EE 437: Designing a 12.5 Gb/s Wireline Transceiver in GPDK45 and FreePDK45

Eugene Liu, *Student Member, IEEE*, Alex Knowlton, *Undergraduate Student, UW*, Matthew Chen, *Undergraduate Student, UW*, Tyler Rahberger, *Undergraduate Student, UW*, Michael Okerlund, *Undergraduate Student, UW*, Diego Cervantes, *Undergraduate Student, UW*

I. Introduction

The University of Washington EE 437: Integrated Systems Spring 2023 capstone project tasked our group with a 12.5 Gb/s wireline transceiver (loosely based on the High-Speed Serial I/O Made Simple reference guide [1]), to be designed within Cadence Virtuoso using both GPDK45 [2] and FreePDK45 [3]. The underlying concept of this course is to learn how data can be transmitted at high-speed through a channel (whether a copper wire or optically). Modern SerDes applications may call for speeds beyond 100 Gb/s, and cutting-edge technology may even call for throughput as high as 400 Gb/s and beyond. For high-speed data to be transmitted through a lossy channel, signals must be filtered, equalized, and properly timed so that they can be understood coherently by the receiver and sent on to their destination.

This capstone incorporated previous undergraduate course material into a single project. While the integrated systems concentration focuses mainly on analog design, this project also requires an understanding of digital circuits and timing, as the result is mixed signal. The transmitter uses NRZ modulation and includes channel equalization circuits.

II. Component Design

The transceiver itself includes a serializer, a feedforward equalizer (FFE) with current mode logic (CML) drivers, a channel model, a continuous time linear equalizer (CTLE), Strongarm comparators, and a deserializer, as shown in Fig. 1.

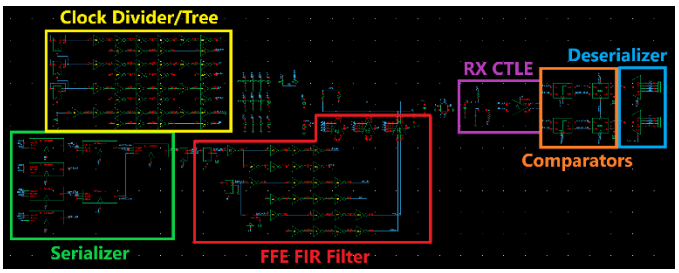


Fig. 1. Block diagram of the entire wireline transceiver. Components include the serializer, FFE, channel model, CTLE, comparator, and deserializer.

Note that clock generation or recovery was beyond the scope of this project. Instead, ideal clocks were generated and then divided or propagated as necessary using CMOS logic and buffers.

A. Serializer (Diego)

In the serializer, this project employed a double edge-triggered flip flop (DETFF) [4] as depicted in Fig. 2. Each DETFF consists of two distinct types of flip flops: a positive edge triggered FF and a negative edge triggered FF.

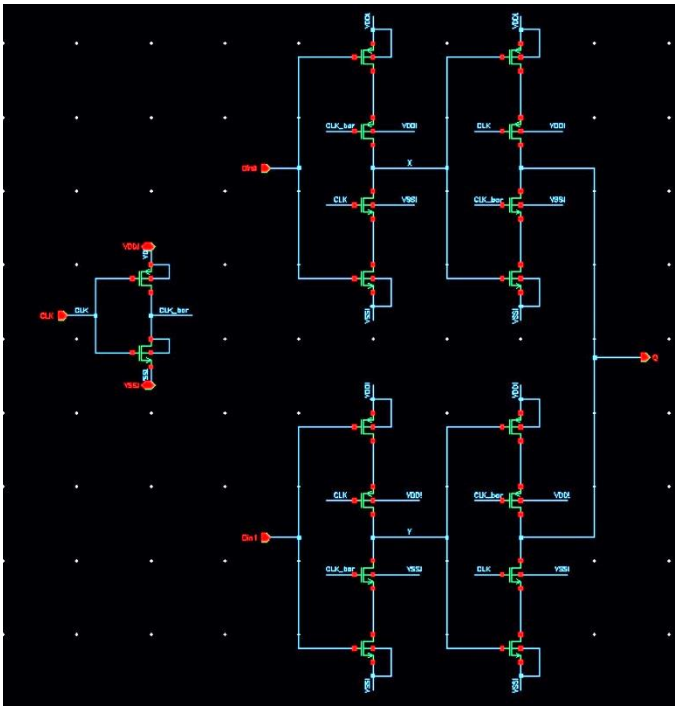


Fig. 2. Schematic of the DETFF with positive and negative edge flip flops.

A simulation for a single block, i.e. a 2:1 serializer simulation was done to test the DETFF. The simulation results depicted in Fig. 3 show an odd dip in data output, but the clocking output remains correct.

Fig. 3. Simulation results for a 2:1 DETFF serializer.

The serializer design found in Fig. 4 includes clock dividers buffered by inverters for logical effort. TSPC flip flops [5], Fig. 6, were used in the clock dividers themselves, as shown in Fig. 5, to make the operation faster.

