IJNRD.ORG ISSN : 2456-4184

INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG

An International Open Access, Peer-reviewed, Refereed Journal

DESIGN OF LOW-POWER SHIFT REGISTER USING GDI TECHNIQUE

[1]K.Kalaichelvi, [2*]C.Nivashini

[1]Assistant professor, Department of Electronics and Communication, VSB Engineering College, Karur.

Abstract **:** For the purpose of testing large-scale GDI circuits with delay-line clocking, propose and demonstrate a shift register. An essential circuit block in cryogenic studies is a pair of serializes, or parallel-to-serial converters, and deserializers, or serial-to-parallel converters. For example, apparatus like a cryostat and cry probe restrict the amount of available input/output (I/O) cables. Therefore, it is essential to minimize the number of I/O cables using a Sir/Des, particularly when testing a large-scale superconductor circuit. Shift registers, which hold data during serial-to-parallel (S2P) and parallel-to-serial (P2S) conversion, are used in RSFQ logic as serializes and deserializers. Regarding GDI logic, we have already suggested and illustrated the feedback-type SR, in which a shift register-like GDI buffer chain with feedback paths functions. However, because feedback routes in delay-line clocking are challenging to create due to the low latency, this Ser/Des was designed for four-phase clocking and does not function in that mode. As a result, we create the GDI hybrid SR, a unique SR for testing delay-line-clocked GDI circuits that combines GDI and RSFQ technology.

Keywords: large-scale GDI circuits, Cryogenic, RSFQ logic, GDI buffer chain.

INTRODUCTION

Power, Area and Speed are main specifications of High Speed serial communication. Depending upon the requirement of these specifications, there are two architecture of PISO.

i. Single Phase Clock based PISO

ii. Multi-Phase Clock based PISO

Single Phase Clock based PISO

Figure 3.1[10] shows the block diagram of a Single-phase clock based PISO. PISO Seri- allies 8 bit parallel data into serial data. The first 8 bit parallel data will convert into 4 bit parallel data using the clock div by 8 after that 4 bit parallel data will convert into 2 bit parallel data using clock div by 4, finally, 2 bit parallel data will convert into serial data using clock div by 2. Here each 2:1 conversion consists of two D flip-flops following by a mux. So for converting 8 bit parallel data into serial data, fourteen D flip flop and seven mux are used but only a single-phase clock is used so it will not be very power-hungry.

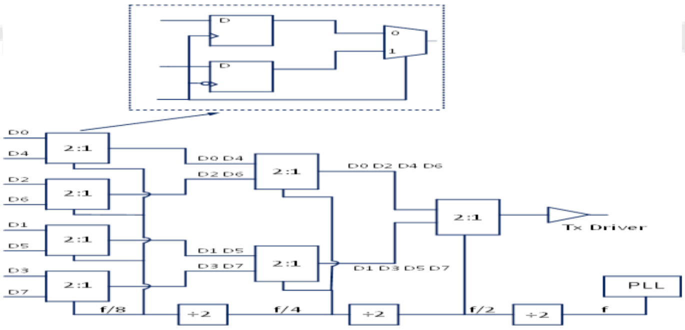


FIGURE 1: Single Phase Clock based PISO

Multi-Phase Clock based PISO

Figure 3.2[11] shows the block diagram of a Multi-phase clock based PISO. PISO Seri- allies 8 bit parallel data into serial data. The first 8 bit parallel data will convert into 4 bit parallel data using the clock div by 8 after that 4 bit parallel data will convert into serial data by using a multi-phase clock. Here first divider block will generate 0, 90, 180, 270 phase clocks, first LSB bit D0, D1, D2 and D3 will serialize with respective phase clock, then MSB bits will be serialized. Here each 2:1 conversion consists of two D flip-flops following by a mux. So for converting 8 bit parallel data into serial data, eight D flip flop and four mux are used, so it will save the area compare to single-phase clock base PISO but to generate different clock phases it will be more power-hungry.

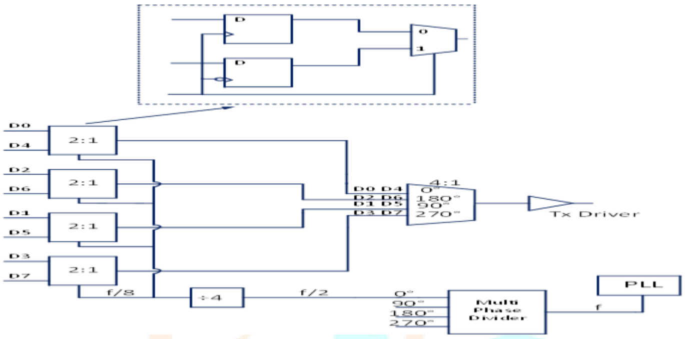


FIGURE 2: Multi Phase Clock based PISO

Comparison between Single and Multi-Phase Clock based PISO

As previously discussed, there are two type of architecture of PISO. One is Single Phase clock based PISO and second is Multi-phase clock based PISO. Each of them has their own challenges in designing. Some of them are listed below:

Single Phase Clock based PISO

Single Phase Clock based PISO has a greater number of D flip-flops and MUX, so these components will increase the overall area of the design. Single Phase clock based PISO will operate at higher clock frequency compare to other architecture, so it will increase the complexity of PLL to generate high frequency of clock Single Phase Clock based PISO is simple binary stage architecture where in each stage you need divide by 2 clocks, so the number of dividers is more in this archi- texture.

Multi-Phase Clock based PISO

Multi-Phase clock based PISO has a smaller number of D flip-flops and MUX compare to single phase clock-based architecture, so it will reduce the overall area of the design.
In this architecture, a divider is needed which generates different phases of clock so it will increase the complexity of overall architecture.
Multi-Phase clock based PISO is more power consuming compare to single phase clock-based architecture.

EXISTING SYSTEM

As discussed in previous chapter, in receiver side, a SIPO block is needed to reduce the bit rate of back end circuit to perform the further signal processing. It is straight forward that SIPO has tree type structure. There are two Architectures of SIPO.

