

2.5Gbps Plesiochronous SerDes with DFE in GPDK45

Aaron Sucov
Cornell University
ans246@cornell.edu

Caden Xu
Cornell University
cx237@cornell.edu

Daniel Kaminski
Cornell University
dgk64@cornell.edu

Wali Afredi
Cornell University
wua3@cornell.edu

Introduction — This Report showcases the design, simulation, and layout of a 2.5Gbps Serializer/Deserializer (SerDes) link designed in GPDK45. The transmitter takes 4-bit streams and serializes them into one differential channel, where we pre-emphasize the signal before it travels through the dispersive channel. The receiver features a continuous time linear equalizer (CTLE) and decision feedback equalization (DFE) to recoup signal integrity of the transmitted bitstream, and a clock and data recovery (CDR) system to synchronize timing for the signal.

I. MOTIVATION

SerDes links are a necessary component of many high-speed communication systems. As data rates get higher, the impact of channel losses becomes more pronounced, degrading the signal integrity. Additionally, dispersion from the channel becomes a critical issue, as the smearing of symbols becomes so significant that symbols overlap with each other, leading to inter-symbol interference (ISI). This manifests as an increased bit error rate (BER). The SerDes ensures that the symptoms of channel losses and ISI are kept to a minimum.

As the necessary data rates required in data centers continue to increase, the need for high-speed wireline communications continues to grow. As a result, the design of fast, effective SerDes links has become crucial for a wide array of high-speed use cases, including data centers and AI, telecommunications, and automotive applications.

II. OVERVIEW

A. High Level Design Description

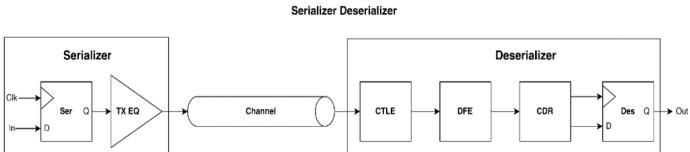


Fig. 1. High Level Block Diagram

The SerDes follows the general signal flow seen in Figure 1. First, the parallel data comes into the serializer, which collates all of the channels, in our case 4, into a continuous data stream at 4x the frequency of each individual channel. Then, the SerDes attempts to equalize for the dispersive effects of the channel using discrete time transmit equalization (TX EQ), which emphasizes the high frequency components of the signal. The signal then passes through the channel, which in our case is modeled to be 50Ω , a typical value used in the real world to standardize high frequency impedance matching. This

channel can be modeled as a set of real and complex poles at different frequencies. The next equalization stage through which the signal passes is the continuous time linear equalizer (CTLE). Unlike the TX EQ, this is a linear filter, using the complex zero response of a source denerated amplifier to once again boost the higher frequencies of the input signal. Following this the signal goes into the last equalization stage, the decision feedback equalization (DFE). This ciruit operates by removing the intersymbol interference (ISI) left in the signal at integer multiples of the unit intervals (in our case, 400ps), after the peak of the data coming in. The equalized signal is sent to the clock and data recovery block, which uses the transitions in the data to generate a precisely matching clock signal. This clock signal is used in both the DFE, and in the next block, the deserializer. This block converts the single data stream back into the four orginal data streams.

B. Key Specifications

Our SerDes takes in 4 input data streams and sends a single data stream over the channel. In order to guide the design of our SerDes, we set several key specifications. The two most critical specifications for our design are the bit rate and the bit error rate (BER). Since most SerDes links are used for high-speed applications, we set our target bit rate at 2.5 Gbps. Additionally, we set our BER target specification to be 10^{-4} . The other important specifications we set for ourselves included a total power consumption below 1W for the Ser and the Des combined, and a data-dependent clock jitter (DDJ) of less than 100 ps rms.

C. Approach + Channel Model

Key to our system design was defining and modelling the channel that our data would be transmitted over. Our goal in determining this was to have a model with significant enough attenuation that all the equalization schemes we wanted to implement would be necessary, but one reasonable enough so that channel equalization was realizable for our desired bit rate.

As a result, we choose to model our channel as a three-pole system, with one, low frequency real pole at 150 MHz, and a complex conjugate pole pair at 800 MHz. The frequency response of our channel, as shown in Figure 2, results in a total attenuation of -55 dB at our desired 2.5 GHz transmitting frequency. These values were also determined to be in accordance with typical backplane channel characteristics, modelling the interconnects between GPUs in a server rack [1].

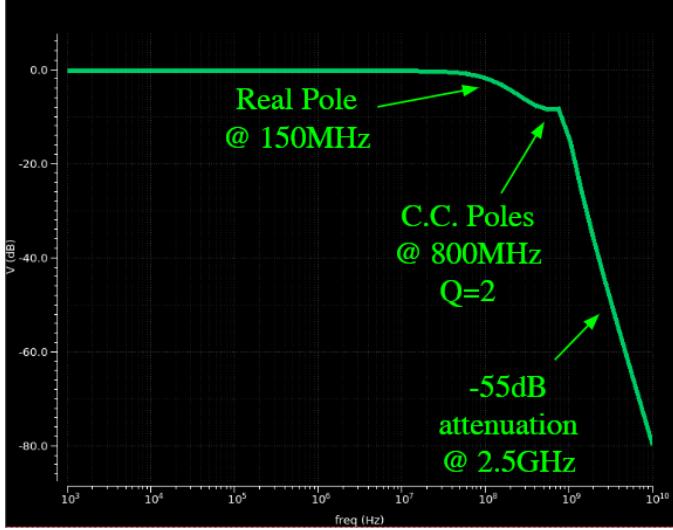


Fig. 2. Channel Frequency Response

Our design approach was to break into teams of two, one working on the equalization schemes, and one working on the CDR and system infrastructure circuitry. This gave us enough parallelism to tackle the large number of circuits required for our project, while still being able to work with a partner to take on complex problems, with multiple eyes on the job.

III. CIRCUIT DESIGN

A. Serializer

The first stage of our system is a serializer, which takes in 4 parallel input data streams at 625Mbps and serializes it into a single 2.5Gbps line. This line then also gets fed into a CMOS to CML level shifter, which creates the differential signal we wish to send over our channel. The serializer is implemented using multiplexers from the ECE4740 Standard Cell Library. This circuit works by using a 4-1 MUX to toggle between the four incoming bit lines. The select signal for this MUX is generated using 2-1 MUXes that choose between VDD and ground based on an input clock signal. Notably, this requires both a clock signal at 625 MHz, as well as a clock signal at double that frequency (1.25 GHz). For the sake of simplicity, we assumed that both of these clocks are generated off chip, but in a more realistic design, it is likely that this higher clock frequency would be generated on chip using a frequency doubling circuit.

Fig. 4 demonstrates the output of the serializer block over four 2.5 GHz clock signals. Over this period, the incoming data received from data sources A through D is 1, 0, 1, and 1, respectively. The serializer successfully outputs 1, 0, 1, and 1 over these four clock cycles.

