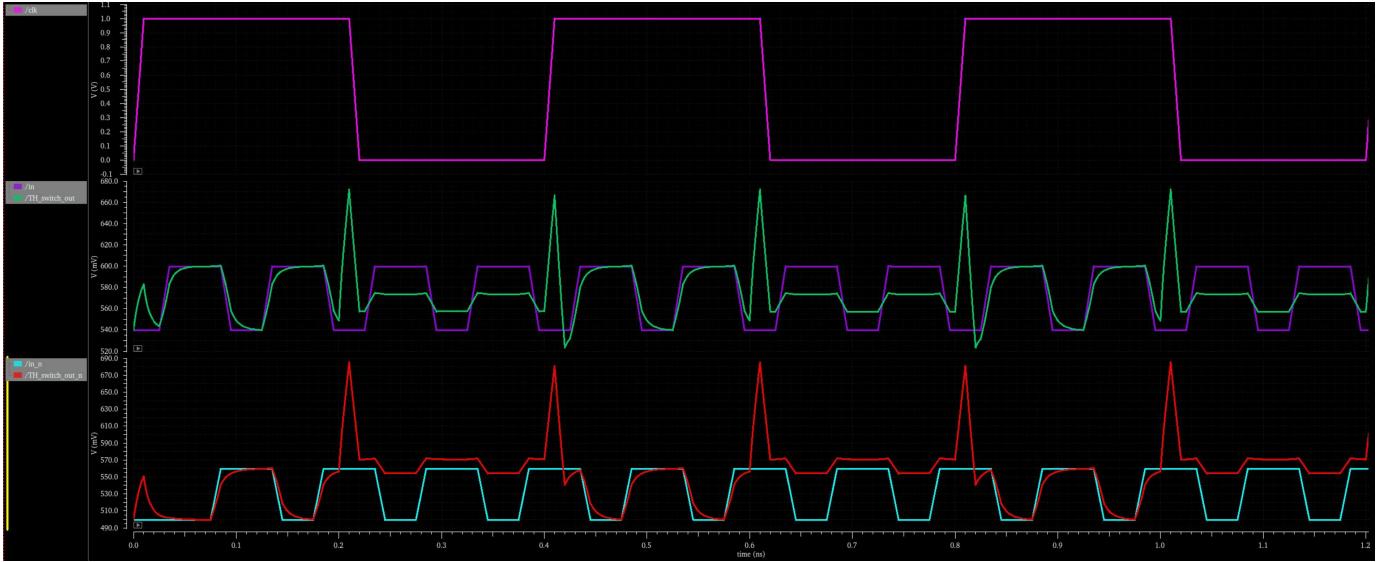


Fig. 27: Simulation waveform of the TH switch.

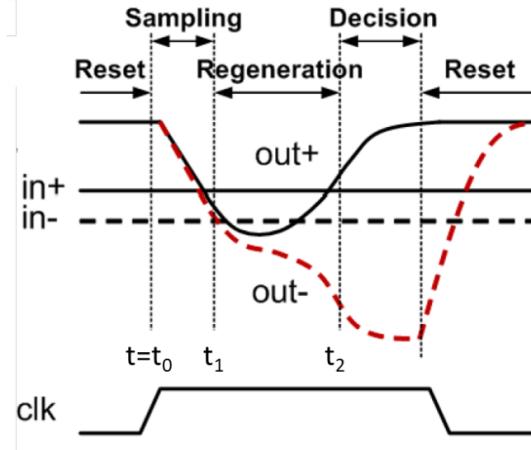


Fig. 28: The operational phases of a StrongARM latch.

C. Synchronizer

The synchronizer merges the four outputs from the slicers into a single clock. As depicted in Figure 33, it consists of six FFs and operates in two stages. In the first stage, the outputs from slicers 0 and 1 are synchronized using the 0-phase clock. Subsequently, all four outputs are synchronized using the 180-phase clock in the second stage.

D. RX Simulation Results

Figure 34 displays the transient waveform showing alternating data transitions at the RX output over six UIs. Figure 35 presents the transient simulation waveform, illustrating that the worst-case data pattern input at the TX is correctly recovered at the RX output. Figure 36 depicts the transient simulation waveform where the error checker output for the PRBS7 data pattern is captured over 10,000 UIs, showing that, aside from the initialization, there are zero errors across the 10,000 UIs.

E. RX Power and Energy Analysis

Power consumption is determined by measuring the average current drawn from the V_{DD} , for the slicers and the synchronizer over a period of 100 UIs. The average current for each component is then multiplied by the V_{DD} to calculate the power. The energy per bit is subsequently calculated by dividing the power by the data rate, which is 10 Gbps. Table XIII provides a summary of the power and energy per bit for each component in the RX.

