

High Speed Receiver Circuit for On-Chip Communications

Yongxin Zhang

High Speed Receiver Circuit for On-Chip Communications

Research Thesis

In partial fulfillment of the requirements for the Degree of Master of Science in
Electrical Engineering

Yongxin Zhang

Submitted to the Senate of
the Technion – Israel Institute of Technology

Tishrei 5775 Haifa October 2014

This research was carried out under the supervision of Prof. Ran Ginosar and Dr. Aharon Unikovski in the Department of Electrical Engineering.
The generous financial help of the Technion Graduate School is gratefully acknowledged.

Contents

List of Figures

List of Tables

Abstract	1
Abbreviations and Notations	3
1 Introduction	5
1.1 Research motivation	5
1.2 FOX2 implementation and research goals	7
1.3 Outcomes of this research	8
1.4 Organization of the thesis	10
2 Asynchronous high-speed bit-serial architecture	11
2.1 Fast clock generator	12
2.2 SR at the TX side	20
2.3 LEDR encoder	22
2.4 Toggle and splitter circuit	24
2.5 SR at the receiver (RX) side	28
3 High speed analog link	31
3.1 Introduction	31
3.2 Transmission line model	31
3.3 Analog transmitter circuit	36
3.3.1 Analog current mode transmitter	36
3.3.2 Simple inverter driver	38
3.4 Analog receiver circuit	38
3.4.1 RGC TIA configuration	38
3.4.2 Simple inverter receiver	44
3.4.3 Differential RGC “TIA”amplifier	45

4 Test circuit	47
4.1 Speed measurement circuit	47
4.2 Test circuit for clock generator	49
4.3 Test circuit for RGC TIA configuration	50
4.4 Test circuit for 2T 2X configuration	51
5 Simulations and test results	53
5.1 Performance of the analog link	53
5.2 Performance of the chip from simulation	56
5.3 Performance of the chip in the lab	62
5.3.1 2T 2X test	62
5.3.2 RGC TIA test	63
5.3.3 CGC test	64
5.3.4 CLKOS test result	66
5.3.5 Link # 0 functionality test	67
5.3.6 Link # 0 performance analysis	67
5.3.7 Interesting result for link # 29	69
6 Summary and conclusions	71
6.1 Research summary	71
6.2 Conclusions	72
6.2.1 Simulation result	72
6.2.2 Test result	73
6.3 Updates from FOX1	73
6.3.1 Digital controller backend process	73
6.3.2 Mixed signal integration method	73
6.4 Recommendations for future research	75
6.4.1 Transmission line model	75
6.4.2 Distributed power supply	75
6.4.3 Use high supply voltage	76
6.4.4 Lef file generation	76
6.4.5 De-cap for fast circuit	76
7 Appendices	77
7.1 FOX2 Digital controller	77
7.1.1 DIN bits format	77
7.1.2 DOUT bits format	78
7.1.3 FSM	79
7.2 Backend flow of the digital controller	80
7.2.1 Introduction	80
7.2.2 Flow	80
7.3 Integration of TX & RX and add de-cap	81

7.3.1	Introduction	81
7.3.2	TX Integration	81
7.3.3	RX Integration	83
7.4	Analog Mixed Signal Integration	84
7.4.1	Introduction	84
7.4.2	Flow	84
7.5	Package and pads	85
7.6	FOX2 Test Environment	90
7.6.1	High Level Testing Architecture	90
7.6.2	PC Interface – Digital LabView Control GUI and Interface	91
7.6.3	PC Interface – Analog LabView Control GUI and Interface	94
7.6.4	FOX2 PCB board schematic	96
7.6.5	Board Testing Procedure	98

Bibliography

101

List of Figures

1.1	Metal stack cross section	7
1.2	Chip layout in Encounter	8
1.3	Chip layout in Virtuoso	9
2.1	System architecture [18]	11
2.2	Clock generator diagram [14] [18]	12
2.3	Clock generator schematic diagram [18]	13
2.4	Schematic of the single delay element	13
2.5	Symbol of the single delay element	14
2.6	Schematic for the voltage controlled current mirror	14
2.7	Symbol for the voltage controlled current mirror	14
2.8	Schematic for delay elements [18]	15
2.9	Delay elements simulation	15
2.10	Layout for delay elements	16
2.11	Schematic of pass-gate based XOR [18]	16
2.12	Simulation of XOR	17
2.13	XOR layouts	17
2.14	Schematic of CGC [18]	18
2.15	XOR symbols	18
2.16	Simulation result of CGC	19
2.17	Layout of CGC	19
2.18	XL schematic and symbol	20
2.19	Schematic for SR at TX side [18]	20
2.20	Schematic of test circuit for TX SRs	21
2.21	Simulation of TX SRs	21
2.22	Layout of TX SR	22
2.23	LEDR protocol	22
2.24	Schematic of LEDR encoder [18]	23
2.25	Simulation of LEDR encoder	23
2.26	Layout of LEDR encoder	24
2.27	Schematic for toggle circuit [18]	25
2.28	Simulation of toggle circuit	25

2.29	Layout for toggle circuit	26
2.30	Schematic for splitter circuit [18]	26
2.31	Simulation of splitter circuit	27
2.32	Layout for splitter circuit	27
2.33	Schematic for RX SR [18]	28
2.34	Simulation of RX SR	29
2.35	Layout for RX SR	29
3.1	Transmission line RLC model	32
3.2	Transmission line in HFSS	32
3.3	Coupled line model	33
3.4	Basic line, M6 line, M5 ground	33
3.5	Increase Z_0 , M6 line, M1 ground	34
3.6	Decrease R, M6&M5 line, M4 ground	34
3.7	Increase Z_0 and decrease R, M6&M5 line, M1 ground	34
3.8	Imported layout in HFSS	35
3.9	Layout geometry at the other side	36
3.10	Basic current mode transmitter	36
3.11	Basic current mode transmitter with adaptive control	37
3.12	TX schematic and layout	37
3.13	Simple inverter driver	38
3.14	Basic common gate (CG) analog receiver	39
3.15	RGC schematic diagram	39
3.16	RGC TIA schematic diagram [23]	41
3.17	Modification of RGC TIA # 1	41
3.18	Modification of RGC TIA # 2	42
3.19	Circuit under optimization	42
3.20	RGC “TIA”after optimization	43
3.21	RGC “TIA”layout	44
3.22	Simple inverter receiver	45
3.23	Schematic diagram of differential amplifier	45
3.24	Layout of differential RGC “TIA”amplifier	46
4.1	Schematic for ring oscillation circuit	47
4.2	Schematic 10 stages D-FF	48
4.3	Simulation of speed measurement circuit	48
4.4	Layout for the speed measurement circuit	48
4.5	Schematic of test circuit for CGC	49
4.6	Simulation of CGC test circuit	49
4.7	Schematic of test circuit for RGC “TIA”	50
4.8	Simulation of RGC “TIA”test circuit	50

4.9	Schematic of test circuit for 2T 2X	51
4.10	Simulation of 2T 2X test circuit	51
4.11	Schematic of test circuit for 2T 2X and RGC “TIA”	52
5.1	Chip test bench	56
5.2	Simulation result for speed measurement circuit	56
5.3	Simulation result for RGC “TIA” and 2T 2X circuit	57
5.4	Simulation result for CGC test circuit	57
5.5	Input data to the digital controller	58
5.6	TX SRs data	59
5.7	Speed control pattern	59
5.8	Waveform on the channel	60
5.9	RX SRs data	60
5.10	Read DOUT from the chip	61
5.11	Read out the speed control bit	61
5.12	Read out the FSM, BER and FER bit	61
5.13	FOX2 test board	62
5.14	2T 2X test circuit	63
5.15	RGC “TIA” test circuit	63
5.16	RGC “TIA” test circuit	64
5.17	CGC simulation result	64
5.18	CGC simulation result	65
5.19	CGC simulation result	65
5.20	CGC test result	65
5.21	CLKOS test result	66
5.22	BER, FER, NTO vs. Frequency	68
5.23	Revised BER vs. Frequency	69
5.24	Receive 0X0040 correctly	70
5.25	Receive 0X0040 wrongly	70
6.1	Digital controller backend process	74
6.2	NMOS De-cap capacitor in FOX2	76
7.1	FOX2 finite state machine	79
7.2	Digital backend flow	80
7.3	Schematic of TX in one link	81
7.4	Layout of TX without de-cap	81
7.5	Schematic of NMOS de-cap	82
7.6	Layout of TX with de-cap	82
7.7	Schematic of RX	83
7.8	Layout of RX without de-cap	83
7.9	Layout of RX with de-cap	83

7.10 Chip package	85
7.11 Package board diagram	88
7.12 FOX2 package	89
7.13 High Level Testing Architecture	90
7.14 FPGA state diagram	91
7.15 Digital Control GUI in LabView – Front Panel	92
7.16 Digital Control GUI in LabView – Backend Panel	94
7.17 Analog Control GUI in LabView – Front Panel	95
7.18 Analog Control GUI in LabView – Backend Panel	96
7.19 FOX2 PCB board layout	97

List of Tables

2.1	Map between signal pins and input bits	21
2.2	Map between signal pins and output bits	28
3.1	Transmission line performance (at the frequency of 5 GHz)	35
5.1	Link configuration and performance summary	54
5.2	CLKOS speed with different corners	66
5.3	Link # 0 test	67
7.1	Digital controller input bits format	77
7.2	Digital controller output bits format	78
7.3	Package information	85
7.4	FOX2 pins	88
7.5	Digital Test board Interface	93
7.6	Analog Test board Interface	96
7.7	Serial link performance table	100

Abstract

As System-On-Chip (SOC) integrates a growing number of modules, and since global On-Chip Interconnect does not scale with technology, it's a big challenge to achieve the long range high-speed on-chip data communication. The typical bit-parallel on-chip communication solutions may suit for the high data rate demand but they induce a high cost of area, noise and power and also growing routing difficulties.

A high-speed bit-serial link that incorporates fast clock generator, two-phase non return to zero (NRZ) Level Encoded Dual Rail (LEDR) asynchronous protocol, serializer and de-serializer using fast asynchronous shift registers, LEDR decoder and differential channel encoding is described. This link can enable one gate (FO4) delay, which is around 10Gbps under the Tower 180nm Technology, and is employed in this research.

The research deals with the design of the high-speed asynchronous analog transmitter and receiver circuit over 10mm transmission line that can satisfy the one FO4 delay requirement. A current mode (CM) analog transmitter and receiver pair is explored. Compared with the voltage mode (VM) circuit, it can give lower swing, lower dynamic power and enable longer distance and faster operation. A typical kind of analog CM TX circuit is adopted and an improved version is proposed for our asynchronous application. Due to the high capacitance of the long interconnect in restraining the bandwidth, a special current mode analog receiver is employed.

Common gate trans-impedance receiver is widely used for the wide bandwidth application. From the optical communication community, we adopted an enhanced common gate trans-impedance configuration – the regulated-cascaded trans-impedance amplifier (RGC TIA) which is more effectively in relaxing the big input capacitance from bandwidth determination. Due to the different applications from the optical communication, mainly because of our large signal consideration, we make some modifications to the circuit. After optimization, we found when no feedback from the second amplification stage, we will get the highest performance.

From the result in RGC “TIA”, we propose a very simple circuit for the high speed operation. Just use 2 stages of inverter as the analog TX and another 2 stages of the inverter as the analog RX. It shows better results both for the power and speed consideration.

To get a reliable link performance from simulation, the RLC model for the transmission line is used. We use HFSS electro-magnetic solver to get the parameters of the RLC model at a certain frequency. For higher accuracy, I exported the layout of the transmission line from Cadence to HFSS to get the S-parameter of it. After getting the S-parameter file, we export it back to Cadence and it should be a very accurate transmission line model.

To prove the concept, a test chip (FOX2) is designed and fabricated which consists of 30 links with different transmission line lengths (the longest in the layout is around 7mm) and different operational modes (CM/VM). Because of limited available metal layers, only M6 layer is used as transmission line. In order to test the chip, the package of the chip and a PCB board that can hold the chip are designed to do the test. We use a NI-sbRIO9642 board to control the board through a LabView Graphic User Interface – GUI.

Test result shows the chip works and it can communicate with the LabView GUI. The speed of the chip corresponding to FO4 delay is around 8.7Gbps based on the result of the RingOS test circuit. However, due to some unknown reasons, many links cannot get the correct data after the link so it's hard to declare the performance of the chip. One voltage mode link with transmission line length of 1.74mm can work without bit error until the frequency of 4.1Gbps under the standard 1.8V supply voltage. So the digital circuit in the link can work at the speed of 4.1Gbps at least.

Abbreviations and Notations

AMS	: Analog Mixed Signal
ASC	: Autonomous Serial Communication
BER	: Bit Error Rate
BW	: Bandwidth
CDL	: Circuit Design Language (a netlist format)
CDR	: Clock Data Recovery
CG	: Common Gate
CGC	: Clock Generator Circuit
CM	: Current Mode
CMNR	: Common Mode Noise Rejection
DFAM	: Differential Amplifier for RGC TIA configuration
DRC	: Design Rule Check
FER	: Frame Error Rate
FF	: Flip-Flop
FOX	: Fast On-Chip Interconnect
FO4	: Fan Out of 4
FPGA	: Field-Programmable Gate Array
FSM	: Finite State Machine
GALS	: Global Asynchronous Local Synchronous
GUI	: Graphic User Interface
IP	: Intellectual Property
LEC	: Logic Equivalent Check
LEDR	: Level Encoded Dual Rail
LVDS	: Low voltage differential swing
LVS	: Layout Versus Schematic
NOC	: Networks on chip
NRZ	: Non Return to Zero
NTO	: Number of Time Out
PLL	: Phase Locked Loop
RGC	: Regulated Cascode
RTZ	: Return to Zero
RX	: Receiver

SCTL	:	Single Cycle Timed Loop
SerDes	:	Serializer – Deserializer
SI	:	Signal Integrity
SOC	:	System On Chip
SR	:	Shift Register
TIA	:	Trans-impedance Amplifier
TX	:	Transmitter
VLSI	:	Very Large Scale Integration
VM	:	Voltage Mode
WAFT	:	Wave front train structure
WP	:	wave-pipelining
XL	:	Transition Latch
2T 2X	:	2 Inverters as TX and 2 Inverters as RX

Chapter 1

Introduction

Thanks to the transistor size scaling, the performance of the VLSI digital logic has been increasing at an exponential rate over decades, which enables the die integrate more transistors and work at higher frequencies, dissipate less power and cost less [2]. This trend enables the decrease of gate capacitive load, which reduces the gate latency and achieves higher performance for micro-processor [18].

However, due to the technology scaling, the interconnect between different modules has become non-negligible even for short distance [18]. Actually the high-capacitance global interconnect has become one of the main sources of the losses over the wire, which degrades the performance in term of throughout and power [11].

Bit-parallel links are able to provide high data rate and they are usually employed to overcome this speed bottleneck [18]. However, parallel links always occupy large area, cause high routing congestion, high noise and dynamic power as well as leakage power over transmission lines [18]. Moreover, as System On-Chip (SOC) integrates a growing number of modules, with the growing number of inter-modular communications, the modules have to turn to serial interfaces [11].

High-speed serial links require less interconnect area and will not result in routing congestion compared with parallel links [18]. For the links that are longer than a certain length, the serial links will outperform the parallel links in terms of power and area and this improvement grows with technology scaling [12] [18].

1.1 Research motivation

Traditional synchronous serial link employs clock signal, whose frequency is limited by clock and data uncertainty, while the clock and data uncertainty are deteriorated with the long link length [11]. Hence, it is not able to satisfy the high speed requirement of the link.

Novel synchronous serial links are often used for off-chip communications but they require complex clock data recovery (CDR) circuits, which usually require a power hungry Phase-Lock-Loop (PLL) [11]. While this PLL often takes a long time to converge on a certain frequency and phase at the beginning of each transition [11] [18]. Moreover, it is also limited to local clock speed which cannot handle the fast transmission

Globally Asynchronous Locally Synchronous (GALS) Networks on chips (NOCs) configuration is very useful for high speed data transmission between different modules on SOC [4] [18]. Hence, it makes sense to construct high-speed asynchronous links for long distance on-chip communication and leave single modules work synchronously. High speed asynchronous links always employ wave-pipelining (WP) [5] [34] [35]. The structure explored in [34] achieves the transmission speed 3.45Gbps on 10mm length interconnect on 250nm technology. It belongs to the GALS scheme and shows more efficient power and area saving with relative simpler design [18]. It also shows the energy efficiency of the WP than non-WP for up to 60% and 50% with respect to area [18] [35]. Wave-front train structure (WAFT) was proposed in [15] to replace the conventional flip-flop (FF) for synchronizing in the serializing stage, which is able to control the wave-fronts propagation speed in the serializer [18]. The result is 3Gbps speed in 180nm technology [15]. An Autonomous Serial Communication (ASC) was shown in [26] [29] and it makes easier to use general interconnect Intellectual property (IP) [18]. Impedance tuning, error correction (Hamming distance 3 code) with a package transfer are incorporated in this research, which enables a 5Gbps link in 250nm technology. A low voltage differential swing (LVDS) with a 3-level asynchronous protocol is introduced in [30]. Under this technique, the power consumption was further reduced and with only one edge is needed for each bit, thus the bandwidth (BW) was also effectively utilized. 1Gbps data rate is achieved for this technology in a standard $1.2\mu\text{m}$ CMOS process [18] [30].

The high capacitance of the long interconnect wire is the main contributor to the signal degradation along the link [11]. At the signaling level, the fast full-swing voltage mode (VM) transitions incur high dynamic current, which dissipate much power and cause cross-talk noise [11] while current mode circuits can support the fast transmission over long links [18]. Current mode sense amplifiers are commonly employed in optical communication, in which diode's small currents are measured and amplified [6] [21] [25] [27]. It was shown that the current mode methods dissipate less power and can achieve faster transmission over longer distance compared with repeater insertion voltage mode method, thus current mode signaling is commonly utilized in interconnect [3] [17] [19] [20] [31] [32] [36]. Current mode LEDR encoding was employed in [19] and it achieved 1Gbps data rate over 5mm transmission line under 130nm Technology.

One architecture that incorporated LEDR encoding scheme, differential current mode signaling and wave-pipelining techniques was proposed in [11]. It can achieve the single

gate (FO4) delay, which is around 67 Gbps for 65nm technology. However, the current mode signaling over the channel was not fully explored and it could just support 28Gbps data rate over 5mm transmission line. In this work, I adopt this architecture and redo the high-speed digital logic parts, for example LEDR encoder, in 180nm technology with FO4 delay around 10Gbps. The novel part of this work is to design a high-speed current-mode analog receiver circuit with low input impedance that can realize the low-swing voltage over the line and satisfy the high speed requirement.

1.2 FOX2 implementation and research goals

FOX2 implementation

We use the Tower 0.18um 1.8V Technology, 6 metal layers. It looks like this:

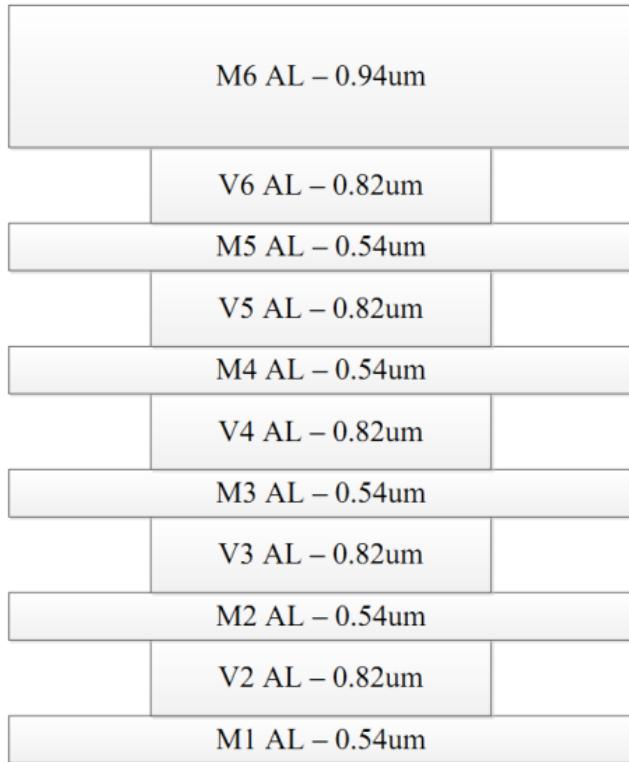


Figure 1.1: Metal stack cross section

Research goals

- Redo the digital logic parts in FOX2 based on the schematics in FOX1 to achieve the FO4 delay speed – around 10Gbps for 180nm Tower technology.
- Get the transmission line model that can well represent the long distance transmission line on the chip.

- Design the analog transmitter and analog receiver circuit that can support the asynchronous high-speed transmission over 10mm transmission line.
- Make a test chip that incorporates the fast analog links and the digital blocks to verify the concept of the asynchronous link.
- Test the chip after it is fabricated: design the test environment [18] to test the functionality and performance of the chip.

1.3 Outcomes of this research

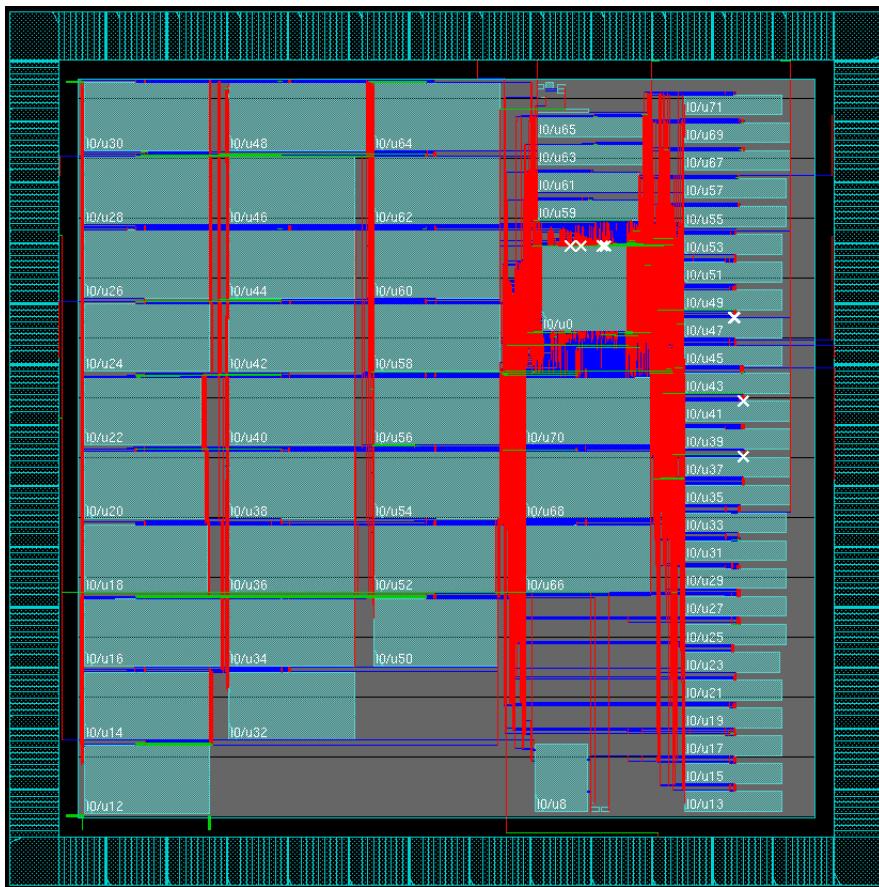


Figure 1.2: Chip layout in Encounter

A test chip has been fabricated: the tap-out date is July, 2014 and the chip has been back at the beginning of Oct, 2014. This is how the chip looks like in encounter after placement and route, but without transmission lines between TXs and RXs.

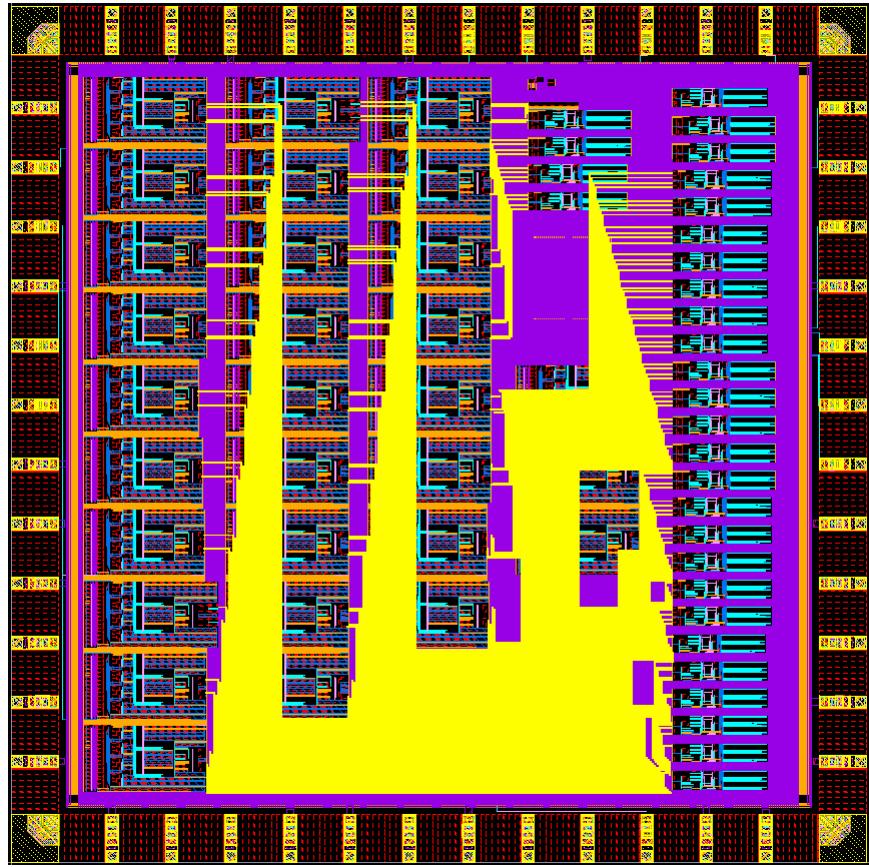


Figure 1.3: Chip layout in Virtuoso

This is the same chip after being imported into Virtuoso and with the transmission lines between TXs and RXs.

I have run the post layout simulation with the layout of the digital controller and it works. After adding the package, we get the chip back on Jan, 2015 and the PCB board that is used to hold the chip was manufactured on May, 2015. Test results show the highest speed for the one 1.74mm transmission line length voltage mode link is around 4.1Gbps and one RGC “TIA” link over 0.42mm transmission line can work until the speed of 2.2Gbps without bit error.

1.4 Organization of the thesis

Chapter 2 introduces the asynchronous high-speed bit-serial link architecture. For example, how we generate the clock, how we achieve the Serializer – Deserializer (SERDES) and the logical simulation results as well the performances of each module. Chapter 3 describes the analog part of the architecture, focusing on the transmission line model and current mode analog TX and analog RX design. After getting the result of the current mode analog TX and analog RX, we proposed another simple TX and RX pair that can also support that high speed asynchronous transmission. Chapter 4 focuses on the test circuit, which can give us the information about the actual speed of the chip. Besides the speed measurement, I also implemented other simple test circuits to test the functionality of some modules to make sure the chip can work properly. Chapter 5 is the simulations and the test results part which includes the performance of the analog links, the overall performance of the chip from post layout simulation and the performance of the chip in the test environment. Chapter 6 is the summary and conclusion, which gives a brief summary of the work and the final results conclusion. Moreover, I add some new techniques used in FOX2 that are different from FOX1 in the chip design process. Finally is the recommendations and the some critical points that will be helpful for the future research in this project.

In the appendices, I attached lots of materials about the FOX2 chip and the details of techniques I used during the design process. For those who want to know how the chip works, how it is tested and how to efficiently design & fabricate a chip can refer to that chapter.

Chapter 2

Asynchronous high-speed bit-serial architecture

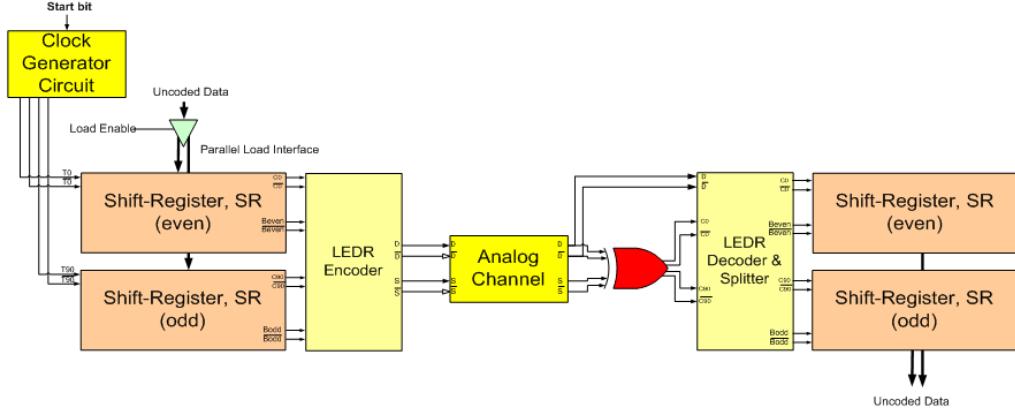


Figure 2.1: System architecture [18]

This is the system level diagram for one link. Before the transition begins, we load the sending data at the input of the 8 bits Shift Register (SR) [9] in parallel and we can see that 2 SRs are used for the even bits and odd bits respectively. Apparently, we can send 16 bits at one transition. After the data is ready, we will enable the load enable signal to the input of the SRs, so the data will be loaded into the SRs. Of course, we need to disable the load enable signal after the data has been stored at the input of the SRs. After the data is ready to be sent, we will give a start bit, which is just a simple step function from 0 to 1. This start bit will generate 2 differential clocks called T0/T0N and T90/T90N. The T0 and T90 have 4 clock cycles or 8 transitions and have the same clock frequency with 90° phase shift and these T0 and T90 will be used to drive the even bits SR and odd bits SR respectively. In this way, we can get the serial 8 bits even data and serial 8 bits odd data with 90° phase shift. The inputs to the LEDR Encoder are the differential signals of the T0 and T90, as well the odd bits and even bits, while the outputs from the LEDR [8] are the serial-differential Data bits and

Strobe bits with the frequency that is twice of T0 and T90 from the output of clock generator circuit [14]. In this way, we achieve the parallel-to-serial transition and we have the high-speed data at the output of LEDR and channel. The analog channel on this diagram represents the analog TX, transmission line and analog RX. Due to the high-speed over the channel, the signal over the channel will be not full swing and we treat it as analog signal. The functionality of the analog channel is to transmit the digital full swing signals at one side and recover the digital signals at the other side correctly. After the analog channel is the LEDR Decoder and we also call it splitter circuit [9], the functionality of this circuit is to take the inputs of the received Data bits, Strobe bits and output the Odd bits, Even bits as well as T0 and T90 again. Clearly, it has the inverse function as the LEDR Encoder. While for the SRs at the RX side [9], it will take the clock and serial data as inputs and after the transition, the Data bits will be saved at the outputs of the SRs in parallel.

2.1 Fast clock generator

As I have mentioned, the circuit we described is an asynchronous circuit, which means no external clock is employed for the data transition. Moreover, in general, the external clock cannot work at such high speed. So we need to generate clock pulses inside of the link.

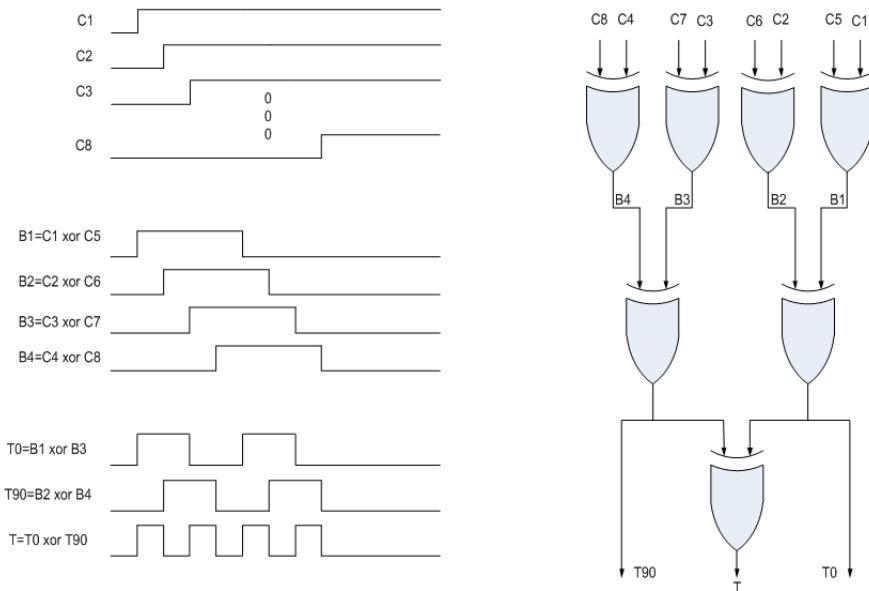


Figure 2.2: Clock generator diagram [14] [18]

This is one example diagram of the clock generator circuit. The input to the clock generator circuit is a simple step signal, and after some delay elements, we can have lots of step signals. By XORing different signals with a certain method, we can get some clock pulses. In general, if we have N ($N = 2^M$) delay elements, we can have N

transitions at most. From this example, we can also see the signals of T0 and T90. Here T0 and T90 have the same frequency, but have 90° phase difference. Moreover, we can see that the number of transitions T0 and T90 have is half of the total transitions T has.