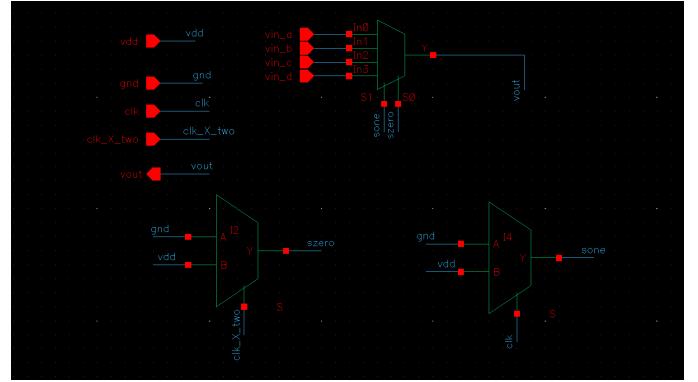


Fig. 3. Serializer Schematic



Fig. 4. Positive Output of Serializer + Level Shifter (in orange) in response to incoming data of 1, 0, 1, and 1

B. Transmit Equalizer and Driver

Before going through the channel, it is desirable to boost the high frequency components of our signal to counteract the expected channel effects. The TX Equalizer pre-distorts the signal by boosting transitions and attenuating lower frequencies. This is realized with a 3 tap FIR filter that increases the signal amplitude during data transitions, ensuring that the received signal is less impacted by the dispersion in our channel. The TX FIR simply has the following H(z) response.

$$H(z) = a_0 + a_1 z^{-1} + a_2 z^{-2}$$

Each tap controls its own differential pair driver, whose tail currents are adjusted to represent tap coefficients. CML flip-flops are used to generate the delays, and current is steered using differential pairs that are loaded by 50Ω . This loading has been modeled with ideal resistors as they may easily be placed off chip (they would not consume additional IO). As we are using current mode drivers, the summing of the FIR taps is handled by the driver itself.

The main constraint in designing our TX driver was ensuring that the multiple differential pairs driving the channel did not limit the bandwidth of our driver such that we were unable to transmit our desired 2.5GHz of signal. This means that the 3dB bandwidth associated with the 50Ω channel and 3 differential pairs was greater than that of our signal. In other

words, the capacitive self-loading of the driver must be small such that:

$$2.5G * 2\pi \geq \frac{1}{50\Omega * C_{self}}$$

This means that the output drivers must be sized minimally to reduce C_{gd} and C_{db} .

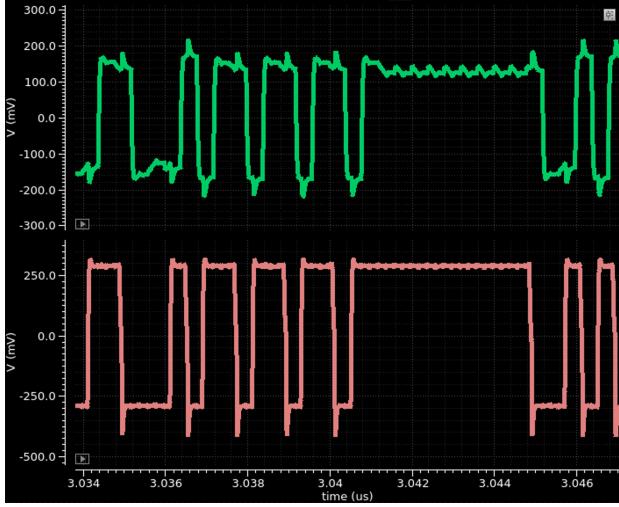


Fig. 5. TX Equalization Transient Simulator

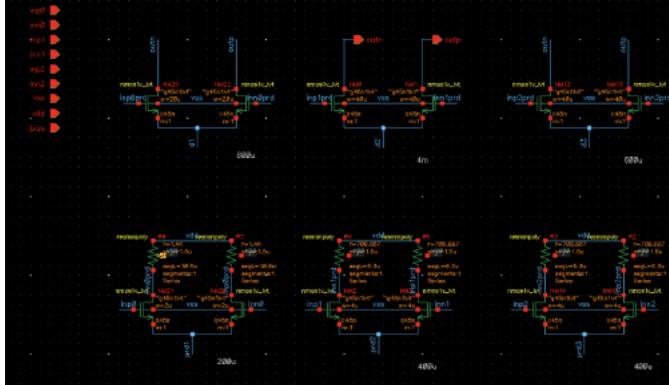


Fig. 6. TX FIR Filter

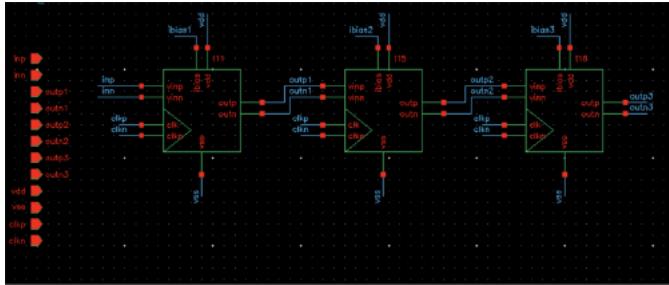


Fig. 7. TX Driver

C. Continuous Time Linear Equalizer (CTLE)

The purpose of the CTLE is to act as a high pass filter to compensate for the low pass response of the channel. It linearly restores the lost high frequency components by using an RC source degenerated differential amplifier.

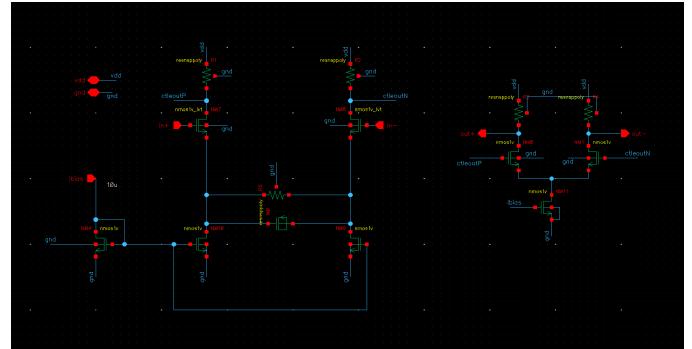


Fig. 8. CTLE Schematic

The frequency response of the CTLE, which was derived using equations for a degenerated CS amplifier, was found to be the following [2]:

$$H(s) = \frac{A_{v0} \left(1 + \frac{s}{z_1} \right)}{\left(1 + R_{deg} C_{deg} \frac{s}{(g_m R_{deg} + 1)} \right) (1 + s R_L C_L)}$$

$$A_{v0} = \frac{g_m R_L}{1 + g_m R_{deg}}$$

This indicates that the location of the zeros can easily be defined with the degeneration capacitance and resistance. The peaking can also easily be found (and is defined) by the location of the zero and the first pole.

$$z_1 = \frac{1}{R_{deg} C_{deg}}, p_1 = \frac{1 + \frac{g_m R_{deg}}{2}}{R_{deg} C_{deg}}$$

$$A_{v,pk} - A_{v,dc} = \frac{p_1}{z_1} = 1 + \frac{g_m R_{deg}}{2}$$

The location of the zero is designed to counteract the primary pole of the channel, and the CTLE pole is pushed as far out as possible to maximize the equalization effect on the channel, as shown below.

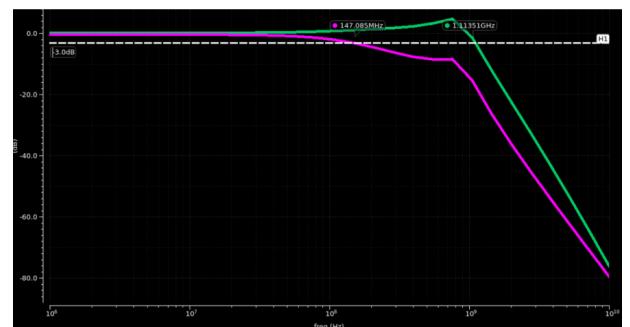


Fig. 9. Channel AC Response (Purple), Channel + CTLE Response (Green)

D. Decision Feedback Equalizer (DFE)

The DFE is a nonlinear equalizer that removes post-cursor ISI by subtracting the known remaining impulse response from previous bits from the incoming signal. The DFE is also able to

remove this without boosting high frequency noise, a major limitation of linear equalizers. The DFE works by making a decision on whether the received bit is a 1 or a 0 with a “slicing” circuit. Then, after a unit interval delay, this decision value is multiplied by a tap weight and fed back into a summing node, where it is subtracted from later bits to delete the remaining effect of the old bit. Our implementation uses a 3-tap filter, which we found compensates plentifully for the postursors in our channel. The functional discrete time output of the DFE is given by $y[n]$ in the following equation:

$$y[n] = u[n] - \sum_{k=1}^N w_k \cdot d[n-k]$$

Where $u[n]$ is your incoming signal from the CTLE, $d[n-k]$ and w_k are the decision and weight of the k th previous bit, and N is the number of taps [3]. The delays are implemented using CML flip flops, and then current is either steered to or away from the signal using FIR taps implemented similarly to the TX FIR filter.