I. Single Stage Architecture based SIPO
II.Binary Stage Architecture based SIPO

Single Stage Architecture based SIPO

Figure 3[6] shows the block diagram of Single Stage Architecture based SIPO. In single-stage architecture, high-speed serial data will directly convert into parallel data. The figure shows a serial data first go into 1:8 DE multiplier (DMUX), after that to synchronized data it will pass through low-speed D flip-flop. Single-stage architecture is easy to implement but it is slow due to large parasitic capacitance at the converging node.

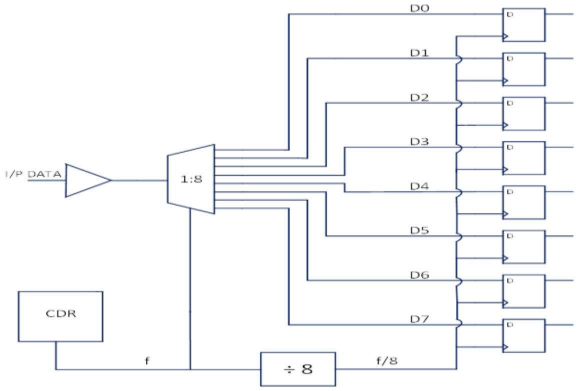


FIGURE 3 Single Stage Architecture based SIPO

Binary Stage Architecture based SIPO

Figure 4[6] shows the block diagram of Binary Stage architecture based SIPO. In single-stage SIPO, speed is low due to high parasitic capacitance at converging node. To solve this problem binary stage architecture is generally used. Here, First serial data converts into two parallel data streams after that each parallel stream further converts into two parallel data streams. So it follows the pattern 1:2:4:8 to convert serial data in parallel data. Binary Stage architecture is faster in among all architecture due to less parasitic capacitance in each stage but the area will be more in comparison to single stage architecture.

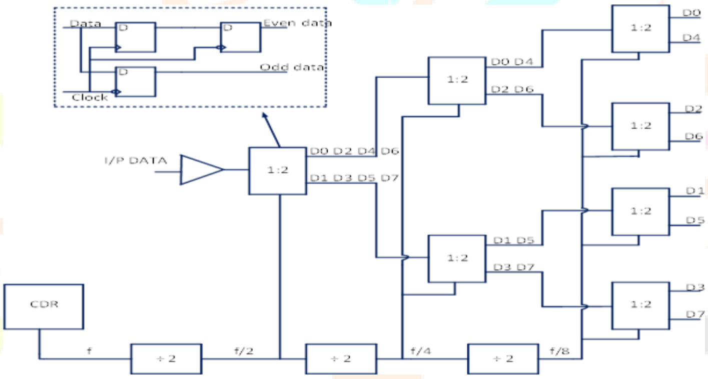


FIGURE 4: Binary Stage Architecture based SIPO

Comparison between Single and Binary Stage architecture based SIPO

As previously discussed, there are two type of architecture of SIPO. One is Single Stage Architecture based SIPO and second is Binary Stage Architecture based SIPO. Each of them has their own challenges in designing.

Some of them are listed below:

Single stage Architecture based SIPO

Single Stage Architecture based SIPO is easy to implement but it is slow due to large number of parasitic capacitance at converging node.

Single Stage Architecture based SIPO has large number of D flip-flop at final stage, so divider will have to generate high driving strength clock so that it will be capable to drive all the flip-flops.Single Stage Architecture based SIPO has no interval stages of converting serial data into parallel data, if there is missing of any data in between than it will become difficult to debug that error after manufacturing the chip.

Multi stage Architecture based SIPO

Binary Stage Architecture based SIPO has a higher number of stages, so overall area is more compare to other architecture.

It has a higher number of dividers to generate clock for each stage, hence it in- creases the complexity of the design for synchronizing each stage data from the previous stage data

PROPOSED SYSTE

High Speed & Low Power Digital Circuits

This Chapter covers different logics to design high speed & low-power custom analog blocks or digital circuits. Section 4.2 covers timing and delay definitions of Flip-Flops and latches. Section 4.3 covers different topologies of Flip-Flop.

Different Logic to design Digital Circuits

CMOS Logic

CMOS logic is by the far most commonly used type of logic circuit. Despite of High speed, CMOS logic is still used in some high speed SR transceiver. The reason is technology scaling down, reduced power supply voltage and robustness of CMOS logic. CMOS logic has pull-up network and a pull-down network. At any time except transitions, either pull-up network is turn on to pull the output voltage to power supply voltage or pull-down network is on to pull down output voltage to ground. Since both the network cannot be turned on simultaneously except during transitions, Therefore, CMOS logic consumes zero static power. Figure 4.1[12] shows a circuit diagram of rail to rail CMOS inverter.

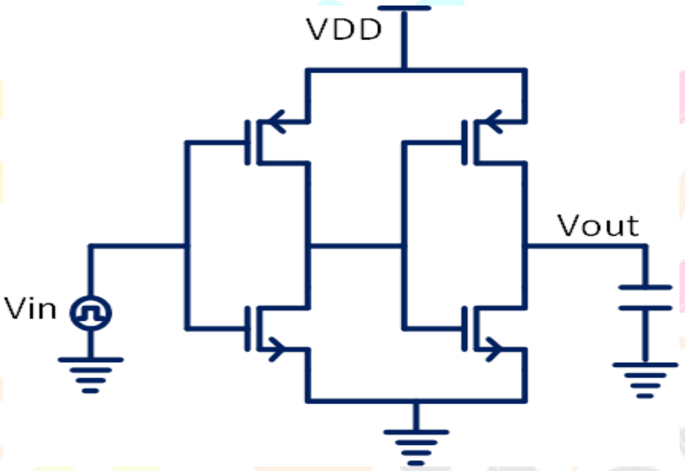


FIGURE 5: Rail to Rail CMOS inverter

The average power consumption of any inverter can be estimated by

P = VddIΔTf = CT otalV 2 f

Some conclusion can be drawn from the basis of simple analysis. Firstly, CMOS logic has to drive large capacitance i.e. parasitic cap, load cap (Pull-up or Pull down), these greatly increases the power consumption according to the equation. CMOS logic consumes more power at high frequencies. CMOS logic is sensitive to common mode noise because it is not differential. Therefore, high-speed CMOS design favors to current mode logic (CML).

