

## ECEN 720 High-Speed Links: Circuits and Systems

### Lab5 –Equalization Circuits

#### Objective

To learn fundamentals of high speed I/O link equalization techniques.

#### Introduction

An ideal cable could propagate all frequency components without any loss. In reality, all the electrical transmitting mediums have finite signaling bandwidth which limits the data rate of binary signaling. Fortunately, engineers have developed equalization schemes both at transmitter side and receiver side to compensate the loss of transmitting mediums and extend a channel's maximum data rate. In this lab, TX feed-forward equalization (FFE) will be studied, which acts as an FIR filter and pre-distorts transmitted pulse in order to invert channel distortion. At the receiver side, RX finite impulse response (FIR) filter, continuous time linear equalizer (CTLE) and decision feedback equalizer (DFE) will be studied, which are implemented as part of the receiver circuits and flatten the system response through conditioning the received signal. In addition to the link equalization, noise sources in high-speed link systems will be introduced in this lab. Figure 1 shows a high-speed electrical link using TX FFE equalization and RX CTLE+DFE equalization.

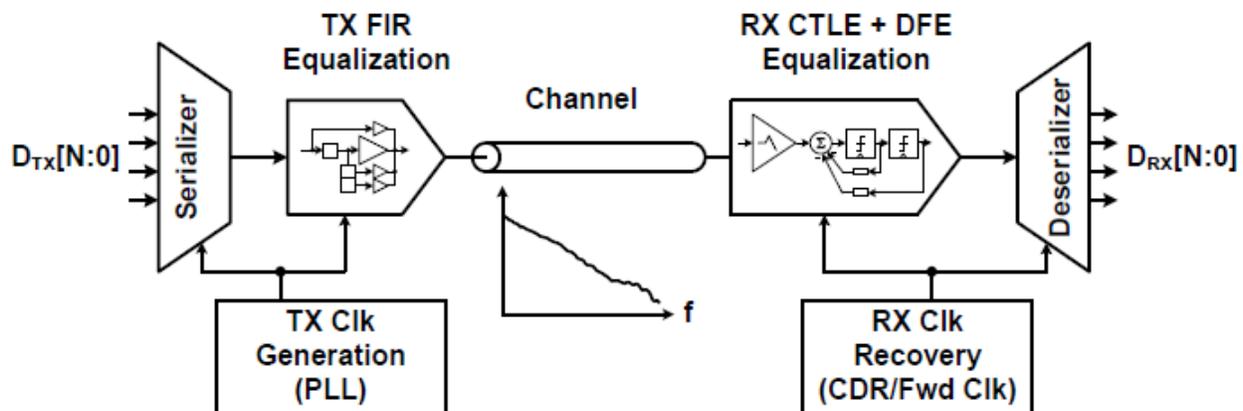


Figure 1 High-Speed Electrical Link with Equalization Schemes

#### TX Feed-Forward Equalization

Transmit equalization is the most common technique in high-speed links design. It is usually implemented using an FIR filter. It pre-distorts or shapes the data over several bit periods in

order to invert the channel loss/distortion. The low frequency components get de-emphasized in order to flatten the channel response. Without a FIR equalizer, the TX driver transmits 1 as a single pulse as the red curve shown in Figure 2(a). The pulse is dispersed by the channel loss/distortion. It is shown as the red curve in Figure 2(b) with pre-cursor and post-cursor ISI components. By using FFE, the pulse is shaped as the blue curve shown in Figure 2(a) based on the channel response. The pulses at time -1 and +1 are generated to cancel the channel pulse response's ISI. The equalized pulse response is shown as the blue curve in Figure 2(b).

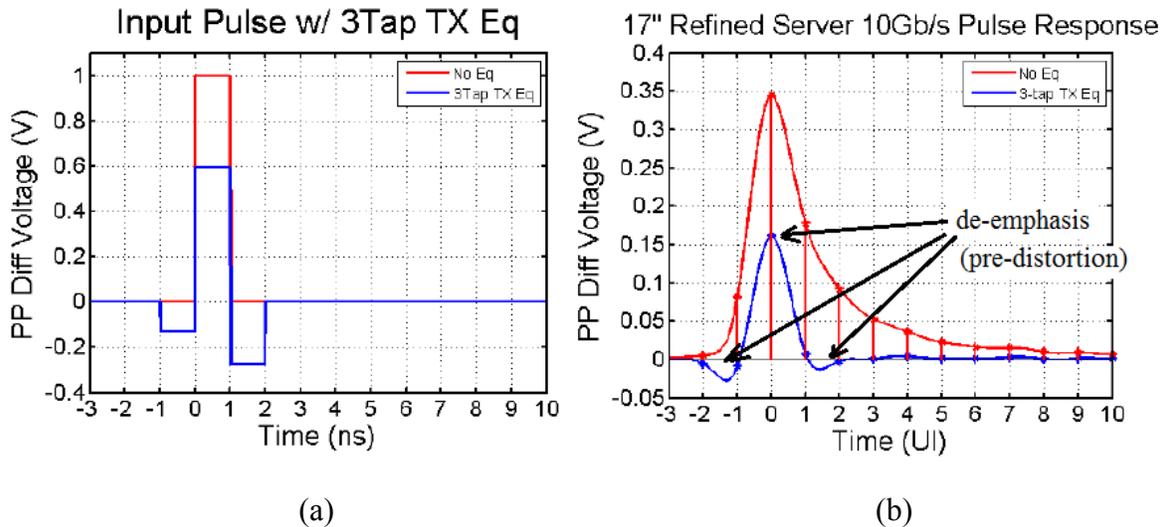


Figure 2 With (blue)/Without(red) FFE Equalizer (a) TX Data Pattern (b) Transmitted Data Pattern at RX Side After The Channel

TX FFE can be implemented as a FIR filter by using unit time delay elements (flip-flops) and current steering DAC circuit as shown in Figure 3(a). Compared with implementing a FIR filter at the receiver side, it is generally easier to build high-speed digital-to-analog converters versus receive-side analog-to-digital converters. However, the transmitter is limited by the peak transmitting power across the channel due to driver voltage headroom. Channel response flattening is realized through attenuating low-frequency signal content as shown in Figure 3(b).

Pros:

- High speed DAC is relatively easy to implement compared to receiver high speed ADC.
- TX FFE can cancel pre-cursor (or sometimes post-cursor) ISI.
- Due to the digital nature of the TX FFE, the noise is not amplified.
- 5-6 bit resolution can be achieved conveniently.

Cons:

- To flatten the channel response, low frequency content is attenuated due to the peak-power limitation.

- To tune the FIR taps, a feedback path from receiver side is required to detect channel response.