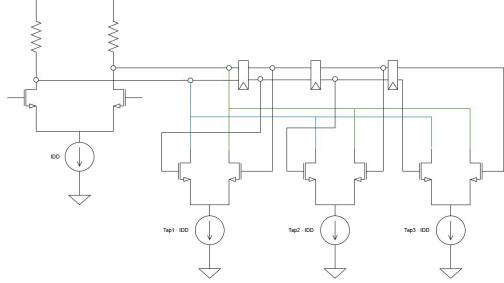


Fig. 10.3 Tap DFE CML Implementation

This imposes a critical timing constraint we must meet to ensure that our decision gets fed back and acted upon in time. If this timing were not to be met, we would subtract the wrong value from the wrong bits, significantly distorting our desired signal. The critical path is defined by the first feedback loop and can be written followed:

$$T_{clk \rightarrow Q} + T_{feedback} + T_{setup} \leq T_{bit}$$

In fact, this constraint is even more stringent, as there needs to be enough time for the summing node to settle at the desired voltage (meaning several τ 's must pass), so the more realistic constraint to shoot for is for the feedback loop to take 25-50% of a unit interval (400 ps).

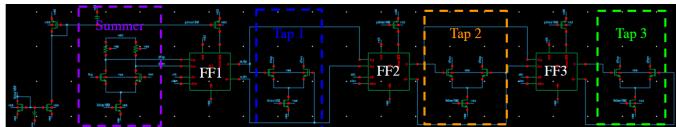


Fig. 11. DFE Schematic

The goal of the DFE is to make the voltage sampled for a zero value to be as close to the nominal zero voltage as possible following a unit length 1 impulse. Our tuned DFE impulse

response is shown. It is clear in Fig. 10 that voltage of the signal at integer unit intervals after the main cursor is within 3% of the nominal zero voltage.

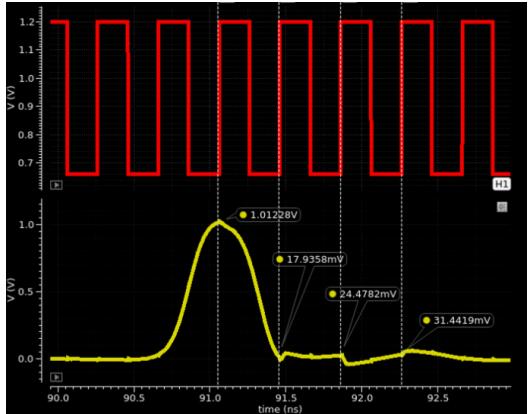


Fig. 12. Post DFE Pulse Response

E. Clock and Data Recovery – Overview

For the DFE block and deserialization process to work effectively, a lock signal that precisely matches the speed at which the data is arriving at is needed. Since the target system, which aims to model most systems (ethernet, USB etc.), does not send a clock along with its data, it is plesiochronous. Here, the local reference oscillator is not a reliable source of timing information for the arriving data, as this oscillator frequently can drift by dozens of MHz (assuming a 50MHz $\pm 50\text{ppm}$ oscillator) [4]. In our design, this reference oscillator oscillates with a frequency of 2.512 GHz. Instead, timing information needs to be derived from the edge information existing in the transitions in the data.

It is important to note that since the data is random, the data does not have a well-defined frequency component to it, and thus, a typical phase-frequency detector (PFD) cannot be used. The PLL architecture that was deemed to be suitable for this application was a dual-loop referenced architecture using a Bang-Bang/Alexander phase detector. At a high level, this architecture uses a current-controlled oscillator (ICO) to generate a reference clock. This signal is then compared against the transitions in the data stream using this bang-bang phase detector – if the clock transitions before the data does, the phase detector sends out an early signal, and if the data transitions first, it sends out a late signal. If the clock is early, then the ICO needs to oscillate slower, meaning it needs less input current, and if the opposite is true, then the ICO needs a greater input current. This change in input current is accomplished through a charge pump and loop filter, which either sources or sinks a fixed amount of current based on the output of the phase detector, and that current gets transformed into a smooth voltage by the loop filter. The output of this loop filter is fed into a transimpedance amplifier, which finally feeds back into the ICO.

As the current controlled oscillator is very sensitive (a small change of input current results in a large output frequency change), the control of the ICO was broken into two paths: a fine path and a coarse path. This makes the loop more well behaved as the K_{ICO} (current to frequency gain) can be reduced for the fine loop.

F. Fine and Coarse Control Loops

Having two separate paths yields a significant drawback, however. The fine path can only make fine adjustments to the ICO, making lock times poor. What exacerbates this issue is that the Bang-Bang phase detector typically has quite a small lock-in range (around the bandwidth of the loop filter), which may prevent phase lock from occurring [5][6]. Although the reference oscillator cannot be used to determine timing information, we are able to depend on it having a frequency *close* to that of the incoming data. Thus, a dual loop architecture with two matched ICOs can be employed as the coarse loop can be locked to a reference, which brings the fine loop close to its locked operating point. A block diagram of this configuration is shown in the figure below:

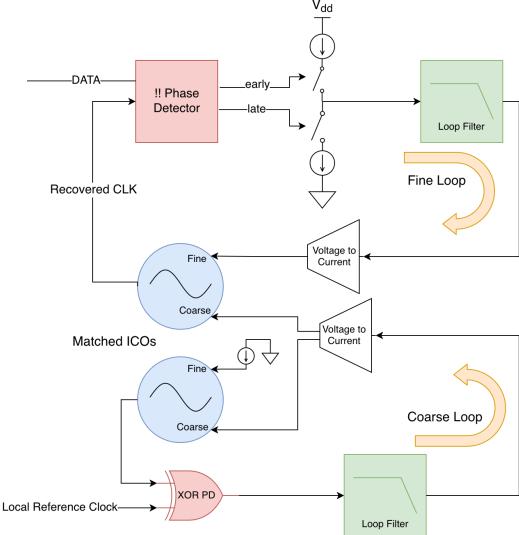


Fig. 13. Complete CDR Architecture

G. Phase Detector

In the fine loop, a Bang-Bang phase detector is used. A variety of phase detectors exist in the SerDes space; two common ones are the Linear/Hogge phase detector and the Bang-Bang/Alexander phase detector [5]. A linear phase detector will linearly feedback the “lateness” of the recovered clock to the data, whereas a Bang-Bang phase detector will only signal “early” or “late”. Although the linear phase detector is simpler to analyze (as the loop should ideally remain linear and typical stability analysis can be employed), a few significant drawbacks of this phase detector include the requirement of a limiting amplifier (to “rail” the input signal from the dispersive channel), and sensitivity to delay within the phase detector itself (which would give rise to dead-zones) [5]. The input of the Bang-Bang phase detector is CML latches, which have

regenerative gain, preventing the need for a limiting amplifier. A schematic of the bang-bang phase detector is shown below.