CML Logic

CML logic is based on differential pairs as shown in figure 4.2[12]. We can replace PMOS transistor with resistance. Therefore, it is faster than CMOS logic. It is fully differential; therefore, it has excellent immunity to common mode noise. When the input voltage is sufficiently large one of the branches can be switched off, while the other takes all the tail current Io, The minimum input voltage can be derived by following equations, The voltage swing is the product of the load current and the tail current. Therefore, it is possible to reduce the voltage swing to improve the speed of circuit.

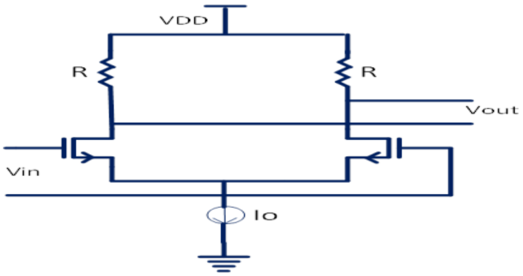


FIGURE 6: CML Logic Differential Pair

The power consumption of CML inverter can be simply estimated by Obviously CML logic consumes static power, the power consumption is independent of frequency. There- fore, CML logic is suitable for high frequency applications in terms of speed and power consumption.

Timing and delay definitions of flip-flops

Building a sequential circuit require memory elements which read a value, save it for some time even if the value has changed. A Boolean logic change its output value as input
change. A memory element has some internal memory and extra circuitry to control it. Internal memory is controlled by clock input, memory element reads input data value according to instructed by clock and stores that in its memory. Memory elements are divided into two major types. Figure 4.3[12][13] shows the difference between positive edges triggered flip-flop and level high latch. As it can be seen from the waveforms latches changes its value when clock is high while flip-flop changes only at the transition of clock from low to high.

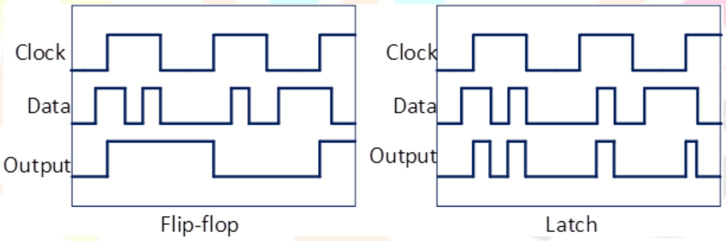


FIGURE 7: Timing Diagram of Flip-flop and Latch

The performance of flip-flop is qualified by three important timings and delays: propagation delay (clock-to-output), setup time and hold time. They reflect in the performance of flip-flop.

Propagation delay

Clock-to-output propagation delay is the amount of time that passes before the output is deemed stable following the arrival of the clock's active edge. The time it takes for an output to change following a clock event is known as clock-to-output. The high-low and low-high transitions have different propagation delays. The flip-flop's propagation delay is therefore the maximum of these two delays by definition.

Setup Time

The period of time before the clock's capturing edge that the synchronous input (D) must appear and be stable is known as the setup time. This is necessary in order for the data to be successfully stored in the storage device. Setup time is by definition the maximum of the values obtained for low-high and high-low transitions, as it may vary for low-high and high-low transitions.

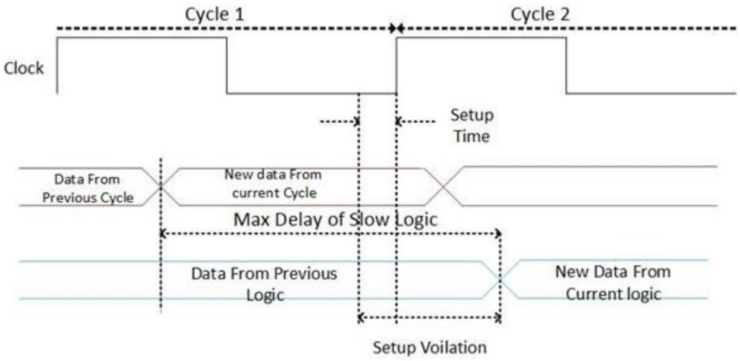


FIGURE 8: Timing Definition of Setup Time

Hold Time

Hold Time is the duration of the synchronous input (D) following the clock's capturing edge that is necessary for the data to be successfully saved in the storage device. Hold time is by definition maximum of the values obtained for low-high and high-low transitions:

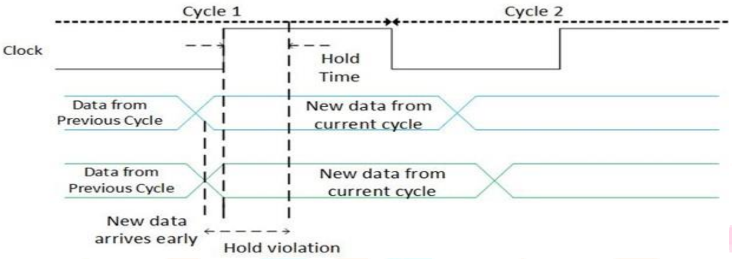


FIGURE 9: Timing Definition of Hold Time

The definitions of setup time hold time and propagation delay are considered as inde-pendent variables. However what happens in reality shows that this parameter is not independent to each other. As it shown in figure 4.6[14] propagation delay increases as data arrives later. When data arrival time is very close to clock edge, clock-to-output delay increases drastically as shown in figure and it is possible that flip-flop function will be changed and it will enter in unstable state called meta-stable state. There are several approaches for setup time definition, one of them is Setup time is the time period before clock edge which causes 5% increase in clock-to-output delay.