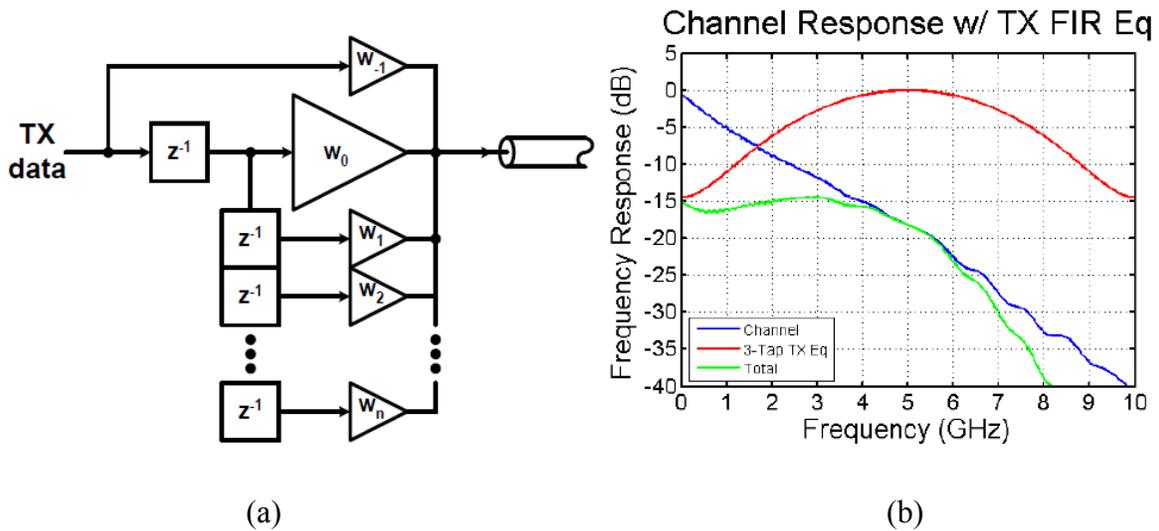


Figure 3 TX Equalization with a FIR Filter (a) FFE Equalizer (b) Channel, TX FIR, and Channel+TX FIR Responses

**Example:**

Given TX FIR  $z$ -domain transfer function (1), find the low frequency response and Nyquist frequency response?

$$W(z) = -0.131 + 0.595z^{-1} - 0.274z^{-2} \tag{1}$$

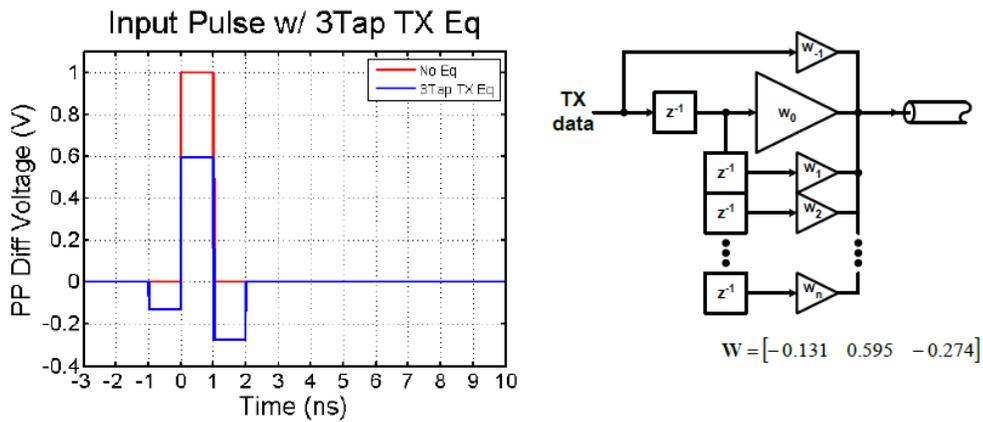


Figure 4 TX FFE Example

At the low frequency, we can assume that data pattern is infinite number of 1 as [...1 1 1 1 1 1...]. The z domain transfer function is  $[w_{-1} \ w_0 \ w_1] = [-0.131 \ 0.595 \ -0.274]$ . The low frequency response can be expressed as

$$[\dots 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ \dots] * [-0.131 \ 0.595 \ -0.274] = [\dots 0.190 \ 0.190 \ 0.190 \ \dots] \quad (2)$$

Alternatively, at low frequency  $f=0$  and in z domain,  $z=\cos(0)+j\sin(0)=1$ , the transfer function can be written as

$$W(1) = -0.131 + 0.595(1^{-1}) - 0.274(1^{-2}) = 0.190 = -14.4dB \quad (3)$$

At the Nyquist frequency, the data pattern is infinite number of -1 and 1 as [...-1 1 -1 1 -1 1...]. The Nyquist frequency response can be expressed as

$$[\dots -1 \ 1 \ -1 \ 1 \ -1 \ 1 \ \dots] * [-0.131 \ 0.595 \ -0.274] = [\dots 1, -1, \ 1 \ \dots] \quad (4)$$

Alternatively, at Nyquist frequency  $f=1/2T_s$  and in z domain,  $z=\cos(2\pi fT_s)+j\sin(2\pi fT_s)=1$ , the transfer function can be written as

$$W(1) = -0.131 + 0.595(-1^{-1}) - 0.274(-1^{-2}) = -1 = 0dB \quad (5)$$

Therefore the FIR attenuates DC by -14.4dB gain and passes Nyquist frequency by 0dB gain.

### RX FIR Equalization

FIR equalization can be realized at the receiver side as shown in Figure 5. Since the receiving signal contains the channel response information, the filter tap coefficients can be adaptively tuned to the specific channel which is the major advantage of receiver side equalization. However, the implementation of the analog delay elements is the major challenging issue in circuit level realization. The high-frequency noise content and crosstalk are also amplified along with the incoming signal. Noise amplification is illustrated in Figure 6. RX FIR equalization can also be realized in digital domain. Due to the speed of the ADC, power consumption can be very high.