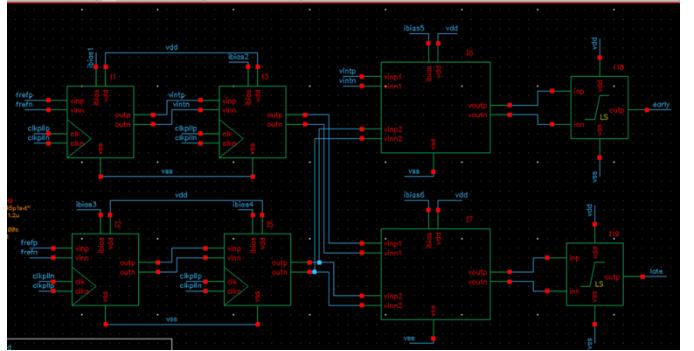


Fig. 14. Phase Detector Schematic

H. Charge Pump and Loop Filter

The role of the charge pump in the PLL is to either source or sink a set amount of current in response to a late or early signal from the phase detector. Thus, the loop filter architecture consists of an NFET and a PFET current mirror, each connected in series to another MOSFET that serves as a switch. This effectively allows the charge pump to source a current if the PFET switch is closed and the NFET switch is open and sink a current if the opposite is true. If both switches are either closed or open, then the charge pump should not source or sink any current. Thus, the current mirrors were tuned to input/output near identical currents when on. Additionally, the current mirrors were placed directly connected to the output, meaning that the switches could be placed on the outside of these current mirrors. This reduces any charge injection that occurs via C_{gd} of the transistors, which reduces the amount of extraneous current that enters the loop filter [7].

As the CDR loop is strongly nonlinear and discrete time (from the behavior of the phase detector), it is difficult to analyze the behavior of the loop filter. However, a typical loop filter design consists of the network shown below [7].

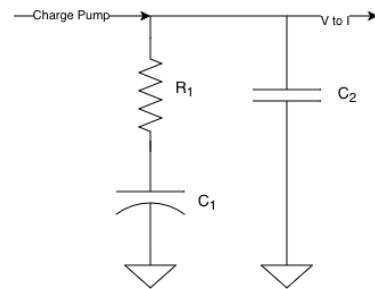


Fig. 15. Loop Filter

Although we did not use analytic equations to determine the loop filter poles and zero due to the difficulty in analyzing the loop, we used the following equation to provide intuition regarding the loop design.

$$\frac{V(s)}{I(s)} = \frac{1 + sRC_1}{s(C_1 + C_2) + s^2RC_1C_2}$$

As the loop filter has a current input from the charge pump and a voltage output, it acts as an integrator (pole at $\omega = 0$). The R in series with C_1 also places a zero in the transfer function, helping stabilize the loop. We chose the coarse loop bandwidth to be larger than the fine loop bandwidth to decrease the lock time of the coarse loop. We chose the fine loop bandwidth to be smaller to reduce the effects of data dependent jitter [5].

As the values of the R and C are very large (dozens of pF), we opted to put these off-chip and thus model these components as ideal components from AnalogLib.

	Coarse Loop	Fine Loop
R_1	$4k\Omega$	$1.5k\Omega$
C_1	$4pF$	$15pF$
C_2	$50fF$	$500fF$

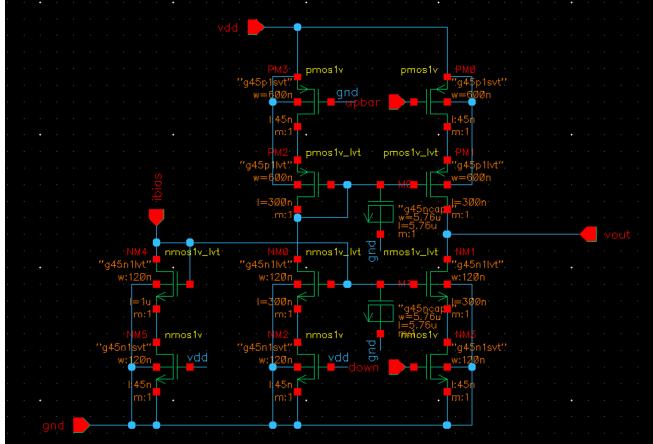


Fig. 16. Schematic of Charge Pump

Charge Pump Currents		
Down	Up	Current
0	0	18.9uA
0	1	257pA
1	0	153nA
1	1	1nA

Fig. 17. Output current of Charge Pump

I. Transconductance Amplifier

This amplifier linearly converts the output of the loop filter to a current for the current controlled oscillator while maintaining a high impedance at the loop filter node (to prevent loading). The design of this amplifier was simply a differential

amplifier, loaded by diode-connected NFETs to form an active current mirror output. Additionally, the differential pair uses source degeneration to increase linearity. The amplifier design uses current mirrors to replicate the current through the differential pair; thus, the output current should just be defined by the input voltage and the tail current.

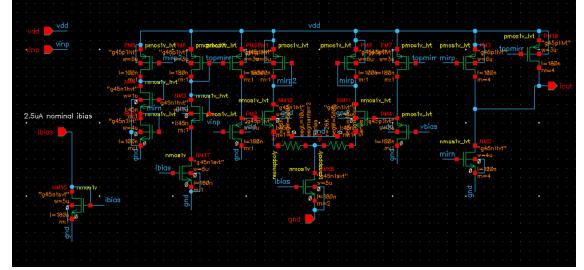


Fig. 18. Schematic of OTA

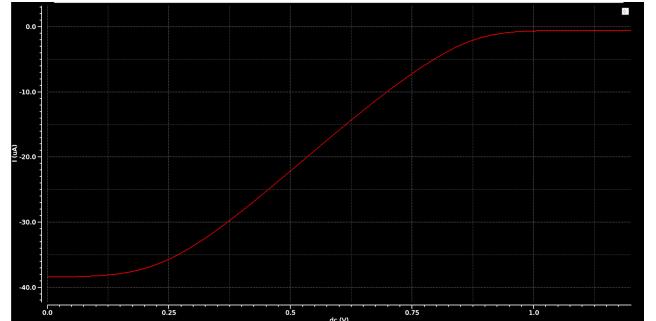


Fig. 19. Voltage to Current relationship of OTA

J. Current Controlled Oscillator (ICO)

The purpose of the ICO is to generate a controlled oscillating clock signal. In wireline applications (which is typically the case for SerDes links), the phase noise is less of a concern, and the tuning range is a larger concern. Thus, we opted to use a ring oscillator topology over an LC oscillator topology, as it has better phase noise and worse tuning range.

The ICO uses a series of three oscillators in feedback to form a standard current starved ring oscillator topology. Two of the three inverters are current-starved, with the exact amount of current set by the sum of the coarse PLL loop and the fine PLL loop. One inverter is not current starved to allow the oscillator to have a larger tuning range. The output of the ring oscillator is AC coupled to a series of two inverters, which are used to generate 50% duty cycle outputs and to increase the fanout of the oscillator. This also ensures that the output of ICO can have CMOS levels.

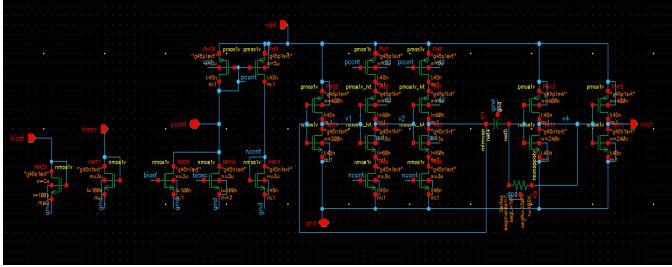


Fig. 20. Schematic of ICO

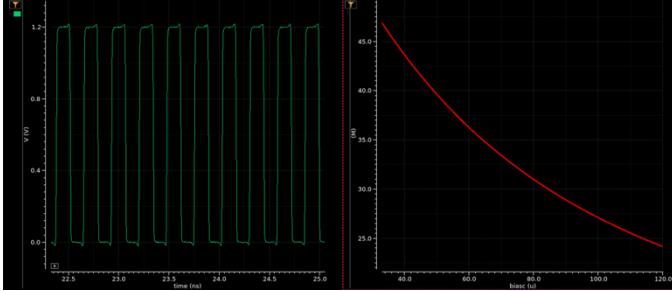


Fig. 21. Output of the ICO (Left), KICO vs Bias Current (Right)

K. Deserializer

The last critical block of this SerDes system is the deserializer, which takes in the fully equalized bit stream and converts it back into the original four-bit streams. The deserializer has two additional valid bit outputs, which identify which data stream output is valid to sample at any time. This circuit first uses a level shifter to convert the CML data back to CMOS. It then employs a 1-4 demultiplexer to separate the data back into four outputs. In order to generate the select signal for this demultiplexer, two simple frequency divider circuits are used, which generate clocks at $\frac{1}{2}$ and $\frac{1}{4}$ the frequency of the PLL-generated clock. The 2-1 multiplexers toggling between VDD and ground also use these $\frac{1}{2}$ and $\frac{1}{4}$ clocks to appropriately match the valid bit signal to the output of the demultiplexer. The results of this are shown in figure 19. As soon as the two valid bits read 00, the output of the deserializer (in purple) matches the incoming data (green) for the duration that the valid bits read 00.