Delay element

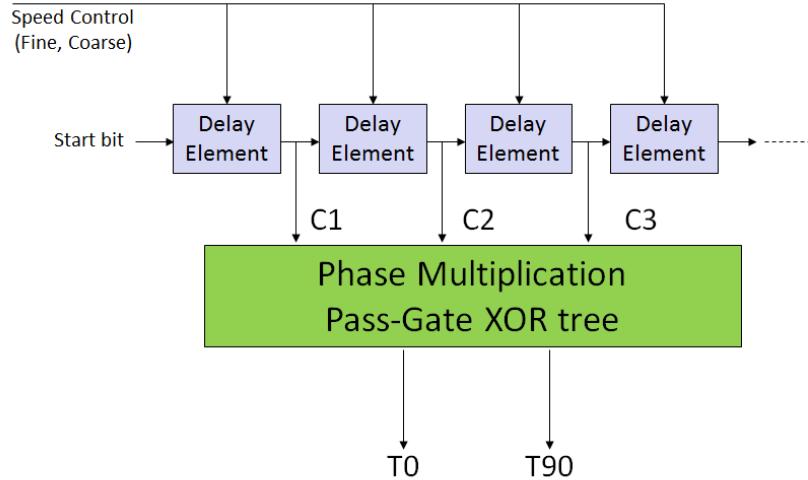


Figure 2.3: Clock generator schematic diagram [18]

This is the diagram of how we generate different delays by using delay elements. The speed control signals we illustrate here are used to control the speed for the link. Because we don't know the exact working frequency of the link on the chip, by utilizing the speed control, we can find it.

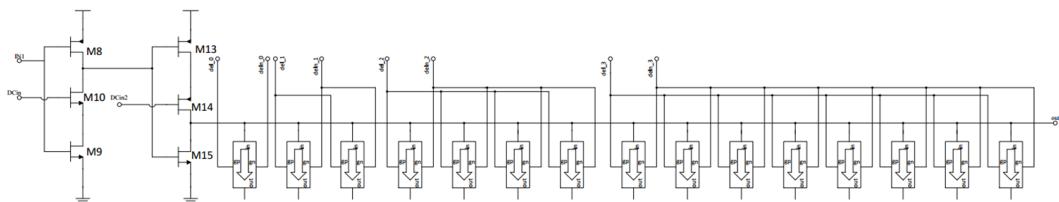


Figure 2.4: Schematic of the single delay element

This is the schematic of the delay element we have for the link. The left side is the current starvation circuit, which is proposed by D. Nahamanny and optimized by G. Albert, I. Shaham and I. The right side is the T-gates, which are used for the fine control [18]. Here we use 4 bits control signals for the fine control part. For the current starvation circuit, by changing the input control voltage of it, we can control the charging and discharging ability of the 2 “inverters”.

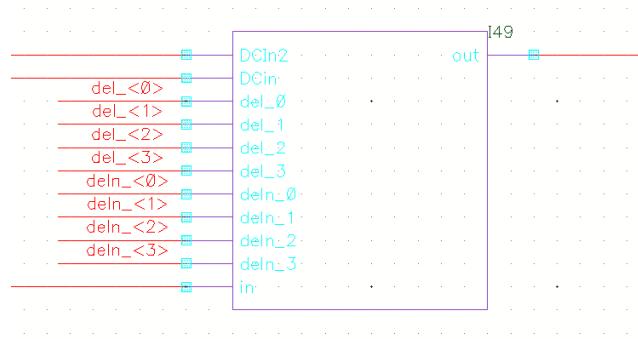


Figure 2.5: Symbol of the single delay element

This is the symbol for this schematic, while for the voltage control part, we make a simple voltage controlled current mirror for it.

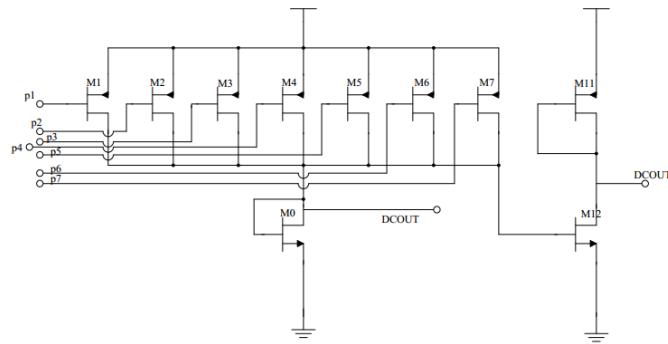


Figure 2.6: Schematic for the voltage controlled current mirror

By controlling the open or close of the PMOS transistors, we can control the output voltage of it.

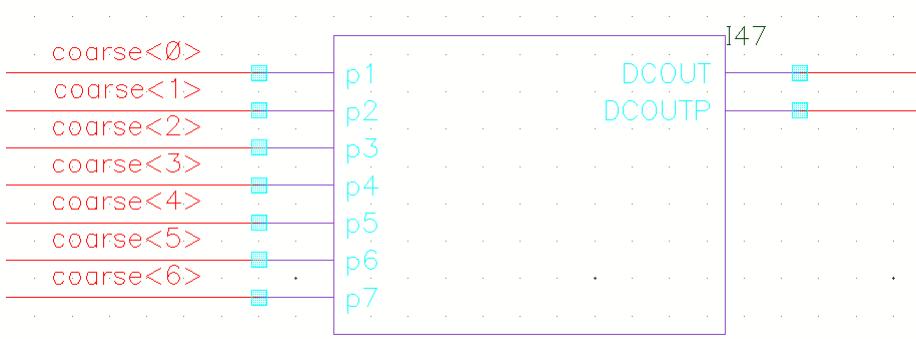


Figure 2.7: Symbol for the voltage controlled current mirror

We have 7 bits for the coarse control for the output voltage. Thus, in total, we can generate around $2^{4+7} = 2^{11} = 2048$ different clock frequencies with pretty high resolution.

As I have mentioned before, we planned to transmit 16 bits at one time, thus, we need to have 16 delays from delay element. Here, in order to make the delays to be the same for the duty cycle consideration, I add another delay element after the last stage as the load for it, so in total, 17 delay elements are used.

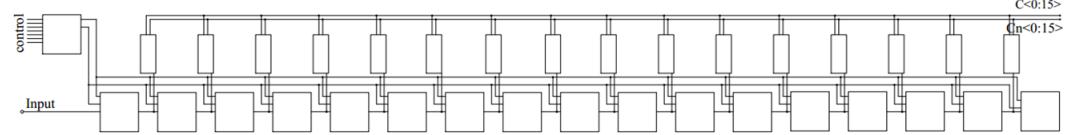


Figure 2.8: Schematic for delay elements [18]

This is the schematic for the delay elements, and all the delay elements have the same speed control signals in order to get 50% duty cycle for the generated clock signals.

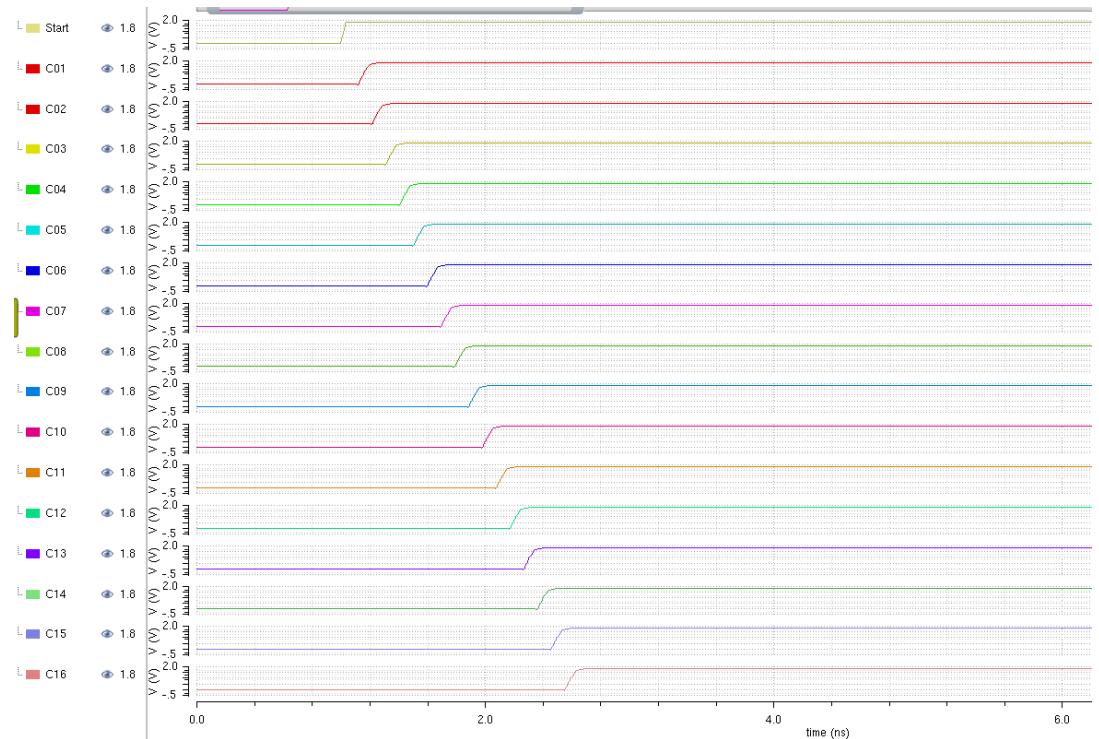


Figure 2.9: Delay elements simulation

This is the simulation results of the delay element, the START signal is the input step function that initialize the delay generation.

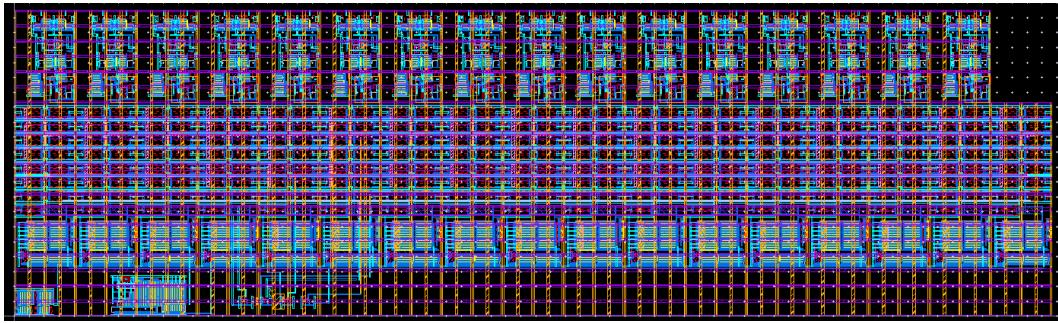


Figure 2.10: Layout for delay elements

This is the layout for the delay element and from the post layout simulation for the delay element, the maximum output speed is around 10.21Gbps and the minimum output speed is around 377Mbps.

XOR

From the diagram of the clock generator circuit, we know that we need to XOR the signals from the outputs of the delay element. Traditional XOR circuit cannot work at very high speed, so we adopt a novel XOR circuit based on pass-gate.

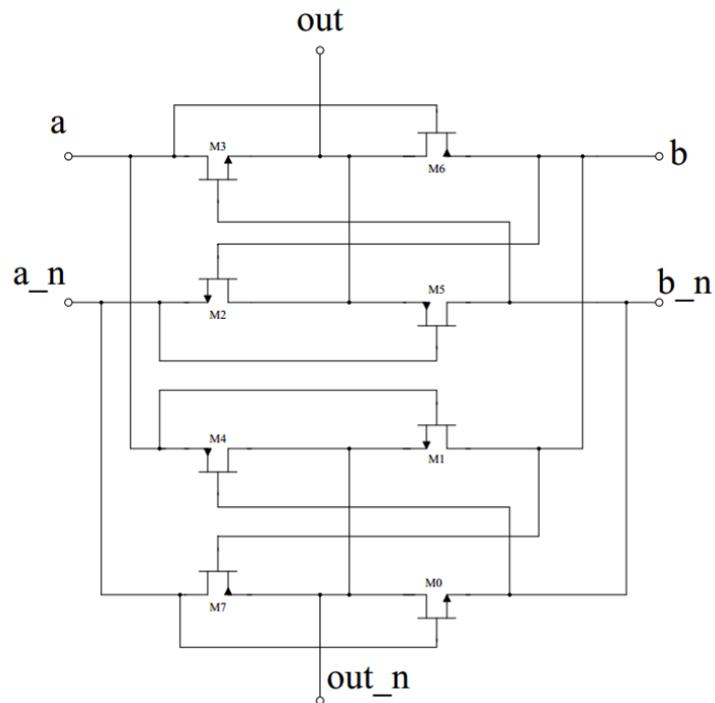


Figure 2.11: Schematic of pass-gate based XOR [18]

The XOR needs the input signals to be differential signals. And it will give differential output signals as well.

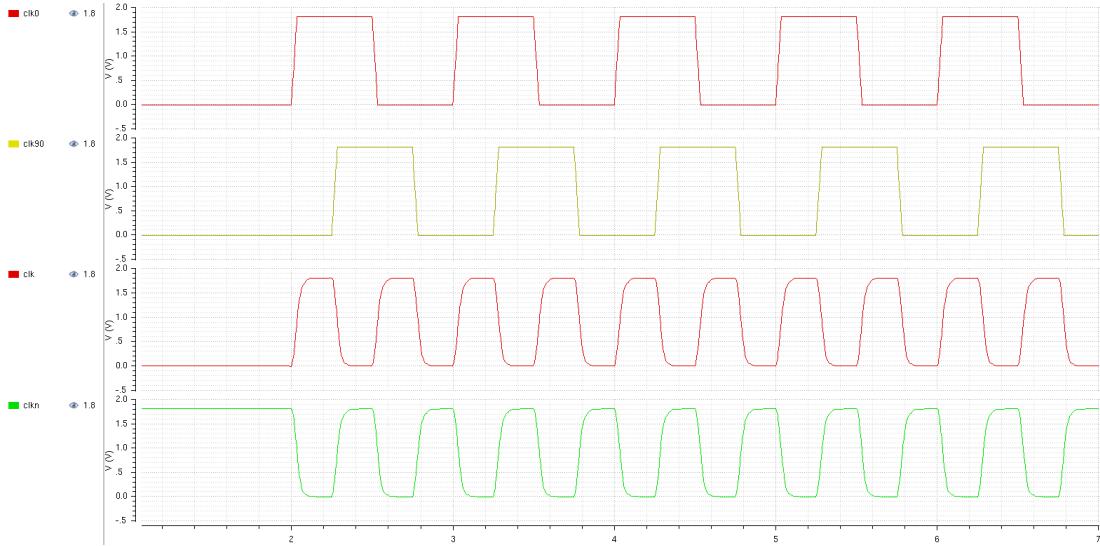


Figure 2.12: Simulation of XOR

This is the simulation result for the XOR circuit and we can see that with the input 90° phase shift clock signals, we can get the maximum speed clock signal from the output.

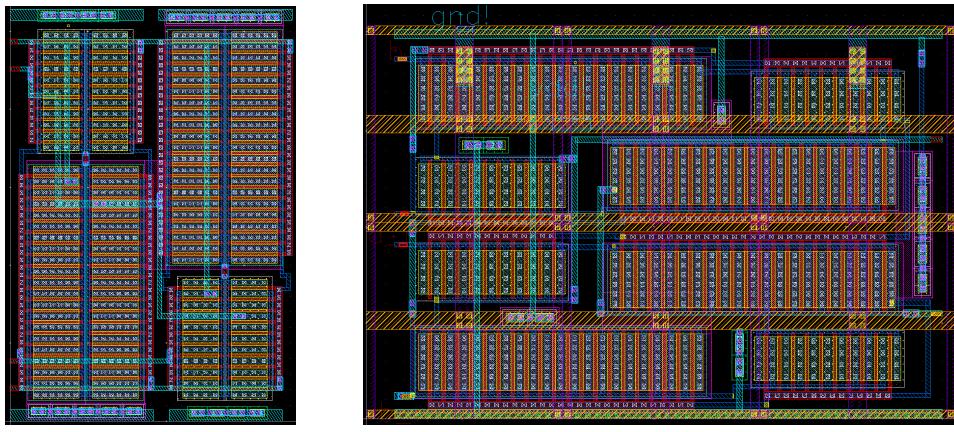


Figure 2.13: XOR layouts

These are the layouts for the pass-gate based XOR. I have 2 versions for different parts in this system. The maximum working speed from the post layout simulation is around 12Gbps.

CGC – integration of delay elements and XOR

The clock generator circuit is the integration of the delay element and XOR circuit. Of course, we need to have some drivers in order to recover the signals.

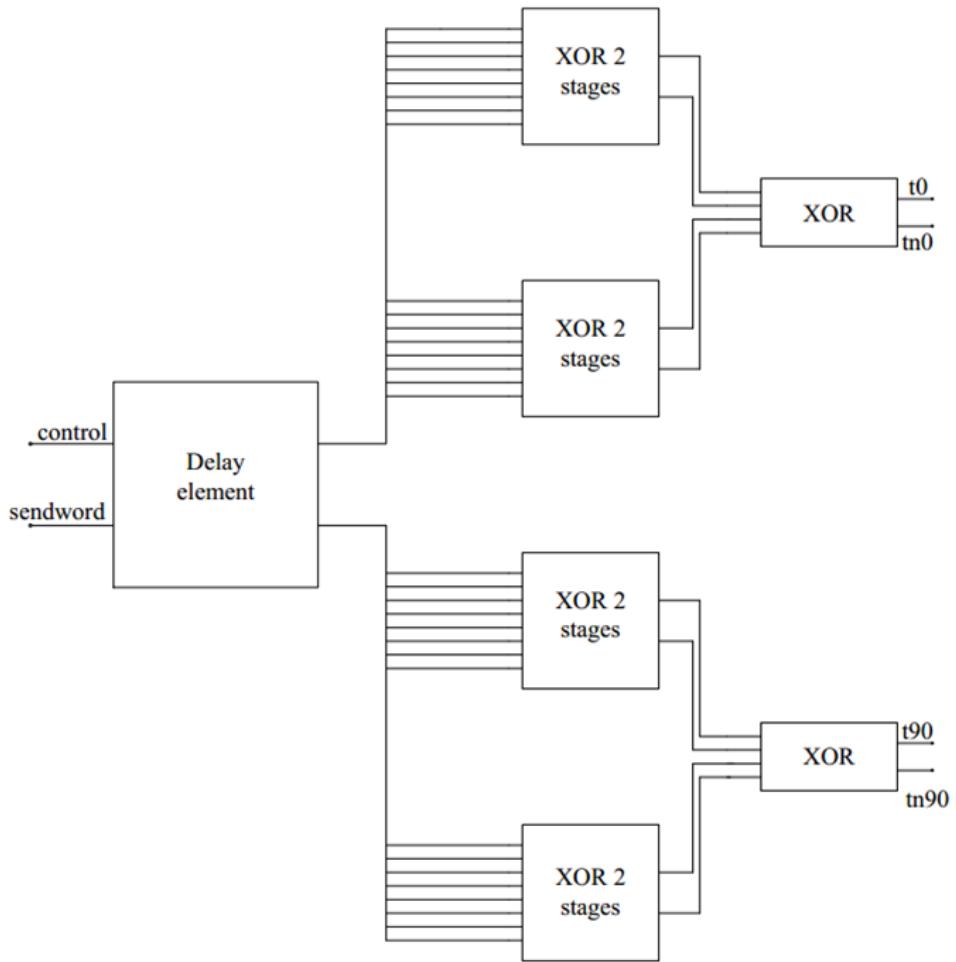


Figure 2.14: Schematic of CGC [18]

This is the schematic for the CGC, whose outputs are T0/T0n and T90/T90n.

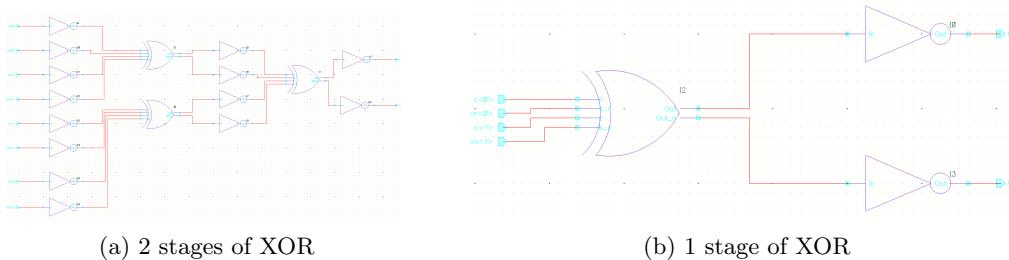


Figure 2.15: XOR symbols

These are the XORs we have in the schematic. In total, we use 3 stages of XOR, which generate 8 transitions (4 cycles) for T0 and T90.

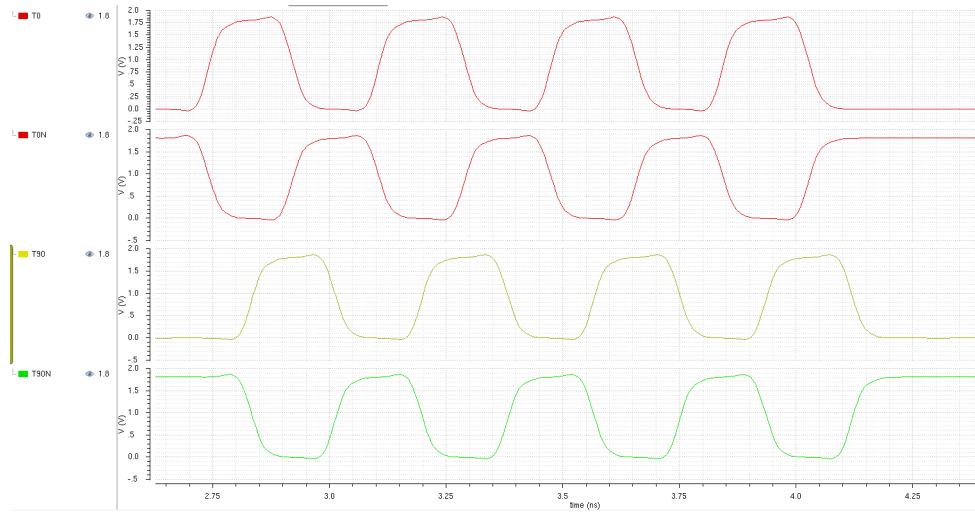


Figure 2.16: Simulation result of CGC

This is the output signals of CGC. We can see that the outputs are differential signals and the 2 signals have 90° phase shift.

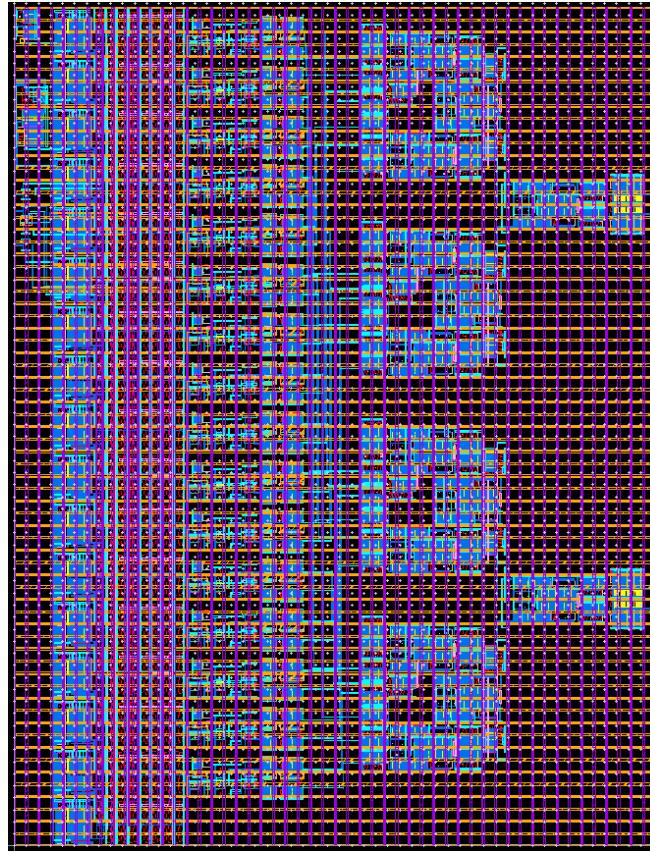


Figure 2.17: Layout of CGC

This is the layout for the CGC. The maximum output frequency from the post layout simulation is around 8.32Gbps. We can see that this result is lower than the frequency

from the delay element, and I think the reason for it is the long wire connections after delay elements, which have large capacitance and thus constrain the speed. Or perhaps because of the large layout area, there is voltage drop over the along the power grid which will make some circuits work slower.

2.2 SR at the TX side

The SR [9] at the TX side achieves the parallel to serial transition for the input data. In order to implement fast transition, we make the SR transmit one bit at each clock change. The basic element in the SR is called transition latch (XL) [9] and here are the schematic and symbol for the XL.

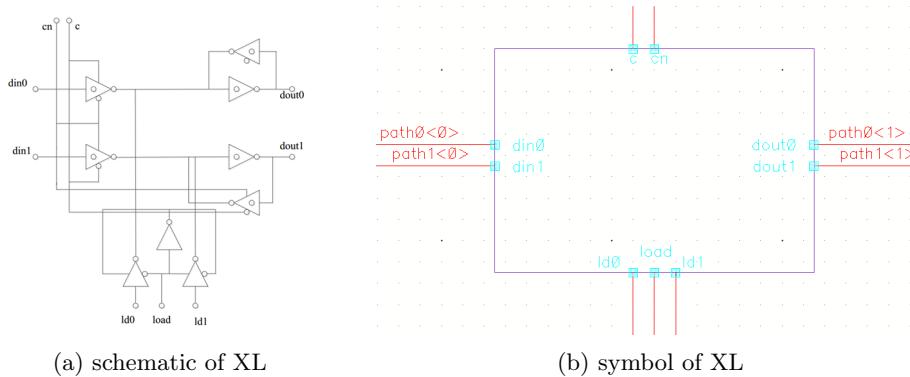


Figure 2.18: XL schematic and symbol

For the XL, we can see that it has 2 separate data paths and each XL is controlled by differential clock signals C/CN. In this way, we can make sure there is one bit transition when the clock changes, no matter from 0 to 1 or 1 to 0. There is load signal at the inputs of the XL. At first, we enable the load signal to load the data into each XL. When it works, we need to disable the load signal in advance to make sure the data is transmitted as planned. The data from the 2 separate data paths will be merged at the output stage of the SR.

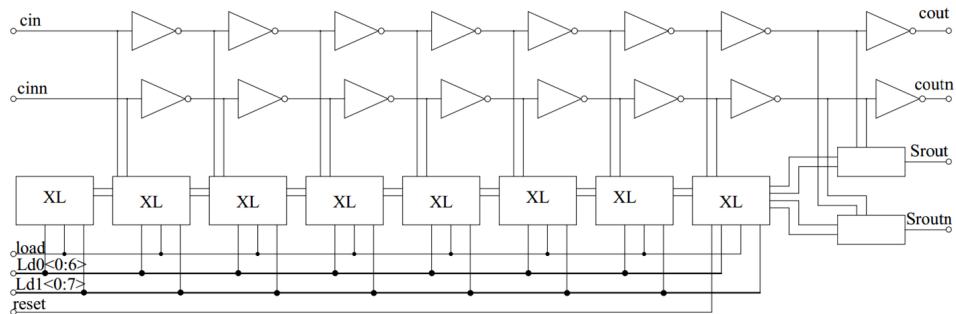


Figure 2.19: Schematic for SR at TX side [18]

This is the schematic for one SR, and we can see it has 8 XLs, which will hold 8 bits (odd bits or even bits). At the output stage of the SR, there is the merge circuit.

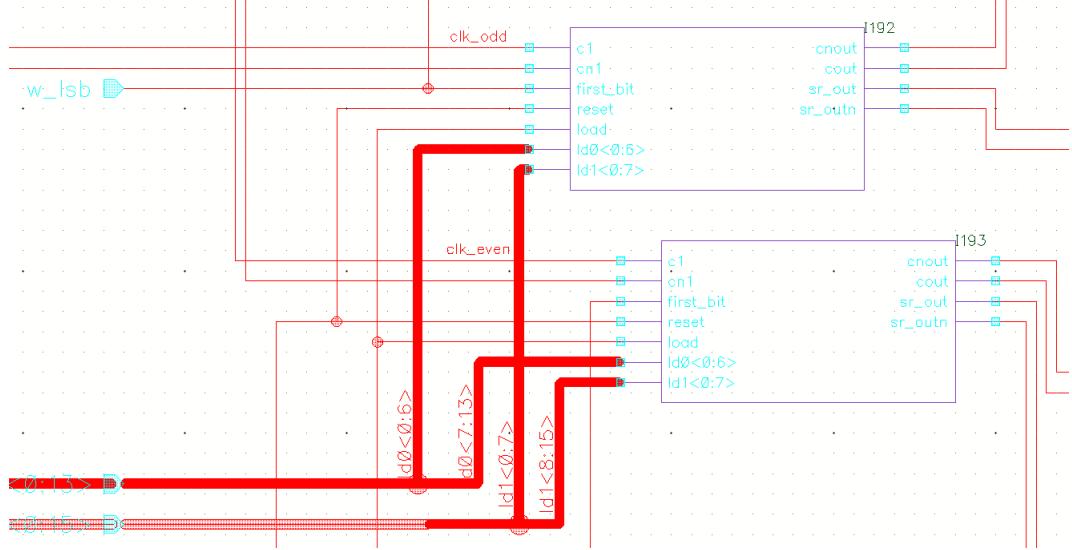


Figure 2.20: Schematic of test circuit for TX SRs

This is the test circuit for the even SR and odd SR together and the signal pins have the following mapping with the transmitted bits.

	bit0	bit1	bit2	bit3	bit4	bit5	bit6	bit7
Even	w_lsb	ld1<6>	ld0<5>	ld1<4>	ld0<3>	ld1<2>	ld0<1>	ld1<0>
Odd	ld1<14>	ld0<12>	ld1<12>	ld0<10>	ld1<10>	ld0<8>	ld1<8>	ld0<7>

Table 2.1: Map between signal pins and input bits

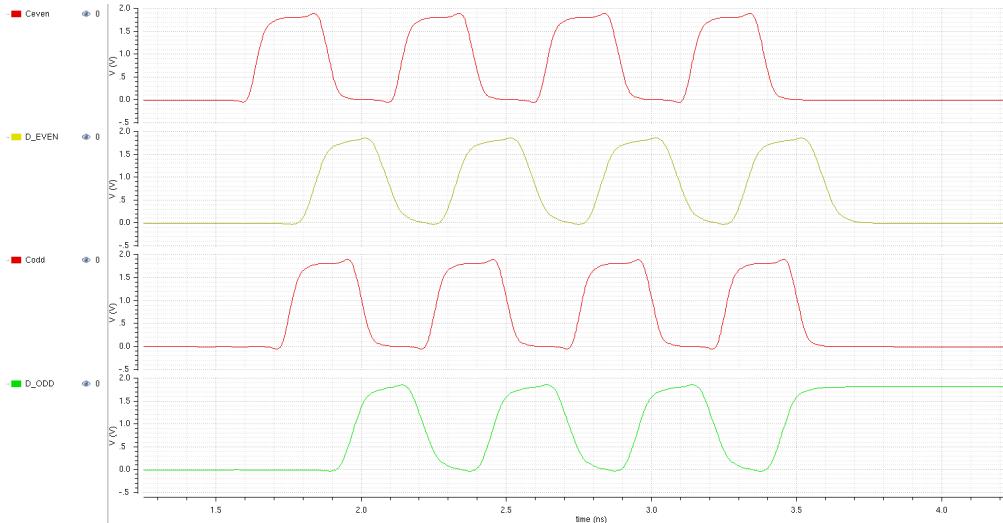


Figure 2.21: Simulation of TX SRs

The input pattern to the test bench is 0011001100110011 and we can see that we get the correct output data bits as well as clock signals.

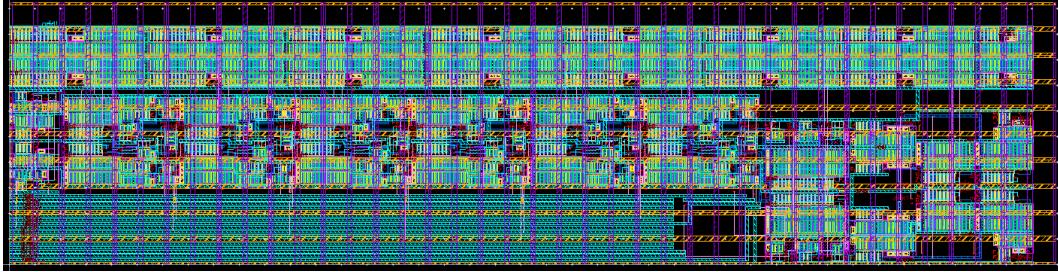


Figure 2.22: Layout of TX SR

This is the layout for the SR at the transmitter side and post layout simulation shows the maximum working frequency for this SR is 8.4Gbps (for some specific patterns, it can work faster).