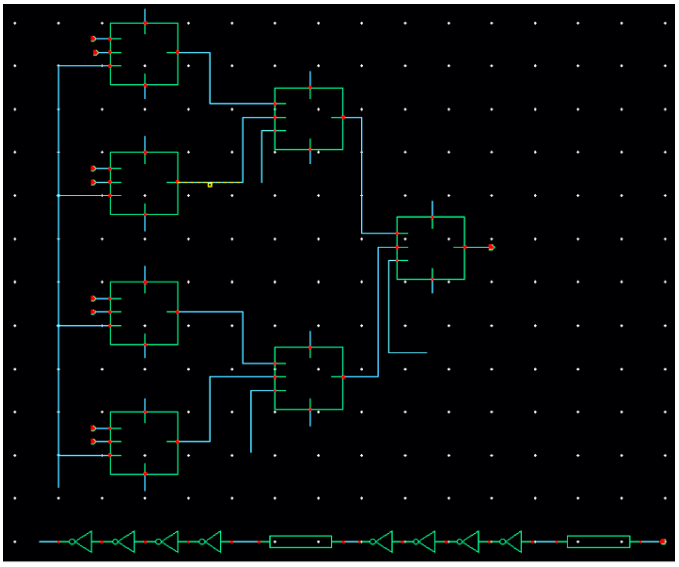


Fig. 4. Schematic of the serializer and clock dividers accompanied by buffers.

The parallel data inputs have been rearranged by the serializer design, resulting in the serialized data sequence at the output of the serializer being obtained as D1, D2, D3, D4, D5, D6, D7, D8.

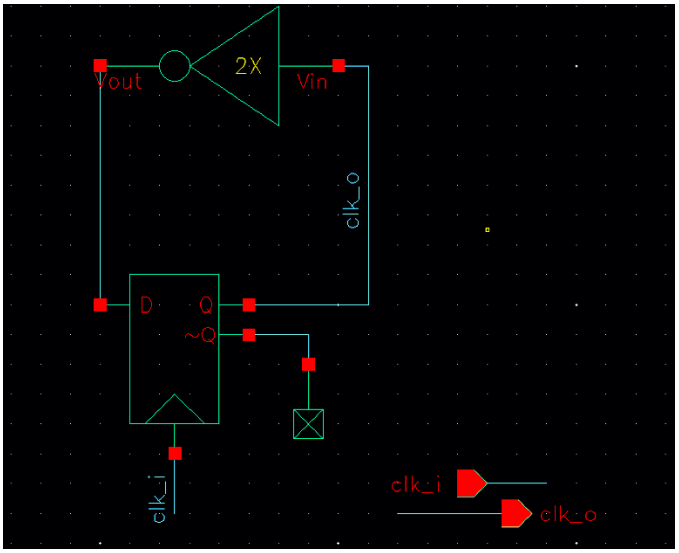


Fig. 5. Schematic of the clock divider. One TSPC FF and a single 2X inverter make up the clock divider.

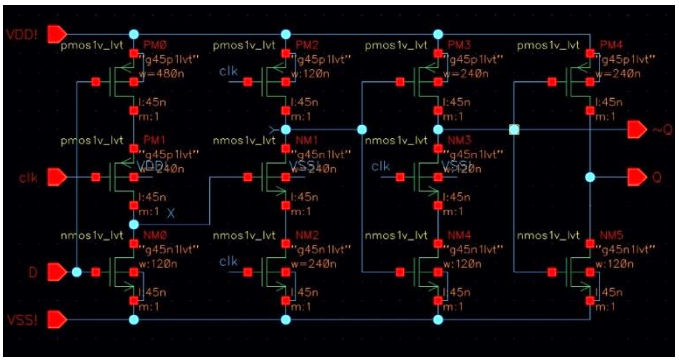


Fig. 6. Schematic of the TSPC flip flops that can operate at 12.5 GHz.

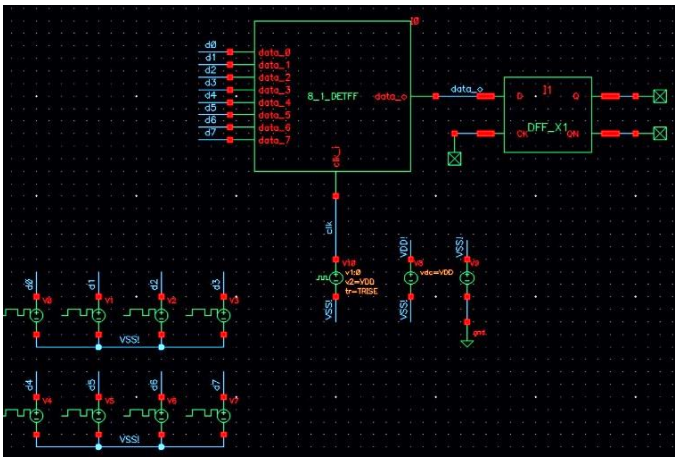


Fig. 7. Schematic of the serializer ready for simulation.

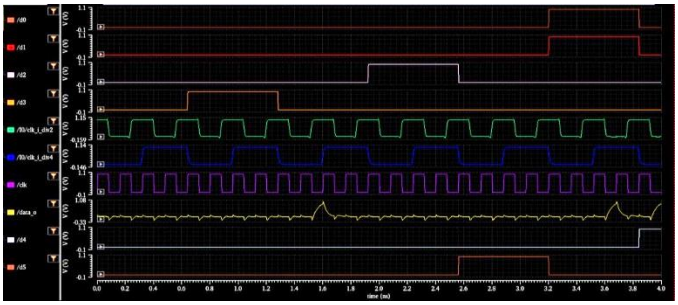


Fig. 8. Simulation results for the 8:1 serializer.