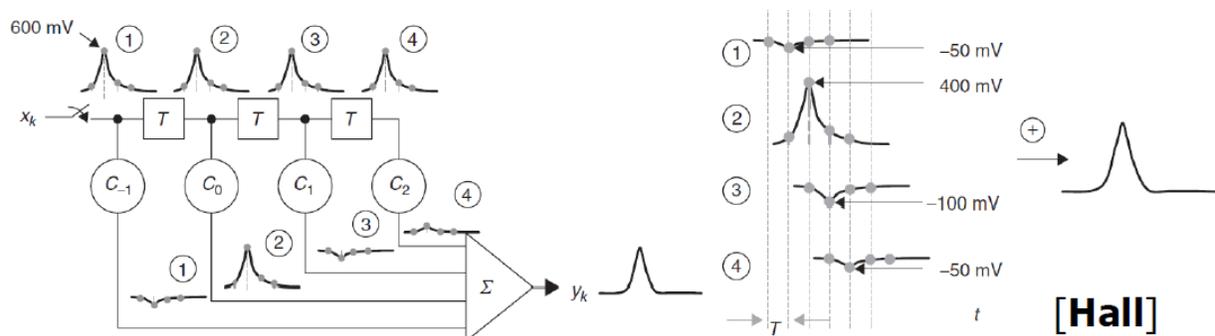


Figure 5 Receiver FIR Equalization

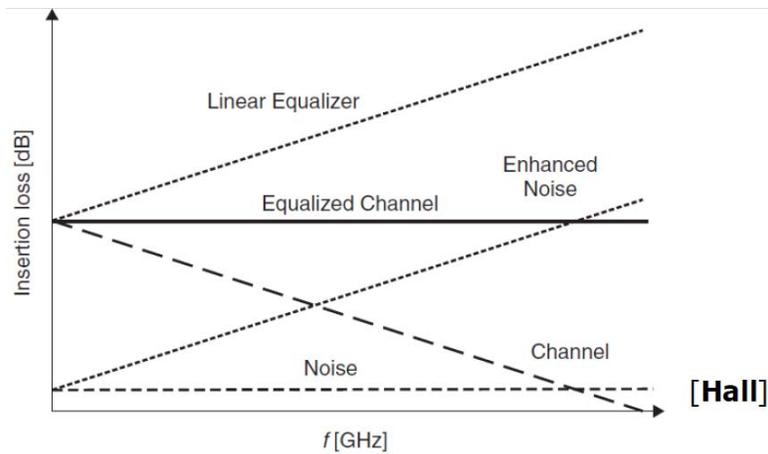


Figure 6 RX Equalization Noise Enhancement

#### Pros

- Amplify high frequency content rather than attenuate low frequency components.
- Cancel both pre-cursor and beyond filter span ISI.
- Filter tap can be adaptively tuned.

#### Cons

- Noise and crosstalk are amplified at the same time
- Analog delays are not easy to implement and tap precision is difficult to meet.

### RX Continuous-Time Linear Equalizer (CTLE)

Both linear passive and active filters can realize high-pass transfer function to compensate for channel loss as shown in Figure 7. Both pre-cursor and long-tail post-cursor ISI can be cancelled using the linear equalizer.

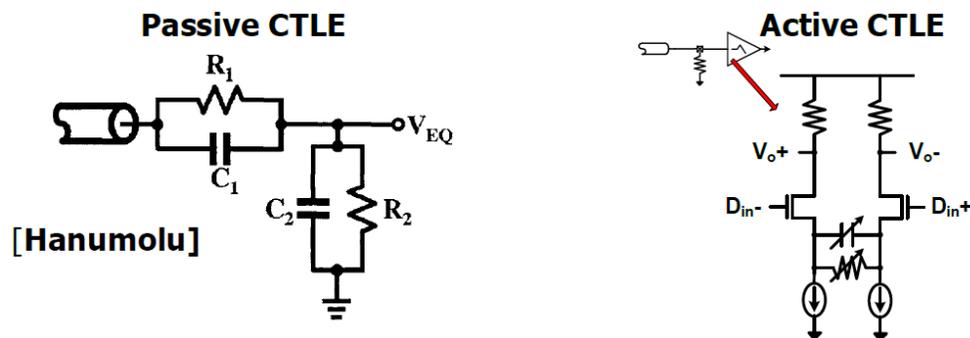


Figure 7. (a) Passive CTLE

(b) Active CTLE

The passive CTLE is the combination of passive low pass filter and high pass filters. The transfer function of passive CTLE shown in Figure 7(a) can be written as

$$H(S) = \frac{R_2}{R_1 + R_2} \frac{1 + R_1 C_1 s}{1 + \frac{R_1 R_2}{R_1 + R_2} (C_1 + C_2) s} \quad (6)$$

At DC, the capacitors can be ignored and the filter becomes a resistor divider circuit. DC gain is written as

$$DC \text{ gain} = \frac{R_2}{R_1 + R_2} \quad (7)$$

At very high frequencies, the capacitors become low impedance elements. The high-frequency AC gain is determined by capacitors only. It is written as

$$AC \text{ gain} = \frac{C_1}{C_1 + C_2} \quad (8)$$

For the passive filter, there is no gain at Nyquist frequency and the peaking is calculated as

$$peaking = \frac{AC \text{ gain}}{DC \text{ gain}} = \frac{\omega_p}{\omega_z} = \frac{R_1 + R_2}{R_2} \frac{C_1}{C_1 + C_2} \quad (9)$$

Active CTLE can be implemented through a differential pair with RC degeneration with gain at Nyquist frequency as shown in Figure 7(b). At high frequencies, degeneration capacitor shorts the degeneration resistor and creates peaking. The peaking and DC gain can be tuned through adjustment of degeneration resistor and capacitor.

Pros

- Active CTLE provides gain and equalization with low power and area overhead.
- Cancels both pre-cursor and long tail post-cursor ISI

Cons

- Equalization is limited to 1<sup>st</sup> order compensation.
- Noise and cross-talk are amplified
- Very sensitive to PVT variations and be hard to tune
- The speed is limited by gain bandwidth of the amplifier.

The transfer function of the active CTLE is written as

$$H(s) = \frac{g_m}{C_p} \frac{s + \frac{1}{R_s C_s}}{\left(s + \frac{1 + g_m R_s / 2}{R_s C_s}\right) \left(s + \frac{1}{R_D C_p}\right)} \quad (10)$$

DC gain is expressed as

$$DC \text{ gain} = \frac{g_m R_D}{1 + g_m R_S/2} \quad (11)$$

Ideal peak gain is equal to  $g_m R_D$ . Ideal peaking can be expressed as

$$Ideal \text{ peaking} = \frac{Ideal \text{ peak gain}}{DC \text{ gain}} = 1 + g_m R_S/2 \quad (12)$$

### RX Decision Feedback Equalization (DFE)

Decision feedback equalizer is commonly implemented in high-speed links receiver-side. Slicer makes a symbol decision without amplifying noise. The results are fed back to the slicer input through an FIR filter to cancel post-cursor ISI. The major challenge in DFE implementation is the closing timing on the first tap feedback, which must be done in one bit period or one unit interval (UI).

Pros

- Boost high frequency content without noise and crosstalk amplification
- Tap coefficients can be adaptively tuned

Cons

- Due to the nature of feedback, pre-cursor ISI cannot be cancelled, but FFE can be used to complement the DFE equalization.
- If noise is large, chance for error propagation is high.
- Critical feedback timing path is less than one UI for first post-cursor cancellation.
- CDR phase detection can be complicated due to the timing of ISI subtraction.