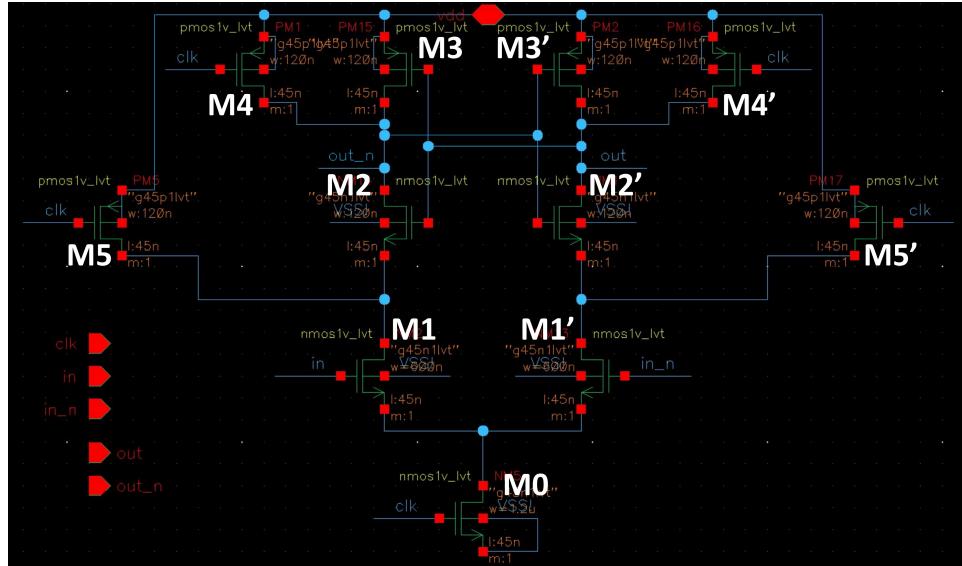


Fig. 29: Circuitry of the StrongARM latch.

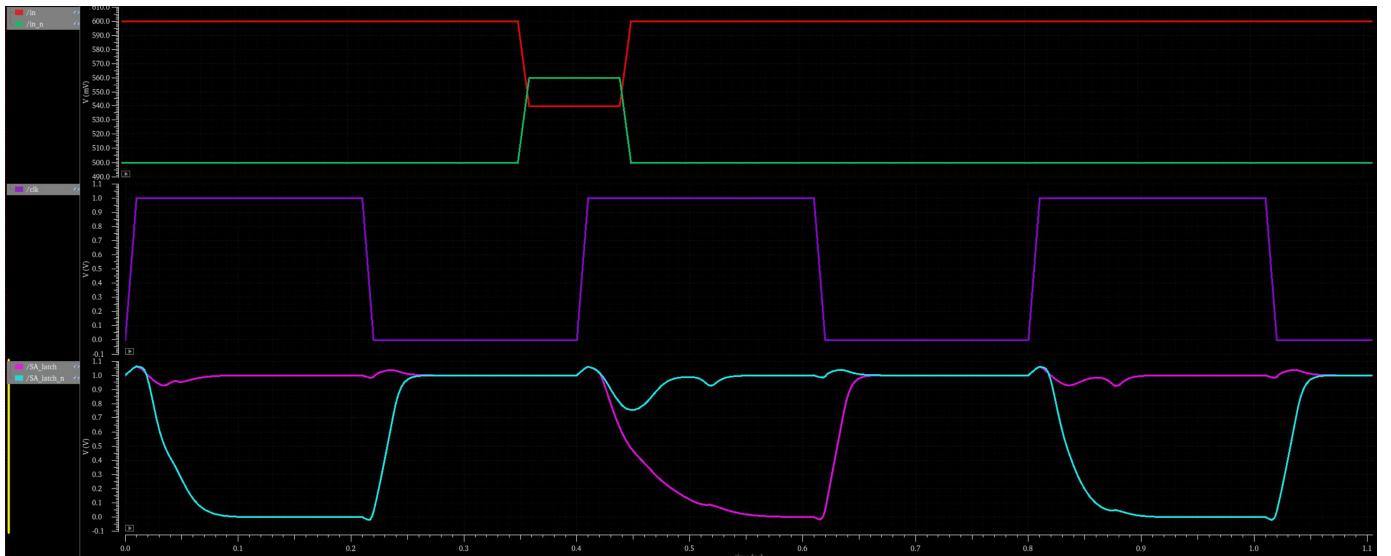


Fig. 30: Simulation waveform of the StrongARM latch.