2.3 LEDR encoder

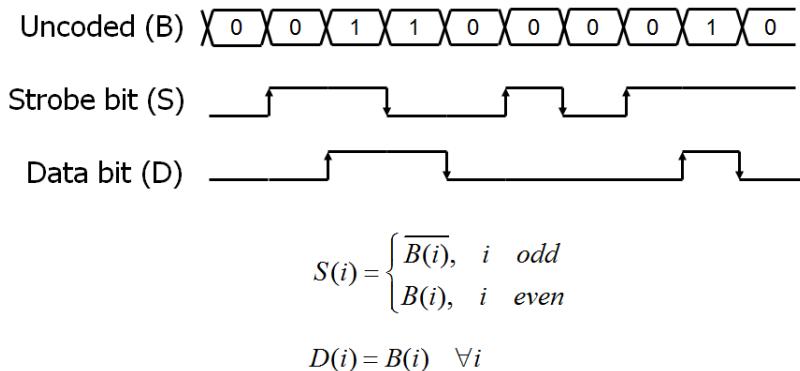


Figure 2.23: LEDR protocol

This is the LEDR protocol [8]. It will encode the uncoded data bits into original Data bit and Strobe bits [7], so it is system code. For the Strobe bits: if we choose the first bit to be the same as the original bit, then the second bit will be the inverse of the original bit. The third bit will be same again and the forth bit will be inverted again, so on and so forth. We can see that when we XOR the Data and Strobe bits, we will get clock signal which has the same speed as the data.

The inputs to the LEDR encoder are the differential even bits and odd bits of the original data, what we need to do is to combine the odd bits and even bits and double the data rate of the data. Here is the schematic for it.

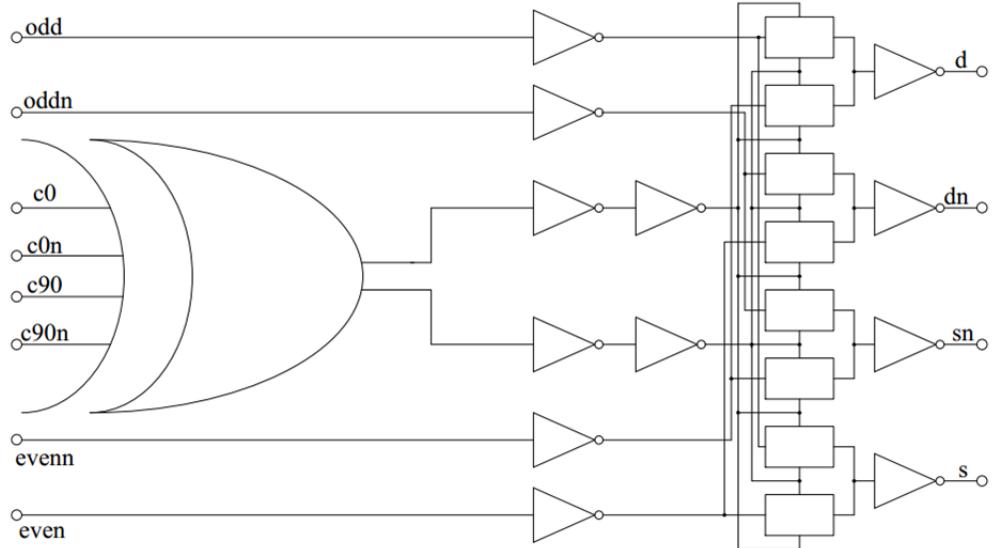


Figure 2.24: Schematic of LEDR encoder [18]

With the inputs T0 and T90, by using the XOR, we can get the maximum frequency clock T. By selecting odd bits and even bits with this clock through controlling the open or close of the pass gate, we can get the desired Data/DataN and Strobe/StrobeN signal at the output.

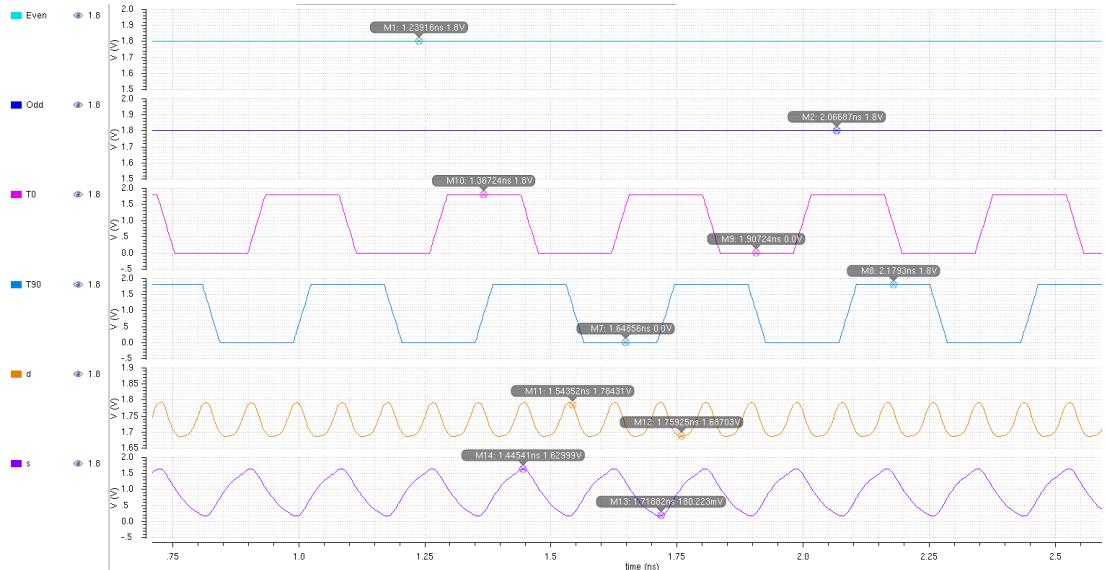


Figure 2.25: Simulation of LEDR encoder

The input odd bits and even bits are VDD signal, thus, the output Data should be always VDD while the output Strobe bits should be clock signal.

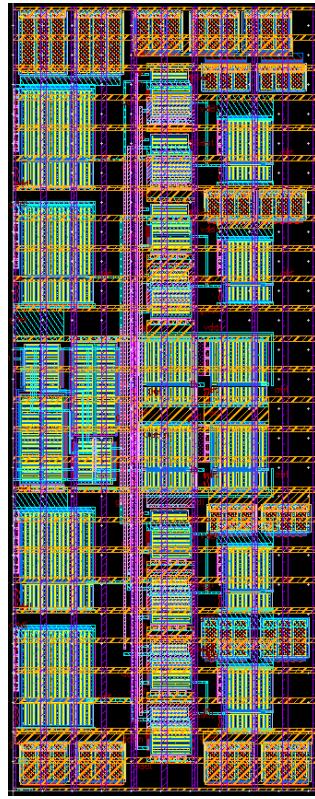


Figure 2.26: Layout of LEDR encoder

Here is the layout for the LEDR Encoder and the post layout simulation shows the maximum working speed is around 11Gbps.

2.4 Toggle and splitter circuit

From the introduction part of the chapter, we know that the Splitter circuit [10] also has the name LEDR Decoder, which means it implements the inverse functionality of the LEDR Encoder. Or equivalently, the inputs to the Splitter circuit are Data/DataN and Strobe/StrobeN signals, while the outputs are even bits, odd bits and the T0/T0N, T90/T90N signals.

Also in the LEDR part, we know that when we XOR Data and Strobe bit, we will get the clock signal. Thus, I will show the Toggle circuit [10] first since Toggle is one consist part of the Splitter and it just deals with the clock signal.

Toggle

The input to the toggle is the differential clock signal. For convenience, I still use T and TN to represent the input clock to the toggle, while the outputs are differential signals of T0 and T90.

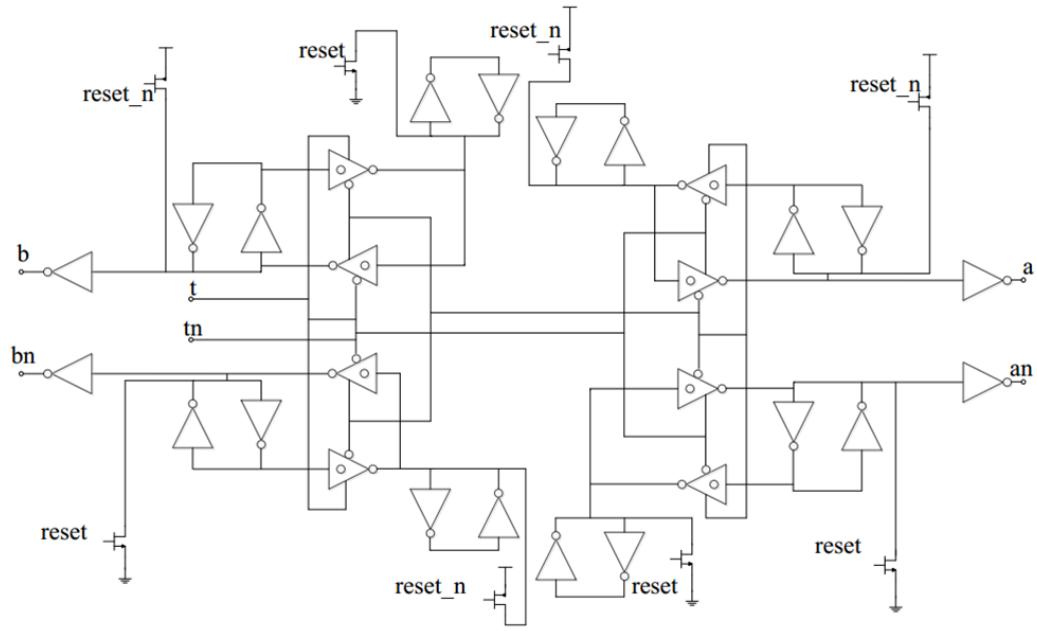


Figure 2.27: Schematic for toggle circuit [18]

This is the schematic for the toggle circuit, which can work at very high speed that can meet the one gate delay requirement.

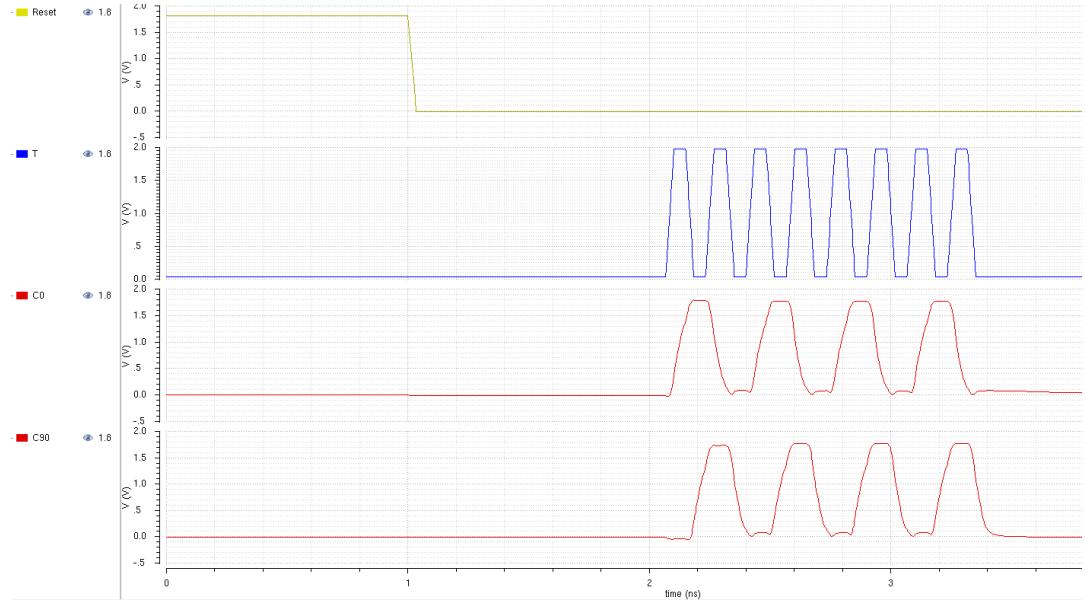


Figure 2.28: Simulation of toggle circuit

The inputs to the toggle are reset signal and clock signal, and we can see that the outputs are 2 clock signals with half the input frequency and with 90° phase shift.

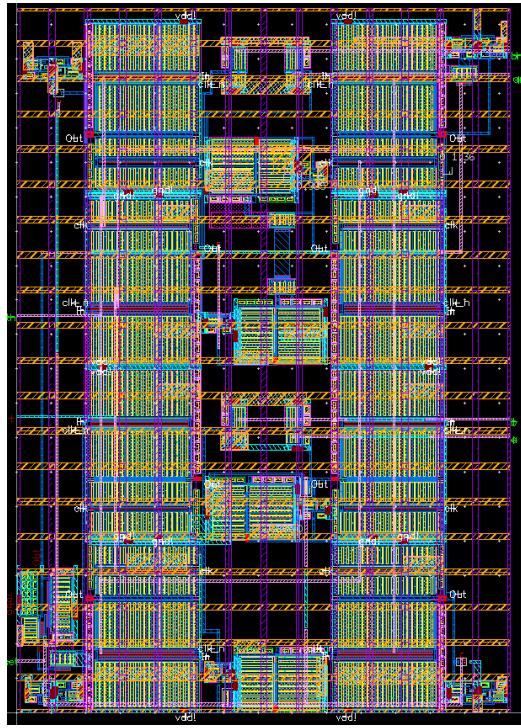


Figure 2.29: Layout for toggle circuit

This is the layout after carefully design. The post layout simulation gives the maximum working speed of the toggle circuit around 12.6Gbps (the layout has a big influence on the overall performance).

Splitter

Since we have gotten the T0 and T90 signal from the toggle, for other parts of the circuit in the Splitter, we only need to splitter the data bits into even bits and odd bits. Of course, in order to use the T0 and T90 drive the odd bits and even bits, we need to make sure the data and clock has the same phase/delay.

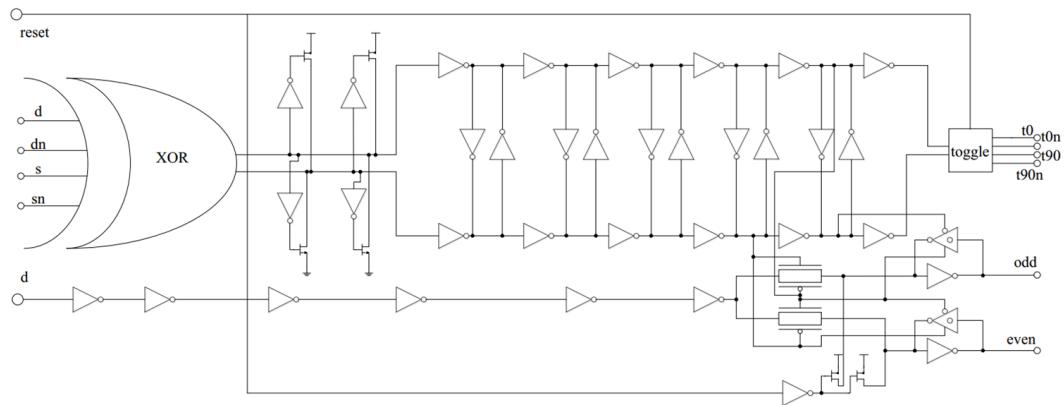


Figure 2.30: Schematic for splitter circuit [18]

This is the schematic for the splitter circuit. The data path is at the lower part of this schematic. The clock T0/T0N generated by the Data and Strobe bits will control the pass gate and in this way, we will get the even bits and odd bits at the outputs of the splitter.

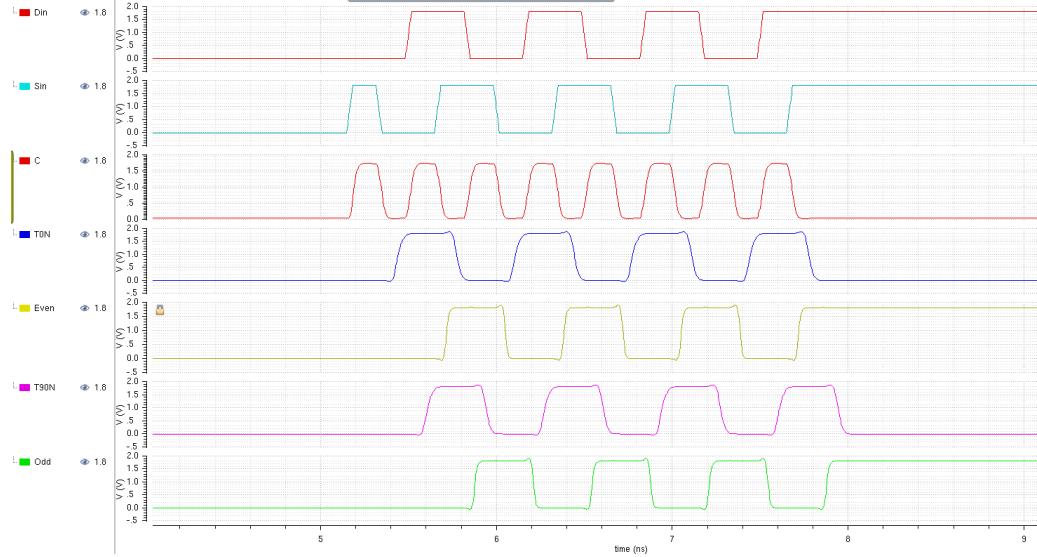


Figure 2.31: Simulation of splitter circuit

The input data has the pattern 0011001100110011 and we can see that at the output, we have ODD bit pattern 01010101 and EVEN bit pattern 01010101.

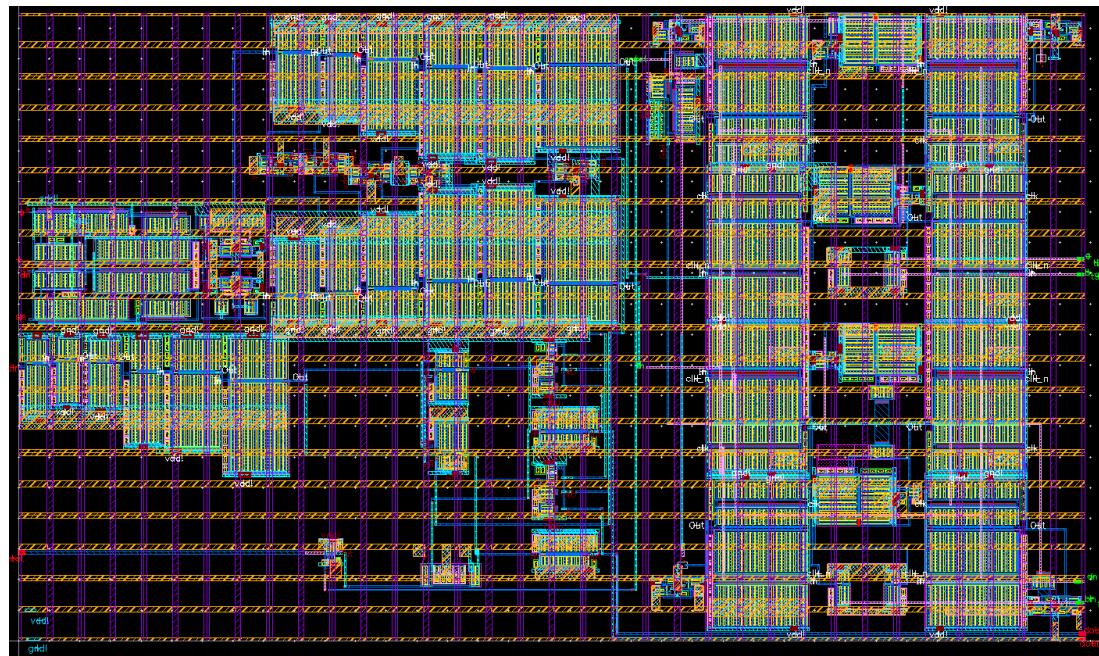


Figure 2.32: Layout for splitter circuit

This is the layout of the splitter circuit. The right side is the layout of toggle. Post layout simulation shows the maximum working speed is around 8.6Gbps. I have checked the bottleneck of this circuit and unfortunately, still the toggle doesn't work well. The input clock signal to the toggle is not ideal clock signal with different rise time and fall time and fluctuations. I think because of the non-ideal input signal that makes toggle work slower.

2.5 SR at the receiver (RX) side

The SR at the receiver side [9] functions inversely compared with the SR at the transmitter side. The inputs to the SR are the even/odd serial bits as well as clock signals (T_0 or T_{90} , depends on the even bits or odd bits), while the outputs of the SR are the parallel odd bits or parallel even bits.

The basic element for the SR is the same XL, which enables the high speed transition. But the control signals for the TX SR and RX SR are different.

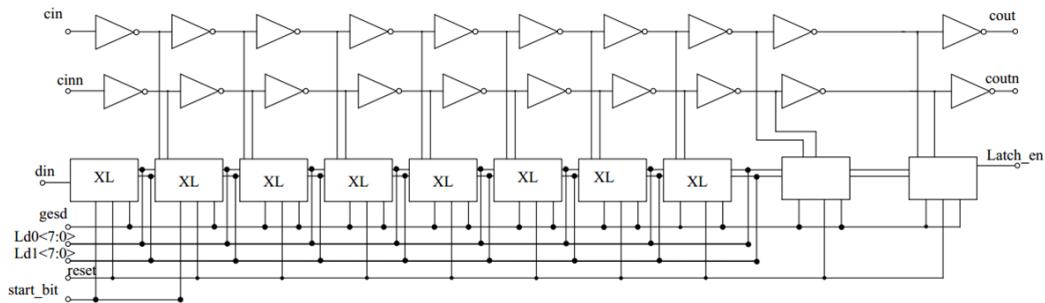


Figure 2.33: Schematic for RX SR [18]

The output bits have the following mapping with the pins.

	bit0	bit1	bit2	bit3	bit4	bit5	bit6	bit7
Even	odd⟨15⟩	odd⟨6⟩	odd⟨13⟩	odd⟨4⟩	odd⟨11⟩	odd⟨2⟩	odd⟨9⟩	odd⟨0⟩
Odd	even⟨7⟩	even⟨14⟩	even⟨5⟩	even⟨12⟩	even⟨3⟩	even⟨10⟩	even⟨1⟩	even⟨8⟩

Table 2.2: Map between signal pins and output bits

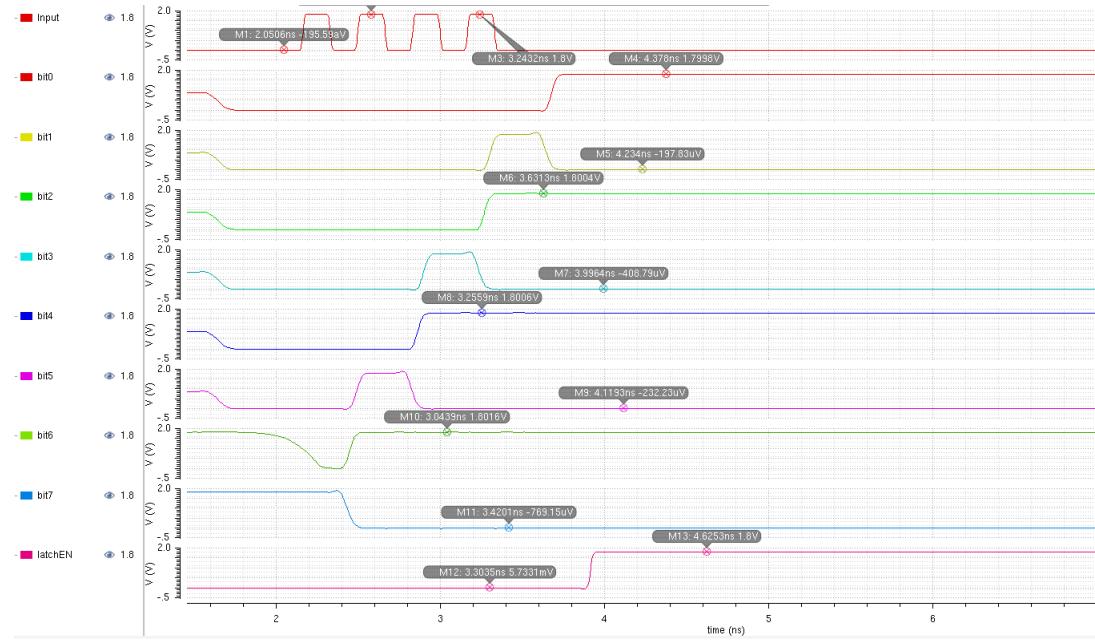


Figure 2.34: Simulation of RX SR

With the input pattern 01010101, we can see that when the transition finished (the latch enable signal is high), the bits that stored at the output of RX SR are 01010101.

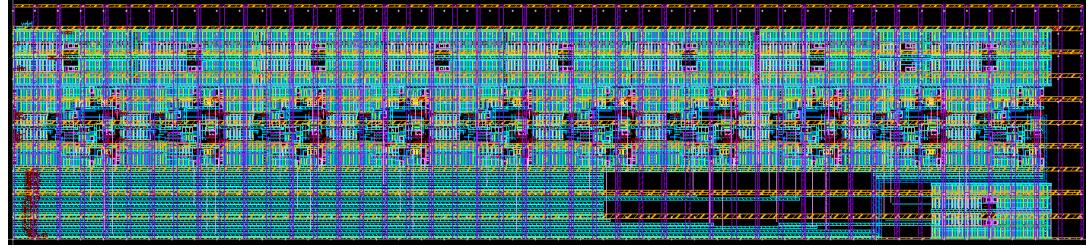


Figure 2.35: Layout for RX SR

This is the layout of the SR at the RX side and the post layout simulation shows the maximum working speed is around 8.1Gbps (still with some specific input patterns, it can work faster).

Chapter 3

High speed analog link

3.1 Introduction

From the diagram of the system architecture, we know that the high speed analog link includes the analog transmitter, the model of the transmission line and the analog receiver. Since the target transmission distance of the link is 10mm, on the designing process of the transmitter and receiver, I use the 10mm transmission line model for the design and performance check. However, after finishing the final layout, because of the placement and route considerations, the actual transmission line length is different from the model. Anyway, before we get the chip back and test it, the evaluation results of the links are all from the simulation, not the true reflection of the layout. After the final layout, I ran another simulation based on the physical dimension and the results are also attached.

3.2 Transmission line model

HFSS for RLC model

Many empirical formulas deal with the transmission line model for the PCB board. While for the on chip transmission line, not many formulas can satisfy the dimension constraint. Moreover, for such formulas, many just can give values of the characteristic impedance, even not include the real part and image part of the it. Since our link works at high speed, I intend to use the RLC model to represent the transmission line. Thus, even with the value of the characteristic impedance, we still cannot get the LC model parameters directly from it.

By modeling the transmission line in HFSS, we can get the characteristic impedance of the line. Moreover, it is also easy to get the real part and image part of it.

From the transmission line model, we can get the this formula:

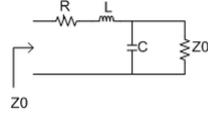


Figure 3.1: Transmission line RLC model

$$Z_0 = R + j\omega L + \frac{\frac{Z_0}{j\omega C}}{Z_0 + \frac{1}{j\omega C}} \quad (3.1)$$

This is the model and the calculation for it. When the unit length is small, we have[16]:

$$Z_0 = \sqrt{\frac{Z}{Y}} = \sqrt{\frac{R + j\omega L}{j\omega C}} \quad (3.2)$$

Here R , L and C are unit length resistance, inductance and capacitance respectively. R can be calculated based on the actual dimension of the aluminum line since no skin effect. Thus, only L and C are unknown. We have 2 parameters and 2 equations – real part and image part of the characteristic impedance, thus we can get the RLC model.

$$\begin{cases} R_0 = \rho L / S \\ C_0 = R_0 / 2\omega Z_{0R} Z_{0I} \\ L_0 = C_0 (Z_{0R}^2 - Z_{0I}^2) \end{cases} \quad (3.3)$$

Here ρ is the electrical resistivity of the metal, L and S are the length and cross-section area of the metal respectively. Z_{0R} and Z_{0I} are the real part and image part of the characteristic impedance (Z_0) of the line while ω is the frequency we used in the HFSS simulation.

The chip we fabricated has the size $4\text{mm} \times 4\text{mm}$, while the transmission line has the length 10mm. Thus, we cannot use ideal micro-stripe line for the model of the line.

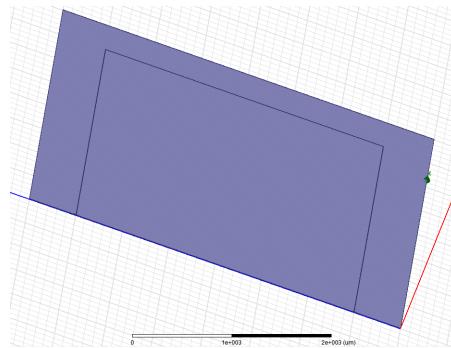


Figure 3.2: Transmission line in HFSS

This is the basic geometry of the line in HFSS, which is “U” shape. Moreover, we can see from the previous part that the signals along the transmission line are differential signals. Thus, perhaps it is better to use the differential line geometry.

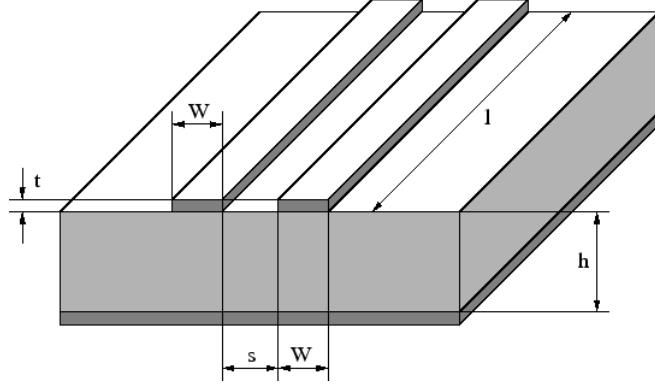


Figure 3.3: Coupled line model

The voltage and current at the output of the transmission line are proportional to $e^{-\frac{Rx}{Z_0}}$ [16]. Hence, we can see that for a certain length, we can reduce the attenuation of the signal by decreasing the resistance of the line, or increasing the characteristic impedance of the line, or both.

At the beginning of the design, I have no need to consider the real geometry for the layout of the transmission line. Thus, based on the metal stack of the technology we use, I combine M6 and M5 as the way to increase the line thickness (to reduce resistance R) and by setting the ground of the transmission line to be M1 in order to increase Z_0 . The default ground is M5 (if we combine M5 and M6, the ground will be M4) and the default transmission line is M6. Thus, for different combinations, we have the following geometry.

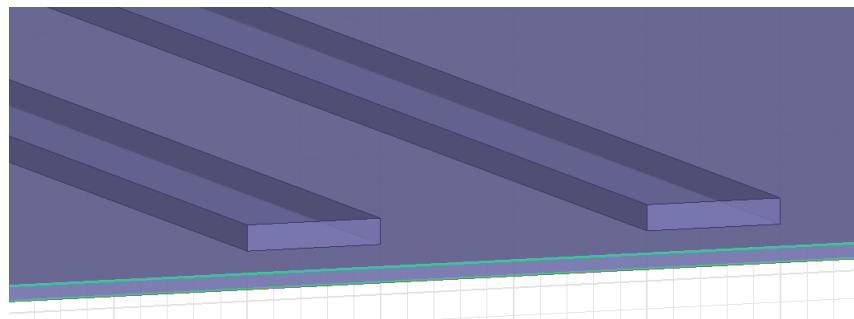


Figure 3.4: Basic line, M6 line, M5 ground

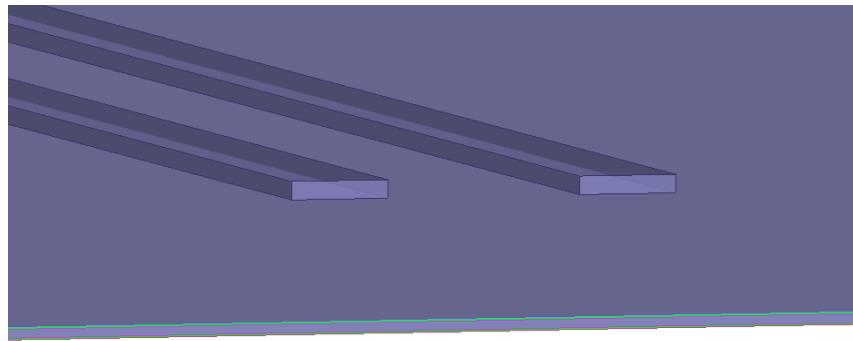


Figure 3.5: Increase Z_0 , M6 line, M1 ground

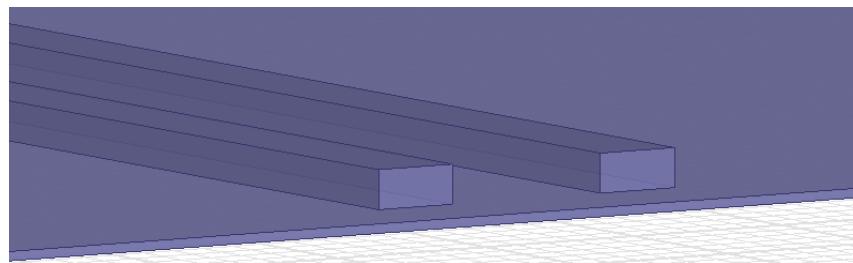


Figure 3.6: Decrease R, M6&M5 line, M4 ground

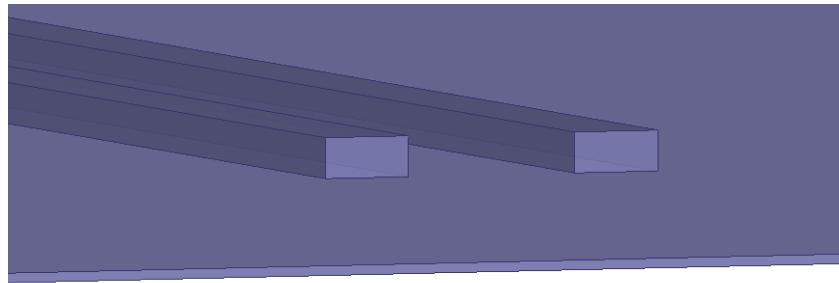


Figure 3.7: Increase Z_0 and decrease R, M6&M5 line, M1 ground

Then we have the following result for the transmission line model in the table.