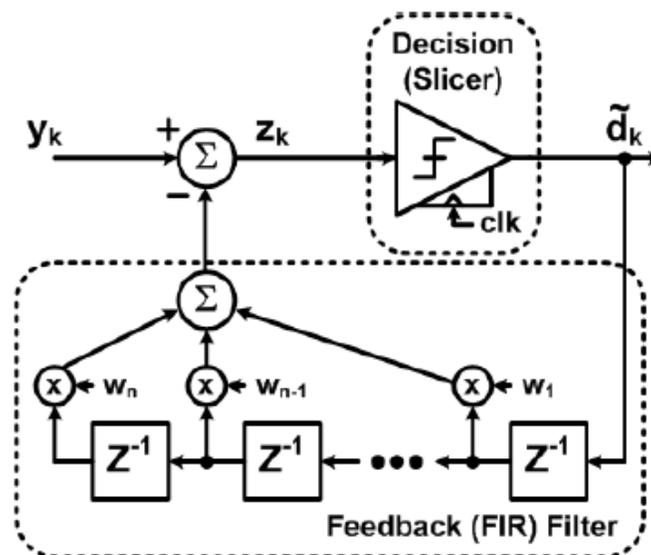


Figure 8 Receiver DFE Equalization

## Direct Feedback DFE Examples

A 6.25Gb/s 4-tap DFE RX from Texas Instruments [2] is shown in Figure 9. Amplifier  $A_1$  works as the main tap of the equalizer which provides linear gain of the input.  $A_2$ ,  $A_3$ ,  $A_4$  and  $A_5$  are the feedback taps, which equalize ISI caused by the channel. All 5 taps are summed into the resistor. The equalized signal is produced at  $RXEQ$ . The TAP1 has the most critical feedback path here. This critical path and its components are shown in Figure 10. The comparator must make decision within  $\frac{1}{2}$  UI in order to meet the timing requirement of adaptive equalization tap values and CDR.  $CLK_{90/270}$  samples the  $RXEQ$  alternatively on top and bottom comparator.  $DFECLK$  shifts the sampled data into the proceeding latches and at the same time selects the DFE polarity based on the data sequence.

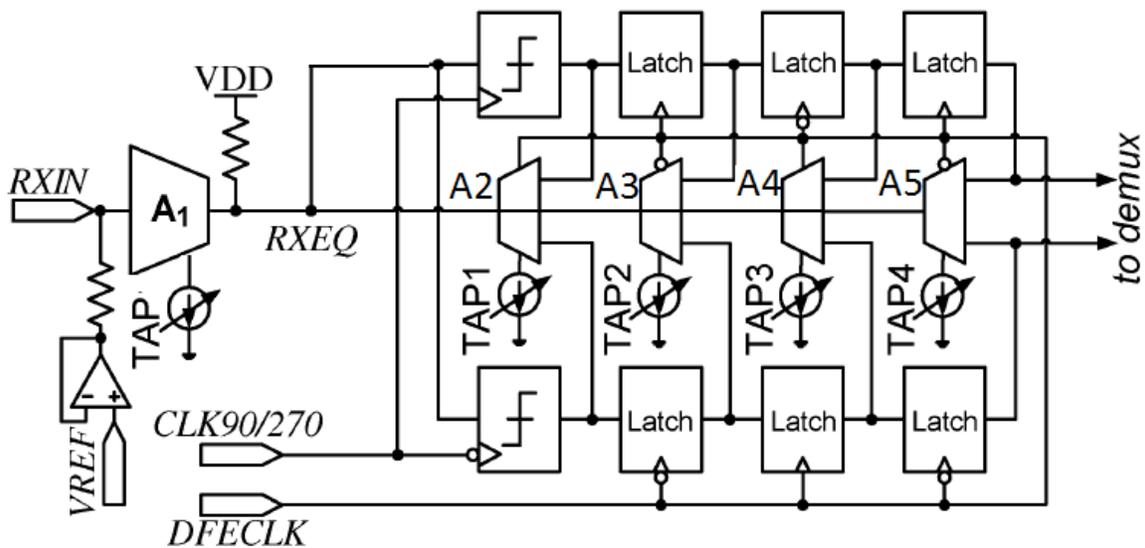


Figure 9 Direct Feedback DFE with Receiver Analog Front End (5-tap DFE for Illustration)

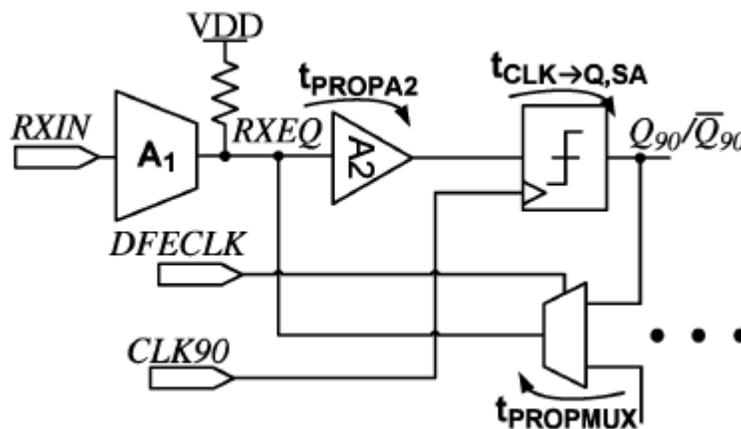


Figure 10 Schematic of the Critical Path of the Direct Feedback DFE



## Pre-Lab

- Assuming a 3-tap TX FIR equalizer with the  $z$ -domain transfer function as the following

$$W(z) = -0.1 + 0.6z^{-1} - 0.30z^{-2} \quad (13)$$

Please find the low frequency response, Nyquist frequency response, and frequency peaking of this 3-tap TX FIR equalizer?

- Design a passive CTLE as shown in Figure 7(a) to realize the transfer function as shown in Figure 12 using  $R_1=900\Omega$ . Please show your schematic.

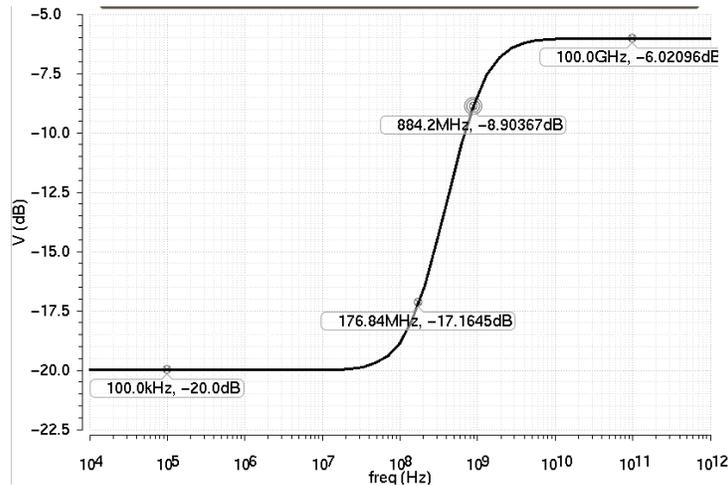


Figure 12 Passive CTLE Transfer Function

- Design a 4-bit parallel PRBS generator with  $2^7-1$  sequence length and an error detector circuit with a testing circuit at 2Gb/s with ideal blocks. Using the test circuit, you should be able to inject error data pattern and observe the error signal. An example is shown in Figure 13. Please show the simulation results. You may refer to the Appendix by Younghoon Song on how to build a parallel PRBS generator and detector.