Tree-style Serializer

A second option for implementing the serializer was to use a tree-style serializer (Fig. 9), where two TSPC flip-flops were followed by a 2:1 tri-state mux and a negative latch. The tri-state mux was chosen for its faster speed, and the two serializers canceled out the inversion.

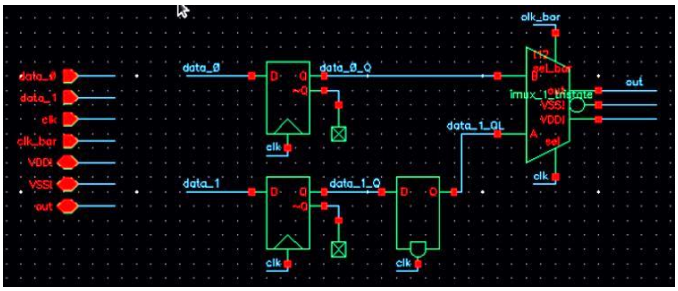


Fig. 9. Single block of the tree-style serializer. This block consists of two TSPC FFs, one negative latch, and a single tri-state mux.

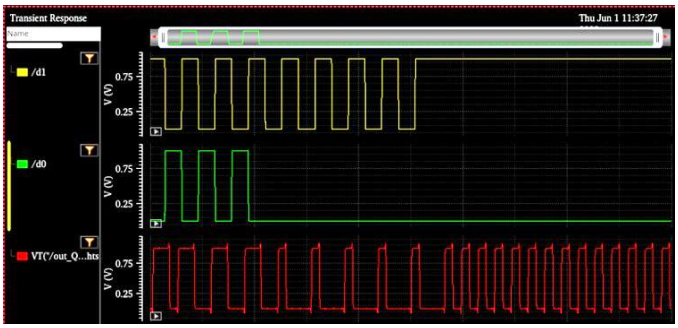


Fig. 10. Simulation results for a 2:1 tree-style serializer.

B. Feedforward Equalizer (Alex and Michael)

The FFE was a 3-tap FIR filter constructed from current mode logic (CML) drivers. The strength of each driver is controlled with 4 bits and are marked as ports tap_0 to tap_3 in the schematic of the CML driver (Fig. 11).

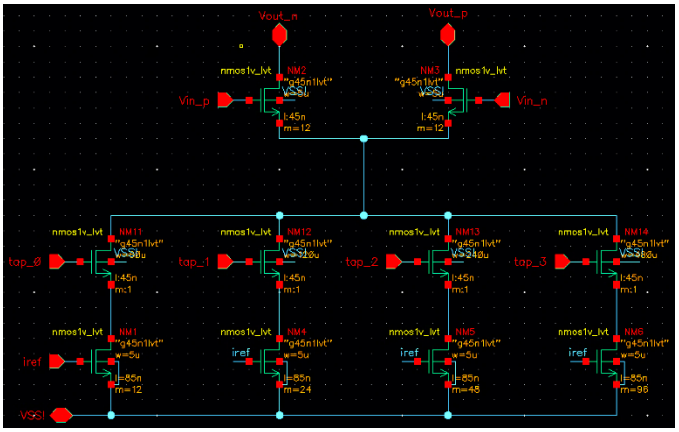


Fig. 11. Schematic of the 4-bit CML driver used as taps in a 3-tap FFE path. The use of multipliers creates the desired overall width.

The configurations of the FFE and the CTLE were optimized simultaneously in MATLAB using several parametric sweeps. 4-tap and 3-tap FIR filters were examined, as well as the number of precursor taps. Ultimately, a filter with 3-taps and no precursor was shown to be best in Fig. 12, which compares the optimal eye opening of different configurations at specific CTLE settings of DC gain and peaking gain.

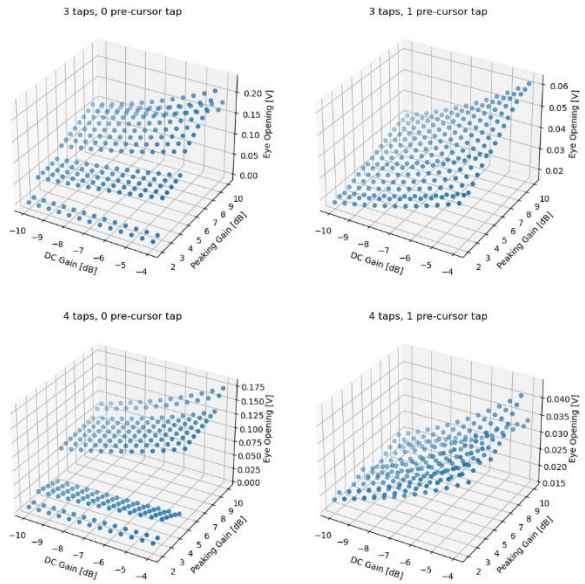


Fig. 12. FFE and CTLE optimization plots. Each 3-D plot is of a different FIR filter configuration (# taps, # precursors). The horizontal axes specify the CTLE's characteristics (DC gain and peaking gain), while the vertical axis shows the eye opening with the FIR filter optimized to that CTLE. The FIR filter with 3-taps and 0 precursor tap produces larger eye openings than the rest.

The driver design called for the use of a differential pair [6] (similar to the design discussed in lecture). In a typical setup, a differential pair might only have a single tail device. However, the desired strength-control for this circuit required the use of four tail currents, sized at 1x, 2x, 4x, and 8x of the unit transistor size (5µm width and 45nm length). Each tail current is accompanied by a like-sized NMOS switch, which controls current strength.