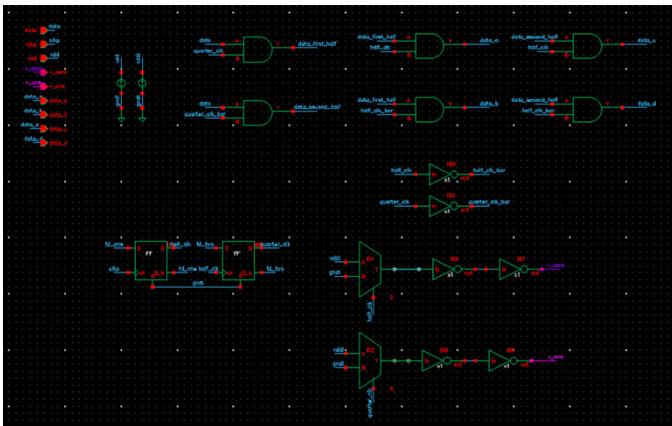


Fig. 22. Deserializer Schematic

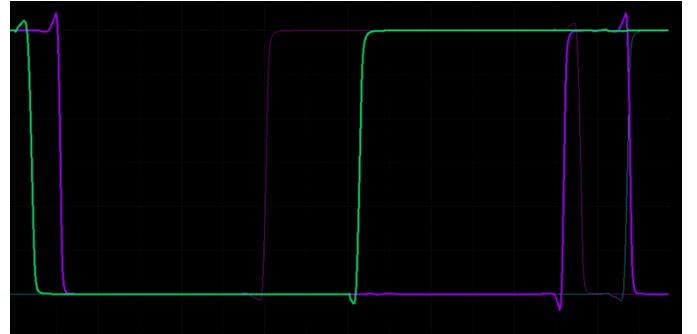


Fig. 23. Deserializer output (green) compared to incoming data (purple)

L. Infrastructure Circuitry

Some circuitry was essential in providing the building blocks or global infrastructure to make our design work; they did not get individual sections, but their designs can be found in the appendix. The designs include a global current mirror to provide bias currents needed in almost every circuit. They also include level shifters to interface between the CML and CMOS portions of our design. Several digital blocks were designed as well, including a CML flip-flop (used by the FIR filters in both the DFE and transmitter, the bang-bang phase detector, and the deserializer) and a CML XOR gate.

M. Overall Results

Full system testing was done to test the functionality of each block. The full schematic diagrams for the transmitter and receiver are shown below.

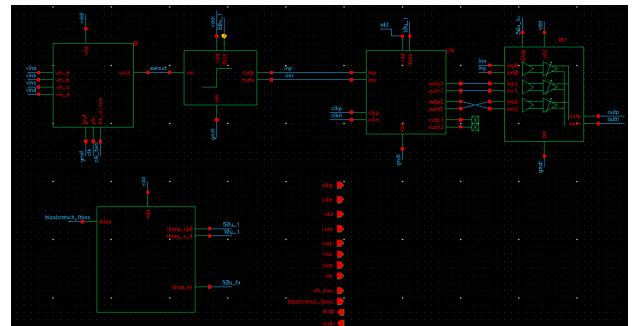


Fig. 24. Full Transmitter Schematic

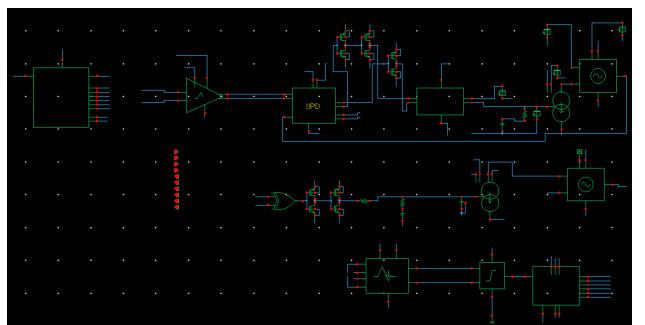


Fig. 25. Full Receiver Schematic

Testing the validity of the transmitter and receiver involved verifying the functionality of the CDR system, as well as the impact of each equalization step on the signal integrity, when all blocks were connected and random input data was received. First, the total clock frequency of the ICO should hover around 2.5 GHz. A plot of the frequency vs time of the ICO is shown below, showing the CDR locking onto the desired 2.5GHz frequency, confirming locking functionality even with ~20 MHz offset from the reference clock.

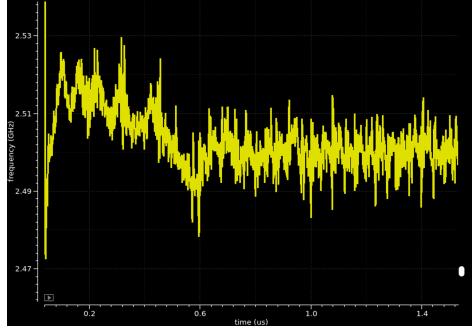


Fig. 26. ICO Frequency vs Time

Additionally, the jitter from this clock signal can be obtained from the frequency vs time plot. This jitter was measured to be 12 ps rms.

To measure signal integrity, eye diagrams were used after the channel, after transmit equalization, after CTLE, and after DFE, and are shown below.

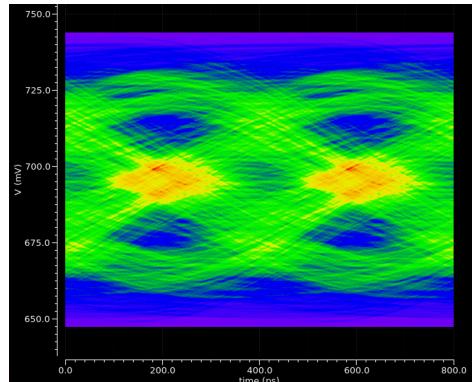


Fig. 27. Eye Diagram After Channel

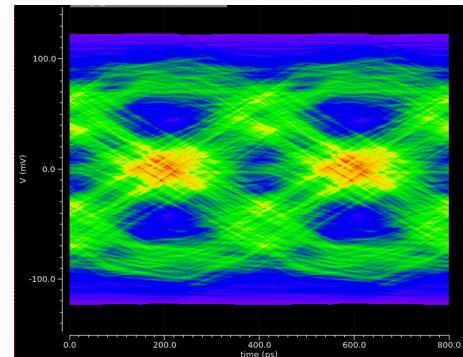


Fig. 28. Eye Diagram After TXEQ

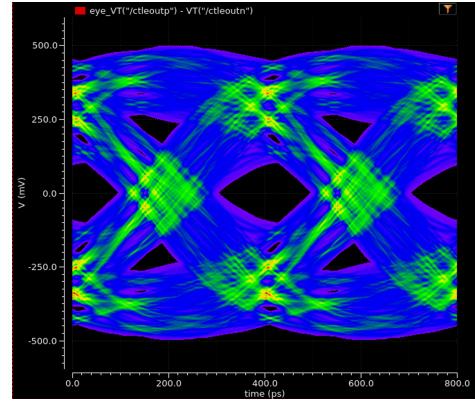


Fig. 29. Eye Diagram After CTLE (Height: 67 mV, Width: 164 ps)

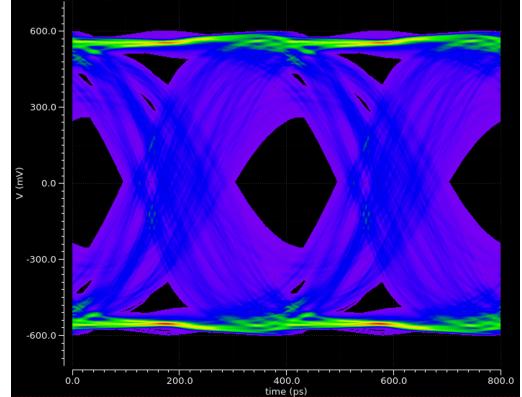


Fig. 30. Eye Diagram After DFE (Height: 458.5 mV, Width: 158.9 ps)

As shown through the progression of eye diagrams, each equalization step significantly improved signal integrity. In total, our eye diagram after the DFE is open enough to ensure a BER of 706×10^{-12} assuming a sampling clock jitter of 12 ps.