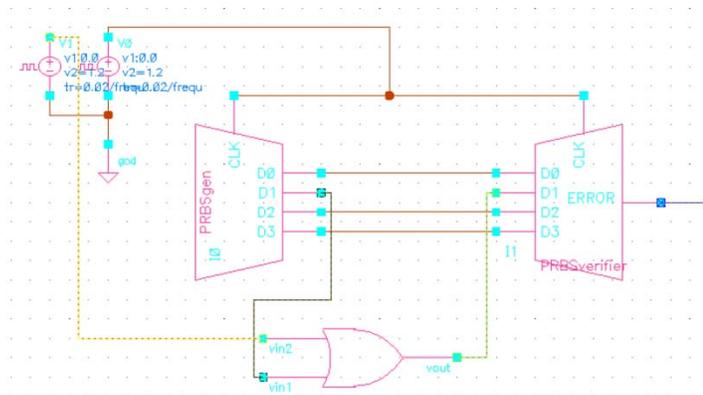


Figure 13 Parallel PRBS Testing Circuit

## Questions

1. **TX FIR Equalization.** This problem investigates TX FIR equalization using the 12” Backplane channel “**peters\_01\_0605\_B12\_thru.s4p**” from course website. For parts (a) and (b), use the example MATLAB code “**channel\_data\_pulse\_pda.m**” and produce the following 2 graphs:
  - a) **Peak-Distortion Eye Height versus FIR tap number at 4G, 8G, and 16Gbps** (3 lines). For the tap numbers, use 1-tap (no equalization), 2-tap (1-post), 3-tap (1-pre and 1-post), and 4-tap (1-pre, 2-post). Don’t restrict the TX equalizer resolution for this graph.
  - b) **8Gb/s Peak-Distortion Eye Height versus Equalizer Resolution with 2, 3, and 4-tap equalization** (3-lines). For the tap resolution sweep, use 3, 4, 5, 6 bits, and also include the infinite resolution data.

Note: The above MATLAB code also requires the “**tx\_eq.m**” function. Also, this MATLAB code is only a reference code. Feel free to modify and improve upon the code as you wish.

- c) **Design an 8Gb/s TX Driver with Equalization** [4]. Modify one of your drivers from Lab3 to include FIR equalization.
  - i. The maximum output voltage swing can be anywhere from 300mVppd (min.) to 1Vppd (max.). This gives you the flexibility to choose whichever driver you wish – from low-swing voltage-mode to current-mode.
  - ii. Use the results from part (a) and (b) to justify your tap number and resolution.
  - iii. **Include two 8Gb/s PRBS eye diagrams** – one without equalization (all weight on main cursor) and one with the proper equalization taps enabled. Import the s-parameter file into your Cadence simulation to produce the eye diagrams. Make sure the channel is properly terminated at both ends. Note, as you will have some additional driver capacitance, the equalization taps may change slightly.
  - iv. The driver and at least one pre-driver stage should be full-transistor level design. The other blocks (PRBS, delay elements, etc) can be macro-models.
  - v. Report transmitter power consumption, power efficiency (mW/Gb/s), and 8Gb/s eye height and width.
2. **RX CTLE Equalization.** Design an 8Gb/s active CTLE to meet the following specifications:
  - a) Min peak gain at Nyquist (4GHz) of 6dB
  - b) Zero frequency tunable from a minimum range of 500MHz to 1GHz
  - c) Minimum tunable peaking (magnitude difference between Nyquist and low frequency response) range of 12dB (Example: +6dB at Nyquist frequency and -6dB at low frequency).
  - d) Load capacitor = 25fF
    - i. **Produce frequency response plots** showing the zero and peaking tunability.
    - ii. **Produce an 8Gb/s PRBS eye diagram** with the 12” Backplane channel output as the input to the CTLE. Optimize the CTLE settings for optimal eye opening.

- iii. Report CTLE power, power efficiency (mW/Gb/s), and 8Gb/s eye height and width.
3. **RX DFE Equalization.** Design an 8Gb/s 2-tap DFE [2][3].
- a) Use one of the comparators you designed in Lab4 in your design. Note, you probably have to speed this design up – as you will need a 4GHz clock if you implement a half-rate design.
  - b) The only thing that has to be transistor level is the comparator. The rest of the blocks (summer, feedback taps, other logic) can be macro-models. Note, for the summer model make sure to capture the RC settling if you use a linear resistive load summer. Feel free to investigate an integrating architecture if you prefer.
  - c) **Produce an 8Gb/s PRBS eye diagram at the summer output** with the 12” Backplane channel output as the input to the DFE. Optimize the DFE settings for optimal eye opening at the input of the comparator (summer output).
  - d) Report DFE power, power efficiency (mW/Gb/s), and 8Gb/s eye height and width.
  - e) Note, you have to synchronize the DFE with the incoming data stream. A good way to do this is with an initial “lone pulse” input pattern. Adjust your comparator clock to sample near the peak of the lone pulse. Then simulate with the PRBS data.

## References

- [1] *Digital Systems Engineering*, W. Dally and J. Poulton, Cambridge University Press, 1998.
- [2] R. Payne, et al., “A 6.25-Gb/s binary transceiver in 0.13-um CMOS for serial data transmission across high loss legacy backplane channels,” *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, Dec. 2005, pp. 2646-2657.
- [3] M. Park, J. Bulzacchelli, and D. Friedman, “A 7 Gb/s 9.3 mW 2-tap current-integrating DFE receiver,” in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2007, pp. 230–231.
- [4] John F. Bulzacchelli, et al., “A 10-Gb/s 5-Tap DFE/4-Tap FFE transceiver in 90-nm CMOS technology”, *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, Dec. 2006, pp. 2885-2900.

# Appendix

Prepared by: Younghoon Song

1. Build a macromodel of a 4-bit parallel PRBS generator and verifier. The PRBS sequence length should be  $2^7-1$ . Refer to Appendix A in Ken Yang's thesis for some intro material on parallel PRBS implementations.

For very high-speed generation of PRBS sequences, it is useful to know which architecture is optimal for a particular application. The different options that can be considered are parallel versus series PRBS generator architectures and the level of multiplexing. The level of multiplexing determines how much slower, relative to the final output, the core generator is operated, thus requiring proportionally less power. However, if the multiplexing level is too deep, too much power might be spent in the multiplexer itself. Figure 1 shows serial PRBS generation and detection configuration.