There were three somewhat "fixed" design considerations for the driver:

- The first was that the differential pair and tail currents had to operate in the saturation region. The CML driver has an output swing that is reduced by $2V_{DSAT}$ across the differential pair and tail currents.
- The second was the general idea that transistor sizing should be kept as small as possible. Bandwidth considerations of a 12.5Gb/s design required the use of smaller transistor sizes to both reduce the effect of parasitic capacitance, and to allow the differential pair to react quickly to the incoming data.
- The third was the current mirror ratio. With a reference current of 100 µA, and a desired current of about 10 mA, a width-scaling factor of 1:100 was the starting point for the tail devices. There was some leeway here, but the unit transistor multiples of 1x, 2x, 4x, and 8x had to be enforced so that the strength control worked properly.

The initial DC simulations showed that all devices were in the triode or sub-threshold operating region. A large V_{DS} across the differential pair, and discrepancies in the V_{TH} between devices were the main causes of these biasing issues. Keeping the "fixed" design parameters in mind, the only reasonable option to reduce the differential pair V_{DS} was to increase the device widths. Increasing the size of the differential pair was somewhat undesirable as this would reduce the high-frequency performance, but it was necessary for biasing.

Since the tail current V_{GS} was mirrored to the 100 µA reference, the only option for fixing the sub-threshold devices was to modify the lengths. The relationship between length and threshold voltage is complex, but through iterative testing, it was determined that increasing the device length reduced the V_{TH}. To maintain the 1-2-4-8 ratios, the lengths were kept consistent for all tail currents. Fig. 13 shows the bitwise strength for each driver and the corresponding current.

Fig. 13. Applying V_{DD} or V_{SS} to the gate of the switch controls the strength of each driver, per the tap values. The (0,1,0,0) output polarity is reversed.

Further adjustments were made once all drivers were connected to the channel and CTLE, but the same biasing strategy was applied. The paper calculations for a current mirror do not take Virtuoso's non-idealities into account, as the initial 1:100 sizing ratio only provided a 7 mA current (10 mA was expected). Maintaining the 1-2-4-8 ratio, the final current mirror scaling factor was 1:180 and resulted in a 12 mA current and an approximate 1 V output swing.

C. Channel Model (Eugene)

The channel model was created in Cadence using the n4port component in the analogLib library, where a s4p file modeling a 'b12' channel was loaded [7]. Additionally, the TX and RX ports were loaded with 300fF capacitors to emulate the parasitic caps of contacts. Note that the channel is AC coupled to the RX chain (see Fig. 14).

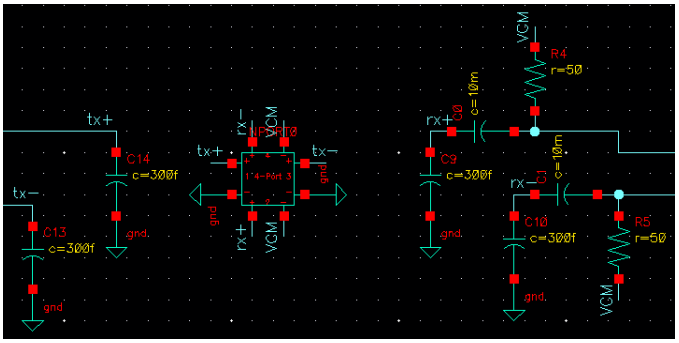


Fig. 14. Schematic of the channel model. The model includes 300 fF loading capacitors, and larger AC-coupling caps on the input of the RX chain. Both ends of the channel were terminated with 50 Ω impedances.

Initially, the AC coupling capacitor was set to a very large value. After a parametric sweep (1 pF step-size) of the capacitor value, it was determined that the coupling capacitors could be as small as 65 pF; values smaller than this reduced the eye height.

D. Continuous Time Linear Equalizer (Eugene)

Optimized in tandem with the FIR filter, the CTLE needed a peaking gain of 10 dB with a peaking frequency at the Nyquist frequency (6.25 GHz). Here, an RC-degenerated CTLE was employed, as seen in Fig. 15.

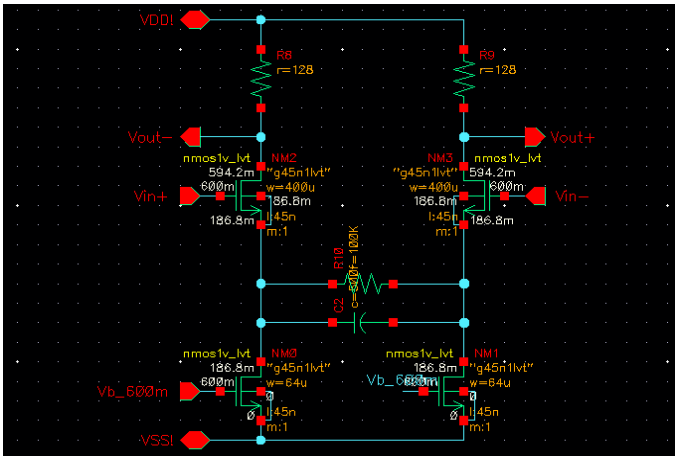


Fig. 15. Schematic of the RC-degenerated CTLE. R8 and R9 contribute to an output pole and were kept small to relax constraints on the input capacitance size of the comparators.