TABLE XIII: Breakdown of Power and Energy per Bit for the RX

Component	Average power (mW)	Energy per bit (pJ)
Slicers	0.04	0.004
Synchronizer	0.03	0.003
Total RX	0.07	0.007

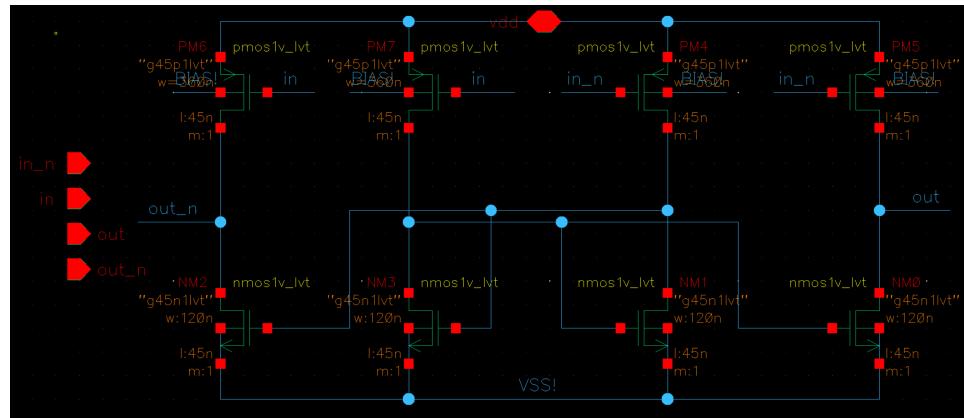


Fig. 31: Circuitry of the SR latch.

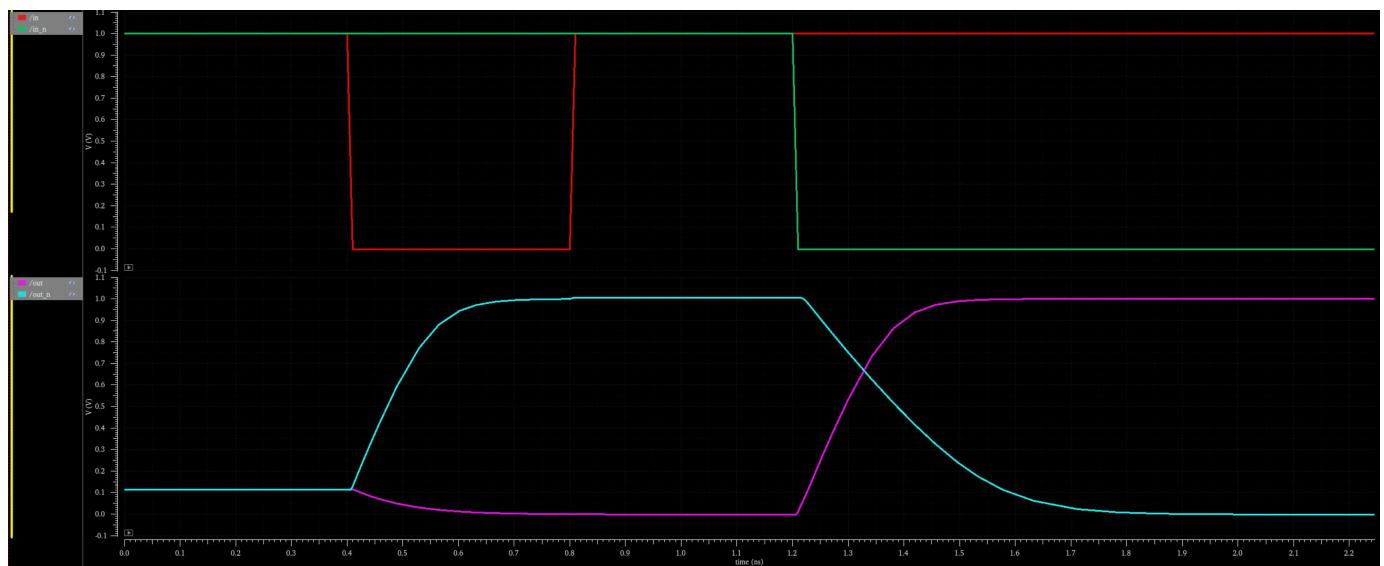


Fig. 32: Simulation waveform of the SR latch.