Type	T(μm)	H(μm)	W(μm)	$Z_0(\Omega)$
Microstripe basic line	0.94	0.82	5	37.8
Microstripe increase Z_0	0.94	6.26	5	76.2
Microstripe decrease R	2.3	0.82	5	32.8
Microstripe increase Z_0 and decrease R	2.3	4.9	5	66.9
“U”shape basic line	0.94	0.82	5	36.5
“U”shape increase Z_0	0.94	6.26	5	77.8
“U”shape decrease R	2.3	0.82	5	32.1
“U”shape increase Z_0 and decrease R	2.3	4.9	5	66.4

Type	T(μm)	H(μm)	W(μm)	$Z_0(\Omega)$
Shield basic line	0.94	0.82	5	34.2
Shield line increase Z_0	0.94	6.26	5	74.3
Shield line decrease R	2.3	0.82	5	27.2
Shield line increase Z_0 and decrease R	2.3	4.9	5	59.3
Coupled basic line	0.94	0.82	5	36
Coupled line increase Z_0	0.94	6.26	5	78.7
Coupled line decrease R	2.3	0.82	5	31.5
Coupled line increase Z_0 and decrease R	2.3	4.9	5	66.5

Table 3.1: Transmission line performance (at the frequency of 5 GHz)

We can see that for the transmission line with same size, even the geometries are different. We will get similar characteristic impedance and thus, similar performance.

Import layout into HFSS

The RLC model in HFSS is easy to get and convenient to use for simulations in virtuoso. However, it is not accurate, both for geometry and dimension. We cannot get ideal “U”-shape line and ideal 10mm line. Thus, the best way to deal with this problem is import the layout of the transmission line into HFSS and then export the result file to Cadence to do the simulation.

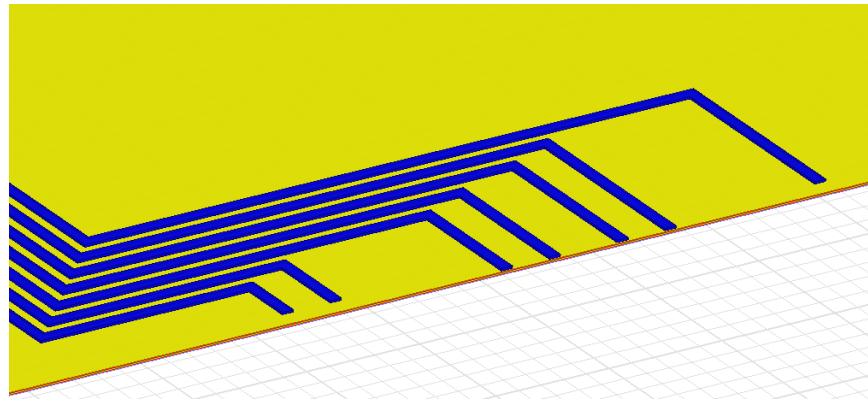


Figure 3.8: Imported layout in HFSS

Each link has 4 lines in the true layout (other lines are like shield, but they have influence on the result).

The blue color here is M6 while the yellow color stands for M5.

This should be used for the final simulation to get the whole link performance. But since during the design stage of the analog transmitter and analog receiver, we don't have and cannot have this. The performances of the analog link we have are still based on the RLC model we got in the previous part.

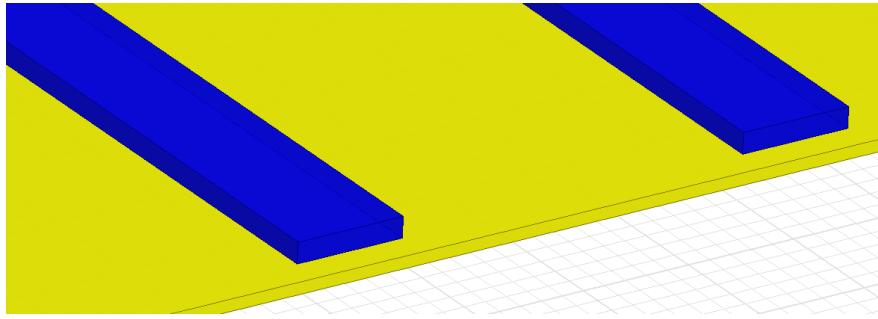


Figure 3.9: Layout geometry at the other side

3.3 Analog transmitter circuit

One big performance drawback of the high-speed on-chip communication is the high capacitance of the long interconnect, which constraints the bandwidth and degrades the signal. In general, current mode (CM) circuit can provide lower swing and dissipate less power compared with voltage mode (VM), which will support the long distance transmission at fast operation.

3.3.1 Analog current mode transmitter

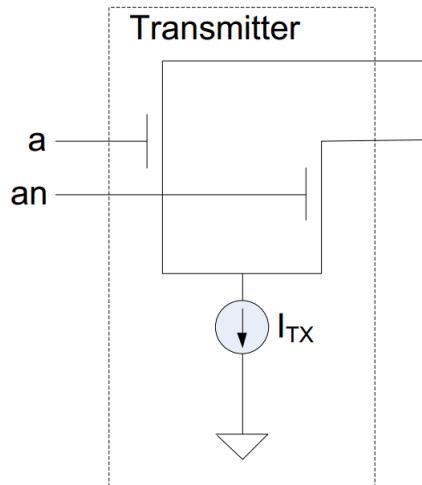


Figure 3.10: Basic current mode transmitter

This is a very basic structure of the analog transmitter circuit [11] with the differential input signals. The drains of the 2 NMOS transistors need to have a certain DC voltage in order to supply a certain current.

The structure is OK for the high-speed transition. However, for us, it is a little problem. We use asynchronous circuit, which means when there is no transition, the circuit and transmission line are at the DC condition, while when there is transition, because of the high speed property, the circuit and the transmission line will work at extremely

high speed. The characteristic impedance of the transmission line is very high at low frequency while at high frequency, it can be pretty low. Thus, from no data transition to transition, the characteristic impedance of the transmission line has an abrupt change, which will distort the first bit at the beginning of the transition.

In order to eliminate this distortion, we make changes to the basic CM circuit and get another circuit called CM transmitter with Adaptive Control.

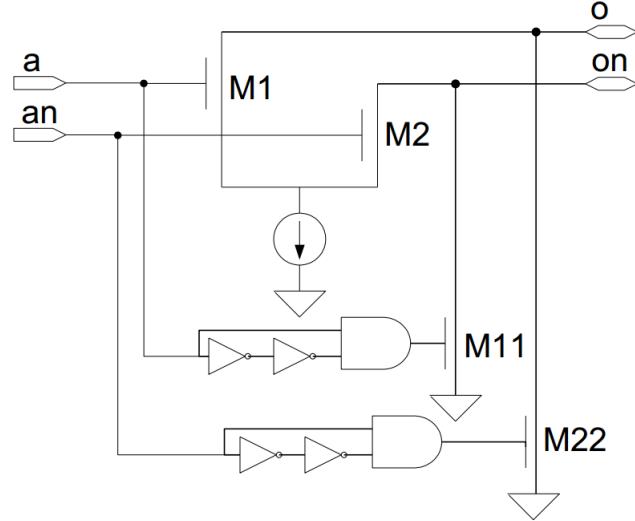


Figure 3.11: Basic current mode transmitter with adaptive control

The M11 and M22 here are used to draw current at the beginning of the first transition [11], which will reduce the characteristic impedance of the transmission line during the slow to fast transition and help eliminate distortion for the first bit.

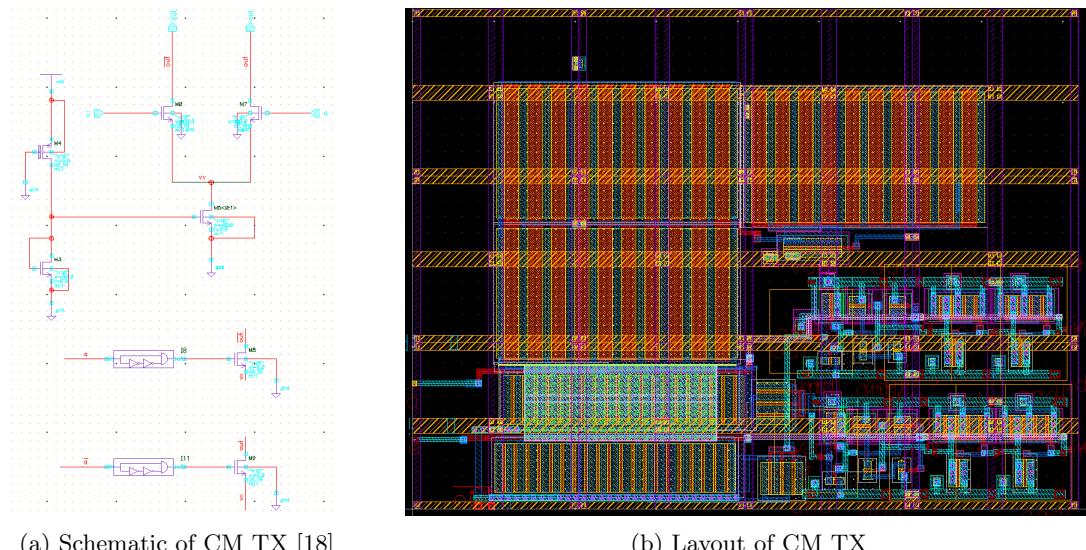


Figure 3.12: TX schematic and layout

This is the schematic and layout of the CM transmitter with adaptive control circuit. For the performance evaluation, I will discuss it at the analog receiver part.

3.3.2 Simple inverter driver

The focus of this research is the analog CM transmitter and receiver. However, for some reasons, I also implemented several links with simple 2 stages of inverter as the analog transmitter and 2 stages of inverter as the analog receiver.

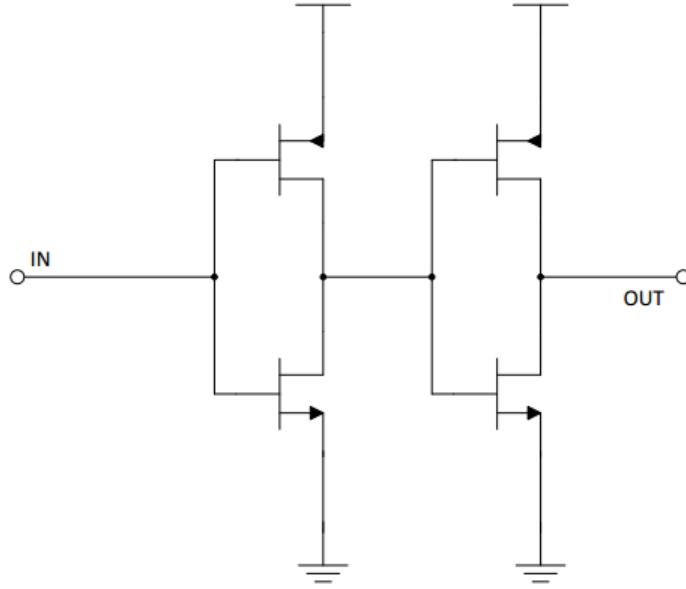


Figure 3.13: Simple inverter driver

The diagram for it is very simple. For the exact size of the inverters, I will optimize it based on the transmission line dimension later.

3.4 Analog receiver circuit

The functionality of the analog receiver circuit is to receive the analog signals emitted from the analog transmitter at the end side of the transmission line and convert the analog signals into digital signals. In the previous part, I mentioned 2 types of analog transmitter circuits. Hence, in this part, I will also introduce 2 types of analog receiver circuit.

3.4.1 RGC TIA configuration

The receiver input is the long interconnect wire. Due to its high ability to isolate the large input capacitance from determining the bandwidth, the common gate (CG) configuration [22] [24] is commonly utilized for high-speed, wide bandwidth operation.

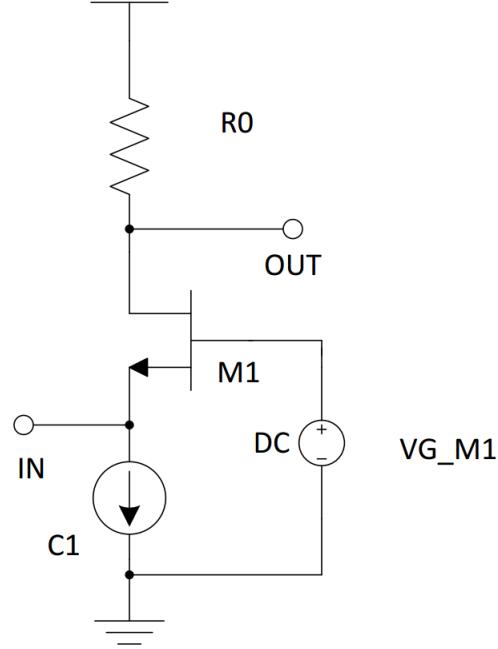


Figure 3.14: Basic common gate (CG) analog receiver

The input impedance for this circuit is $Z_{in} = \frac{1}{g_{m1}}$ for the low frequency input.

While from the optical communication community, an enhanced CG configuration, called Regulated cascode (RGC) trans-impedance configuration is commonly used [23]. The advantage of the RGC configuration is that it can isolate the input impedance more without harming the bandwidth [28].

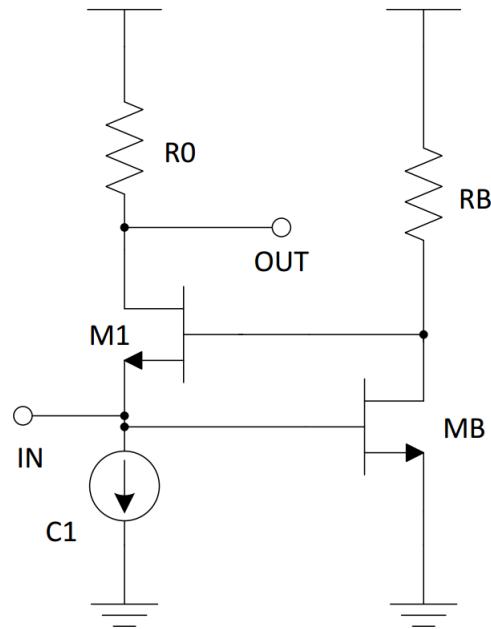


Figure 3.15: RGC schematic diagram

This is the schematic diagram of the RGC configuration. Compared with the CG configuration, there is local feedback at the input stage. With this local feedback, the input impedance in this case is $Z_{in} = \frac{1}{g_{m1}(1+g_{mB}R_B)}$ [23].

Since the RGC input stage is a relatively independent stage compared with the following TIA stage, I optimize this stage first. The input node also needs to supply a DC voltage to the drain of the NMOS transistor for the analog transmitter, and in order to get a proper current for that analog TX, the DC voltage at the input should be larger than a certain value. While for the RGC stage, the lower the input DC voltage, the better the overall performance. So I choose the constraint DC voltage from the analog TX side for the input point DC voltage. For the output resistor R0, I give it an initial value (for the sweep, I will do it in the TIA stage).

One parameter we need to sweep is the DC current for C1. When it is large, we can have high trans-conductance of M1, which can provide small input impedance and wide bandwidth. However, for a certain output resistance R0, too high DC current means no enough output range.

Another parameter that needs optimization is the DC gate voltage for M1. High voltage means we can use small size of M1 to provide the same current, which is good for the output bandwidth, while high voltage will influence the local loop gain of the RB and MB pair.

The last parameters we need to optimize are the size of transistor MB and the value of RB. For a certain DC gate voltage of M1, we will sweep the size of MB and RB to get that DC voltage first, but it will give different output signal swing due to the different local feedback bandwidth [23].

The criterion for the optimal solution set is the output signal swing (the output has a certain capacitance load of course) at a certain frequency. After that stage, we will get the sizes or values of all the components except R0.

The following diagram is the original schematic diagram of RGC TIA receiver. We called the RGC stage the input stage, while the main amplification stage is trans-impedance amplifier – TIA stage.

M3 transistor is used for the common source amplification [23]. Hence the size for it can be pretty big in order to get enough amplification factor. However, the large size of the M3 also contributes large input capacitance at the gate of transistor M3. Since the output point of the RGC input stage determines the bandwidth, we use a source follower M2 to isolate the M3 capacitance from determining the bandwidth of the circuit [23] [28]. However, the insertion of the source follower M2 will decrease the open-loop gain and influence the circuit linearity [23]. The size of M2 should also be optimized for linearity and bandwidth consideration [23].

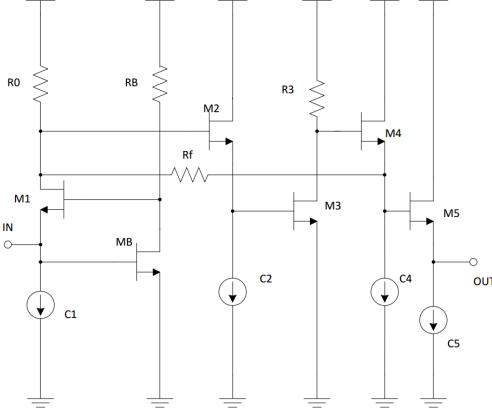


Figure 3.16: RGC TIA schematic diagram [23]

The feedback resistor is applied back to the drain of M1 for 2 reasons[23]. Firstly, since the dominant pole is determined by large R_0 , without applying the feedback resistor to the drain of M1 will result in narrow bandwidth[23]. Due to Miller effect, the value of R_f can be made several times larger than R_0 , thus we can get higher trans-impedance gain[23]. Secondly, it is better not to alter the DC bias conditions for each component after adding the feedback resistor[23]. We should keep the DC voltage at each node of R_f as close as possible[23]. Hence, the feedback resistor R_f is fed back to the drain of M1 [23].

Here, M4 and M5 are 2 source followers to change the DC voltage for a certain application.

However, for our application, the output DC voltage is too low. I need to use the output signal to drive the digital circuit (inverter) directly. Hence it is desirable to set the output DC point of the circuit to be around $V_{DD}/2$ and I need to make modifications for this configuration.

RGC TIA modification # 1

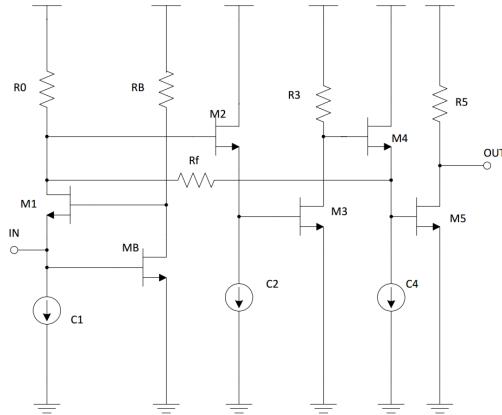


Figure 3.17: Modification of RGC TIA # 1

Change the output stage from source follower to common source stage. It can satisfy the output DC point easily. However, after the second common source stage, we have very high gain, but the bandwidth is highly limited.

RGC TIA modification # 2

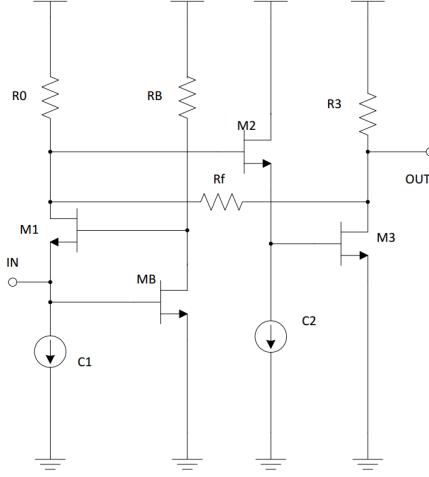


Figure 3.18: Modification of RGC TIA # 2

We eliminate the last 2 source follower stages and apply the feedback between the output point and the drain of M_1 . It satisfies the output DC point and bandwidth condition simultaneously.

For the components in the TIA circuit, as well as the R_0 in the RGC stage that I didn't optimize at the RGC stage. I will do the optimization in the following process.

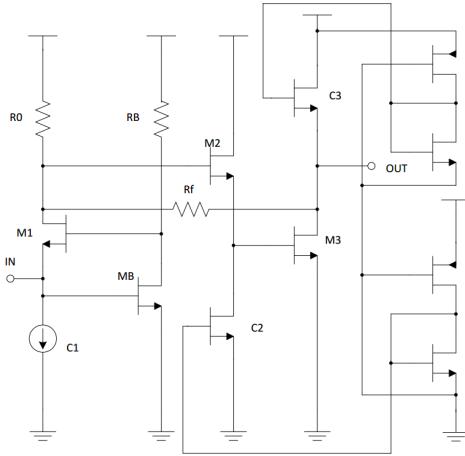


Figure 3.19: Circuit under optimization

Here I give another schematic for the optimization. I replaced the current source with a NMOS transistor that works in the saturation region. While replace the R_3 resistor

with a PMOS that work in the linear region (function as resistor, but I am not sure which gate voltage for it is the best).

Firstly, set an initial value for width of transistor M2; set an initial DC gate voltage for M3; set an initial DC gate voltage for C2 to make it work in saturation region; set an initial value for resistor Rf and set an initial DC gate voltage for transistor C3.

Secondly, sweep the size of M3 for maximum output swing. The size of transistor C3 is changeable to make the DC output voltage to be half VDD. Then we will not change that size.

Thirdly, sweep the DC gate voltage for transistor C3. Still, the size of C3 is changeable to make the output DC voltage to be half VDD. After optimization, I find that when the gate voltage is 0, it will give the best performance.

Fourthly, sweep the value Rf. To keep the output DC voltage to be half VDD, we still need to change the size of C3. After optimization, I find that when the resistance is infinity, we will get the best performance. No feedback is optimal!

Fifthly, sweep the input DC voltage to the TIA stage by changing R0. We can get the optimal R0 here.

Sixthly, sweep the DC gate voltage of M3 by changing the size of C2. Make sure C2 still works in the saturation region.

Finally, sweep the size of M2 for maximum output swing. Maintain constant DC gate voltage of M3 by changing the size of C2.

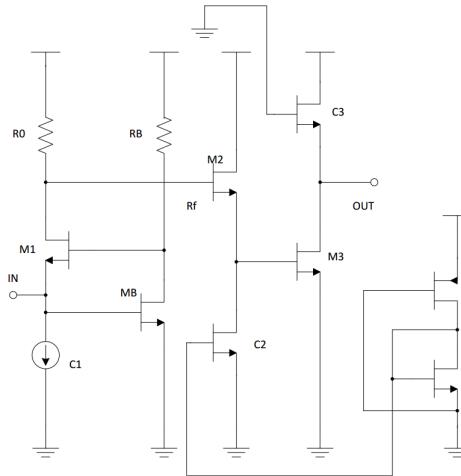


Figure 3.20: RGC “TIA” after optimization

This is the schematic diagram of the RGC TIA configuration after optimization. No feedback exists in the “TIA” stage. The circuit we are exploring needs to work at the limit speed for that technology.

The feedback in this situation will limit the frequency for the loop has to close in order to make the circuit work. This requires higher current in order to work fast but still works slower than the open loop.

Hence, the result is no feedback for the RGC TIA configuration and it is not appropriate to call it trans-impedance amplifier any longer.

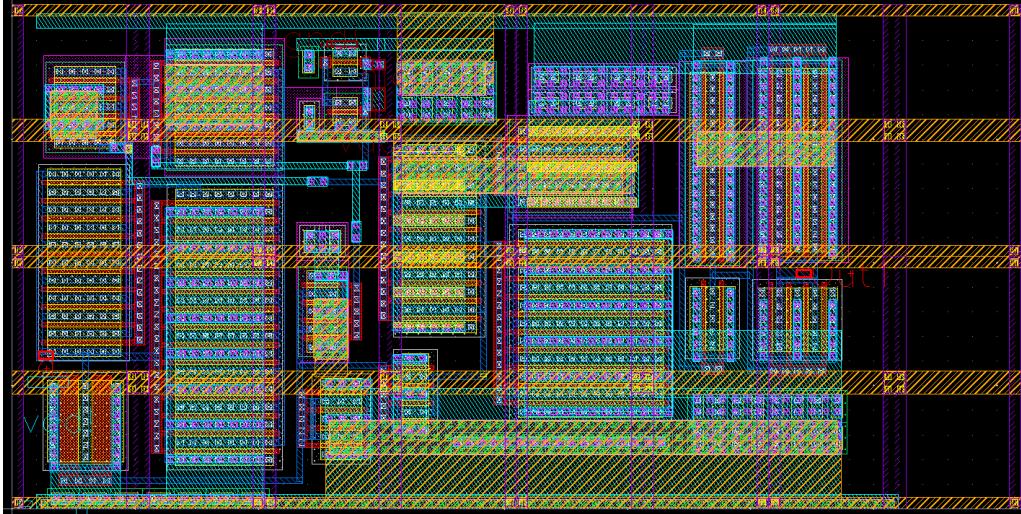


Figure 3.21: RGC ‘TIA’ layout

This is the layout for the RGC ‘TIA’ circuit, here I use PMOS transistor working at linear region as resistors for different corner considerations.

3.4.2 Simple inverter receiver

From the RGC TIA part, we see that after optimization, the RGC ‘TIA’ configuration is not current mode trans-impedance amplifier because of the no feedback in the TIA stage. Thus, perhaps simple voltage mode circuit can also work well for this speed. My advisor A.Unikovski tried several different kinds of circuits first and finally got this configuration. This is the reason why we make this simple inverter configuration as analog transmitter and analog receiver.

The schematic for it is not new. For the optimization, the size of the load for the receiver is fixed (the input to the splitter stage). So we can just use 2 stages of inverters with certain ratio and make sure it works at the target speed. While for the optimization of the inverters at the analog transmitter side, still, we give it initial sizes with certain ratio along with the transmission line model. With different transmission line models, we will get different sizes for the inverters. For example, we have several links that has no transmission line connection, and in this case, there is no need to use the analog transmitter and analog receiver. We can connect the output of LEDR encoder with Splitter circuit directly.

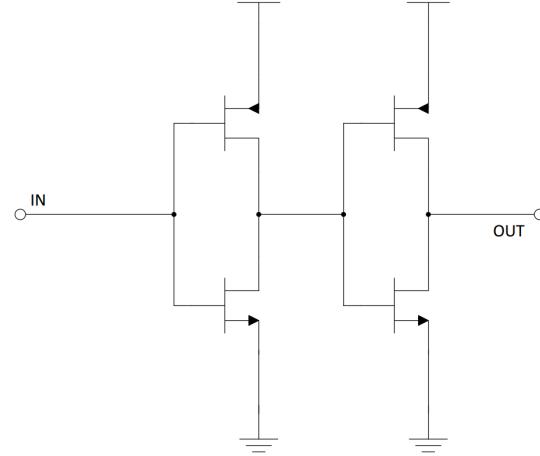


Figure 3.22: Simple inverter receiver

3.4.3 Differential RGC “TIA” amplifier

As we have mentioned before, the differential circuit has better performance for common mode noise rejection compared with the single ended configuration. Since the link we have is also differential link, it's better also to have differential receiver [18].

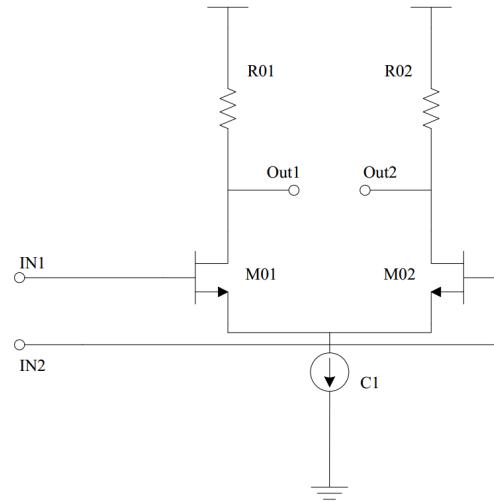


Figure 3.23: Schematic diagram of differential amplifier

This is the differential amplifier we use. The inputs to the differential amplifier are from the output of RGC TIA. But for this case, since the output of the RGC TIA will not be used to drive the digital circuit directly, we can change the output DC voltage of the RGC TIA stage. The outputs of the differential amplifier stage need to have the DC voltage around half VDD.

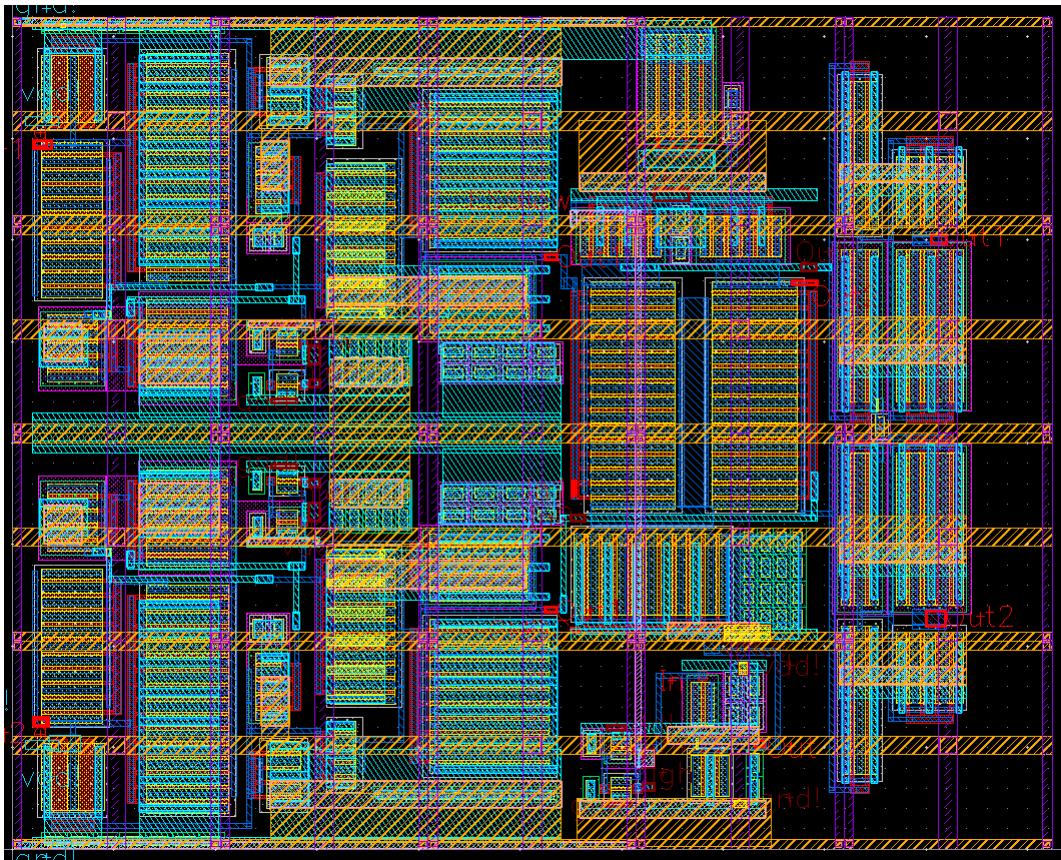


Figure 3.24: Layout of differential RGC “TIA” amplifier

Here is the layout of RGC “TIA” with the differential amplifier. Actually we have 2 RGC “TIA”’s in this layout.

Chapter 4

Test circuit

The chip is for on-chip communication which has many different links for different transmitter-receiver configurations and for different transmission lines. However, the control for each link needs many signals while we don't have enough I/O pads for it. We make a digital controller inside of the chip to control the functionality of the chip, thus we just need to control the digital controller through the I/O pads of the chip.

However, in this test configuration, if the digital controller is not well designed, we will get no information from the chip and we cannot figure out any problems regarding to the failures. So we make some test circuits for the test of the digital blocks, analog blocks as well as to get an idea of how fast the chip really works.

4.1 Speed measurement circuit

The digital controller of the circuit works at low frequency with external input clock signal, while the link works at very high speed under the internal clock generated by clock generator circuit. Due to the high working speed, we cannot output the internal signals directly to the I/O pads. In order to measure the actual working speed of the link, we design a speed measurement circuit. This circuit is used to generate a constant low frequency clock signal. From the simulation, we can get a table of frequencies for the clock generator circuit and this speed measurement circuit. After the chip is fabricated, we can get the clock frequency of the speed measurement circuit. Hence, using the clock frequency we get from the the speed measurement circuit after testing the chip and the frequency mapping table from the simulation, we can get an approximate working frequency of the link [18].



Figure 4.1: Schematic for ring oscillation circuit

This is the schematic for the ring with 41 stages of inverters. The post layout simulation shows the output frequency for this ring is around 259MHz, which is still too fast for the I/O pads to output directly.

So I use another 10 stages of frequency divider circuit to get slower output frequency. The basic element for this frequency divider is Edge triggered D-FF [1].