A thorough analysis of this CTLE topology revealed that its parameters (C_S, R_S, g_m, R_D, C_P, and I_D) were not fully constrained, despite requirements for peaking gain/frequency and DC gain. Fig. 16 shows these CTLE parameters and how they relate to one another.

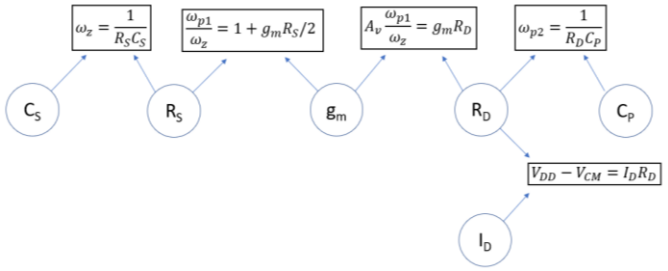


Fig. 16. A diagram depicting the under constrained nature of the CTLE circuit. CLTE circuit parameters are contained in blue circles, while constraining equations are boxed. The left-hand side of each equation is constant, determined by desired design specifications, while the right-hand side relates two parameters together.

Using Fig. 16 to design the CTLE was straightforward. For example, given the input capacitance C_P of the comparators that load the CTLE, R_D can be determined by maintaining the second pole at the Nyquist frequency. From here, a suitable I_D can be found to create the desired output common mode (CM) voltage. At the same time, g_m can be chosen to generate the required DC gain. With g_m selected, R_S can then be sized to make the correct peaking gain. And finally, C_S can be adjusted to place the zero at its designated frequency.

The completed CTLE had an output CM voltage of 600 mV and a minimal output impedance to avoid creating a low-frequency pole when loaded with two Strong-Arm comparators. Additionally, the CTLE met every response requirement, as illustrated in Fig. 17.

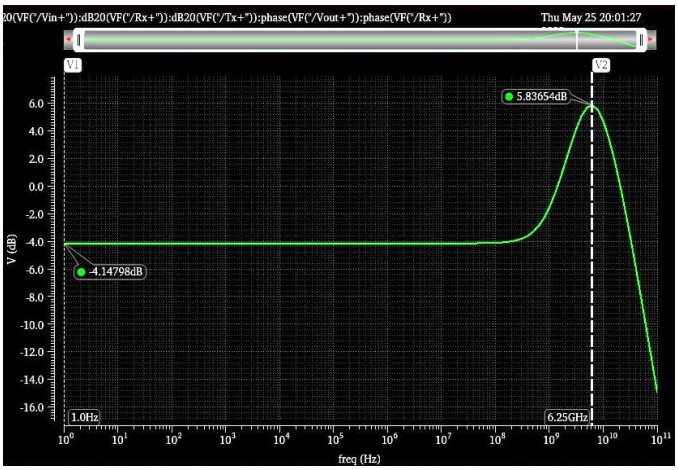


Fig. 17. Magnitude response of the CTLE shows a peak at the Nyquist frequency of 6.25 GHz, with a peaking gain of 10 dB and a DC gain -4 dB. This fulfills the requirements set by the MATLAB optimization.

E. Comparator (Matthew)

The output of the CLTE block requires processing prior to deserializing the received data bits. A comparator [8] and latch are designed to amplify the CLTE outputs to the rail voltages; the comparator block output is then fed to the deserializer. The Strong-Arm comparator [9] was chosen because it can amplify the differential voltages and consumes zero static power.

The Strong-Arm comparator consists of a differential input pair, back-to-back CMOS inverters, a current tail, and precharge transistors set at the inputs of the inverters. During the negative clock edge, the current tail will inhibit current through the differential branches so that the inputs of the inverters can be precharged up to V_{DD} by transistors PM2 and PM3, as shown in the schematic in Fig. 18. This is known as the *precharge* phase of the comparator.

Using the two inverters connected in a positive-feedback manner, the differential input signal can be amplified to the supply rail voltages. Given that V_{ip} of the differential input signal is higher than V_{im}, this causes the left V_{ip} branch to discharge faster than the right V_{im} branch. Both branches will discharge until the output of the left V_{ip} branch falls below $V_{DD} - V_{th,p}$ and causes PM0 to turn on. This causes V_{op} to rise back up to V_{DD}, maintain a high bias voltage for the left CMOS inverter, and causes V_{om} to continue to discharge. This is known as the *sampling* phase of the comparator. The opposite would occur if V_{im} was higher. Again, all of this happens during the positive clock edge when the precharge transistors are off and the current tail is on. The input voltages of the inverters will be recharged during the *precharge* phase.

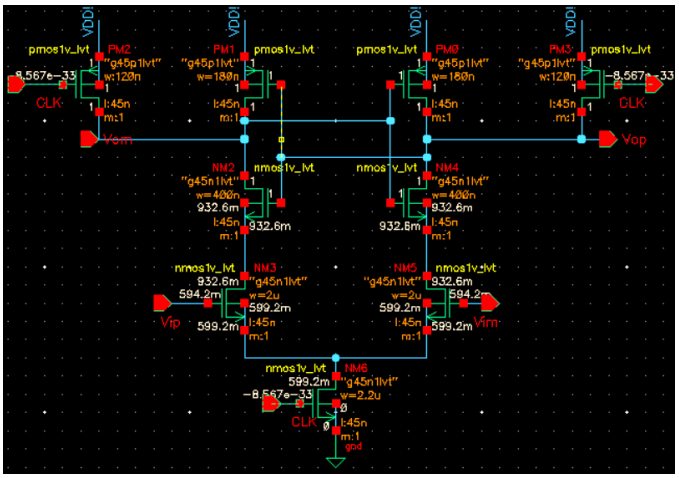


Fig. 18. Schematic of the Strong-Arm comparator. Two comparators are required and utilized to operate at the positive and negative edges of the clock, resulting in 1:2 deserialization.