**A. Serial PRBS Generation and Error Detection**

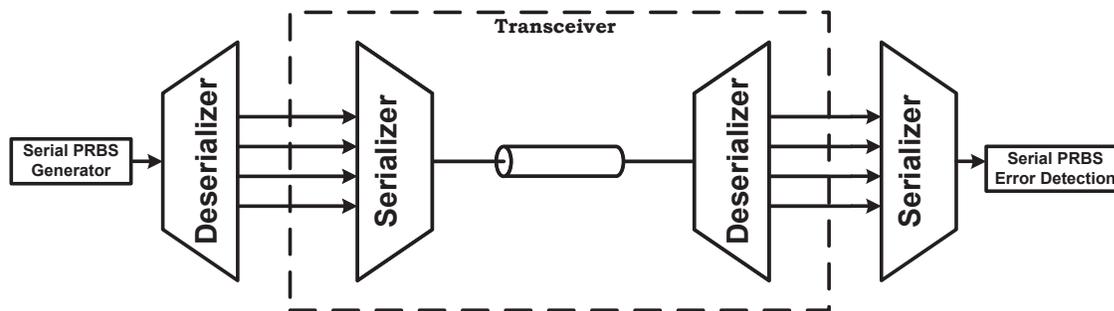


Figure 1 Serial PRBS generation and Error Detection

Series PRBS generators are linear feedback shift registers, where the length of the register  $n$  and the feedback function determine the length of the sequence  $p=2^n-1$ [1]. For multiplexing the sequence to  $q$  times the original bit rate, original sequences, spaced apart by  $(p-1)/q$  bits in phase are required [2]. An efficient algorithm exists for obtaining the phase shifts, nevertheless, the number of XOR gates required to implement the phase shifts in hardware grows exponentially with  $q$ , which is shown in Figure 2.

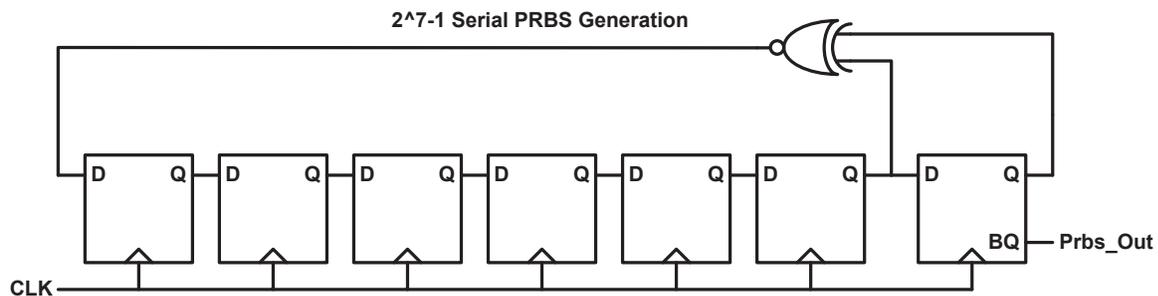


Figure 2  $2^7-1$  Serial PRBS Generation

In simulation, the initial condition was zero, therefore the register remains in a degenerate state. XNOR gate was used instead of some method of initialization.

The error detection is based on the principle of multiplication by a reciprocal polynomial. The technique uses the same length shift register chain as in the generating polynomial. In Figure 3, the serial detection scheme is shown. The shift register is continuously fed by the incoming bit stream and the shift register outputs are XNOR in the same manner as in the generating polynomial. The XNOR output, which is inverse with Din compare with Din XOR gate. If there bits are inverse, the checker generates a no error flag, BQ, for the incoming data. If the bits same, an error is flagged. The technique allows self synchronization to the incoming bit stream, without using any additional hardware.

**ERROR Detector for PRBS =  $2^7 - 1$**

\*Self-synchronization with incoming data sequency

\*Not Accurate with High BER

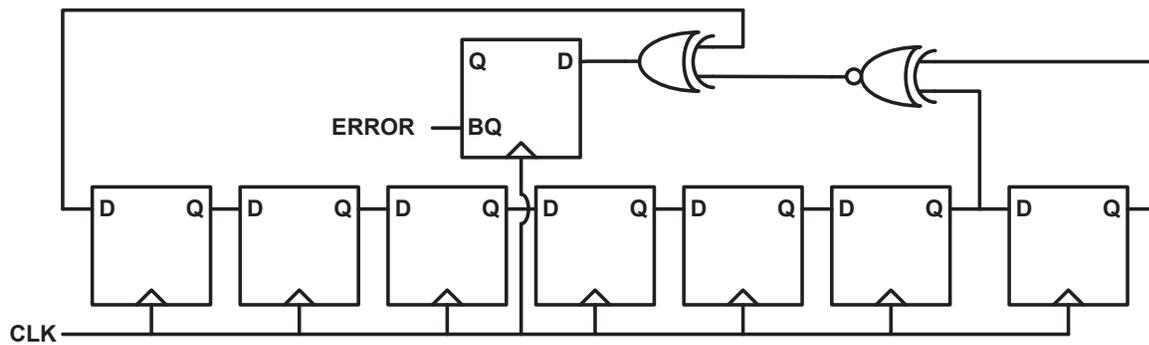


Figure 3 ERROR Detection for PRBS =  $2^7-1$

**Simulation Result: PRBS Generation, 1 to 4 Demux, 4 to 1 MUX, and PRBS detection**

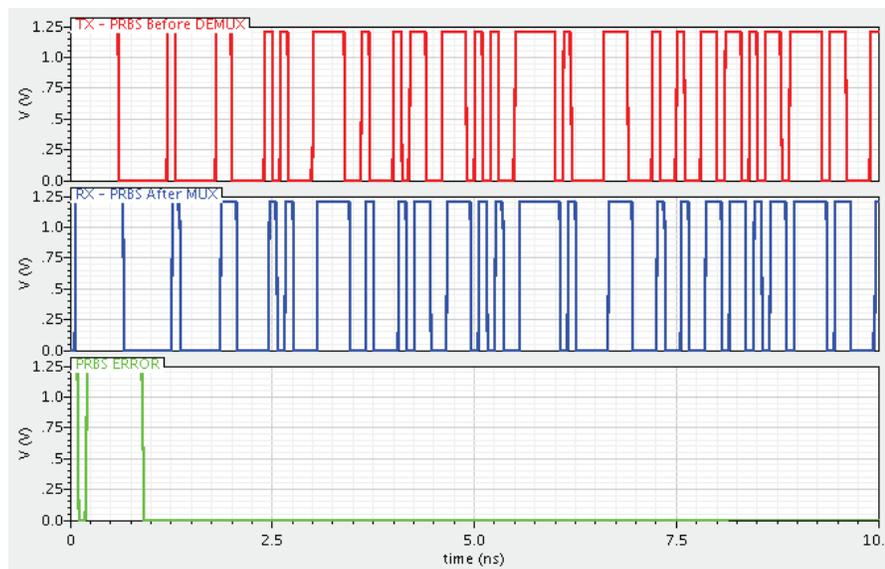


Figure 4 10Gbps  $2^7 - 1$  PRBS Generation and Detection

Figure 4 shows simulation result. Initially error exists; however it is not valid time for error detection. Therefore we can ignore those errors.

#### **B. 4 bits Parallel PRBS Generation and Purposed Error Detection**

In the parallel PRBS generator and detector architecture, which is shown figure 5, the phase shifted sequences are available directly from the generator. The  $n \times m$  transition matrix  $T$ , which can be obtained from the characteristic polynomial of the PRBS, proves useful for constructing parallel PRBS generators. A procedure for translating into the PRBS generator schematic with parallel outputs is given in [3]. The resulting outputs are phase shifted appropriately for direct multiplexing.