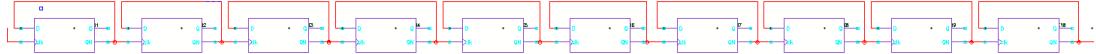


Figure 4.2: Schematic 10 stages D-FF

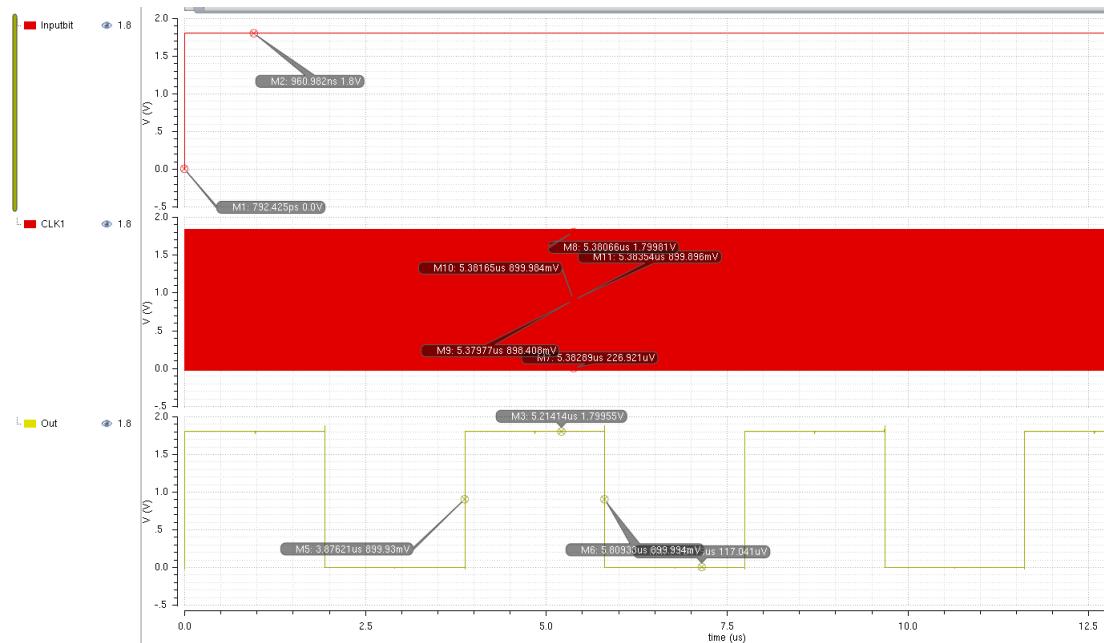


Figure 4.3: Simulation of speed measurement circuit

We can see that the output clock cycle is around 3.95us (253 KHz), which is slow enough to be outputted.

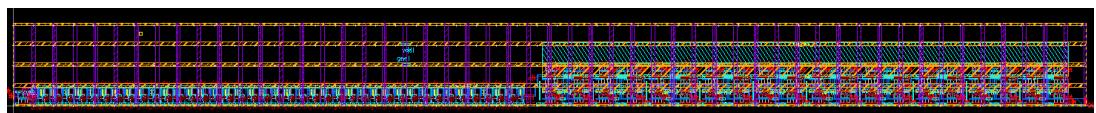


Figure 4.4: Layout for the speed measurement circuit

The left half part is the layout of the 41 stages of inverters (40 stages of inverter and 1 NAND2 circuit) and the right half is the layout of the 10 stages D-FF.

4.2 Test circuit for clock generator

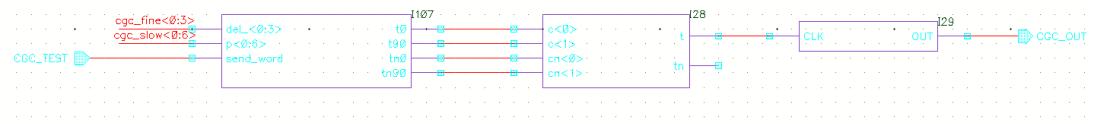


Figure 4.5: Schematic of test circuit for CGC

The purposes for the test circuit of clock generator are twofold. Firstly, we can verify the logic of the clock generator circuit. Secondly, we can get some output clock signal from the output. If the I/O pads can work at a certain frequency, we can get the true frequency of the CGC circuit for a certain input speed control bits.

Moreover, we can get more interesting information about the clock generator circuit. The start bit for the link is one input step function from 0 to 1 transition. However, when we finish one data transition, we need to disable the start bit to reset it to 0 and then for another transition, we will set another 0 to 1 step function. For the delay element, when there is signal change from 1 to 0, there will be delays as well. Thus, the clock generator circuit will generate another clock whose frequency is determined by the 1 to 0 transition delay of the input signal.

After the clock generator circuit, I use 3 stages of the frequency divider circuit. Since the clock generator generates 8 cycles of clock, we can get one output cycle from at the output of the test circuit.

The test for it is a little bit tricky. The input signal to it is a slow clock signal outside of the chip while the control signals for the clock generator are still from digital controller.

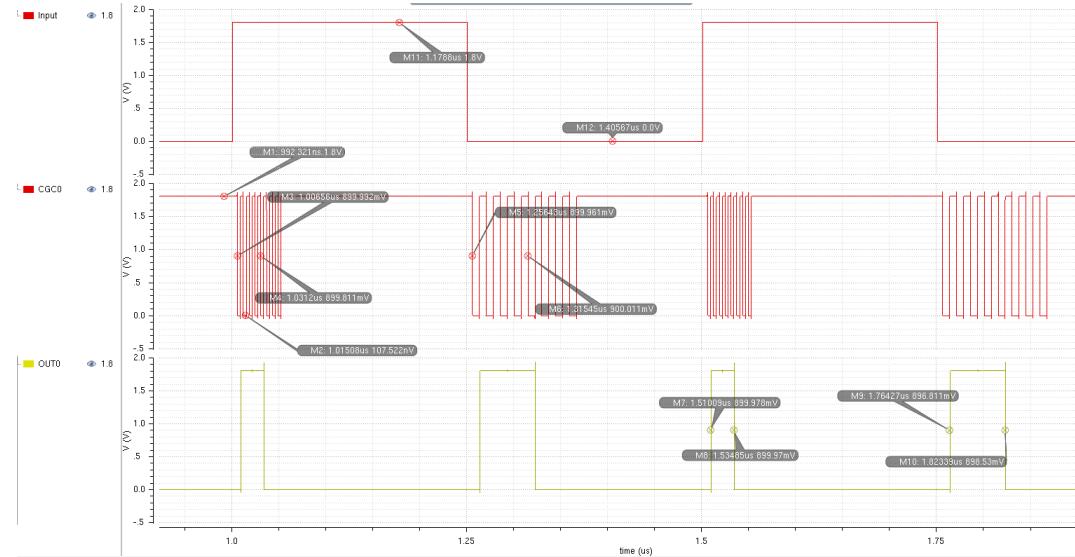


Figure 4.6: Simulation of CGC test circuit

We can see that by using the slow clock signal, we actually generate 2 clock signals: one is from the regular clock generation circuit and one is from the falling edge of the input signal.

4.3 Test circuit for RGC TIA configuration

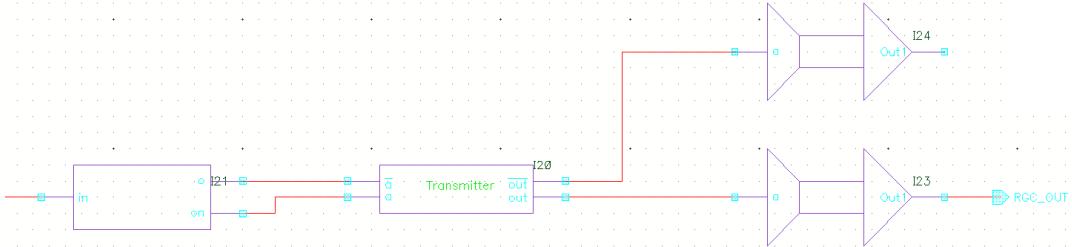


Figure 4.7: Schematic of test circuit for RGC “TIA”

The RGC “TIA” is used for the analog channel, so we also need to include the analog transmitter inside. Since we don’t have enough I/O pads on the chip, we just use one input signal and then use one simple S2D circuit (single to differential converter) to get the differential signal for the analog transmitter circuit. The transmitter will connect with the receiver directly with one output from the receiver connected with one output pad for the chip.

Still the input is slow clock signal, so that we can check the output signal directly. The purpose for this test circuit is to make sure the RGC TIA configuration works on the logic level.

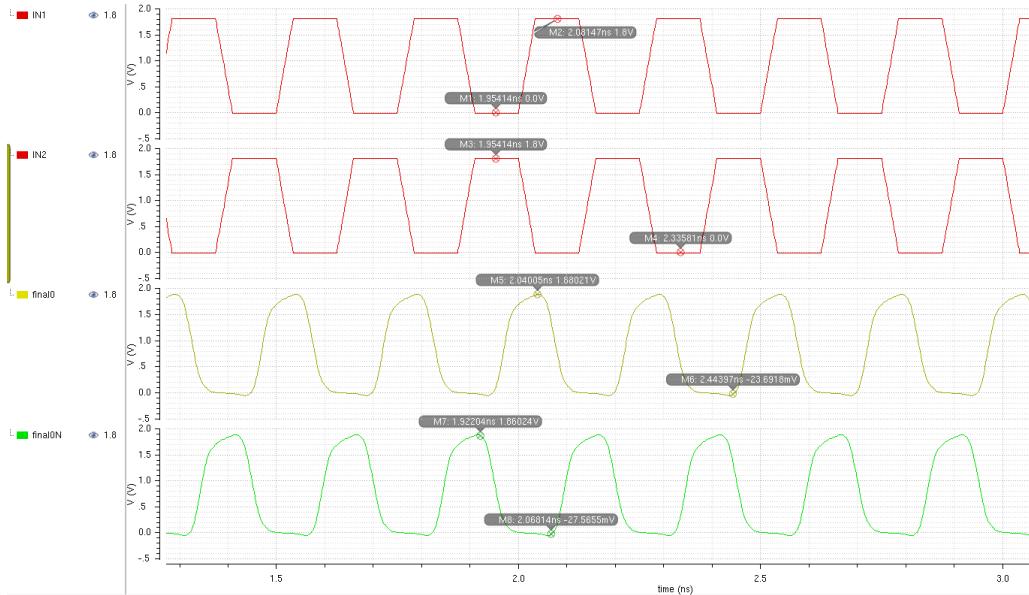


Figure 4.8: Simulation of RGC “TIA” test circuit

We can see that this is actually a very simple test circuit, with input of clock signal, we want the output to be also clock signal. This can also be used to make sure that the Fab works correctly.

4.4 Test circuit for 2T 2X configuration

Similar with the RGC TIA, I also implement the test circuit for the 2T 2X configuration. The only difference is that the 2T 2X is single ended with one input and one output. But the idea is the same: with the input slow frequency signal, we verify the functional behavior of this circuit.

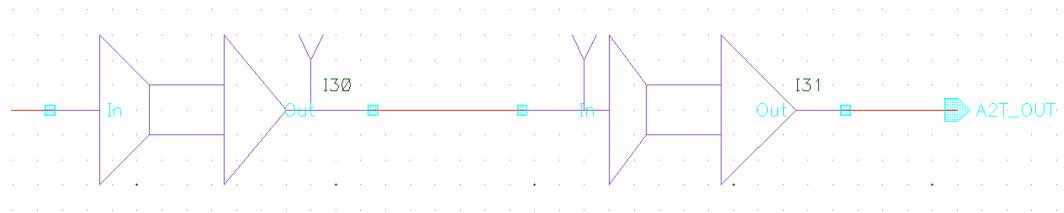


Figure 4.9: Schematic of test circuit for 2T 2X

This is the test circuit part, and to make good use of pads, the 2T 2X configuration and the RGC TIA configuration share the input signal.

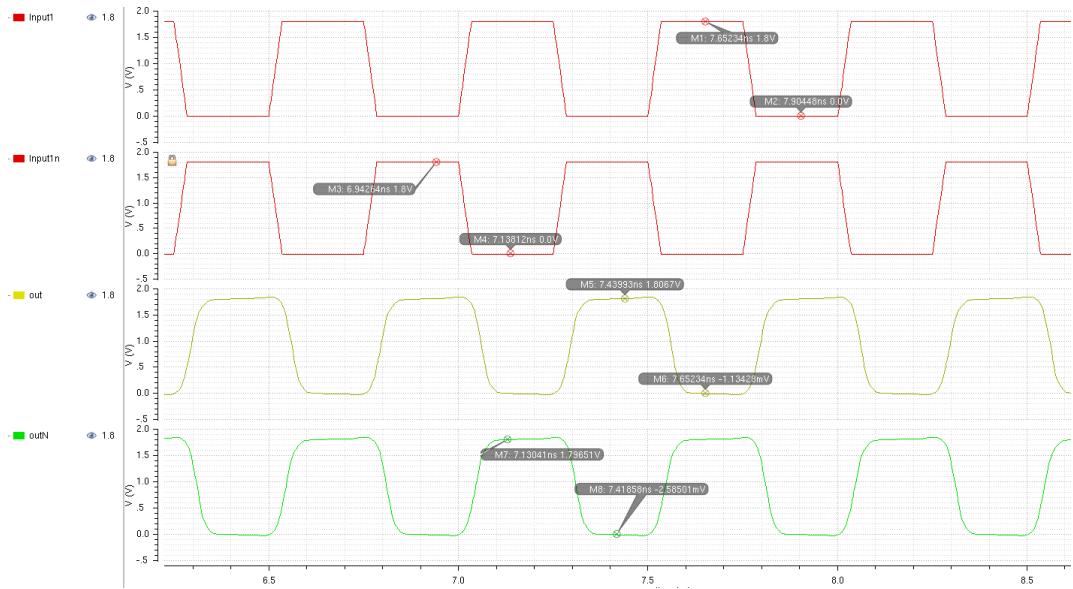


Figure 4.10: Simulation of 2T 2X test circuit

This is the simulation waveforms of the 2T 2X inputs and outputs. We can see that like the RGC “TIA” configuration, this one is really a simple circuit as well.

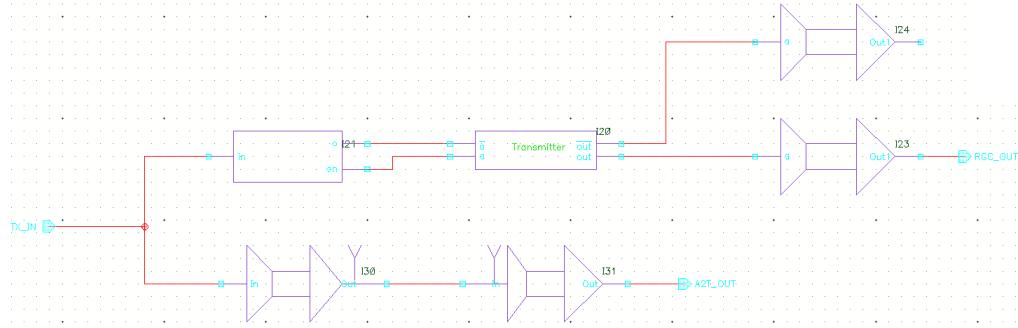


Figure 4.11: Schematic of test circuit for 2T 2X and RGC “TIA”

The TX_IN input is one pad while the RGC_OUT and A2T_OUT are 2 outputs. The reason to give the pin name A2T_OUT instead of 2T_OUT is that for the verilog netlist, it is not legal to use a variable name start with number.

Chapter 5

Simulations and test results

For the simulation result, first we care about how fast the analog link can work (the maximum working speed of the analog transmitter and analog receiver over a certain transmission line model). Secondly, we need to verify the functionality of the chip: the logic of the digital controller, the functionality of the test circuit and the performance of the analog link from the post layout simulation result of the chip.

For the test result, after we get the chip back. We need to test the output frequency of the speed measurement circuit, and the true functionality of the RGC “TIA” and 2T 2X configuration. Moreover, by the frequency mapping, we can approximate the true working frequency inside of the link.

5.1 Performance of the analog link

For the performance of the analog link, the line model I use is the 4 different line configurations as I show before: the basic line, the line with increased characteristic impedance, the line with decreased resistance and the link with increased characteristic impedance and decreased resistance. But for the true geometry, the line length will be different as planned before due to the limited metal layers for layout design. The planned performance is on the left side of the table while the simulation result for true layout is on the right side with the true transmission line length. Still due to the limited metal number, all the transmission line are in the Z_{0a} mode.

Link #	Conf	Fre[G]	Pow[mA]	line[mm]	Fre[G]	pow[mW]	E[pJ/bit]
1	Z0a AL	8.6	5.72	2.61	13.9	70.7	5.65
2	Z0a 6AL	11.5	6.8	3.06	13.8	74.2	5.96
3	Z0a AL 2S	10.2	11.97	3.46	14.3	120.7	9.41
4	Z0b AL	12.9	3.53	3.98	10.6	40.9	4.35
5	Z0b 6AL	10.8	1.66	4.42	7.1	23.0	3.61
6	No line			4.87	8	21.1	2.97

Link #	Conf	Fre[G]	Pow[mA]	line[mm]	Fre[G]	pow[mW]	E[pJ/bit]
7	RGCTIAZ0aAL	8.3	20.8	5.28	10	37.4	4.2
8	RGCTIAZ0a6AL	9.8	21.5	5.72	10.1	37.7	4.2
9	RGCTIAZ0bAL	11.4	22.43	6.16	8.2	36.8	5.0
10	RGCTIAZ0b6AL	12.2	23.33	6.61	7.9	36.6	5.1
11	RGCTIANoline	11.5	23.35	2.88	10.8	38.8	4.0
12	DFAMZ0aAL	8.5	24.16	3.34	10.6	44.1	4.68
13	DFAMZ0a6AL	9.8	24.72	3.78	11.2	44.2	4.45
14	DFAMZ0bAL	11	25.54	4.22	9.6	43.1	5.10
15	DFAMZ0b6AL	11.3	26.01	4.66	9.1	42.7	5.30
16	DFAMNoline	11	25.98	5.10	8.9	42.6	5.38
17	Z0c AL	10	5.85	5.56	12	74.6	6.94
18	Z0c 6AL	12.4	7.48	6.06	11.7	75.7	7.11
19	Z0c AL 2S	11.4	13	6.38	12.9	124.3	10.5
20	Z0d AL	12.47	3.46	3.11	11.7	43.42	4.02
21	Z0d 6AL	10.8	1.66	2.74	9.2	23.65	2.93
22	RGCTIAZ0cAL	8.8	20.99	2.39	9.7	37.57	4.23
23	RGCTIAZ0c6AL	10.4	21.92	2.16	11.5	39.19	3.80
24	RGCTIAZ0dAL	11.3	22.42	0.90	10.8	39.2	4.00
25	RGCTIAZ0d6AL	12	23.23	0.67	10.7	39.15	4.04
26	DFAMZ0cAL	9.2	24.47	0.42	11.0	45.16	4.39
27	DFAMZ0c6AL	10.8	25.18	0.23	12.0	45.16	4.64
28	DFAMZ0dAL	10.8	24.45	2.26	10.7	44.46	4.64
29	DFAMZ0d6AL	11	25.83	2.03	10.8	44.55	4.62
30	AHCM	9.4	42.11	1.74	15.4	72.74	5.25

Table 5.1: Link configuration and performance summary

For the simulation, I use 3 different line lengths and 4 different line geometries. The lengths I choose are 10mm transmission line, 6mm transmission line and no line. The a, b, c, and d here stand for basic transmission line, the line with increased characteristic impedance, the line with decreased resistance and the line with both the increased characteristic impedance and the decreased resistance respectively.

We can see that for the basic 10mm length transmission line, the 2T 2X configuration works at the speed of 8.6Gbps with 9.32mW power dissipation. For the RGC “TIA” configuration, the best result is 8.3Gbps with 37.44mW power consumption. Here I also give the result of the results of the RGC “TIA”differential amplifier. Still for this length line model, the highest speed for it is around 8.5Gbps with the power consumption of 43.49mW. Hence, under this line model, both for the speed and power

consideration, the 2T 2X configuration gives better result. Perhaps 2T 2X configuration is more suitable for this on-chip communication for that Tower 0.18um Technology unless other novel current mode configurations are explored.

5.2 Performance of the chip from simulation

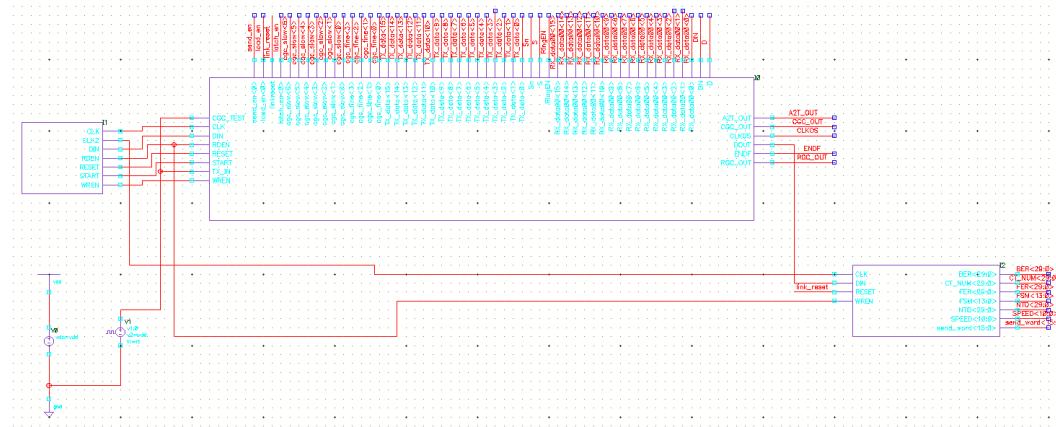


Figure 5.1: Chip test bench

This is the test bench for the chip. I use digital blocks to generate the input control bits and data bits to the chip and use another digital block to keep the data outside of the chip.

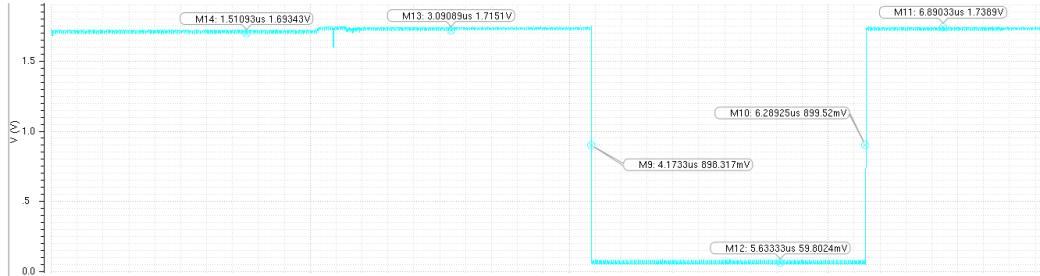


Figure 5.2: Simulation result for speed measurement circuit

This is the clock signal from the ring oscillation circuit and we can see that the output frequency is around 235.8 KHz, not exactly the same as the simulation result. However, we can see that the voltage range is not 0 to 1.8. I think it is because of the voltage is not idea that causes this problem. Anyway, it works.

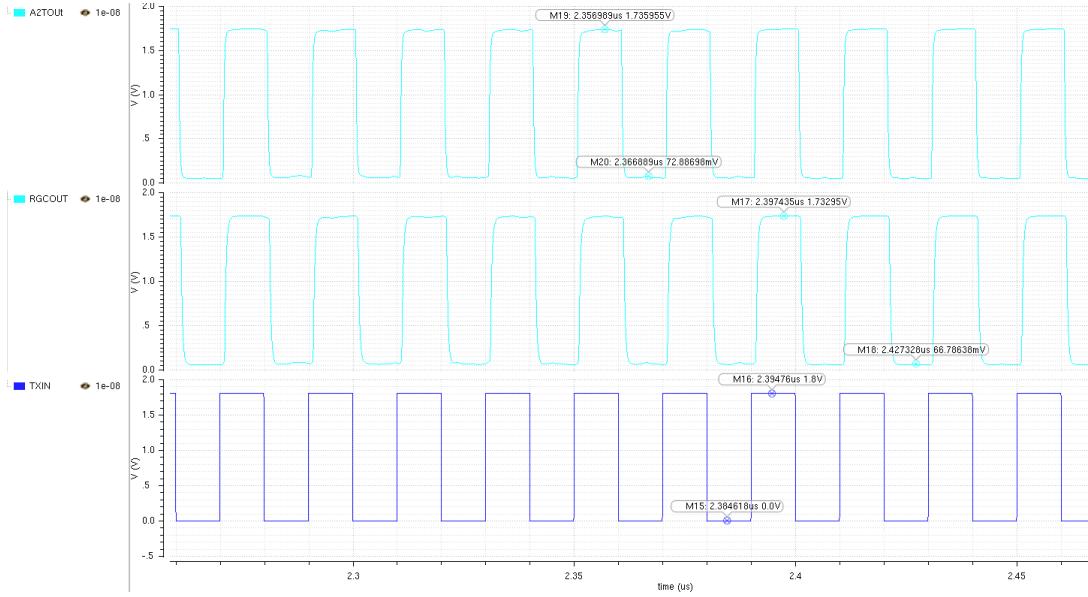


Figure 5.3: Simulation result for RGC ‘TIA’ and 2T 2X circuit

This is the test circuit for RGC ‘TIA’ and 2T 2X configuration, we can see that even with the ideal low frequency input signal, the output signals are not full range from 0 to VDD. Still, both of these 2 configurations work.

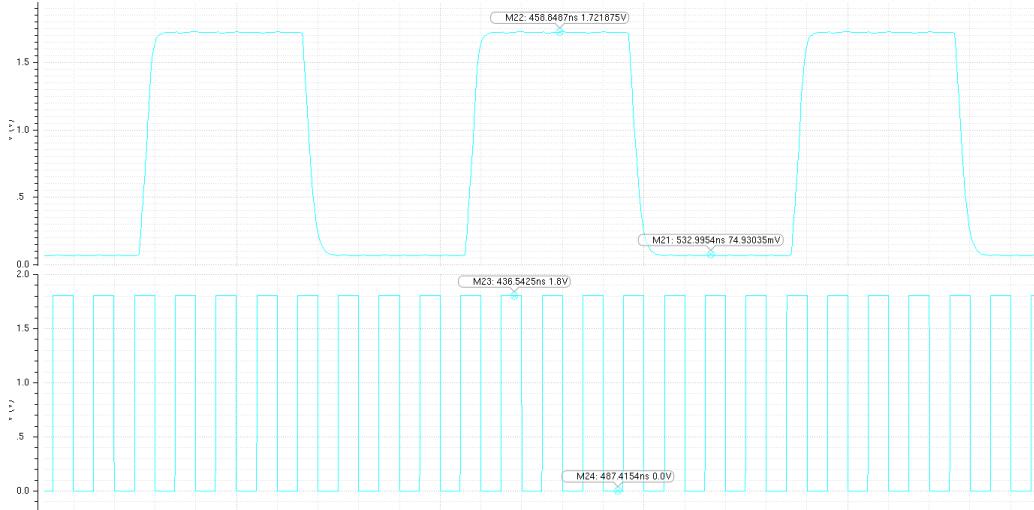


Figure 5.4: Simulation result for CGC test circuit

The input clock frequency is not low enough, so the output signal cannot represent the true property of the clock generator circuit. Anyway, it will give us some result and for the true test, we can modify the input signal frequency (Perhaps need to update it, but post layout simulation for the chip is not a good idea (20 days simulation)).

For the whole chip simulation, I want to verify 3 parts: write data to digital controller; the functionality of the analog link; write the data to digital controller.

Figure 5.5: Input data to the digital controller

We can see that the input data bits are 0000000000111111 (0X0003F) and the speed control bits are 1110000-1101. Link 0 is selected and the test data number is just 1 (for the easiness of checking the simulation result and use short time to get the final result).

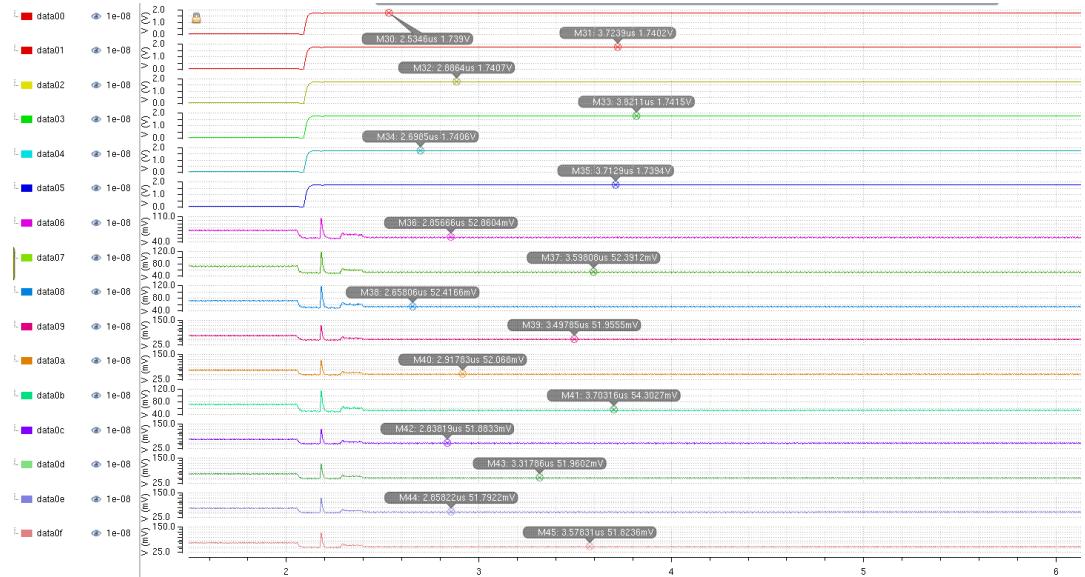


Figure 5.6: TX SRs data

This is the data at the inputs of the TX SRs that is ready to be sent, and we can see that this is exactly the same data that is loaded to the input of the digital controller.

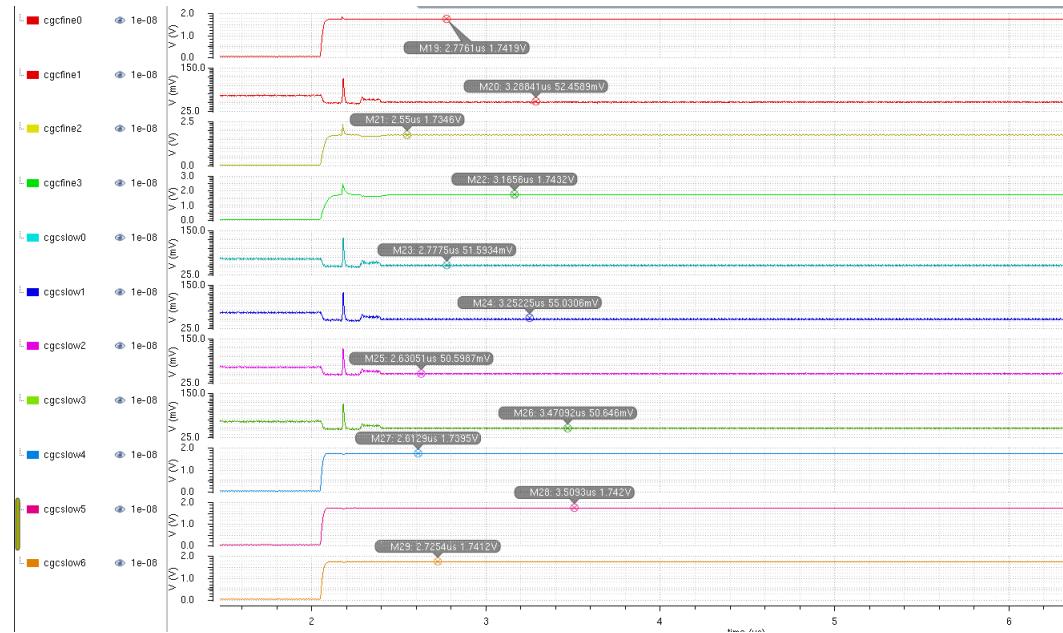


Figure 5.7: Speed control pattern

Here is the speed control pattern at the output of the digital controller and also the inputs to the analog links. We can see that these signals are the same as the input signals to the digital controller.

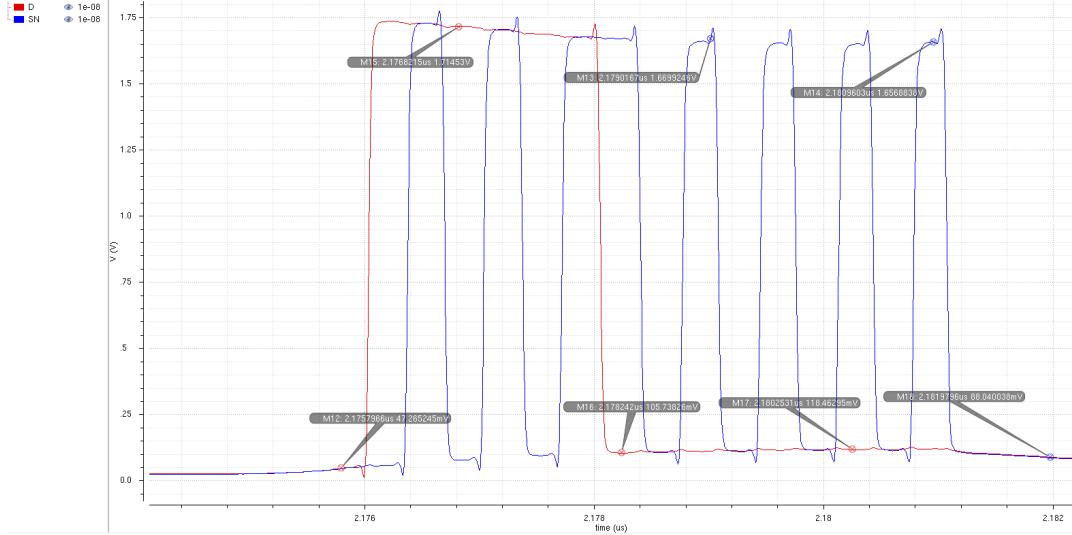


Figure 5.8: Waveform on the channel

We can see that on the channel, we have the correct data bits and strobe bits (I mistook S and SN in the label, thus, here SN is the real data of S). From link configuration, we know that link 0 is the 2T 2X link, thus, the data on the channel is the real data we transmitted from the TX side. Moreover, we can make it certain that the digital controller successfully loaded the input data to the link.