For the processing of the CLTE output, two Strong-Arm comparators are required and used so that there are no gaps in the data amplification during the precharge phase. The two comparators are triggered at the positive and negative edges of the clock, resulting in a relaxed sampling period and a 1:2 deserialization. If only one comparator was used, the clock would need to be double the frequency so that the comparator can go through the precharge and sampling phases within one bit period (80 ps) so that no data bits are lost. This would require more area and power so that the precharge and discharge phases occur fast enough.

There were four main considerations when designing the Strong-Arm comparator:

- Proper operation with a peak-to-peak differential input signal of 80 mV using square and sine waves.
- Common mode voltage of 600 mV to match CTLE.
- Small aperture time of equal to or less than half of the comparators' sampling period.
- Rail-to-rail differential output for data deserialization

The comparator output, as seen in Fig. 19, contains the *precharge* and *sampling* phases, which resembles a signal that is *not* fully differential. Thus, another circuit block is required to convert the comparator output to a differential signal prior to deserialization.

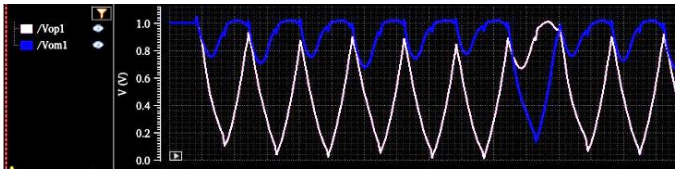


Fig. 19. Output of the Strong-Arm comparators. The *precharge* and *sampling* phases are seen in each clock cycle and allow the V_{op} to be differentiated from the V_{om}. Additional circuit blocks are required so that a digital output is sent to the deserializer.

The Strong-Arm comparator has large input transistor pairs to allow for larger gain. The inverter transistors in the middle of the comparator are sized so that the *sampling* phase is quick. The precharge transistors on the left and right sides are sized as small as possible to reduce the load capacitance on the output of the comparator. It was verified through simulation that minimum-sized devices were able to precharge the inverter input branches at a sufficient speed.

Initially, an RS-latch realized with FreePDK45 digital NAND blocks was used; the output waveform is shown in Fig. 20. However, the rise and fall times led to bit widths to be narrower than desired. Also, the logic transitions from high to low and low to high were too slow. Therefore, another digital block was designed on the transistor level so that rise and fall times between the digital states could be faster.

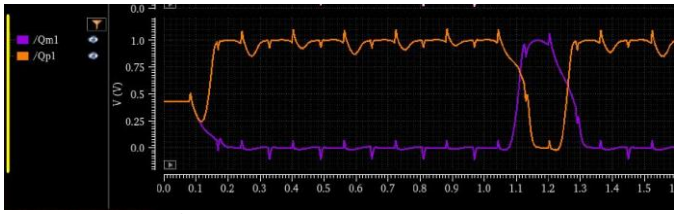


Fig. 20. Output of the RS-latch built using FreePDK45 cells. The RS-latch can remove the precharge phase of the comparator output from Figure 18, but transitions are too long due to the fixed rise and fall times in the FreePDK45 cells.

Fig. 21 (below) shows a dynamic-to-static converter, also known as a tri-state buffer [10]. This buffer helps to produce the desired output. The V_{op} and V_{om} outputs of the Strong-Arm comparator are fed into the left and right branches of the converter and become a fully differential Q_p and Q_m output that resembles the serialized data after the first serializer block of the transmitter. These results are discussed later in the Top-Level Results section.

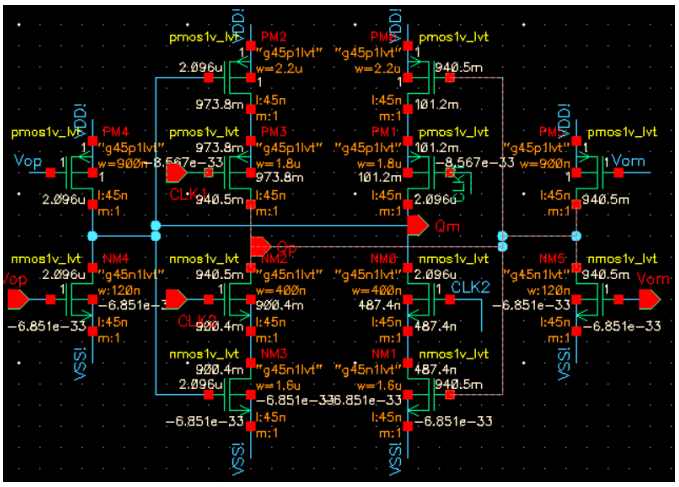


Fig. 21. Schematic of the Dynamic-to-Static Converter (D2S). D2S converter removes the regen phase of the comparator output and feeds the resulting rail-to-rail differential signal to the deserializer.

The converter consists of a double inversion buffer utilizing two sets of CMOS inverters. The first inverter is realized in the outer branches, and the second inverter is realized in the top and bottom transistors within the inner branches. The other two transistors in each half of the buffer are used as tri-state buffer controls to allow high impedance when the data clock is low (which prevents contention). This allows the double-inverted data to be latched at the output for the entire clock cycle, and new data will be held in the next clock cycle.