Finally, the power consumption of the transmitter and receiver sides combined was measured to be 36.67 mW.

N. Layout

The final step of this project involved laying out both the transmitter and receiver side. These layouts are shown below:



Fig. 31. Layout of Reciever Chip (428.6x503.9 μm^2)

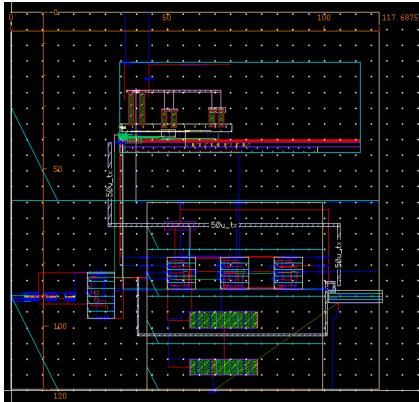


Fig. 32. Layout of Transmitter Chip (120x117um²)

O. Conclusion

In summary, this project involved the design of a 2.5 Gbps SerDes link with both decision feedback equalization and clock and data recovery. This project was successful in meeting all of our desired specifications, and a table summarizing our end results is provided below:

System Goals and Achieved Results		
Specification	Goal	Achieved
Data Rate	2.5Gbps	2.5Gbps
Power Draw	1W	36.67mW
CDR Clock Jitter	100ps	12ps
Bit Error Rate	10^{-4}	$0.7 \cdot 10^{-9}$

While we were largely successful in our design, there are a number of key improvements that could be made in the future to make this chip viable to tape-out. One major improvement could be the analysis of noise; no noise analysis was done during this design process, and since many of our systems are very sensitive to voltage and current (notably, the CDR system), designing with noise in mind would be crucial to ensure that this SerDes would work if it were to be taped-out. Additionally, PVT simulations would need to be conducted to verify the robustness of the SerDes design. Finally, running extraction after layout would be crucial to ensure that this SerDes design would be fully functional.

P. Discussion

Overall, this project proved extremely challenging. The complexity of the circuits and the large number of different blocks, coupled with tight time constraints, made accomplishing our initial goals a hugely difficult task.

There were a number of important takeaways and surprises from this project, including but not limited to:

- The difficulty of tuning and analyzing transient-based circuits
- The number of disagreements or inconsistencies in literature regarding design techniques
- The use cases of logic families other than CMOS
- The time needed to simulate large, complicated designs

One specific unexpected phenomenon occurred during the design of the CDR circuitry. When the PLL locked, there were steady-state oscillations. Initially, this was believed to be the loop filter going unstable, but no loop filter parameters seemed to remove these oscillations. After a literature search, this appeared to be typical operation in strongly nonlinear phase detector circuits [8].

Another unexpected phenomenon was the TX equalizer increasing the data dependent jitter. Although we were unable to find a definitive answer for this, we believe it is the TX equalizer interacting with the CTLE poorly. Namely, we believe that it is the TX equalizer attempting to cancel the same real pole as the CTLE, resulting increased ISI. Thus, we chose the TX taps to only lightly equalize the transmit response.

REFERENCES

- [1] “Sam Palermo – ECEN 720: High-Speed Links Circuits and Systems.” Available: <https://people.engr.tamu.edu/spalermo/ecen720.html>
- [2] B. Razavi, “The Design of an Equalizer—Part One [The Analog Mind],” IEEE Solid-State Circuits Mag., vol. 13, no. 4, pp. 7–160, 2021, doi: 10.1109/MSSC.2021.3111426.
- [3] B. Razavi, “The Design of an Equalizer—Part Two [The Analog Mind],” IEEE Solid-State Circuits Magazine, vol. 14, no. 1, pp. 7–12, 2022, doi: 10.1109/MSSC.2021.3126997.
- [4] B. Razavi, “Challenges in the design high-speed clock and data recovery circuits,” IEEE Communications Magazine, vol. 40, no. 8, pp. 94–101, Aug. 2002, doi: 10.1109/MCOM.2002.1024421.
- [5] B. Razavi, *Design of integrated circuits for optical communications*, Second edition. Hoboken, N.J: Wiley, 2012.
- [6] K. Park and D. Jeong, “Analysis of frequency detection capability of Alexander phase detector,” Electronics Letters, vol. 56, no. 4, pp. 180–182, Feb. 2020, doi: 10.1049/el.2019.3488.
- [7] “EE 290C: High-Speed Electrical Interface Circuit Design (Spring 2011, UC Berkeley).” Available: <http://infocobuild.com/education/audio-video-courses/electronics/EE290C-Spring2011-Berkeley/>
- [8] A. Teplinsky, R. Flynn, and O. Feely, “Limit cycles in bang-bang phase-locked loops,” in *2006 IEEE International Symposium on Circuits and Systems*, Island of Kos, Greece: IEEE, 2006, p. 4. doi: 10.1109/ISCAS.2006.1693524. Available: <http://ieeexplore.ieee.org/document/1693524/>. [Accessed: Dec. 20, 2025]
- [9] H. Wang and J. Lee, “A 21-Gb/s 87-mW Transceiver With FFE/DFE/Analog Equalizer in 65-nm CMOS Technology,” IEEE Journal of Solid-State Circuits, vol. 45, no. 4, pp. 909–920, Apr. 2010, doi: 10.1109/JSSC.2010.2040117.
- [10] T. Beukema et al., “A 6.4-Gb/s CMOS SerDes core with feed-forward and decision-feedback equalization,” IEEE Journal of Solid-State Circuits, vol. 40, no. 12, pp. 2633–2645, Dec. 2005, doi: 10.1109/JSSC.2005.856584.