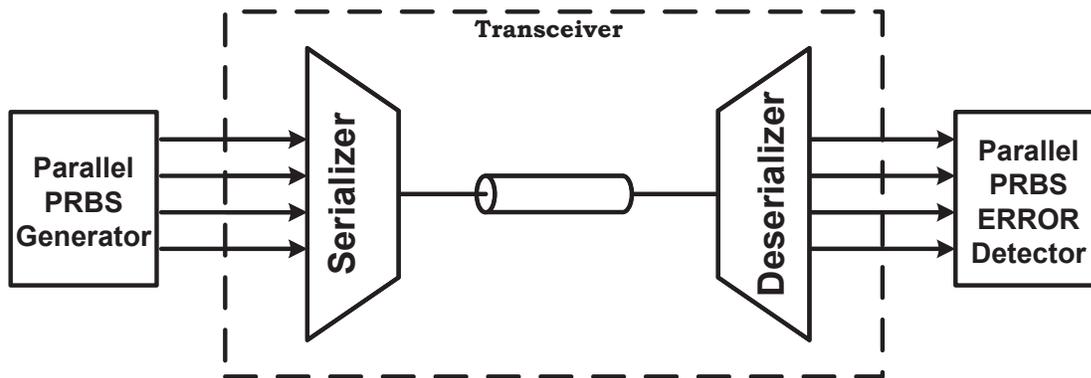


Figure 5 Parallel PRBS generator and detector architecture

The circuit requires  $n$  flip-flops as in the traditional method, but  $m$  XOR gates are needed to generate parallel  $m$ -bit data streams. Figure 6 shows the connections among flip-flops and XOR gates to form a parallel generator. The flip-flops are connected as a parallel register with the output fed back to the input through the XOR gates. The XOR gates are connected so each bit of the new word is generated according to a given polynomial equation. In this paper, we have used 4-to-1 multiplexer to form a single high-speed bit stream using the low-speed parallel data generated by 4-bit parallel PRBS generators. The XOR gates are connected so each bit of the new word is generated according to the polynomial equation. For example, on the first clock cycle, bit six of the PRBS is generated by XOR bit 1 and bit 3, which are presently held in the register. At the same time, bit 7 is generated from bits 2 and 4. A slight irregularity occurs at the bottom of the register. In order to calculate bit 9, bits 4 and 6 are required. Bit 6 is not held in the register, so it must be obtained from an earlier XOR, where it is being calculated.

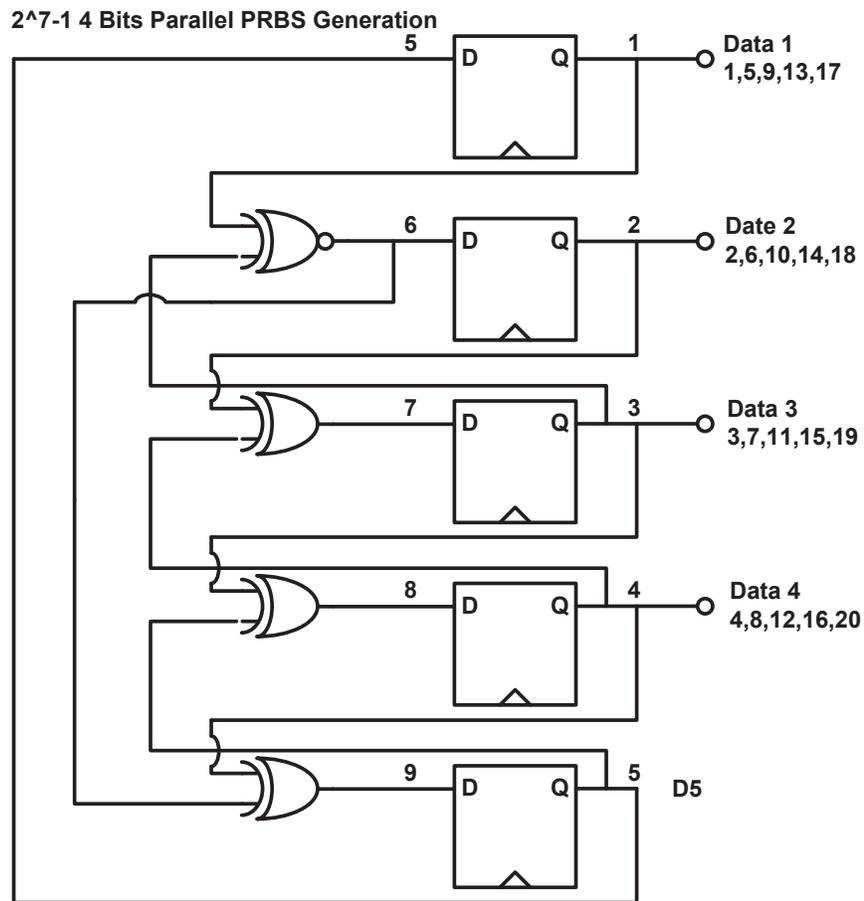


Figure 6 N=5, W=4 2<sup>7</sup>-1 Parallel PRBS Generator [3]

Again, to prevent it remains in a degenerate state. XNOR gate was used instead of some method of initialization.

$$\text{Data 1}[n-1] = \text{D5}[n]$$

$$\text{Data2}[n-1] = \text{XNOR}(\text{Data1}[n], \text{Data3}[n])$$

$$\text{Data3}[n-1] = \text{XOR}(\text{Data2}[n], \text{Data4}[n])$$

$$\text{Data4}[n-1] = \text{XOR}(\text{Data3}[n], \text{Data5}[n])$$

$$\text{D5}[n-1] = \text{XOR}(\text{Data4}[n], \text{XNOR}(\text{Data1}[n+1], \text{Data3}[n+1]))$$

D5 internally used for 4 bit Parallel PRBS generation.