Resultantly, V_{op} and V_{om} data that is inputted from the comparator will be converted to a rail-to-rail fully differential output signal and is held for the entire clock cycle; one of these differential outputs is shown in Fig. 22. Compared to the RS-latch, the D2S converter provides the desired output from the comparator to the deserializer: rising and falling edges that are both cleaner and faster.

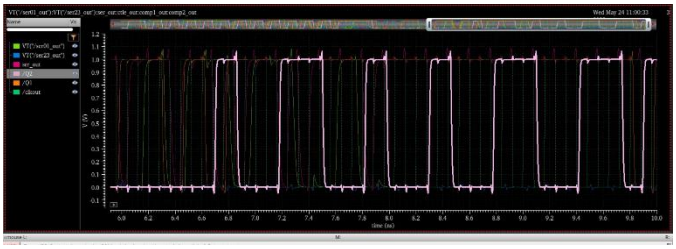


Fig. 22. One of the differential outputs of the Dynamic-to-Static Converter (D2S). The edges are cleaner and faster, and the bit width is maximized.

F. Deserializer (Tyler)

In the design of the deserializer, a shift register architecture [4] was chosen, as seen below in Fig. 23. This design was simple and only required the use of 4 flip-flops (and only 4 more for the parallel load), with a single deserializer connected to each comparator in the RX chain. This architecture was also selected as it is less prone to timing issues since it only uses two clocks.

The standard cell flip-flops in FreePDK45 were used as they operated properly at 6.25 GHz. Utilizing these standard cell flip-flops instead of the TSPC CML flip-flops stood as a better option as the TSPC design has floating node outputs which could be susceptible to discharge and leakage which can hinder their operation at lower frequencies. As a result, the CMOS FFs were also chosen for the clock divider circuit.

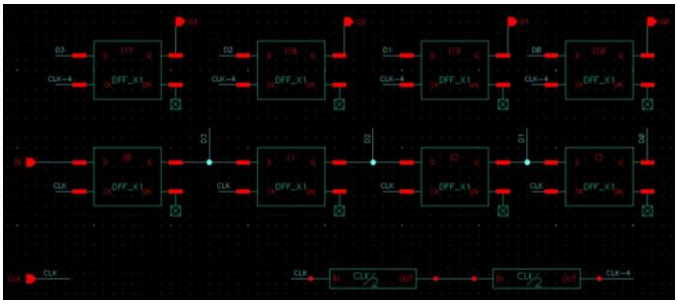
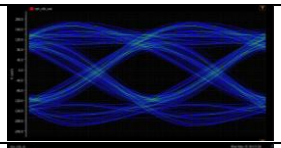
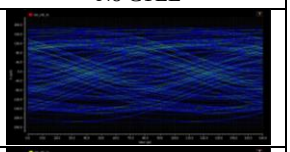
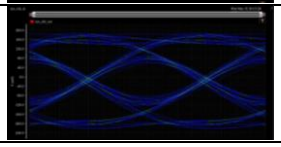
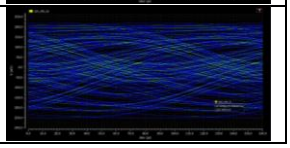


Fig. 23. Schematic of the shift register deserializer. The shift register increments the data through with each clock cycle and is loaded in parallel into another register at one fourth the original clock frequency.

III. Toplevel Results

To examine the performance of the FFE and CTLE, 4 eye diagrams were created with/without the FFE/CTLE. These results are outlined in Table I.

TABLE I.
EFFECT OF FFE AND CTLE ON THE EYE DIAGRAM

	With CTLE	No CTLE
With FFE		
No FFE		

In the case where no FFE and CTLE were used, the eye diagram was completely closed, indicating the need for channel equalization techniques. A larger plot of this eye diagram can be seen in Fig. 24.

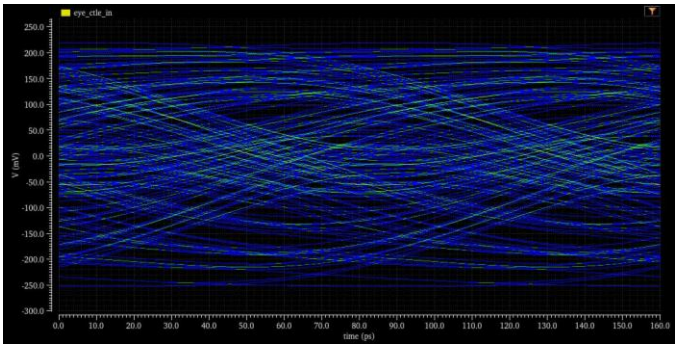


Fig. 24. Eye diagram without channel equalization techniques. The eye is fully closed, creating the need for equalization.

With both the FFE and CTLE, the eye diagram is significantly improved, creating an open eye. This result can be examined in Fig. 25.

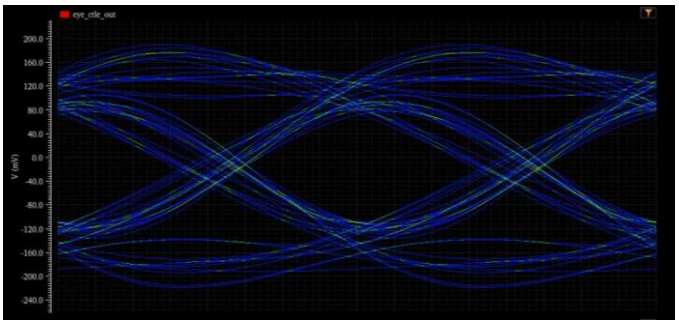


Fig. 25. Eye diagram with both the FFE and the CTLE. The eye is completely open with an eye height of 160 mV.