Figure 5.9: RX SRs data

We can see that when we get the latch enable signal, the output bits from the RX SRs are exactly the same as the input data. Thus, we can claim that the link works fine.

I made a mistake in the reading out of the DOUT when test the circuit (but this is not the dysfunction of the chip). So there is one bit shift from the label, but never mind, we can get a glance of the output data.

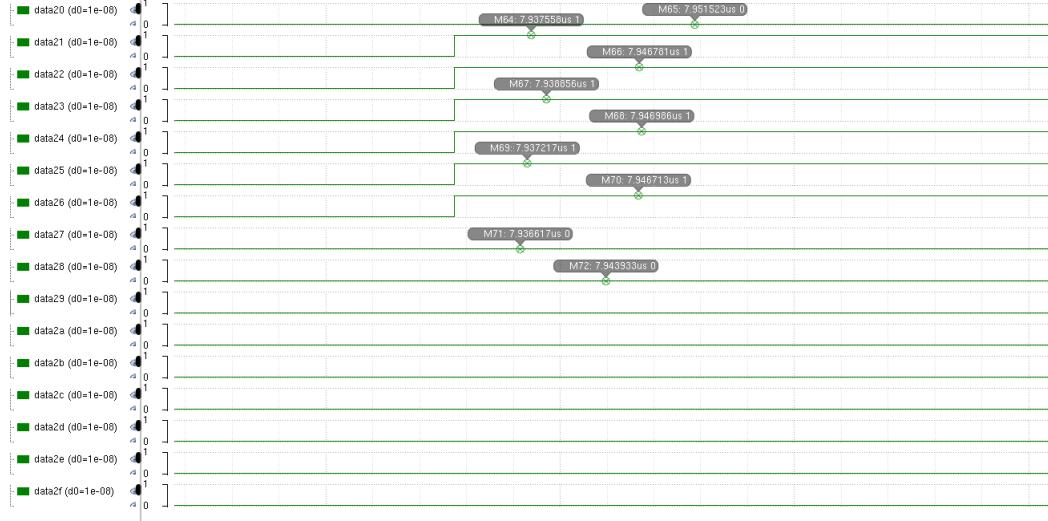


Figure 5.10: Read DOUT from the chip

We read out the data from the chip and I made a mistake in the phase of the clock signal for the read out module, just one bit shift. But we can see that the data is correct.

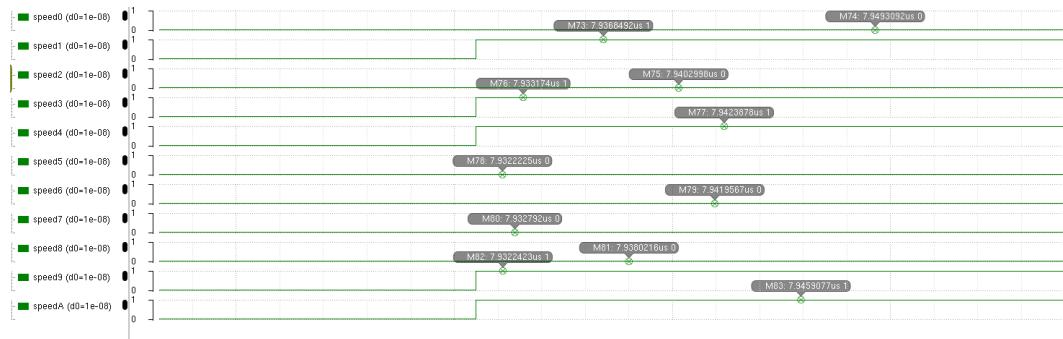


Figure 5.11: Read out the speed control bit

As the case for the data bits, we can get the data, but there is bit shift from the test bench and since it took more than 20 days for this test, I would like to leave it.

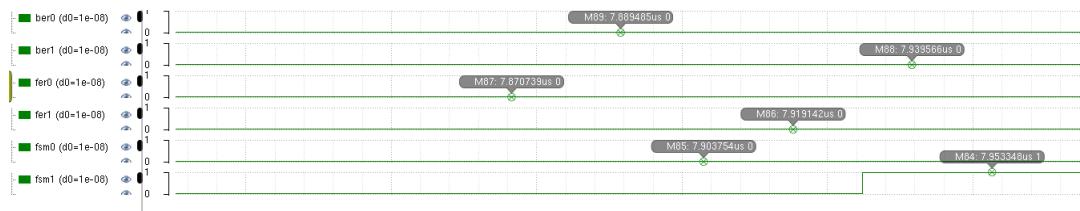


Figure 5.12: Read out the FSM, BER and FER bit

Finally is the finite state machine status (FSM), bit error rate (BER) count and frame error rate (FER) count. For the details of it, please refer to the appendix 7.1.

5.3 Performance of the chip in the lab

We have fabricated the chip and the test PCB board of the chip has been ready. Here is the test board:

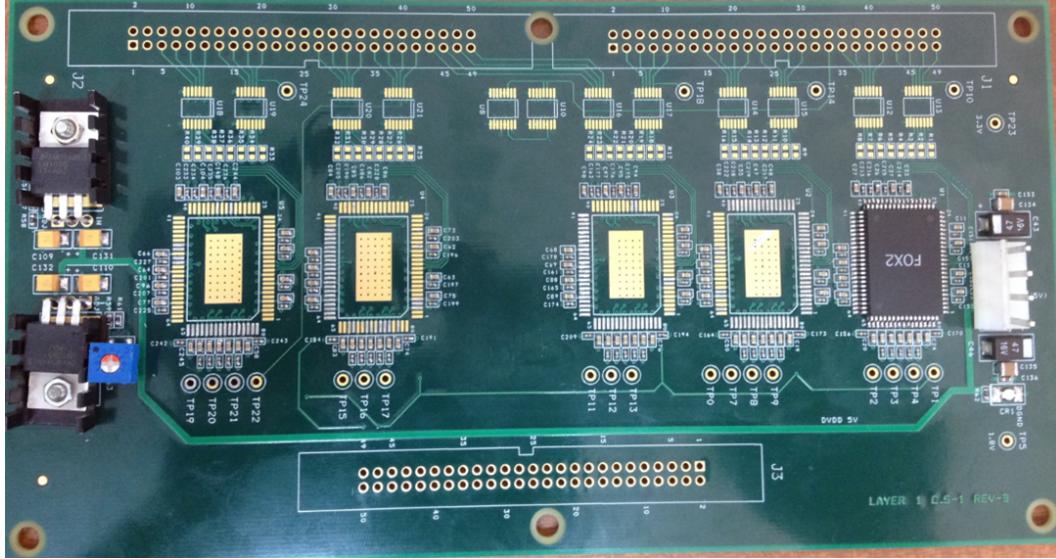


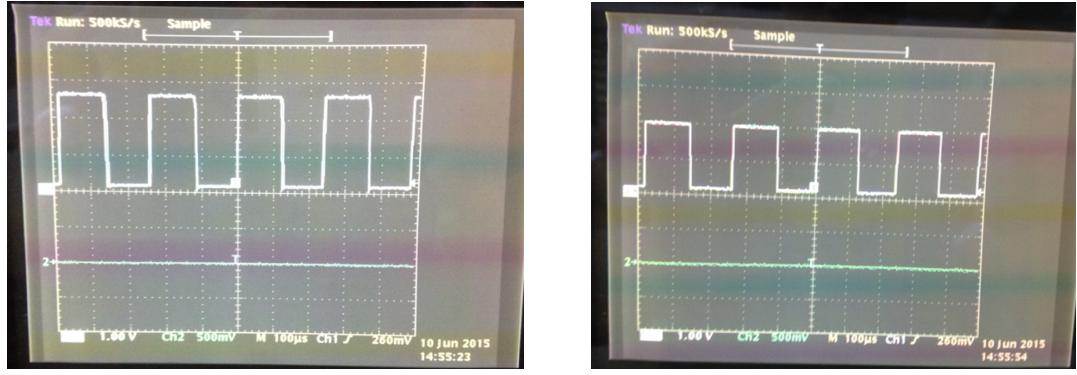
Figure 5.13: FOX2 test board

Actually the board is designed to accommodate 5 FOX2 chips. However, from the constraints of the voltage regulator circuit, it can only provide around 3.0A current while the DC current consumption of the FOX2 chip is around 1.0A. In order to make sure the regulator is not too hot when it works, we just solder 1 or 2 chips on the board.

The order of the test is in the following. First is the 2T 2X test circuit with slow input signal. Secondly, is the RGC “TIA” test circuit with slow input signal. Then we will check the output signal from the CGC output port. After that, we will get the Clock cycle of the CLKOS output. Finally the functionality and performance of the link will be added.

5.3.1 2T 2X test

Because the input to the 2T 2X is from the shift level driver directly, thus, the signal level is not 0 to Vdd (1.8V) but 0 to 3.3V, but still, we can get the correct 0 to Vdd signal at the output. From the test environment constraint, the input frequency is just around 200KHz. But from the test result, we are pretty sure that this technology works and if the chip doesn't work as we expected, then there must be something wrong within the design.



(a) 2T 2X input

(b) 2T 2X output

Figure 5.14: 2T 2X test circuit

5.3.2 RGC TIA test

The RGC “TIA” test here is harder to verify compared with the 2T 2X configuration since it’s analog circuit and output is highly depending on the output DC voltage before the inverter pair. Here is the test result.

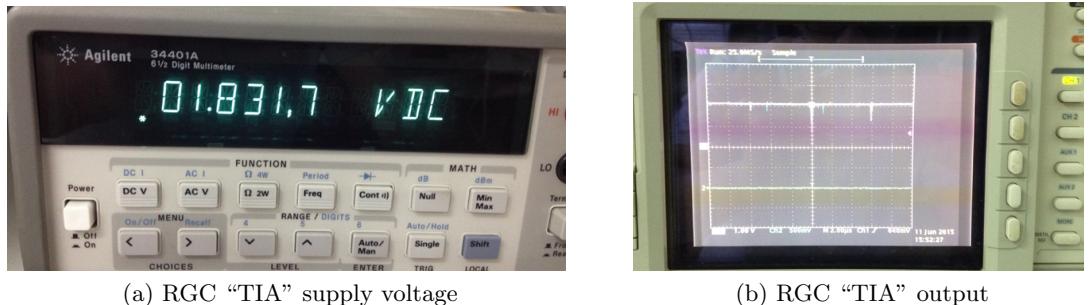


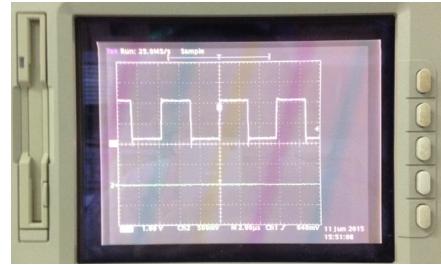
Figure 5.15: RGC “TIA” test circuit

We can see that when the supply voltage to the chip is around 1.832V, we cannot get the correct output signal even the input frequency is very slow. But in this case, we can see that the output signal has glitches. And from simulation, I knew that when increasing the supply voltage, the DC output voltage of the RGC “TIA” stage will be lower, thus will result in lower output signal level. Hence, I tried to increase the supply voltage and check the result.

We can see from the test that when the supply voltage to the chip is around 1.86V, the RGC “TIA” circuit will also give the correct signal. Thus, the RGC “TIA” configuration also works. However, due to the fact that it is so sensitive to the supply voltage. I cannot guarantee that it will give good performance when we test the chip. For example, it will be even very hard for us to find what supply voltage is best for the RGC “TIA” configuration for the performance consideration. Thus, the performance analysis of the



(a) RGC “TIA” supply voltage



(b) RGC “TIA” output

Figure 5.16: RGC “TIA” test circuit

chip will be based on 2T 2X links.

5.3.3 CGC test

The next circuit of the test is the CGC test, the input control signal for the CGC is also from the digital controller of the chip. Thus, in order to test it, we need to make sure the digital controller works fine. Luckily, I have known in advance that the digital controller works.

Simulation of the CGC in the previous part of this chapter is not very practical since the input frequency is at really high speed. However, for us, we can just inject slow input signal for the CGC circuit. In this case, the simulation gives the following result.

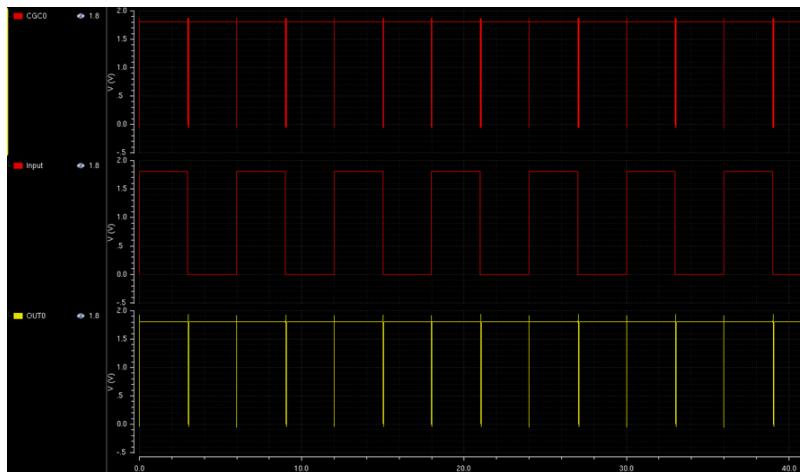


Figure 5.17: CGC simulation result

In this case, we can see that the output signal is just negative pulses. After zooming in the picture, we can see the signals like this.

We can see that the output signal is the negative pulse, while the CGC0 signal is the signal after the 3 stages of XOR after the delay elements which has 8 cycles. Moreover, from the CGC circuit, we know that when a step function comes, it will generate a

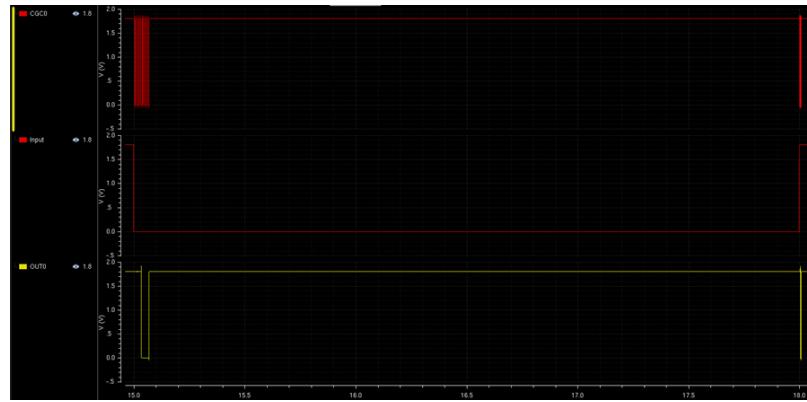


Figure 5.18: CGC simulation result

clock pulse, while when this signal is reset, we will get another clock signal with the same number of pulses. But since the negative step function has different circuit path, the delays from the output will be different. Here is the simulation result.

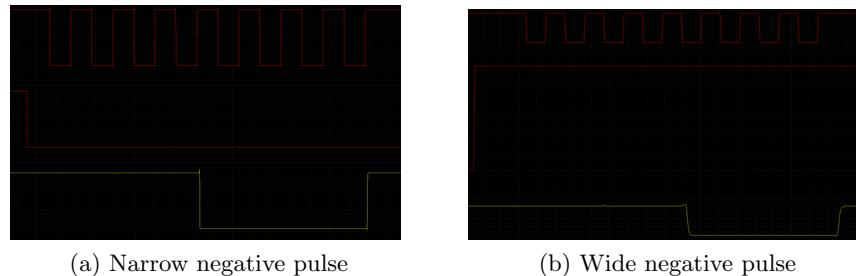


Figure 5.19: CGC simulation result

Here we can see 2 negative pulses with different signal width. From simulation, the width of the 2 negative pulses are 58.9ns and 24.5ns respectively while from the oscilloscope, the 2 negative pulses have width 62.5ns and 35ns.

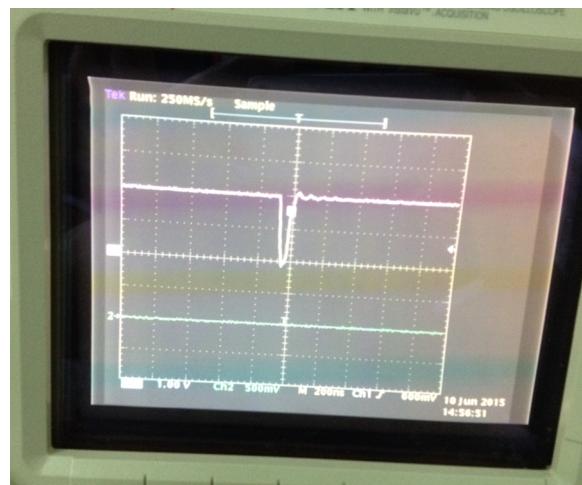


Figure 5.20: CGC test result

The signal at the output of the oscilloscope is not very ideal since the rise time and fall time of the signal is pretty large in the order of ns. But anyway, we can see that the simulation result matches the test result well.

5.3.4 CLKOS test result

As I have mentioned in the previous section of this chapter, the CLKOS signal from simulation is around 235.8KHz. For the test, the output clock frequency is slow enough, thus we can test the CLKOS signal directly.

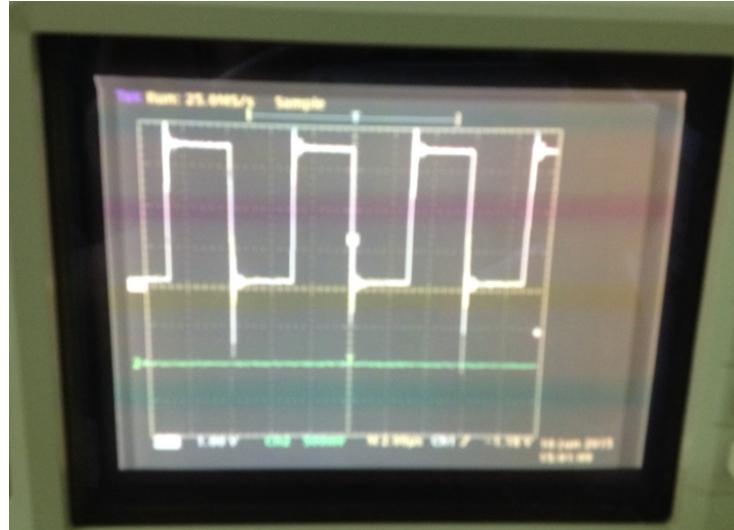


Figure 5.21: CLKOS test result

From the test, we can see that the clock cycle is around $6\mu s$, which corresponds to 166.7KHz and it is pretty different from the simulation result. But since we don't know the details of the conditions that the chip work with, for example, the environment temperature, the uncertainty of the technology, etc. Here I add more simulation result of the CLKOS circuit with different corners.

Corner\Temperature	25°	80°	100°
Nominal	235.8KHz	217.6KHz	211.8KHz
Slow	189.0KHz	172.8KHz	167.8KHz

Table 5.2: CLKOS speed with different corners

Here we can see that with the slow corner and 100° temperature, the CLKOS speed from simulation is almost the same as the result we get from the test. Perhaps in reality, we have slow corner for the chip and the working temperature is around $80 \sim 100^\circ\text{C}$.

5.3.5 Link # 0 functionality test

Since the primary analog interface is the 2T 2X configuration, the functionality analysis and performance analysis of our link will be focused on 2T 2X configuration. The functionality test is based on slow clock frequency, for example, I used the slowest possible speed control pattern which is around 250KHz for the 1.8V supply voltage.

However, the word length in our research is 16 bits, thus, in total, we can have 2^{16} different words/bit patterns. It's impossible to list all the words. Here I just listed some "hard" words, for example, the one just have 1 bit 1 while all other bits are 0s. Because in this kind of word, it's very easy to lose this 1.

Input data	Correct?	Received data
0000000000000001	Y	0000000000000001
0000000000000010	Y	0000000000000010
00000000000000100	N	0000000000000001
000000000000001000	Y	000000000000001000
00000000000010000	Y	00000000000010000
000000000000100000	Y	000000000000100000
0000000001000000	N	0000000000010000
0000000010000000	Y	0000000010000000
0000000100000000	Y	0000000100000000
0000001000000000	Y	0000001000000000
0000010000000000	N	0000000100000000
0000100000000000	Y	0000100000000000
0001000000000000	Y	0001000000000000
0100000000000000	N	0101000000000000
1000000000000000	Y	1000000000000000

Table 5.3: Link # 0 test

We can see that even for the slow speed situation, due to some unknown reasons of our serial architecture, we can not avoid bit error and frame error for our system.

5.3.6 Link # 0 performance analysis

For the performance analysis, I will begin with the slow speed pattern as well. But in this case, I will not try to check what the output data with a certain input data pattern. I start with a simple data and then use multiple words transmission method from the

GUI (graphic user interface) of the LabView. After the transmission is finishing, I will read the BER, FER and NTO of this link.

The total words I plan to send is around 2^{20} , which is around 1 million words. As I mentioned in the previous section, since the word is 16 bits long, we can have 2^{16} different words. From the property of the linear feedback shift register (LFSR) in our digital controller, we can get that with any seed signal besides all 0s, we can generate all the other $2^{16} - 1$ words. Thus, for the case that we set the number of transmitted words to be 2^{20} , all the words appear at the same probability to the entire transmission and the BER, FER and NTO will be the average of the all the transmitted data on the basis that all the words have the same transmission probability.

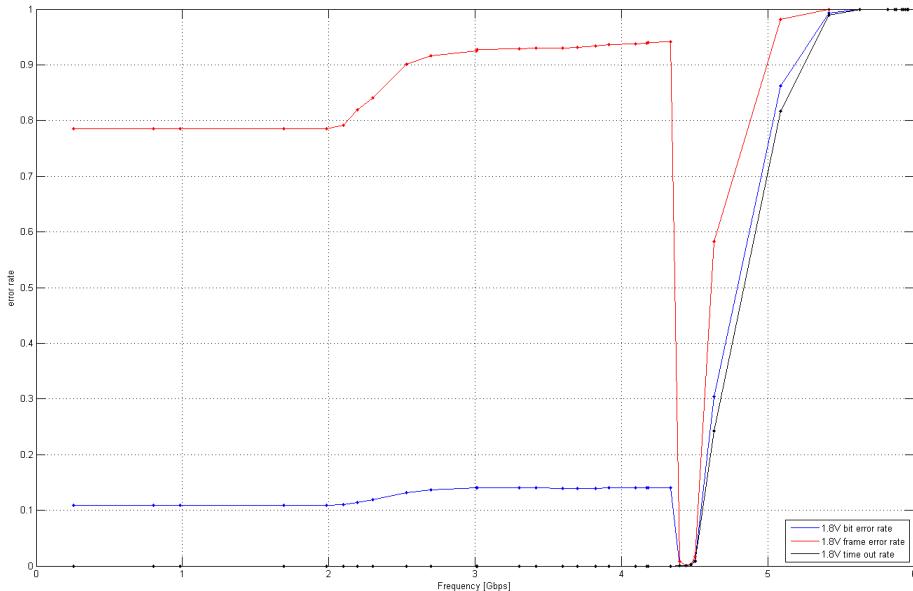


Figure 5.22: BER, FER, NTO vs. Frequency

From the calculation we can see that at low frequency, we have some constant BER, FER and no NTO. The BER and FER will change at the frequency around 2Gbps. Without doubt, the NTO should always increase with frequency since the higher the frequency, the higher probability that the clock cannot be recovered. But here we have a very strange result – at the frequency around 4.5Gbps, we have a low BER and FER point, but at this frequency, the NTO is not 0... Currently, I have no idea about how to explain it.

But in theory or from simulation, the data transmitted and received at the low frequency should be correct. I will do another analysis based on this assumption now. Assume the transmitted 1 million words form one long word A , and the received long word is C . Then the Hamming distance between A and C is the total bit error. Under the assumption that C is the correctly received word, then we need to add something to the

received word in order to make the Hamming distance between these 2 words to be 0. Thus, I add $A - C$ to the received word. Thus, the revised Hamming distance between the transmitted words and the received word would be $| A - C + A - C | = 0$. Then for the case that we received word D , the BER is $| A - D + A - C |$, from the theory of metric, we can get $\| A - C \| + \| A - D \| \leq \| A - D + A - C \| \leq \| A - C \| + \| A - D \|$. The upper bound is always true and will be a pretty big number in our result. Here, we assume the lower bound holds and using the data we get before data time out occurs, we can get the following result.

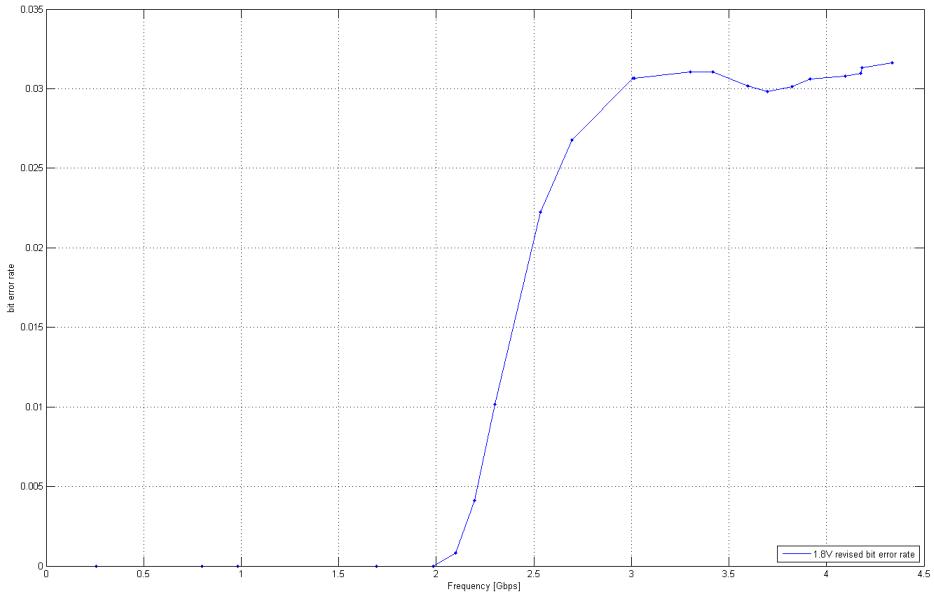


Figure 5.23: Revised BER vs. Frequency

We can see from this result that the BER is 0 at low frequency and at about 2.0Gbps, we begin to have BER and BER always increases with frequency. At around 4.5Gbps, we begin to have NTO, so I cannot use this method to get the BER at that high frequency. I would like to say that the performance for this link is around 2.0Gbps under 1.8V supply voltage.

5.3.7 Interesting result for link # 29

When I was analyzing the performance of all the link, I got some strange results regarding to several links. For example, for link # 29. We know that for the 2T 2X links, when the input data is 0000 0000 0000 0100 at low frequency, the received word pattern is 0000 0000 0000 0001. However, for the # 29 link, when the input is 0000 0000 0000 0100, the received data can be correct. For example, I conducted a test for 2 words within which the second word is 0000 0000 0000 0100, when the first word is 0011 0011 0011 0011, we can recover the data correctly. However, if the first word is 0000 0000

0000 0000, the received word will be error again. Currently I have no idea about why this happened and how to explain it. Here is the results in the GUI.

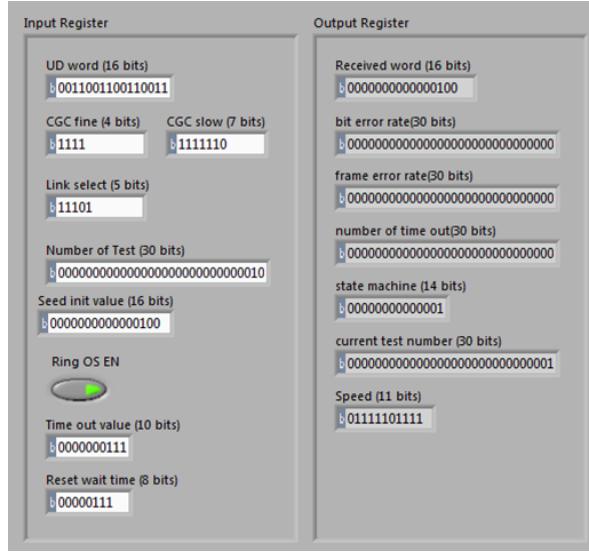


Figure 5.24: Receive 0X0040 correctly

Here when the first word is 0X3333, we can get the correct result.

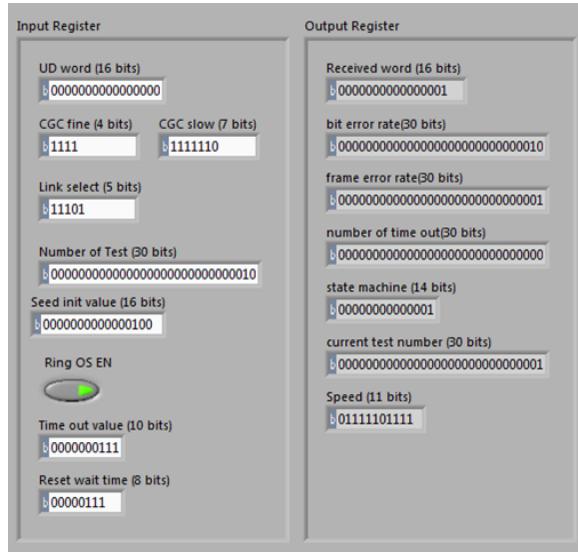


Figure 5.25: Receive 0X0040 wrongly

When the first word is 0X0000, it will give error.

Chapter 6

Summary and conclusions

6.1 Research summary

This research is based on a PhD research of “Asynchronous Current Mode Serial Communication”[11] (<http://goo.gl/HFej7B>) of Doctor R. Dobkin. For this research, I fully adopted his idea of clock generating, parallel to serial transition, merging, splitting and serial to parallel transition. The novel parts of this research are the modifications of the delay element, the new methods to get the transmission line model, the design of the RGC “TIA”analog receiver and the idea of using 2T 2X pair as analog transmitter and analog receiver.

It's a big project, which took us around 2 years (more than 20 months) from the time we began the FOX2 project to the time we submitted the final gds file to Tower for fabrication. Many undergraduate students participated into this project as the project they took from the Lab. G. Albert and I. Shaham optimized the delay element for FOX2 after taking the schematic from FOX1, and due to the large area of CGC/delay element in FOX1, we made some modifications and they updated the new configuration. Another group of students D. Poliakov and S. Weiner implemented the layout of the delay element and clock generator circuit. A. Cohen and V. Vax optimized the schematic for the XOR circuit in FOX2 and A. Cohen and I made 2 different versions of layout for XOR for different application part. A. Stolero and Y. Oren optimized the schematic for Shift Register Circuit for the transmitter side and T. Carbone implemented the layout for the this SR. A. Cohen and V. Vax also optimized the schematic for the LEDR Encoder, and after that A. Cohen implemented the layout for it. For the analog transmitter part, D. Perez adopted the schematic from FOX1 and optimized it with a certain version of analog receiver. I got the transmission line model with the suggestions from D. Nahamanny and A. Unikovski. For the analog receiver part, I optimized the RGC input stage and then made changes to the TIA stage. With the results we got from RGC TIA, A. Unikovski suggested the 2T 2X voltage mode configuration. A. Karako and A. Giterman optimized the toggle circuit and after that they implemented

the layout for it. I optimized the schematic for the other parts of the splitter circuit and implemented the layout for it. The Shift Register circuit at the receiver side is a little bit forgotten, so I made small optimization, but changed the control signals and interfaces based on the Shift register at the transmitter side and again, T. Carbone implemented the final layout for it.

I got the original VHDL code for the digital controller from R. Dobkin. After understanding the differences between FOX1 and FOX2, I re-wrote it in verilog. With the help of G. Samuel, I went through the back end flow of the digital controller and passed the gate level simulation with R. Dobkin's help. Moreover, the post layout simulation for the digital controller also works. For details of it, please refer to the appendix 7.2.

For the integration of the transmitter and receiver, I implemented step by step as the layout I did for the other blocks – appendix 7.3. After that, I had the layout of transmitters, receivers, digital controller and test circuits. By following appendix 7.4, I got the final layout of the chip. Of course, in order to test it, we have to make package for the chip, the package and pin location of the chip can be found in appendix 7.5.

The chip has been returned from Tower at the beginning of October, 2014. Package of the chip is added at the beginning of Jan, 2015 and a PCB board was fabricated to hold the chip on May, 2015. In order to “talk” to the chip, we make a LabView GUI by controlling a NI sbRIO – 9642 board. In that way, we are able to write command data to the chip and read data from the board.