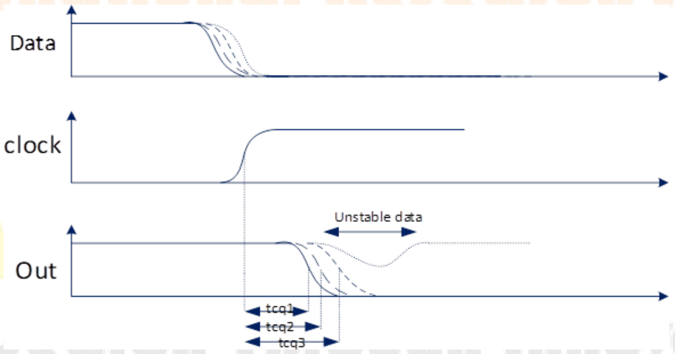


FIGURE 10: Real Timing Definition of Propagation Delay

PISO Architecture

Figure 11 shows the top-level architecture of proposed Serialize. In this design both single and multi-phase architecture are used. Here 40-bit parallel data from PMA will be loaded in serializer block for every rising edge of the clock Tx par clk in. Tx par clk in and sync en are used to enable the synchronizer block to generate sync 2p and sync 2n signals, which will be used to enable shift registers, dividers. Parallel to serial conversion happens successively 40→20→4→2→1. First stage of PISO is 40→20 which is low speed stage and implemented using single phase clock-based architecture. Second stage 20→4 is implemented using multi-phase clock and shift registers-based architecture. Final stages 4→2→1, which are high speed and most power consuming stages are implemented using Differential D flip-flop which has very low timing requirements and less power- hungry compare to other flip-flop topologies. Detail description of data flow and design of each individual block is further explained in following sections.

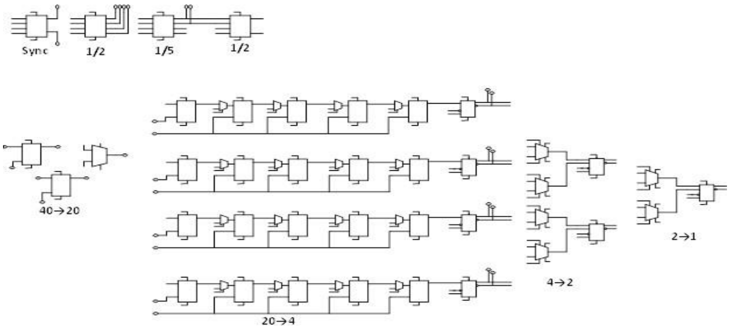


FIGURE 11: Top Level Block Diagram of Proposed PISO

PISO

Figure 12 shows the block diagram of first stage of PISO, Here 40 bit parallel data will be serialized into 20 bit serial data. Parallel data launched from PMA is sampled with internally generated clk40p clock, needed for synchronization between data and clock. MSB 20 bits are further sampled and delayed by falling edge clk40n. Synchronized LSB 20 bits passes through mux when clk40n is low and delayed MSB 20 bits passes when clk40n is high. 40 bit parallel data is serialized into 20 bits after first stage of PISO.

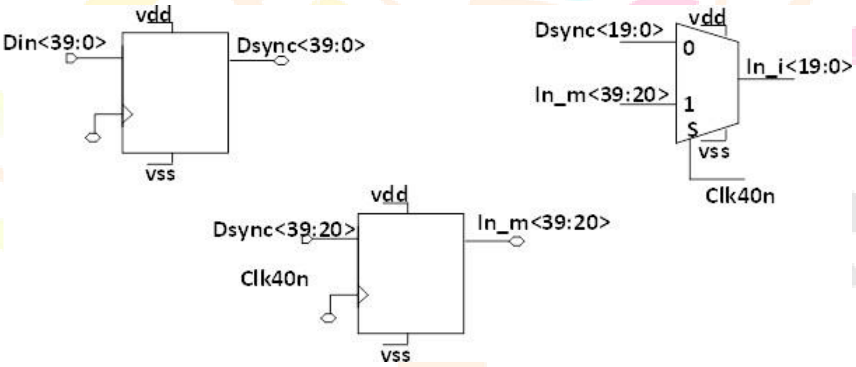


FIGURE 12: Block Diagram of PISO

PISO 20→4

PISO first stage converts 40 bit parallel data into 20 bit data. Figure 5.3 shows the block diagram of second stage of PISO. Here data is loaded on to 20 flip-flops, based on the selection signals which are phase shifted by 90◦. Selection signals Sel dig0◦, 90◦, 180◦, and 270◦ are high for one clock cycle of Clkp 0◦,90◦, 180◦, and 270◦ respectively and low for four clock cycles. When selection signals are high all the parallel data is loaded on the flops and when it is low data is pushed out of the shift register serially. The 20→4 stage has two purpose:

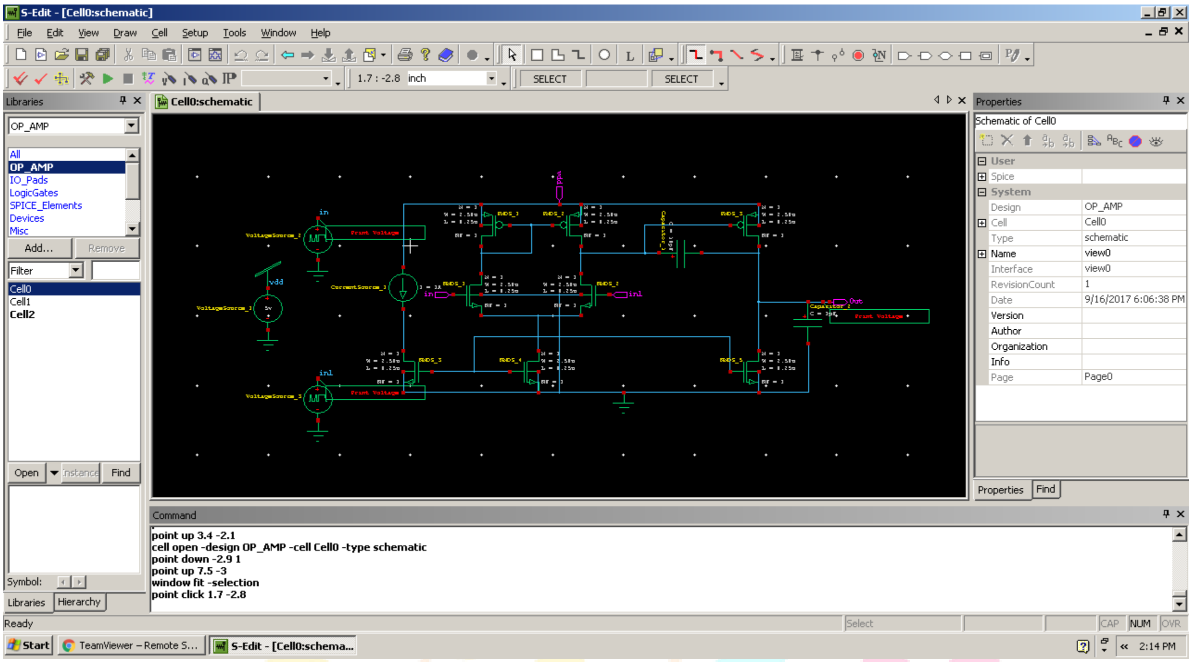
1. Analog (from previous bit) or digital data select (with Sel dig0), Each Mux flop cells shift its previous cells data or (when sel dig is up) select new data.
2. After the Mux flop shift register array we have high speed latch which verify that the data is its appropriate phase. Also high speed latch is used because of higher time margins should be available to avoid post layout variations in setup and hold of flops and latches.

FIGURE 13: Block Diagram of PISO 20→4

SEDIT DESIGIN IMPLEMENTAION

CML Logic

CML logic is based on differential pairs as shown in figure 4.2[12]. Resistance can be used in place of PMOS transistors. It is therefore quicker than CMOS logic. Because it is entirely differential, common mode noise is effectively suppressed. One of the branches can be turned off and the other can soak up all of the tail current Io when the input voltage is high enough. The following formulas can be used to determine the minimum input voltage: The product of the tail current and the load current is the voltage swing. Consequently, it is feasible to increase circuit speed by lowering voltage swing.