Additionally, a BER analysis of this improved eye diagram revealed an eye width of 22 ps at a BER of 10^{-12}, as shown in Fig. 26.

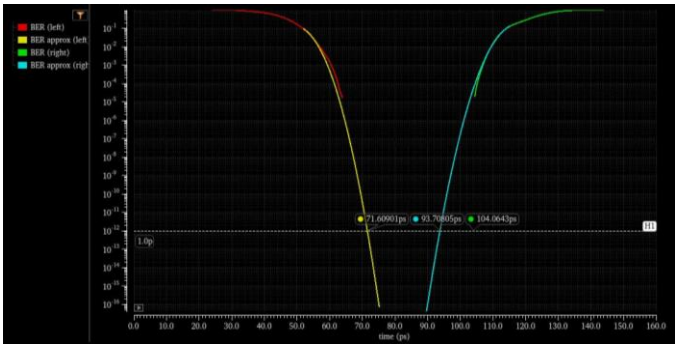


Fig. 26. BER curve with both FFE and CTLE in use.

The creation of a clean eye diagram ultimately means nothing if any component in the transceiver chain fails. To test the transceiver in its entirety, a transient simulation was run where a set of data was passed into the serializer, then read at the deserializer. As expected, the TX data matched the RX data, as portrayed by the waveform in Fig. 27.

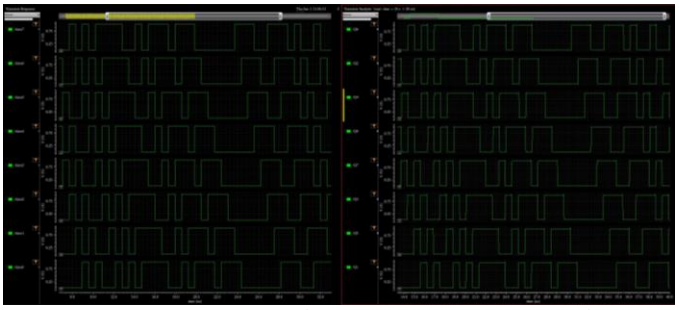


Fig. 27. Transient simulation of the entire transceiver. The left window shows the TX data, while the right window shows the RX data, which matches the TX data. Note that the RX window is delayed by 8 ns compared to the TX window due to delays in the channel model.

IV. Conclusion

For most of the group, this was their first mixed-signal project, so the learning curve was sufficiently high. However, with each circuit block connected, and after extensive testing, we found that our design met the project specifications at a 12.5 Gb/s transfer rate. At a 10^{-12} BER, an eye-width of 22 ps was measured. After passing signals through the channel, the 160 mV eye-height indicates a clear distinction between logic levels. Furthermore, a comparison of the input and output waveforms demonstrates that each transmitted bit is successfully deciphered at the output.

V. References

[1] "High-Speed Serial I/O Made Simple," xilinx.com, https://www.xilinx.com/publications/archives/books/serialio.pdf (accessed Jun. 1, 2023).

[2] Cadence Design Systems. *GPDK 45 nm Mixed Signal GPDK Spec.* (version 6.0, September 9, 2019) Accessed: May 1, 2023

[3] R. Davis, P Franzon, M. Bucher, S. Basavarajaih (NCSU), J. Stine, I. Castellanos (OSU). *FreePDK 45nm.* (version 1.3, March 3, 2009) Accessed: May 1, 2023

[4] N. Jaiswal and R. Gamad, "Design of a New Serializer and Deserializer Architecture for On-Chip SerDes Transceivers," *Circuits and Systems*, vol. 06, no. 03, pp. 81–92, 2015. doi:10.4236/cs.2015.63009

[5] B. Razavi, "TSPC Logic," *IEEE Solid-State Circuits Magazine*, vol. 8, no. 4, pp. 10–13, Nov. 2016. doi:10.1109/mssc.2016.2604198

[6] S. Moazeni (2023). University of Washington EE437/538B: Integrated Systems Capstone/Design of Analog Integrated Circuits and Systems: Lecture 3: Overview of Basic Transmitter (Tx) Blocks [PowerPoint slides].

[7] S. Palermo. *12" Backplane S-Parameter Data.* (version unknown) Accessed: May 1, 2023 [Online] Available: https://people.engr.tamu.edu/spalermo/ecen689/peters_01_0605_B12_thru.s4p

[8] B. Razavi, "The Design of a Comparator [the Analog Mind]," *IEEE Solid-State Circuits Magazine*, vol. 12, no. 4, 2020, Nov. 2016. doi:10.1109/mssc.2016.2604198

[9] B. Razavi, "The Strongarm Latch [a Circuit for All Seasons]," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 2, pp. 12-17, Nov. 2015. doi:10.1109/mssc.2016.2604198

[10] K. T. Settaluri et al., "Demonstration of an optical chip-to-chip link in a 3D integrated electronic-photonic platform," ESSCIRC Conference 2015 - 41st European Solid-State Circuits Conference (ESSCIRC), Graz, Austria, 2015, pp. 156-159, doi: 10.1109/ESSCIRC.2015.7313852.