2<sup>7</sup>-1 4 Bits Parallel PRBS ERROR Detection

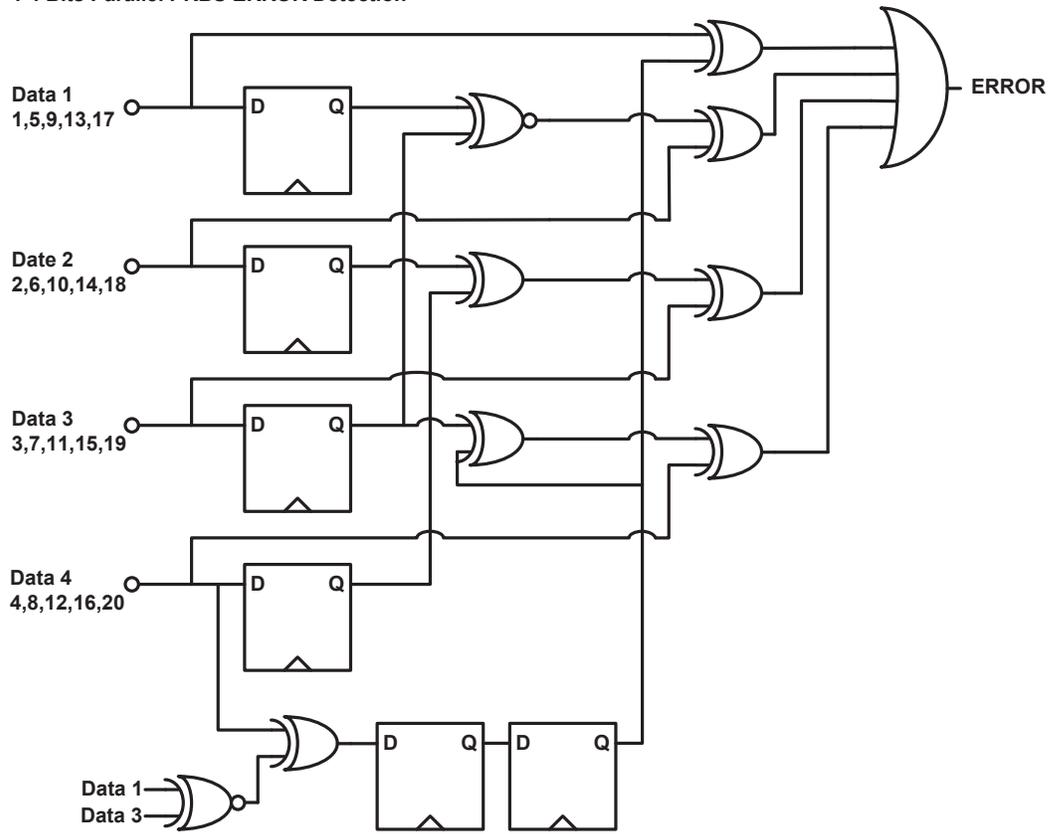


Figure 7 2<sup>7</sup>-1 4 Bits Purposed Parallel PRBS ERROR Detection

Purposed Error detection configuration is simple. We know how to generate Data 1 – 4 based on PRBS generation equation. Therefore it simply compares incoming data and internally generation data. They are XORed, thus it will generate Zero, if they are identical. Finally 4 data are ORed, therefore if one of data has error, it will generate ERROR signal. However, this configuration has to be well design for timing such as match gate delay.

### Simulation Result: 2.5Gbps 4 bits parallel PRBS Generation and PRBS Error detection

$2^7-1$  PRBS 2.5Gbps 4 bits was generate. Figure 8 show if generation data match with receiving date, as we expect, there have no error.

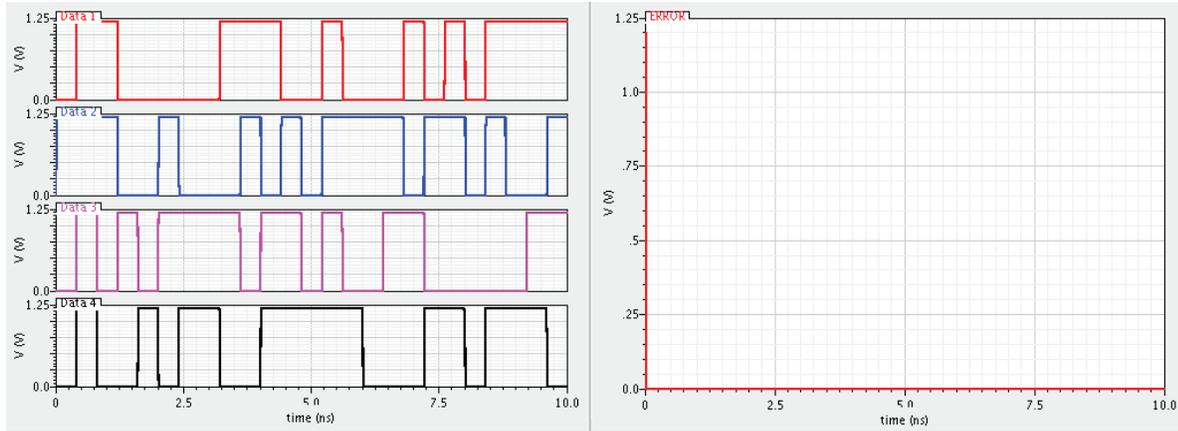


Figure 8 PRBS generation and Detection without ERROR

To see error detection performance, RX data 2 error bit was generated, which is shown in Figure 9. As we can see PRBS error detection detects this ERROR.

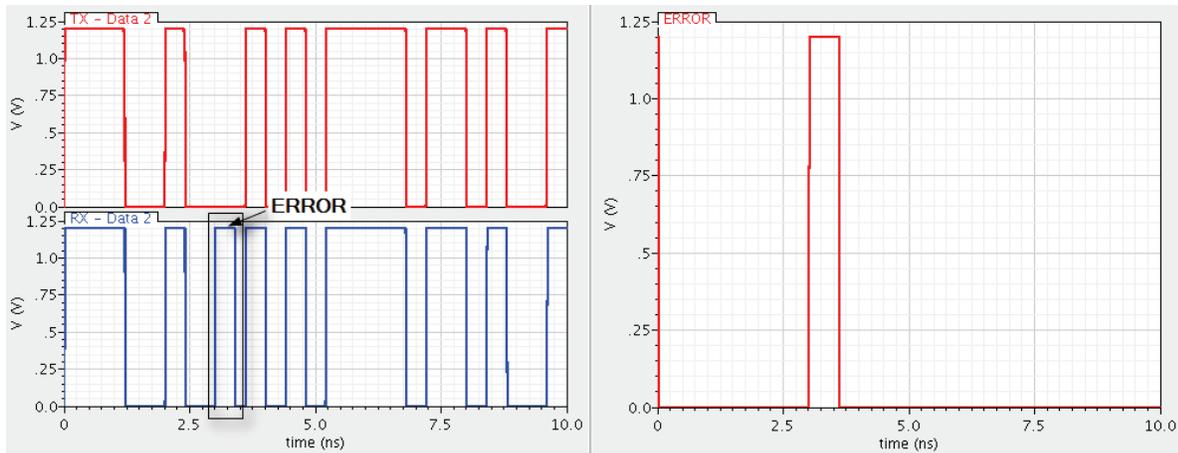


Figure 9 PRBS generation and Detection with ERROR

## Reference

- [1] S. W. Golomb, *Shift Register Sequences*. San Francisco, CA: Holden-Day, 1967.
- [2] F. Sinnesbichler, A. Ebberg, A. Felder, and R. Weigel, "Generation of high-speed pseudorandom sequences using multiplex techniques," *IEEE Trans. Microw. Theory Tech.*, vol. 44, no. 12, pp. 2738–2742, Dec. 1996.
- [3] Seongwon Kim "45-Gb/s SiGe BICMOS PRBS Generator and PRBS Checker" IEEE CICC 2003