6.2 Conclusions

I have finished the design of the chip and the final layout for it. The post layout simulation result indicates the chip can work properly, including the digital controller part. Test chip shows the maximum working frequency for the SERDES can be more than 4.1Gbps, while for the performance of different links, please refer to the appendix also.

6.2.1 Simulation result

From the post layout simulation, we are almost sure that the chip works. The test circuit also works. With the simulation result we have now, the maximum working frequency for the 2T 2X configuration with 10mm transmission line can work at the data rate of 3.85Gbps. While for the case of supply voltage to be 2.0V, simulation result gives the maximum working speed is 4.07Gbps. While from the test, one voltage link can work at the speed of 4.1Gbps without incurring bit error.

For the case of the RGC “TIA”link, it doesn't work in the whole link post layout simulation with 10mm transmission line and I don't know the reason well.

For the RGC “TIA” with differential amplifier, it works at 1.8V power supply with the maximum speed of 3.42Gbps while for the case with 2.0V power supply, it cannot work without bit errors.

6.2.2 Test result

The chip works and the maximum working speed is around 4.1Gbps for one voltage mode link with 1.74mm line length. For the RGC “TIA” links, only when the supply voltage is more than 1.85V, it begins to work and one link with 0.47mm transmission line can work until the frequency of 2.2Gbps without bit error.

6.3 Updates from FOX1

The FOX1 test chip doesn’t work well. From the test environment, we found that the output bits are not correct because the result it gave was not valid. So we believed we had mistakes in the digital controller of FOX1. After reviewing the process, we found that we forgot the clock tree generation during the digital process. So we need to fix this problem in FOX2.

After getting the layout of the digital controller and the layout of other blocks (transmitter, test circuits and receiver), we need to integrate them together on a single die. In FOX1, we didn’t have that manual, so we did it manually. However, due to the big routing difficulties, it took really long time for that integration. For FOX2, we cannot afford such long time for that global routing. Thus, we need to find an automatic way to do it.

6.3.1 Digital controller backend process

Fig 6.1 shows the digital process I used in FOX2 [13]. In FOX1, we skipped the clock tree generation stage when using encounter. I think because of the timing violation the digital process in FOX1 that makes the circuit unable to work. Moreover, in this FOX2 process, after that LEC (logic equivalent check), I also did the LVS (Layout versus schematic) check in virtuoso to make sure there is no problem in the digital process. Finally, I passed the DRC (Design Rule Check) check and LVS check for the final layout of the digital controller.

6.3.2 Mixed signal integration method

We did the automatic placement and route for the digital controller in Encounter. Thus, we also want to use Encounter to do the automatic placement and route for the top level of the chip. For the digital controller, the cells that Encounter uses are all available at the library while the library doesn’t contain the cells for the top level circuit. Hence we need to make files that contain the cell information and include these files in the library.

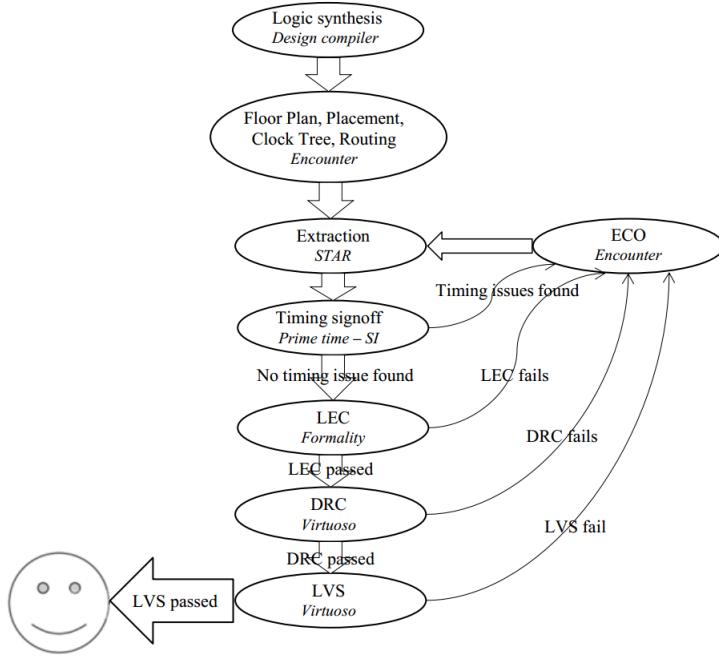


Figure 6.1: Digital controller backend process

Here is the process for the process

1. Make lef files for all the analog blocks
2. Make lef file for the digital controller
3. Make the top.io file based on the chip configuration
4. Make top level schematic for the chip
5. Make an equivalent verilog file for the top level schematic
6. Add the lef files into the library
7. Connect the pads with the inputs and outputs of the top level circuit
8. Encounter stage for the io adding, flow plan and route
9. DRC check (add the external and dummy)
10. LVS check (get cdl file for the top level schematic; get the cdl for the pads, then combine them together)

The overall process is like what we did in the digital controller backend process. But we needn't to do the clock tree generation, extraction, timing check and LEC. During that process, I had some problems for the antenna DRC check because the lef files I generated are not perfect. For the DRC errors, I fixed it manually. For the details of it, please refer to Appendix 7.4.

For the LVS stage, we had the layout. But for the schematic, it is not trial to get since we don't really have the schematic for it because of the digital controller. So we generate the CDL file for the top level schematic and replace the digital controller part by the CDL we used for the LVS check of the digital controller. Finally, we passed the DRC and LVS check (still some DRC errors regarding to dummies, but have no idea how to deal with it).

6.4 Recommendations for future research

6.4.1 Transmission line model

From the time I started the analog transmitter, analog receiver design, I didn't know the method to use HFSS to extract the S-parameter directly. So I used the RLC model for the design and all the performance results of the analog channel are based on that RLC line model. But I think it is not accurate enough. So for those who want to do similar work, it is better to use the HFSS to get the S-parameter directly.

6.4.2 Distributed power supply

In the Virtuoso, we assume the power we provide is ideal, which means for the VDD and GND power line, there is no voltage fluctuation and the power supply can provide any current immediately when there are transitions in the circuit. However, for the actual case, this cannot be true. The power supply is not ideal, it needs time to correspond to the current change. Thus, there will be fluctuations along the power grid line.

One way to deal with this issue is to get the distributed model for the power grid. For example, we have power pads and set that point to be ideal power supply while for other parts, we can get the RC model between this ideal VDD input point to the local VDD point. Since the circuit works at high speed, the fluctuation along the line should have the same frequency as the circuit working frequency. Thus, we should also use HFSS to get this model. But since we have many power I/O pads, I have no clear idea about how to get the exact model for it.

From the extracted view of the layout, actually we can see the netlist for the power grid. It is not ideal metal connection, but with parasitic resistors and capacitors. Perhaps this is the true power grid RC model. Moreover, from the post layout simulation result, I found that for some specific circuit, even when it works at pretty low frequency, the output swing is not from 0 to VDD. If the power grid line is ideal, this couldn't have happened. But still, we can just add one pin at the layout for the chip, while we have many power pins. In this case, the location of the power pin will make a big difference on the performance of a certain part of the circuit. I need to consult more professional person about it.

6.4.3 Use high supply voltage

In general, increasing the supply voltage for the circuit will make the circuit work faster. I made several simulations for the extracted view and found that under the same condition (same control signals, same temperature and same transistor corner), the higher voltage will give faster working speed. But for a certain speed control pattern, increasing the supply voltage cannot guarantee it works although it can work at higher frequency.

6.4.4 Lef file generation

In the FOX2 top level integration, we need to generate the lef file for the transmitter and receiver as well as the test circuits in order to use the encounter to do the placement and route job. The default setting to generate the lef file from the virtuoso doesn't contain the area information for the input pins and output pins. The area information is very important for the route and it is very easy for us to make antenna errors without it. I suffered this problem, but since I had some manually fixing part on the layout, I cannot do it again from the beginning. It's annoying to fix these antenna errors by hand. However, I heard from Goel there is a way to get lef file with the area information, consult him for further information.

6.4.5 De-cap for fast circuit

For the circuit that works at high speed, the power supply is not fast enough to supply the current for the high speed circuit. In contrast, the local capacitors will supply the current during the fast transition. So it makes sense to add local capacitors near the fast circuit.

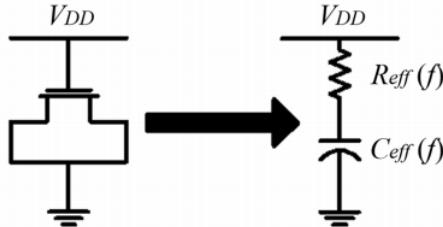


Figure 6.2: NMOS De-cap capacitor in FOX2

This is the local capacitor I choose for FOX2, here I just use NMOS for performance consideration. The length of the NMOS transistor I use for the 180nm technology is 2um [33].

Chapter 7

Appendices

7.1 FOX2 Digital controller

7.1.1 DIN bits format

#	Bits	Width	Name	Description
1	[15:0]	16	UD_WORD	User-defined word to send
2	[19:16]	4	CGC_FINE	CGC fine control bits 2^4
3	[26:20]	7	CGC_SLOW	CGC slow control bits 2^7
4	[31:27]	5	LINK_SEL	link selection signal
5	[61:32]	30	NUM_OF_TEST	Number of test to be performed. First word is the user defined word and when it is greater than 1, other words are from the internal random generator, while the initial seed is defined by SEED parameter
6	[77:62]	16	SEED	Input seed for random word genera- tion
7	[78]	1	Ring_OS_EN	Test enable signal for Ring Oscilla- tion Circuit
8	[88:79]	10	Time_OUT	wait time for latch enable signal
9	[96:89]	8	WT_RST_TIME	Wait reset time: idle time between compare state and the next transi- tion state

Table 7.1: Digital controller input bits format

7.1.2 DOUT bits format

#	Bits	Width	Name	Description
1	[15:0]	16	RS_WORD	If latch enable is high, equals to the latest received word. If not, equals to 0
2	[45:16]	30	BER	Number of single bit errors
3	[75:46]	30	FER	Number of word errors
4	[105:76]	30	NTO	Number of time out, time out means for one transition, latch enable signal is not high
5	[119:106]	14	FSM	Finite state machine one hot code 0: idle state 1: load data state 2: set parameter state 3: wait start state 4: test reset state 5: test wait 1 state 6: test load data stat 7: test wait 2 state 8: test send data state 9: test wait latch enable state 10: test compare state 11: wait before next run state 12: set out SR state 13: set out wait state
6	[149:120]	30	CT_NUM	Current Test number -1 (number of words sent so far)
7	[160:150]	11	Speed	{CGC_SLOW, CGC_FINE} (verify the functionality of state machine)

Table 7.2: Digital controller output bits format

7.1.3 FSM

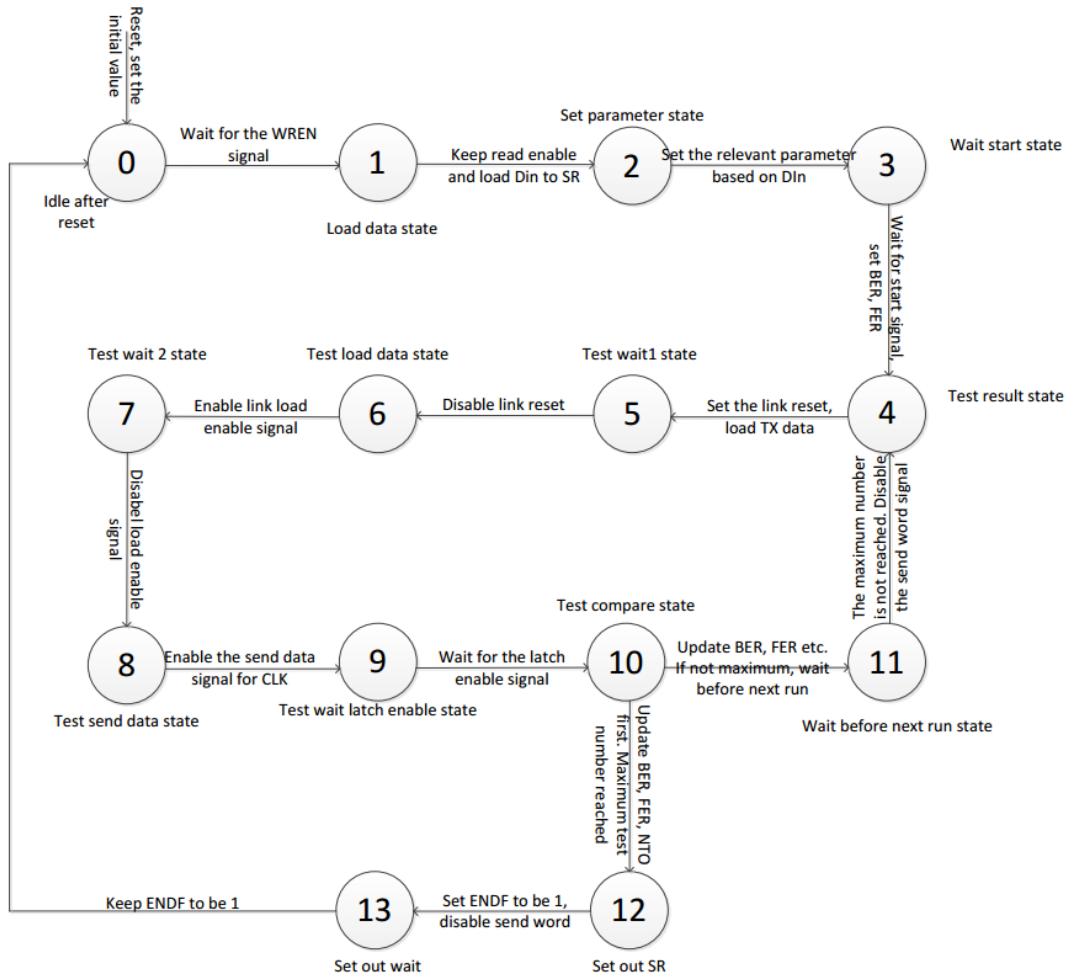


Figure 7.1: FOX2 finite state machine

7.2 Backend flow of the digital controller

7.2.1 Introduction

In this part, we took behavioral Verilog code (RTL) and created data base in GDS format which will be sent to Tower FAB for fabrication [13].

7.2.2 Flow

For the main stages of the digital process, we have a brief summary:

1. Logic Synthesis
2. Floor Plan
3. Placement
4. Clock Tree Synthesis
5. Routing
6. Extraction
7. Timing signoff with PrimeTime -- SI
8. ECO flow with encounter
9. Logic Equivalent Check (LEC)
10. DRC & LVS check

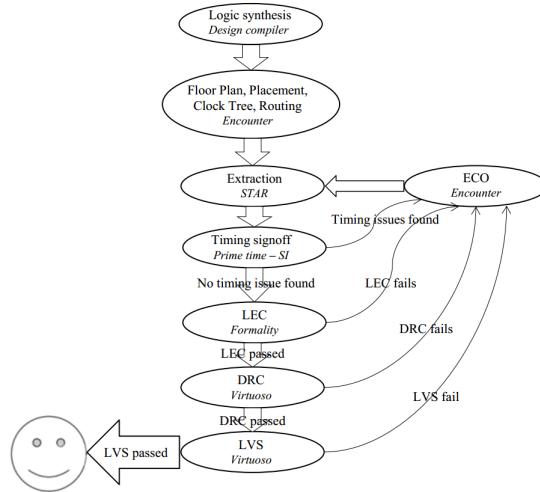


Figure 7.2: Digital backend flow

For the details of 1 – 9, please refer to the manual of digital backend flow [13]. While for part 10 – the DRC & LVS check, please refer to the manual [37].

7.3 Integration of TX & RX and add de-cap

7.3.1 Introduction

In Chapter 2, we went over the schematics and functional units of the individual part for the TX and RX respectively. But in the integration part, the TX or RX alone is like a module and together they form a link. In this part, I will illustrate how the schematics of TX (analog TX is just one part) and RX look like and how the layouts look like. Moreover, I mentioned the de-cap capacitors before. So in this part, we can see what are the layouts of TX and RX look like when NMOS de-cap is added.

7.3.2 TX Integration

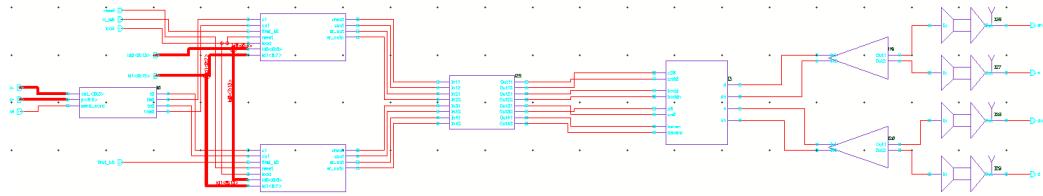


Figure 7.3: Schematic of TX in one link

This is the schematic of one TX in a link, from left to right: CGC, 2 SRs, Inverter chains between SRs and LEDR, LEDR, Inverter chains between LEDR and analog TX, analog TX. The Inverter chains here are used to recover the signal for the drive strength consideration.

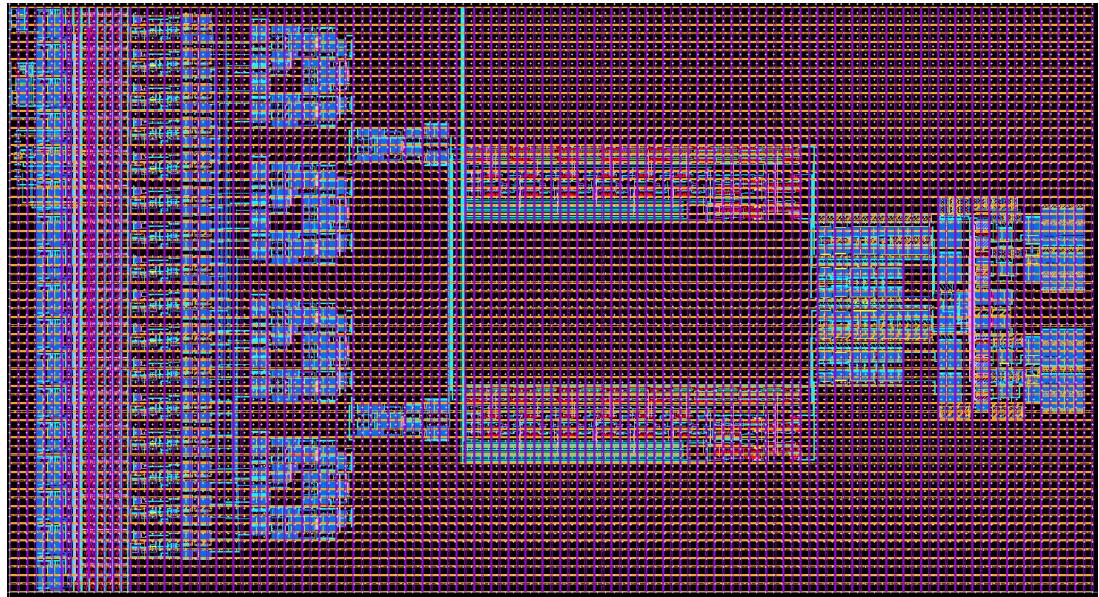


Figure 7.4: Layout of TX without de-cap

This is the layout without de-cap and we can see that at the first glance, it is not very

compact. I also feel not very nice since lots of area is not properly utilized. However, since the de-cap will help to improve the performance, some empty part can help the performance.

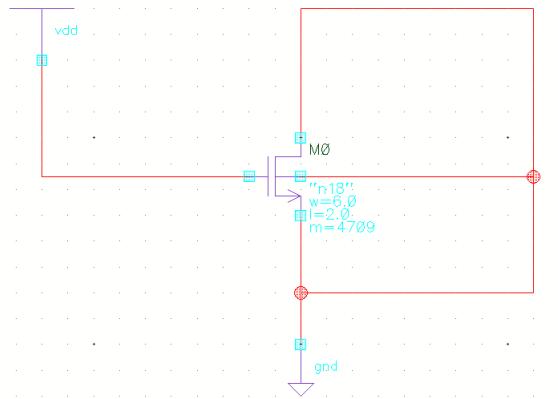


Figure 7.5: Schematic of NMOS de-cap

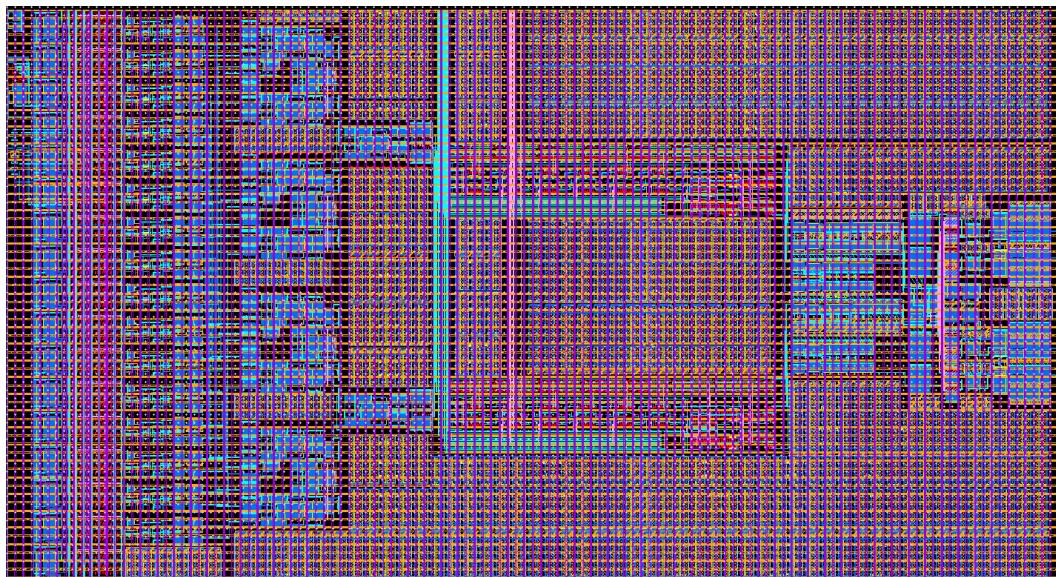


Figure 7.6: Layout of TX with de-cap

We can see that when the de-cap is added, the layout looks very different and becomes compact now.

7.3.3 RX Integration

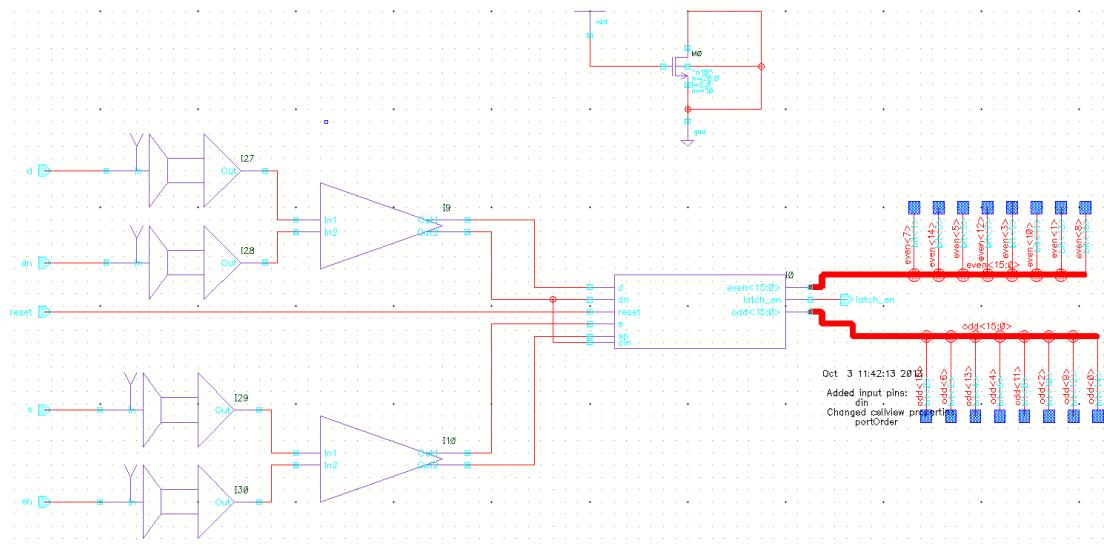


Figure 7.7: Schematic of RX

This is the schematic of the RX in one link, from left to right is the analog RX circuit, the inverter chains and the block at the right side is the integration of Splitter circuit and SRs.

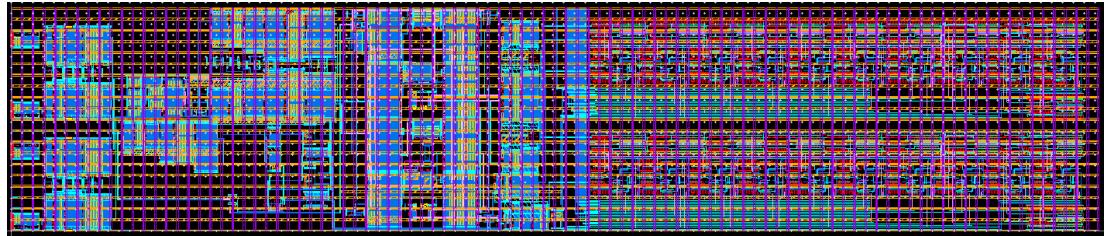


Figure 7.8: Layout of RX without de-cap

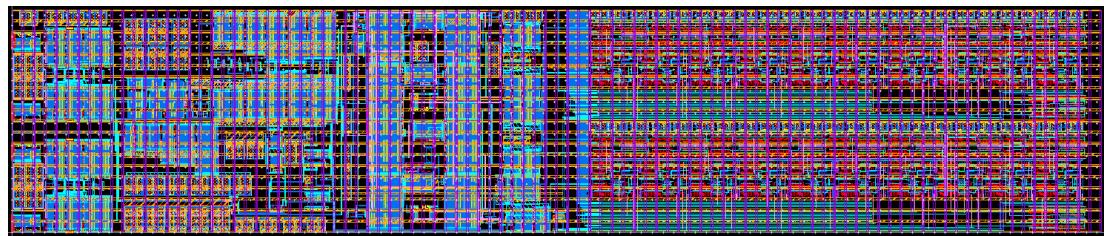


Figure 7.9: Layout of RX with de-cap

Compared with TX, we can see that RX is more compact. One reason for that is at the RX side, there is no CGC, which takes lots of area in the TX. But anyway, I managed to add some de-cap for RX also.

7.4 Analog Mixed Signal Integration

7.4.1 Introduction

Analog Mixed Signal (AMS) integration is the stage after getting the layout of TX, RX, digital controller and test circuits and combining all these blocks together to form the layout of one chip. In FOX1, we made a manual route for the layout which took more than half year.

7.4.2 Flow

This is a little detailed summary of the mixed signal integration, for the details of the flow, please refer to the manual [38].

1. Make lef file for the analog blocks
 - Open the analog block layout.
 - Generate abstract file.
 - Export lef file.
 - Edit the lef file.
2. Make lef file for the digital controller
3. Make the top.io file
4. Make the top level schematic of the chip
5. Make an equivalent verilog file for the chip
6. Connect the pads with IN/OUT of the chip
7. Encounter stage
 - Load file.
 - Add iofill and run in the script.
 - Power line, placement and route.
8. DRC check
 - Dummy fill (if needed).
 - Run DRC.
9. LVS check
 - Get cdl file.
 - Make changes to cdl file.
 - Get cdl file for pads.
 - Run LVS.

7.5 Package and pads

The chip has 48 pins. However, for some reasons of the fabrication stage, the 48 pin package is not available for us. So I choose a 80 pins package instead. Of course, 32 pins for this package is NC (No connection). The pins are arranged in the following way.

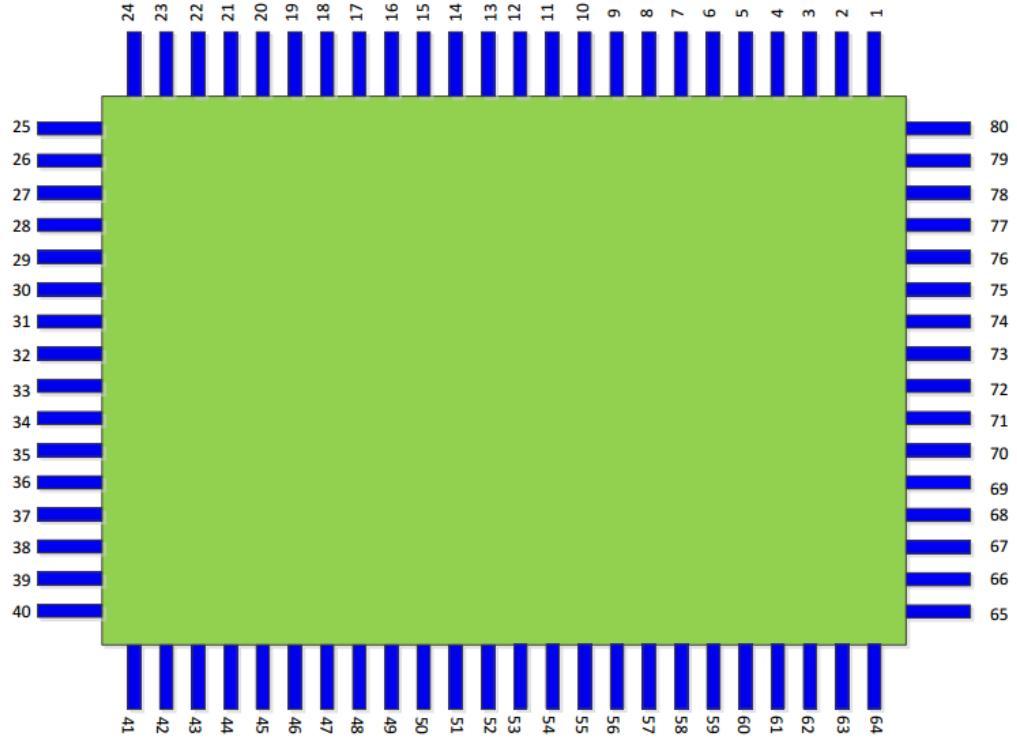


Figure 7.10: Chip package

The 1st pin is at the right top of the chip and the direction of the pins is anti-clockwise. The property of the package is summarized in this table:

Type	MQFP
Data source	http://goo.gl/QPCF9t
Name	MQFP80B
Supplier	I2A
Pin number	80
Size (W×L×H) (mm)	14×20×2.7
FP (mm)	3.2
Cavity (mm)	6.6×6.6
Bonding Diagram	DRBS-QFP1420-002
Marketing Outline	DRMO-QFP1420-001

Table 7.3: Package information

For the functionality and property of the chip, I make a table for it.

Pin/Package No.	Name	Property
1	NC	NC
2	NC	NC
3	GND	PWR
4	NC	NC
5	VDD	PWR
6	NC	NC
7	GND	PWR
8	NC	NC
9	VDD	PWR
10	NC	NC
11	GND	PWR
12	START	IN
13	DIN	IN
14	WREN	IN
15	NC	NC
16	VDD	PWR
17	NC	NC
18	CLK	IN
19	NC	NC
20	VDD	PWR
21	NC	NC
22	RESET	IN
23	NC	NC
24	NC	NC
25	NC	NC
26	NC	NC
27	ENDF	OUT
28	GND	PWR
29	CLKOS	OUT
30	GND	PWR
31	DOUT	OUT
32	RDEN	IN
33	VDD	PWR
34	VDD	PWR
35	GND	PWR
36	VDD	PWR
37	GND	PWR

Pin/Package No.	Name	Property
38	VDD	PWR
39	NC	NC
40	NC	NC
41	NC	NC
42	NC	NC
43	GND	PWR
44	NC	NC
45	A2T_OUT	OUT
46	NC	NC
47	CGC_OUT	OUT
48	NC	NC
49	VDD	PWR
50	NC	NC
51	RGC_TIA_OUT	OUT
52	SLOW_CGC_IN	IN
53	VDD	PWR
54	GND	PWR
55	NC	NC
56	VDD	PWR
57	NC	NC
58	GND	PWR
59	NC	NC
60	VDD	PWR
61	NC	NC
62	GND	PWR
63	NC	NC
64	NC	NC
65	NC	NC
66	NC	NC
67	VDD	PWR
68	GND	PWR
69	VDD	PWR
70	GND	PWR
71	VDD	PWR
72	VDD	PWR
73	VDD	PWR
74	GND	PWR
75	VDD	PWR

Pin/Package No.	Name	Property
76	2T&RGCTIA_IN	OUT
77	VDD	PWR
78	GND	PWR
79	NC	NC
80	NC	NC

Table 7.4: FOX2 pins

Here IN means input pins, OUT means output pins while PWR means power pins and NC means no connection. We can see that 14 out of the 48 pins are not Power pins while 8 pins are for the digital controller and 6 pins are for the test circuit.