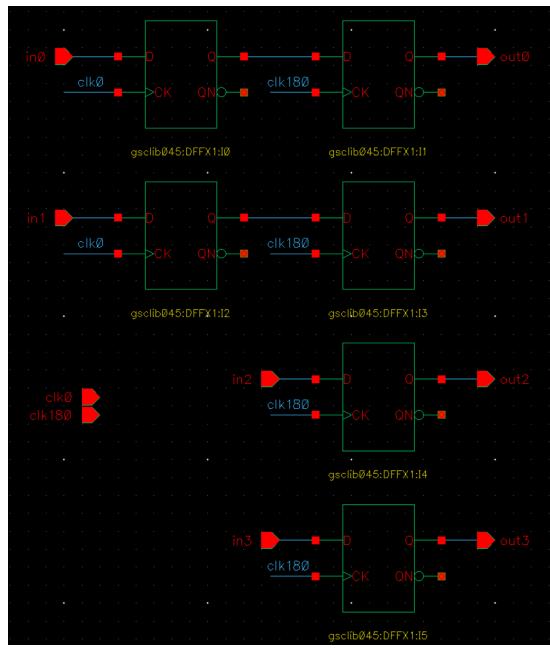


Fig. 33: Synchronizer architecture.

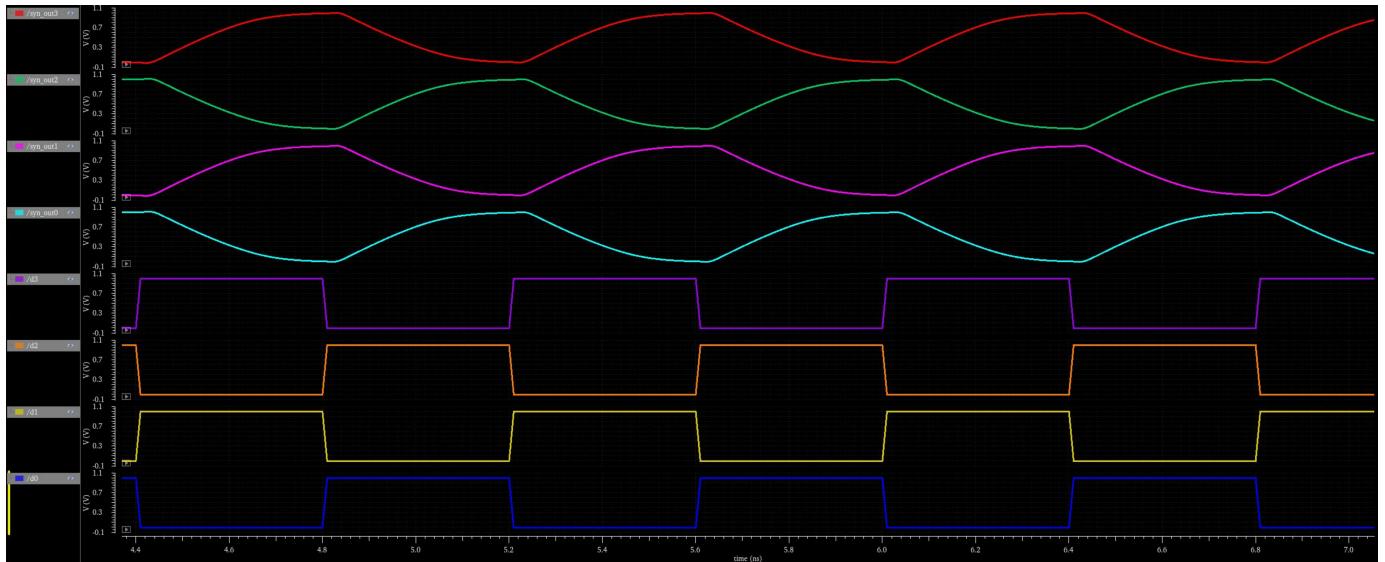


Fig. 34: Transient waveform showing alternating data transition at RX output for 6 UIs.