IV. APPENDIX

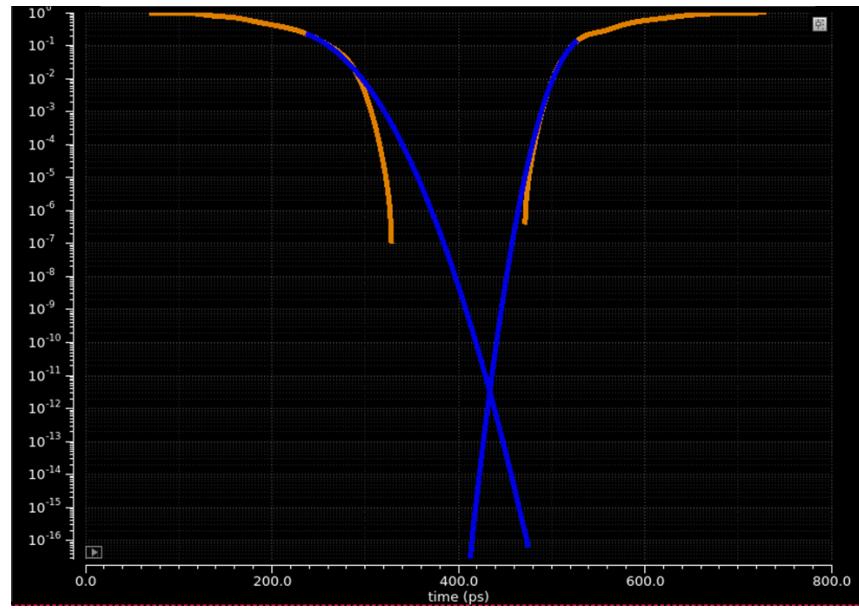


Fig. 33. Bit Error Rate Curve Post DFE (BER vs. Sample Time)