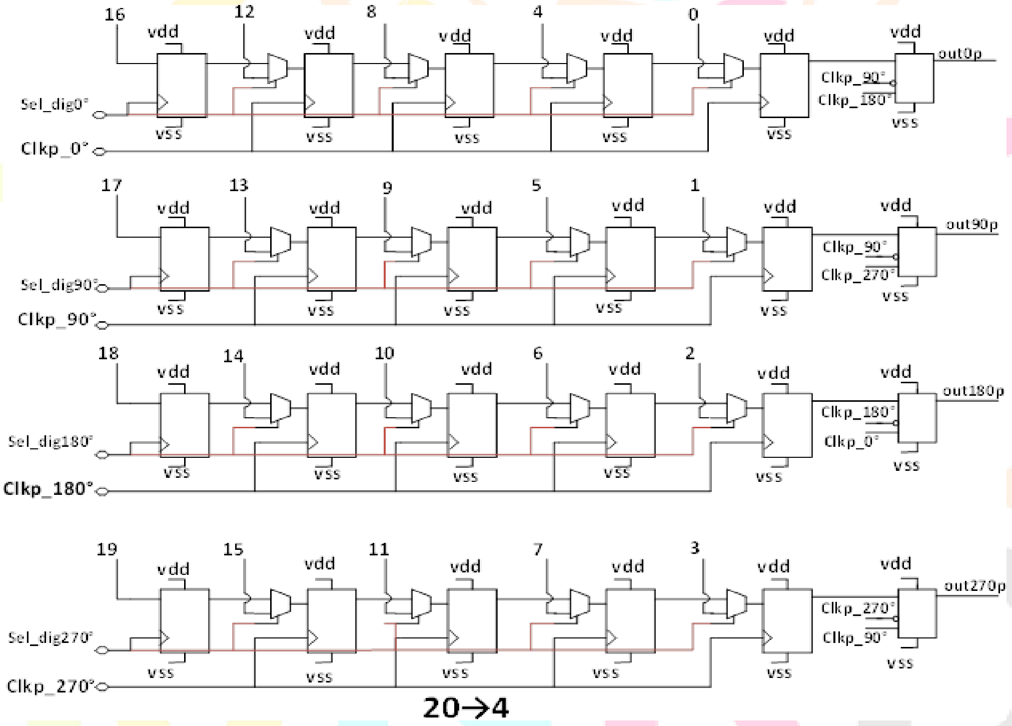


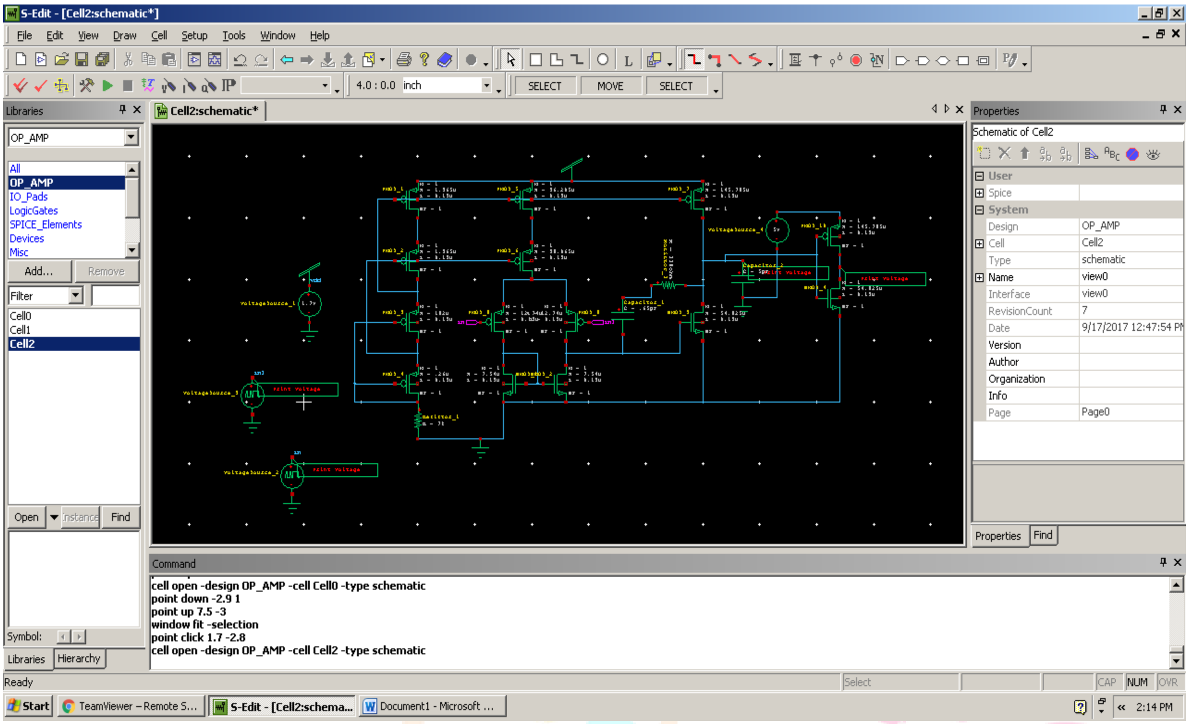
1.7 : -2.8 inch

Schematic of Cell0

Design	OP_AMP
Cell	Cell0
Type	schematic
Name	view0
Interface	view0
RevisionCount	1
Date	9/16/2017 6:06:38 PM
Version	
Author	
Organization	
Info	
Page	Page0

```
point up 3.4 -2.1
cell open -design OP_AMP -cell Cell0 -type schematic
point down -2.9 1
point up 7.5 -3
window fit -selection
point click 1.7 -2.8
```





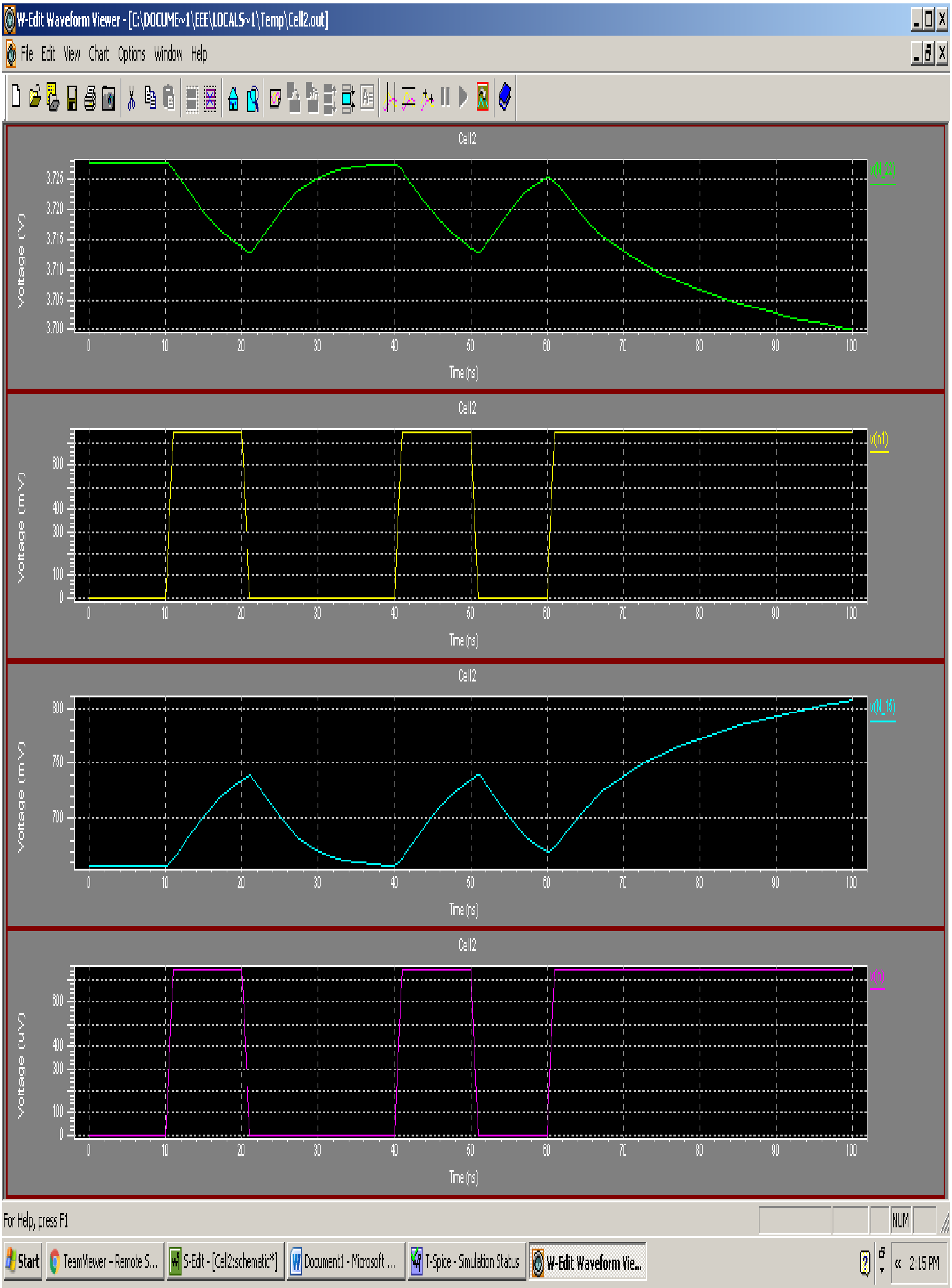
Properties

Schematic of Cell2

⊟ User	
⊞ Spice	
⊟ System	
Design	OP_AMP
⊞ Cell	Cell2
Type	schematic
⊞ Name	view0
Interface	view0
RevisionCount	7
Date	9/17/2017 12:47:54 PM
Version	
Author	
Organization	
Info	
Page	Page0