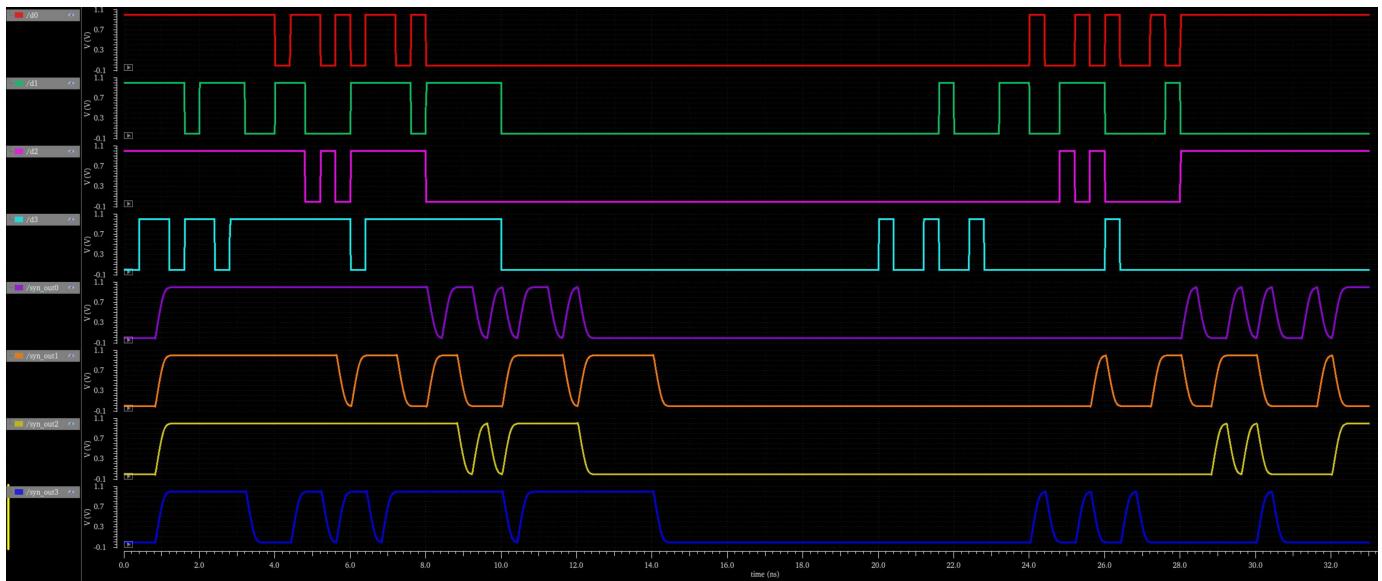


Fig. 35: Worst case data pattern recovered at the RX's output.

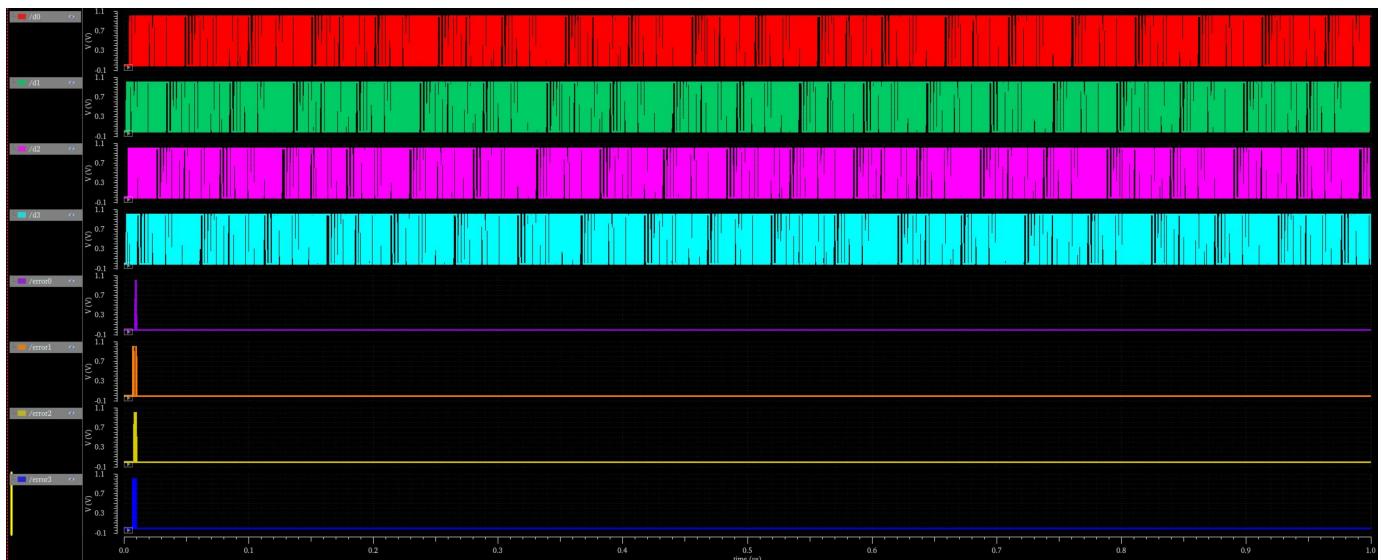


Fig. 36: Error checker output for PRBS7 data pattern captured over 10,000 UIs.