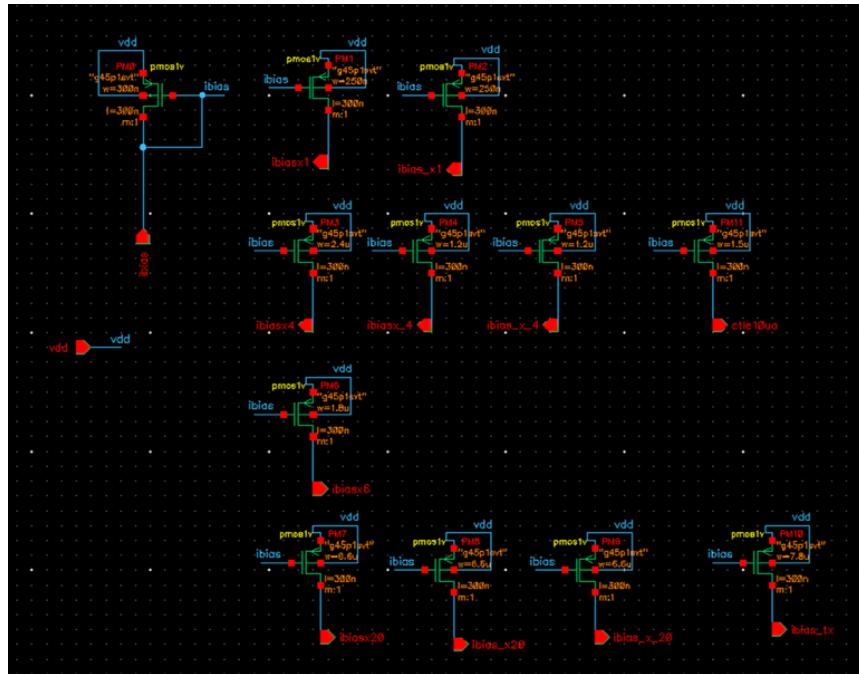


Fig. 34. Global current mirror Schematic

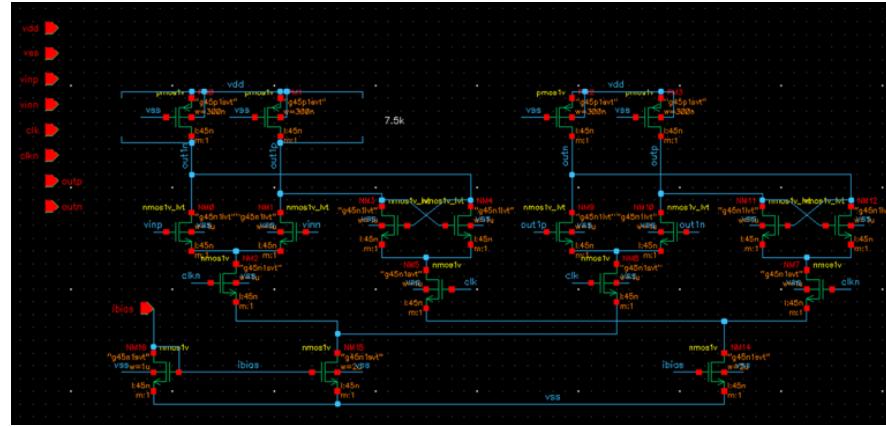


Fig. 35. CML Flip Flop Schematic

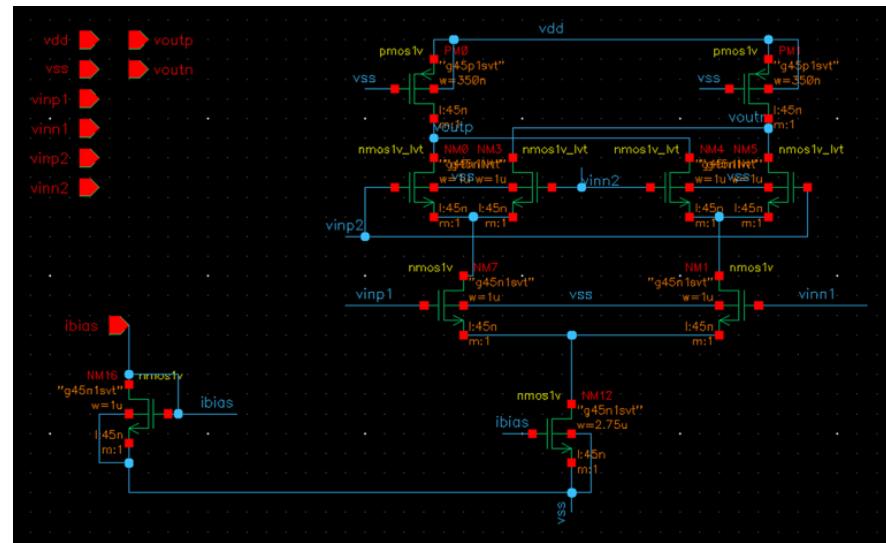


Fig. 36. CML XOR Schematic

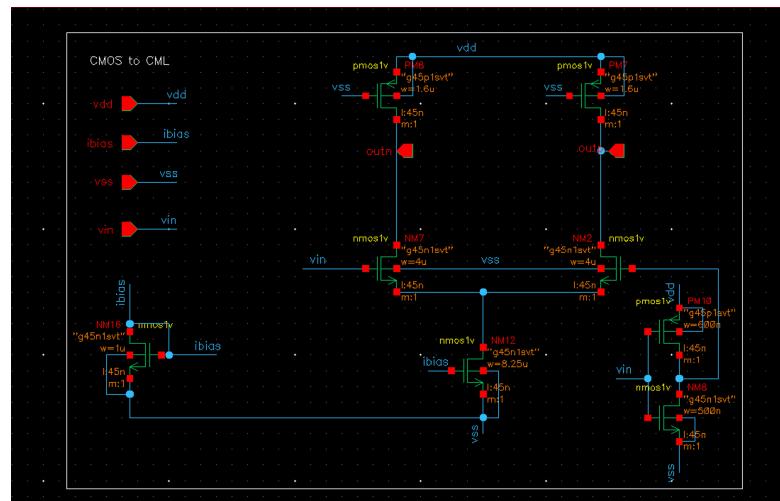


Fig. 37. CMOS to CML Level Shifter Schematic

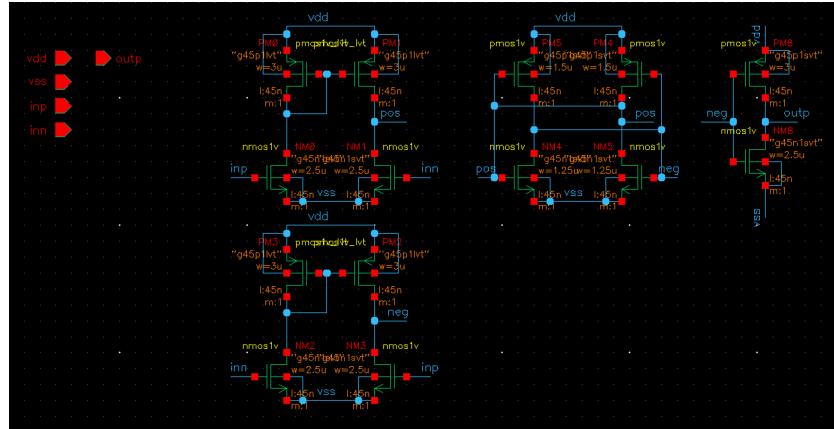


Fig. 38. CML to CMOS Level Shifter Schematic

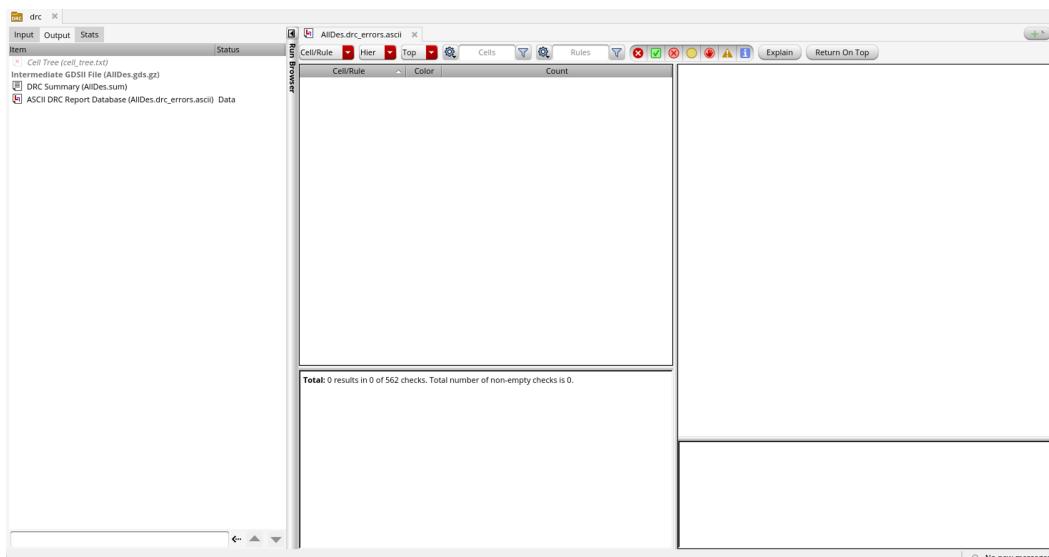


Fig. 39. Deserializer DRC Clean

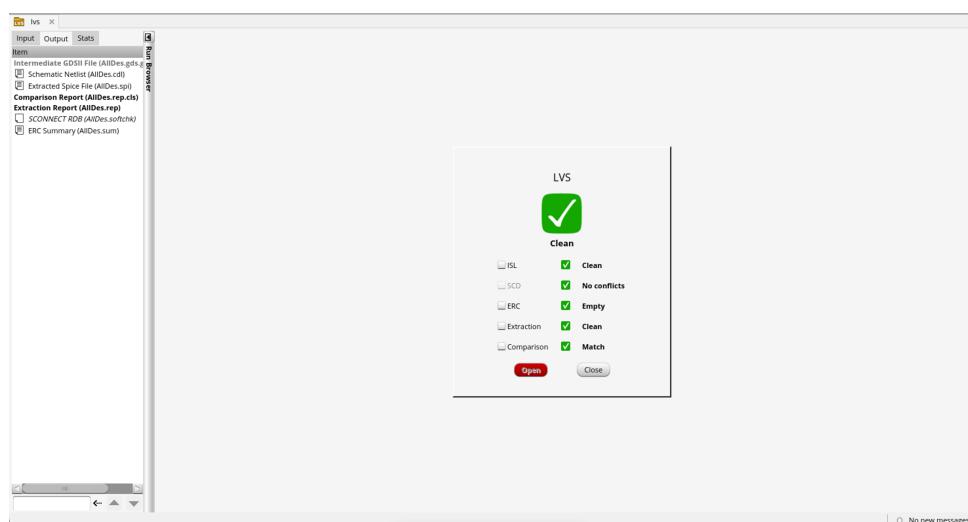


Fig. 40. Deserializer LVS Clean

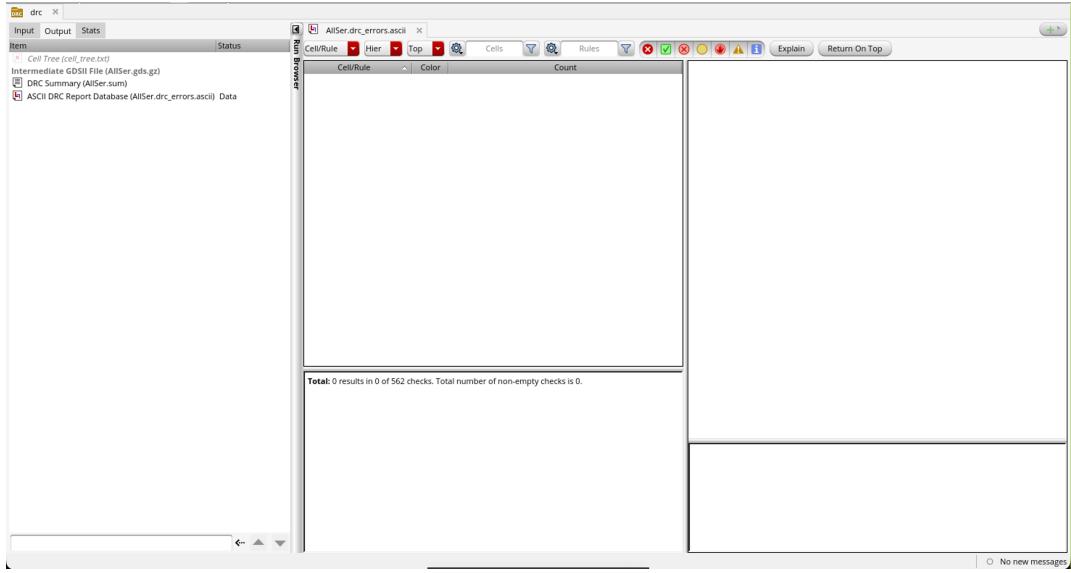


Fig. 41. Serializer DRC Clean

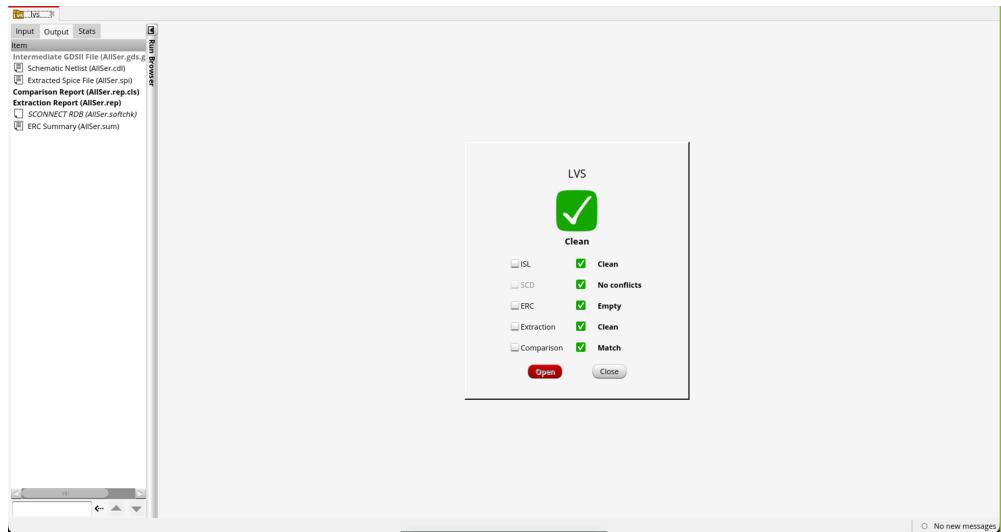


Fig. 42. Serializer LVS Clean