Command

```
cell open -design OP_AMP -cell Cell0 -type schematic
point down -2.9 1
point up 7.5 -3
window fit -selection
point click 1.7 -2.8
cell open -design OP_AMP -cell Cell2 -type schematic
```



Hold Time is the duration of the synchronous input (D) following the clock's capturing edge that is necessary for the data to be successfully saved in the storage device. Hold time is by definition maximum of the values obtained for low-high and high-low transitions:

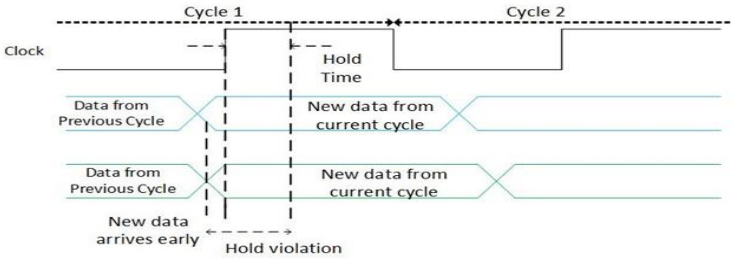
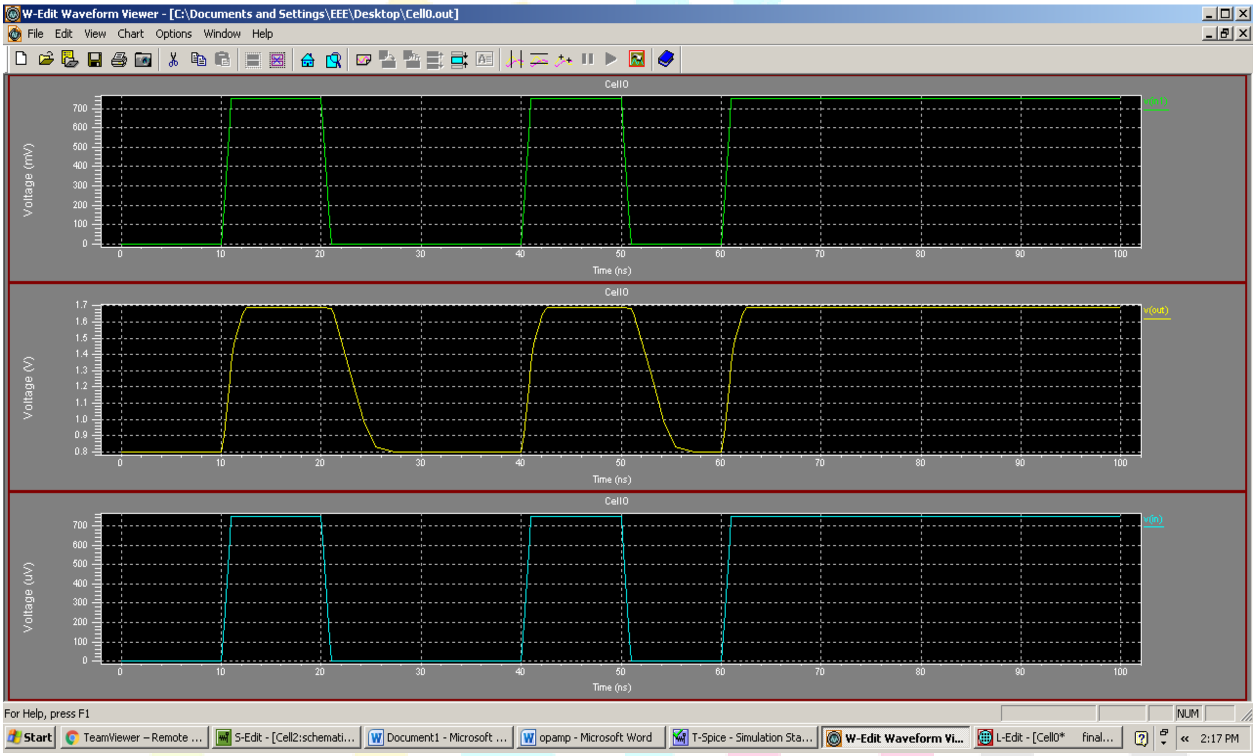


FIGURE 9: Timing Definition of Hold Time

Setup time, hold time, and propagation delay definitions are regarded as independent factors. But actual events demonstrate that these parameters are not independent of one another. Figure 4.6[14] illustrates how propagation delay rises with later data arrival. The clock-to-output latency grows sharply when the data arrival time approaches the clock edge, as seen in the image. This can cause the flip-flop function to alter and put it in an unstable state known as the meta-stable state. One method for defining setup time is to define it as the amount of time prior to the clock edge that results in a 5% increase in the clock-to-output delay.



TEDIT CODE IMPLEMENTATION

```
* Circuit Extracted by Tanner Research's L-Edit Version 13.00 / Extract Version 13.00 ;
* TDB File:  C:\Program Files\Tanner EDA\Tanner Tools v13.0\opamptest.tdb
* Cell:  Cell0  Version 1.08
* Extract Definition File:  ..\..\..\Documents and Settings\EEE\My Documents\Tanner EDA\Tanner Tools v13.0\L-Edit and LVS\Tech\Generic0_25um\Generic_025.ext
* Extract Date and Time:  09/21/2017 - 14:14
.include "E:\software\software\Tanner Tools V 13 With Crack\T88F.md"
```

NODE NAME ALIASES

```
*       3 = vdd (43.8 , 8.25)
*       5 = out (52.3 , 2.1)
*       6 = Gnd (45.95 , -4)
*       13 = in (46.85 , 2.1)
*       14 = in1 (49.5 , 2.05)
M1 4 in1 8 2 PMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (48.1 1.95 48.85 2.2)
M2 10 9 11 2 PMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (41.3 2.95 42.05 3.2)
M3 8 9 12 2 PMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (43.45 2.95 44.2 3.2)
M4 Gnd Gnd 9 2 PMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (41.3 -3.4 42.05 -3.15)
```

```
M5 9 Gnd 10 2 PMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (41.3 -0.45 42.05 -0.2)
M6 11 10 vdd 2 PMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (41.3 5.8 42.05 6.05)
M7 12 10 vdd 2 PMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (43.45 5.8 44.2 6.05)
M8 out 10 vdd 2 PMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (45.7 5.8 46.45 6.05)
M9 7 in 8 2 PMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (45.6 1.95 46.35 2.2)
M10 Gnd 4 out 1 NMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (51.85 -0.75 52.6 -0.5)
M11 Gnd 7 4 1 NMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (48.1 -1.95 48.85 -1.7)
M12 Gnd 7 7 1 NMOS L=250n W=750n AD=562.5f PD=3u AS=562.5f PS=3u    $ (45.6 -1.95 46.35 -1.7)

VVoltageSource_1 vdd Gnd  DC 1.7
VVoltageSource_4 N_20 Gnd  DC 5
VVoltageSource_2 in Gnd  BIT({0100101111} ON=750u )
VVoltageSource_3 in1 Gnd  BIT({0100101111} ON=750m )
.PRINT TRAN V(in)
.PRINT TRAN V(out)
.PRINT TRAN V(in1)
********* Simulation Settings - Analysis section *********
.tran 1ns 100ns start=0ns
.option prtdel=0ns
```

TIME ANALYSIS

```
* Total Nodes: 14
* Total Elements: 12
* Total Number of Shorted Elements not written to the SPICE file: 0
* Output Generation Elapsed Time: 0.015 sec
* Total Extract Elapsed Time: 1.265 sec
.END
```

Conclusion

We proposed the GDI hybrid SR toward the testing of large-scale GDI circuits with delay-line clocking. The hybrid SR is made up of GDI interfaces for data transmission between the shift registers and the GDI CUT and RSFQ shift registers for data storage during S2P and P2S conversion. Moreover, all the component circuits in the hybrid SR are seamlessly clocked by a single excitation current to synchronize the GDI and RSFQ parts. We fabricated and tested an 8-to-3 encoder integrated with the hybrid SR, which is the largest delay-line-clocked GDI circuit ever designed, at 4.2 K up to 4.5 GHz, thereby demonstrating that the hybrid SR enables the testing of delay-line-clocked GDI circuits with only a few I/O cables at high clock frequencies. Our next step is to demonstrate even larger GDI circuits using the hybrid SR.

Future Work

The current work, Serialize and Deserializer are supporting data rate up to 28Gbps with power consumption less than (< 10mw) for PISO and less than (< 4mw) for SIPO. This work can further be improved for higher data rate and reduce power consumption. In PISO this work can extended by tapping pre, main and post data. In SIPO, less number of flip-flops can be used so that it will reduce area for given data rate and power.

Reference

[1] Zexian Li. Design of serializer and deserializer operating in 65 nm cmos technology for high-speed serial link (hssl) applications. Senior Thesis in Electrical Engineering University of Illinois at Urbana-Champaign.

[2] Rishi Ratan. Design of a phase locked loop based clocking circuit for high speed serial link applications. University of Illinois at Urbana-Champaign, 2014.

[3]F.M.Gardner. Charge-pump phase-lock loops. IEEE Trans.Commun, COM-28 (11):18491858, 1980.

[4] Hanqiao Zhang Steven Krooswyk Jeff Ou. High speed digital design, design of high speed interconnects and signaling. Elsevier Inc., 2015.

[5] Razavi. Design of integrated circuits for optical communication. 2003.

[6] H. Tao. 40-43-gb/s) oc-768 16:1 mux/cmu chipset with sfi-5 compliance. IEEE J. Solid- State Devices, 38(12):2169–2180, 2003.

[7] M. Chen A. A. Hafez and C. K. Yang. A 32-48 gb/s serializing transmitter using multiphase serialization in 65 nm cmos technology. IEEE J. Solid-State Circuits, 50(3): 763–775, 2015.
[8] S. Sidiropulos Horowitz M., Chih-kong Yen Yang. High-speed electrical signaling: Overview and limitations. IEEE Micro, 18(1):12–24, 1998.
[9] shraghian K. Weste N. H. E. Principles of cmos vlsi design, a systems perspective. second edition, 1994.References
[10] Stojanovic V. Oklobdzija V.G. Comparative analysis of master-slave latches and flip- flops for high-performance and low-power systems. IEEE, 34:536–548, 1999.
[11] Nikolic B. Rabaey J. M., Chandrakasan A. Digital integrated circuits, a design perspec- tive. second edition, 2003.
[12] Brodersen R.W. Markovic D. Nikolic B. Analysis and design of low-energy flip-flops.
[13] International Symposium on Low Power Electronics and Design, pages 52–55, 2001.
[14] Yuan J. Sevensson C. New single-clock cmos latches and flip-flops with improved speed and power saving. IEEE Journal of Solid-State Circuits, 32:62–69, 1997.
[15] M. Chen and C. K. Yang. A 5064 gb/s serializing transmitter with a 4-tap, lc-ladder- filter-based ffe in 65 nm cmos technology. IEEE J. Solid-State Circuits, 50(8):1903– 19016, 2015.Amr Elshazly Yan-Yu Huang Hang Song Kai Yu Frank OMahony Jihwan Kim, Ajay Bal- ankutty.
[16] A 16-to-40gb/s quarter-rate nrz/pam4 dual-mode transmitter in 14nm cmos. IEEE International Solid-State Circuits Conference, pages 60–63, 2015.