For the Bond Diagram, it looks like that:

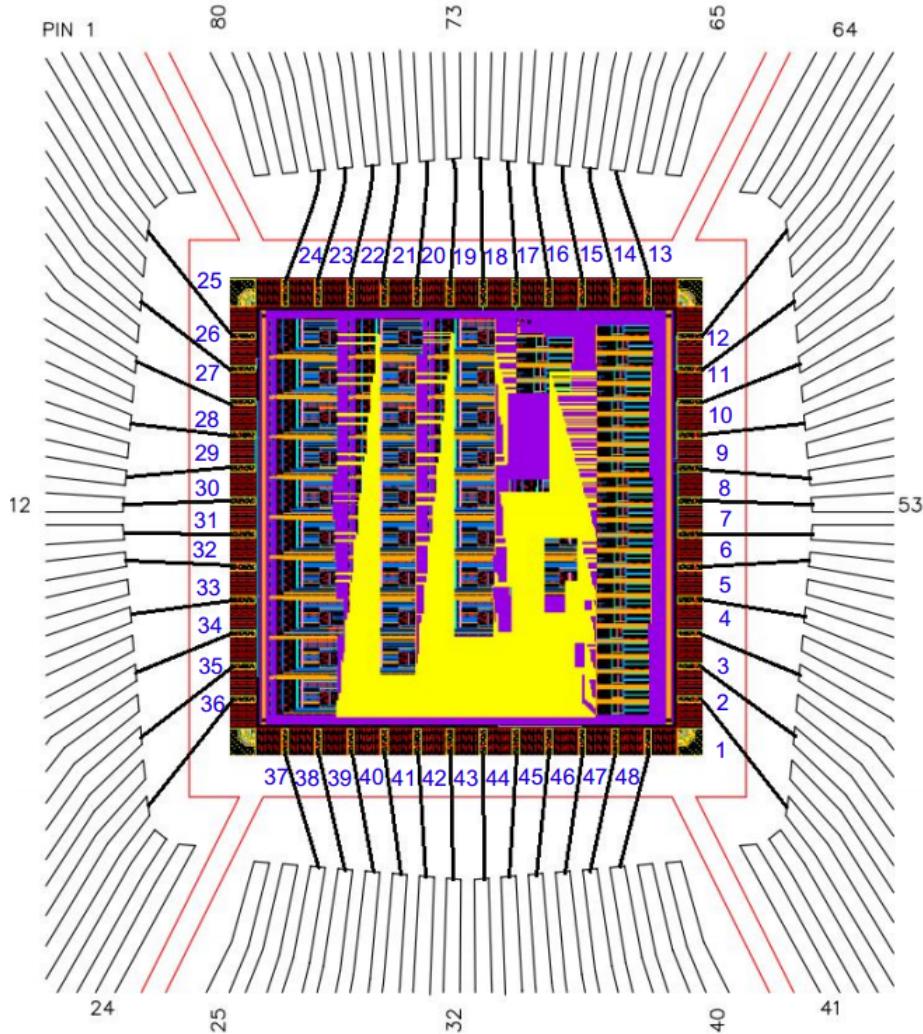


Figure 7.11: Package board diagram



Figure 7.12: FOX2 package

This is what the chip looks like in final after it has been fabricated.

7.6 FOX2 Test Environment

7.6.1 High Level Testing Architecture

The test system consists of a PC, an interface board made by National Instruments (NI), and an analog PCB board that holds the FOX2 chips.

The PC is connected to the NI sbRIO9642 board, which includes a Spartan 3 Xilinx FPGA, via an Ethernet cable. We design a VI to control the analog board in the LabView, which is running on the PC. The VI includes a digital netlist that generates the control signal for the FOX2 chip on the analog board and reads the signals from the chip back to the VI panel.

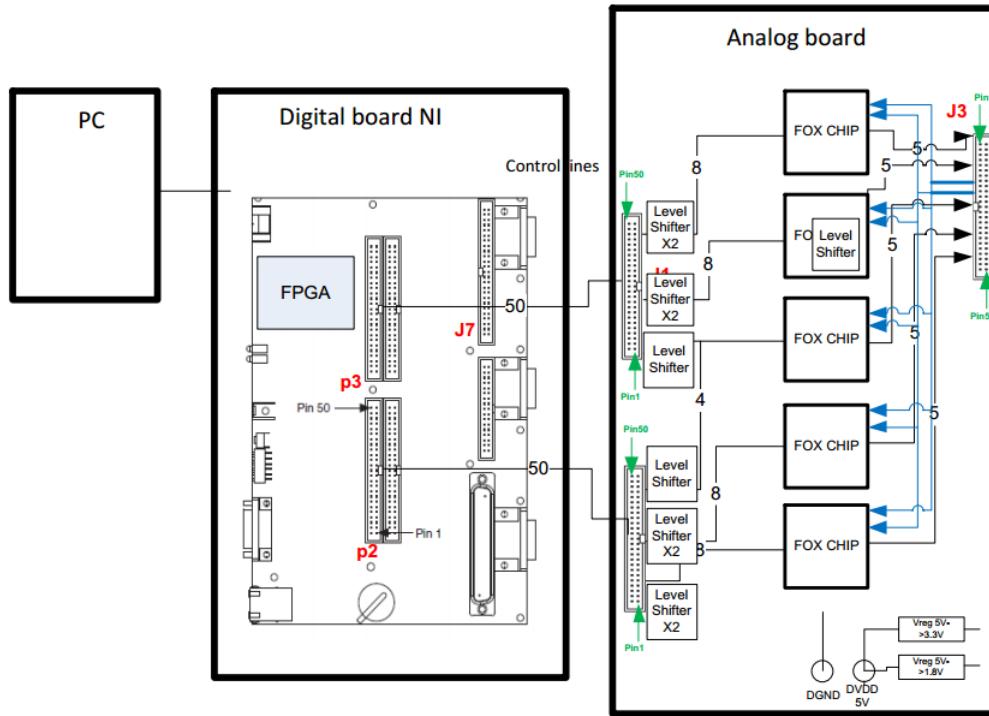


Figure 7.13: High Level Testing Architecture

The FOX2 PCB board is custom designed, which contains 5 FOX2 chips and various interface circuits and indicator LED lamps. All signals from the NI board to the FOX2 PCB board are slow digital signals operating at the frequency around 5MHz. All fast signals are generated within the FOX2 board, but there is no fast I/O interface between the chip and the FOX2 PCB board. We also add some Test points in the board to test some slow signals through oscilloscope.

In the following of this part, we will describe the details of PC interface, the FOX2 PCB board and the steps to use the system to test the functionality and performance of the chip.

7.6.2 PC Interface – Digital LabView Control GUI and Interface

The Spartan 3 Xilinx FPGA manages the communication between the GUI (LabView GUI in the PC) and the FOX2 chips in the FOX2 Test board.

The Finite State Machine (FSM) of the module is in the following. The write operation that controls the writing data to the FOX2 chip is triggered by GUI. While the read operation can be controlled from GUI or by the running stage (When ENDF = 1, means the FOX2 chip has finished operation).

Here is the state diagram of the FSM for the Xilinx FPGA.

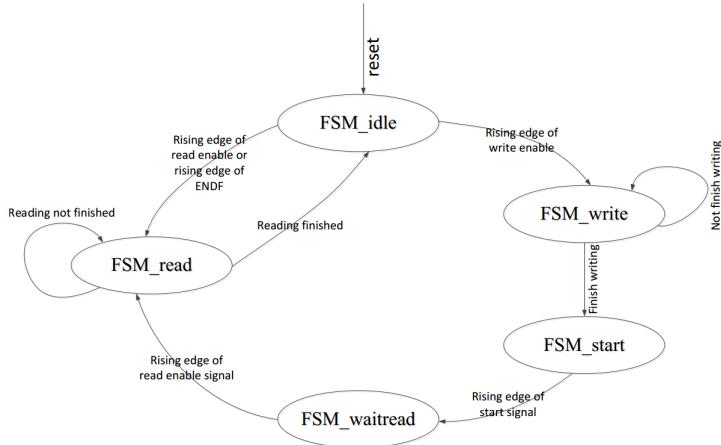


Figure 7.14: FPGA state diagram

Using the LabView GUI to start the operation as described in the diagram.

1. **System reset** – select the wanted chip number, set the relevant shift register data and push the Reset/PC_reset Button. The relevant reset chip led will be turned on (in the GUI).
2. **Verify the GUI** – Turn off the Reset button. Push the Test Mode button to make sure the GUI can work properly. The Test Mode is just show the data in the input register at the output register in the GUI.
3. **Verify connection** – Push the Read button to make sure the connection with the chip is right. The output fields in the output register should be always 0 or 1 depends on the input data.
4. **Write operation** – Press the Write button to create the data stream to the target chip. After the operation finished, the Write data ready signal will be high on the GUI.
5. **Start transmission** – Press the Start button on the GUI to enable the start signal for the chip. The start signal will be high for the target link.

6. **Read operation** – Since the transmission is always fast compared with the time we have, we should press the Read enable signal manually in order to read the bits. After the read stage is finished, the Rdata ready signal will also be high on the GUI. The output register will us the result when the Data ready indicator is on.

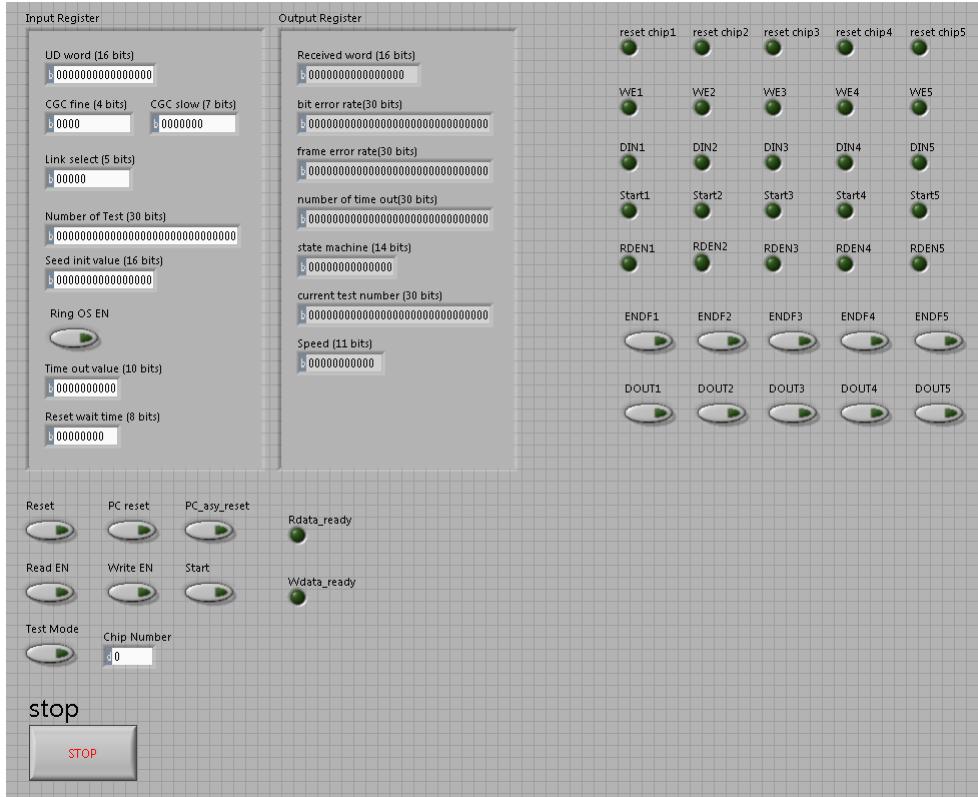


Figure 7.15: Digital Control GUI in LabVIEW – Front Panel

In the following Table, we will see the digital signals connection from FPGA I/Os in the NI board through the connectors on the analog FOX2 PCB board and into the chip pins. This table can help to test and debug the FOX2 PCB board. We can follow a signal path from the input connector on the FOX2 PCB board and to the chip pin to verify the signal connectivity.

Signal	FPGA pin #	Digital I/O	FOX2 board	(chip #, pin #)
DOUT1	Port9/DIO7	P3 47	J1 47	(1, 31)
RDEN1	Port9/DIO6	P3 45	J1 45	(1, 32)
ENDF1	Port9/DIO5	P3 43	J1 43	(1, 27)
START1	Port9/DIO4	P3 41	J1 41	(1, 12)
DIN1	Port9/DIO3	P3 39	J1 39	(1, 13)
WREN1	Port9/DIO2	P3 37	J1 37	(1, 14)

Signal	FPGA pin #	Digital I/O	FOX2 board	(chip #, pin #)
RESET1	Port9/DIO1	P3 35	J1 35	(1, 22)
CLK1	Port9/DIO0	P3 33	J1 33	(1, 18)
DOUT2	Port8/DIO7	P3 27	J1 27	(2, 31)
RDEN2	Port8/DIO6	P3 25	J1 25	(2, 32)
ENDF2	Port8/DIO5	P3 23	J1 23	(2, 27)
START2	Port8/DIO4	P3 21	J1 21	(2, 12)
DIN2	Port8/DIO3	P3 19	J1 19	(2, 13)
WREN2	Port8/DIO2	P3 17	J1 17	(2, 14)
RESET2	Port8/DIO1	P3 15	J1 15	(2, 22)
CLK2	Port9/DIO0	P3 33	J1 33	(2, 18)
DOUT3	Port7/DIO7	P3 7	J1 7	(3, 31)
RDEN3	Port7/DIO6	P3 5	J1 5	(3, 32)
ENDF3	Port7/DIO5	P3 3	J1 3	(3, 27)
START3	Port2/DIO8	P2 49	J2 49	(3, 12)
DIN3	Port2/DIO7	P2 47	J2 47	(3, 13)
WREN3	Port2/DIO6	P2 45	J2 45	(3, 14)
RESET3	Port2/DIO5	P2 43	J2 43	(3, 22)
CLK3	Port9/DIO0	P3 33	J1 33	(3, 18)
DOUT4	Port6/DIO8	P2 39	J2 39	(4, 31)
RDEN4	Port6/DIO7	P2 37	J2 37	(4, 32)
ENDF4	Port6/DIO6	P2 35	J2 35	(4, 27)
START4	Port6/DIO5	P2 33	J2 33	(4, 12)
DIN4	Port6/DIO4	P2 31	J2 31	(4, 13)
WREN4	Port6/DIO3	P2 29	J2 29	(4, 14)
RESET4	Port6/DIO2	P2 27	J2 27	(4, 22)
CLK4	Port9/DIO0	P3 33	J1 33	(4, 18)
DOUT5	Port6/DIO8	P2 17	J2 17	(5, 31)
RDEN5	Port6/DIO7	P2 15	J2 15	(5, 32)
ENDF5	Port6/DIO6	P2 13	J2 13	(5, 27)
START5	Port6/DIO5	P2 11	J2 11	(5, 12)
DIN5	Port6/DIO4	P2 9	J2 9	(5, 13)
WREN5	Port6/DIO3	P2 7	J2 7	(5, 14)
RESET5	Port6/DIO2	P2 5	J2 5	(5, 22)
CLK5	Port9/DIO0	P3 33	J1 33	(5, 18)

Table 7.5: Digital Test board Interface

We can get the backend panel of the LabView GUI based on this table.

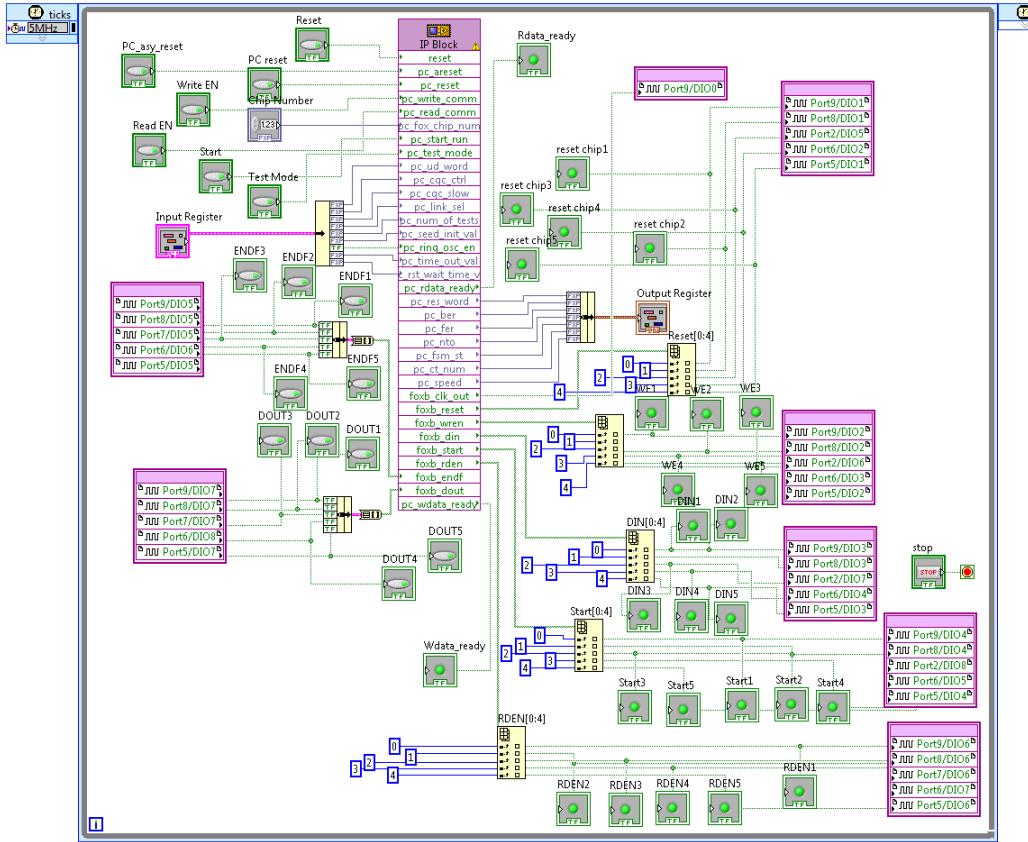


Figure 7.16: Digital Control GUI in LabView – Backend Panel

7.6.3 PC Interface – Analog LabView Control GUI and Interface

From the section of the Test Circuit, we know that all the test circuits are independent of the digital controller and all the input signals and the output signals have chip pin location to it. Thus, it's OK to control/test the test circuit directly.

Test circuits are analog signals and the SCTL (Single Cycle Timed Loop) method is no necessary for it since no external clock signal is needed. We have 6 kinds of analog signals as shown in the previous section:

1. Speed measurement circuit – output signal (CLKOS)
2. CGC test circuit – input and output signals
3. RGC TIA circuit – input and output signals
4. 2T 2X circuit – input and output signals

Note that since the input to the RGC TIA circuit and the 2T 2X circuit can be the same, I just shorted it in the FOX2 PCB board. But later, I found if the frequency for the input clock is slow enough, we are able to use the same test signal for CGC also. Thus, in the LabView front panel, I also shorted the input signal for the CGC test

circuit. Thus, in this situation, only one input signal is needed but for the outputs, we need to allocate FPGA I/O for each one. For convenience, I generate the input test signal through the FPGA for the test.

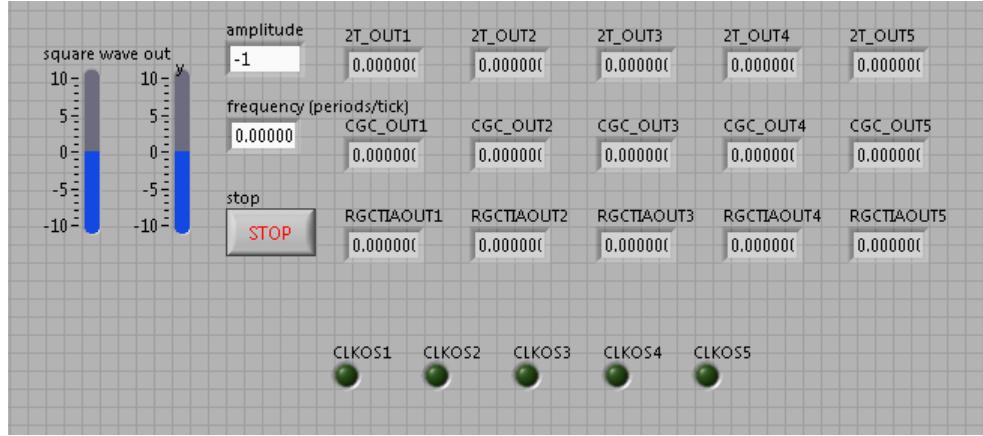


Figure 7.17: Analog Control GUI in LabView – Front Panel

The following table shows the analog signals' connections from FPGA I/Os in the NI board through the connectors on the analog FOX2 PCB board and into the FOX2 chip pins. Since this table is about the analog signals for the test circuits. I have added Test Points in the board for these signals so that we can detect the signal from the oscilloscope. This part is mainly used for debug and verify the FOX2 PCB board and the fabrication process of the chip (for example, if we cannot get the correct RGC TIA output signal with certain input signals, we cannot expect the digital part will work).

Signal	FPGA pin #	D/A I/O	FOX2 board	(chip #, pin #)
SLOW_CGCIIN	AO0	J7 49	J3 49	(1/2/3/4/5, 52)
2T&RGCTIAIN	AO1	J7 47	J3 47	(1/2/3/4/5, 76)
CLKOS1	Port9/DIO8	P3 49	J1 49	(1, 29)
2T_OUT1	AI0	J7 2	J3 2	(1, 41)
CGC_OUT1	AI8	J7 3	J3 3	(1, 42)
RGC_TIAOUT1	AI9	J7 4	J3 4	(1, 46)
CLKOS2	Port8/DIO8	P3 29	J1 29	(2, 29)
2T_OUT2	AI10	J7 8	J3 8	(2, 41)
CGC_OUT2	AI11	J7 9	J3 9	(2, 42)
RGC_TIAOUT2	AI3	J7 10	J3 10	(2, 46)
CLKOS3	Port7/DIO8	P3 9	J1 9	(3, 29)
2T_OUT3	AI5	J7 15	J3 15	(3, 41)
CGC_OUT3	AI6	J7 17	J3 17	(3, 42)
RGC_TIAOUT3	AI14	J7 18	J3 18	(3, 46)
CLKOS4	Port2/DIO4	P2 41	J2 41	(4, 29)

Signal	FPGA pin #	D/A I/O	FOX2 board	(chip #, pin #)
2T_OUT4	AI16	J7 22	J3 22	(4, 41)
CGC_OUT4	AI24	J7 23	J3 23	(4, 42)
RGC_TIAOUT4	AI25	J7 24	J3 24	(4, 46)
CLKOS5	Port5/DIO8	P2 19	J2 19	(5, 29)
2T_OUT5	AI26	J7 28	J3 28	(5, 41)
CGC_OUT5	AI27	J7 29	J3 29	(5, 42)
RGC_TIAOUT5	AI19	J7 30	J3 30	(5, 46)

Table 7.6: Analog Test board Interface

With this information, we can get the Backend Panel of the Analog Control GUI in LabView:

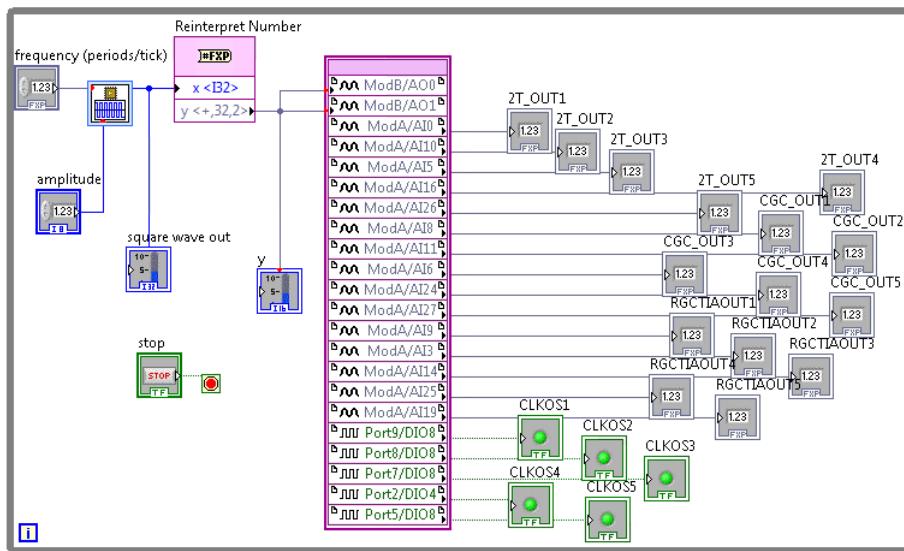


Figure 7.18: Analog Control GUI in LabView – Backend Panel

7.6.4 FOX2 PCB board schematic

We make this FOX2 PCB board in order to make the chip work on this board so that we can test it. We put 5 chips on each board and the board will also provide the DC voltage for the FOX2 chip and other components in the board. The supply voltage for the chip is 1.8V. But in general, we are required to generate $\pm 10\%$ around the standard supply voltage, so I add a potentiometer in the voltage regulator part.

FOX2 TEST CHIP BOARD LAYOUT

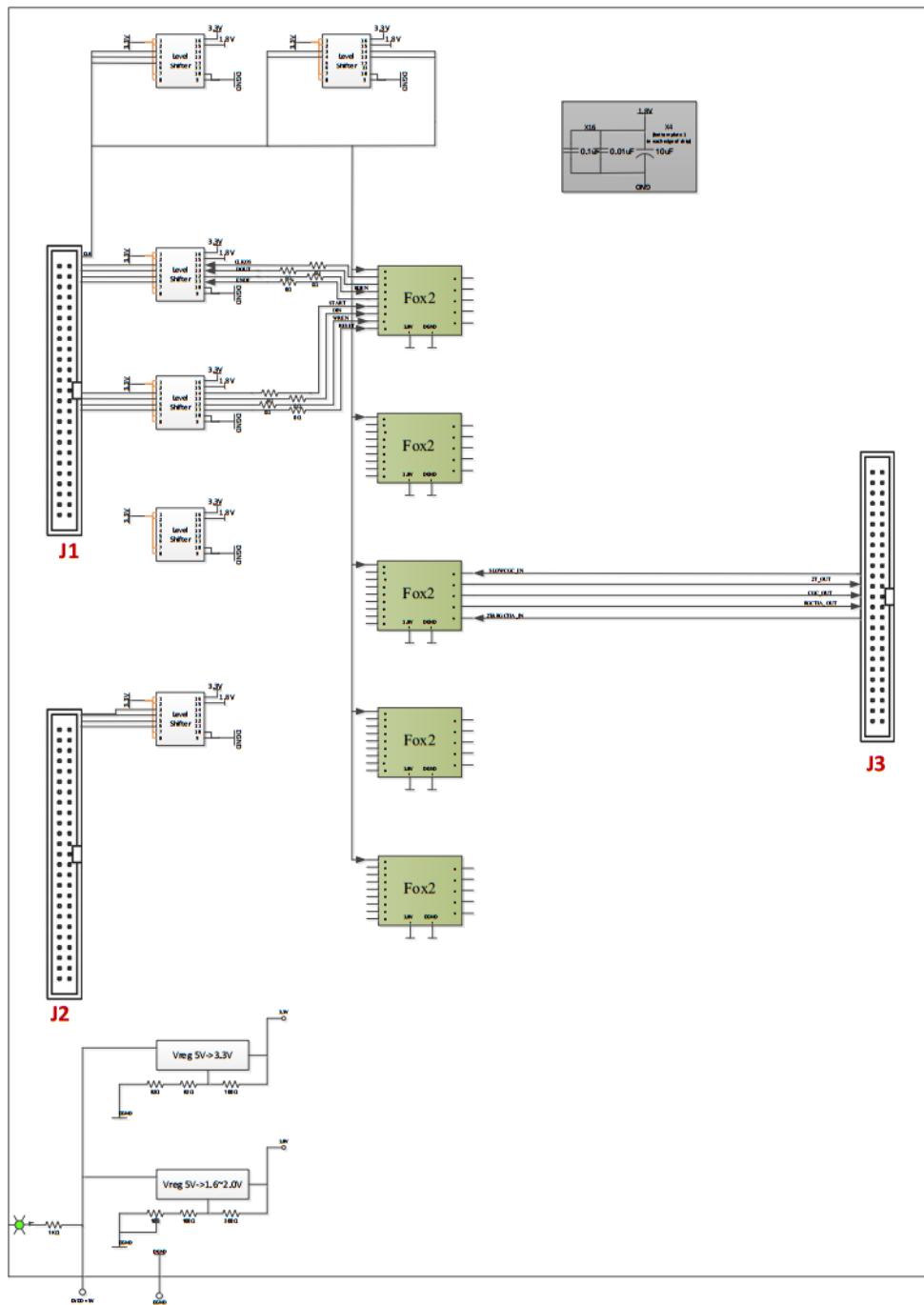


Figure 7.19: FOX2 PCB board layout

7.6.5 Board Testing Procedure

The following procedure is set for testing a board with 5 chips. It includes visual and electrical circuitry verification, steps for power-up the PCB board connections, clock signal testing, test circuit verification, speed measurement and serial-link testing:

1. Check visual connections (power connections, all digital-analog board connectors and all boards components)

(a) Check for wanted and unwanted shorts between the following points

- DGND, DVDD
- Each Chip power supply
- Connectors signals and power lines

(b) Connect power sources to board and verify voltages at test points of 1.8V (1.6V ~ 2.0V) and 3.3V.

2. Check the signals for the test circuit

(a) Set the analog GUI in LabView and using the oscilloscope to measure the relevant output signals

- Set the CGC test circuit output signal with input slow clock signal
- Check the RGC TIA circuit output with input slow clock signal
- Check the 2T OUT circuit output with input slow clock signal

3. Test the serial link

(a) Connect digital connectors and verify link between boards

(b) Perform Test mode

- Reset the link and set initial word in the digital GUI for the test
- Press the Test Mode button and to make sure the output bits are the same as the input relevant bits

(c) Speed measurement

- Reset the chip and reset the relevant link
- Set the OSEN part on the GUI to be 1
- Press the Write button, when the write operation is finished, we will be able to get the CLKOS signal
- Measure the output Clock frequency using oscilloscope

(d) Send and Receive data

- Set the Input register at the front panel of the GUI
- Reset the chip
- Reset the relevant link
- Begin the Write and when it finishes, we should have
 $\text{WriteReady} = 1$
- Press the Start button to initialize the send work process
- We can receive the $\text{ENDF} = 1$ signal when it finishes to start the read data stage or we manually press the read button to start the read process
- Check the BER/FER/NTO information to measure the performance
- Compare the relevant section in the Output Register to make sure the chip work well

(e) Summarize the result in the following form

Link #	Length [mm]	Voltage [V]	Speed [Gbps]	BER	FER	NTO
0	2.61	1.804	4.44	8.76×10^{-6}	1.40×10^{-4}	1.28×10^{-4}
1	3.06	1.802	4.44	2.0×10^{-4}	2.1×10^{-4}	2.0×10^{-4}
2	3.46	1.802	—	—	—	—
3	3.98	1.802	4.26	1.2×10^{-3}	7.7×10^{-3}	5.3×10^{-5}
4	4.42	1.802	3.70	1.9×10^{-4}	2.0×10^{-4}	1.8×10^{-4}
5	4.87	1.802	—	—	—	—
6	5.28	2.035	2.42	0.17	0.99	6.5×10^{-5}
7	5.72	2.035	2.42	0.24	1.0	2.6×10^{-5}
8	6.16	2.035	2.42	0.23	1.0	2.6×10^{-5}
9	6.61	2.035	1.39	0.23	1.0	8.6×10^{-6}
10	2.88	2.035	1.64	0.23	1.0	0
11	3.34	2.035	1.14	0.19	0.96	0
12	3.78	2.035	0.30	0.38	1.0	3.9×10^{-2}
13	4.22	2.035	2.72	0.33	1.0	0
14	4.66	2.035	2.53	0.32	1.0	2.1×10^{-2}
15	5.10	2.035	2.52	0.30	1.0	0
16	5.56	1.802	4.18	4.2×10^{-4}	4.3×10^{-4}	4.1×10^{-4}
17	6.06	1.802	4.14	2.8×10^{-4}	2.9×10^{-4}	2.8×10^{-4}
18	6.38	1.802	—	—	—	—
19	3.11	1.802	4.18	5.2×10^{-4}	5.3×10^{-4}	5.2×10^{-4}
20	2.74	1.802	4.01	5.1×10^{-4}	5.2×10^{-4}	5.1×10^{-4}
21	2.39	2.035	—	—	—	—

Link #	Length [mm]	Voltage [V]	Speed [Gbps]	BER	FER	NTO
22	2.16	2.035	0.30	1.0	1.0	1.0
23	0.90	2.035	2.0	0.19	0.96	0
24	0.67	2.035	3.0	0.30	1.0	5.8×10^{-2}
25	0.42	2.035	2.2	0	0	0
26	0.23	2.035	1.6	0	0	0
27	2.26	2.035	1.64	0.19	0.97	0
28	2.03	2.035	3.1	0.14	0.92	0
29	1.74	1.802	4.1	0	0	0

Table 7.7: Serial link performance table

Bibliography

- [1] Frequency Divider method. http://en.wikipedia.org/wiki/Frequency_divider.
- [2] International Technology Roadmap for Semiconductors 2005 [Online]. www.itrs.net/Links/2005itrs/Home2005.htm.
- [3] Rizwan Bashirullah, Wentai Liu, and Ralph K Cavin. Current-mode signaling in deep submicrometer global interconnects. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 11(3):406–417, 2003.
- [4] Luca Benini and Giovanni De Micheli. Networks on chips: a new soc paradigm. *Computer*, 35(1):70–78, 2002.
- [5] Wayne P Burleson, Maciej Ciesielski, Fabian Klass, and Wentai Liu. Wave-pipelining: a tutorial and research survey. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 6(3):464–474, 1998.
- [6] Wei-Zen Chen, Ying-Lien Cheng, and Da-Shin Lin. A 1.8-v 10-gb/s fully integrated cmos optical receiver analog front-end. *Solid-State Circuits, IEEE Journal of*, 40(6):1388–1396, 2005.
- [7] Barry M Cook. Ieee 1355 data-strobe links: Atm speed at rs232 cost. *Microprocessors and Microsystems*, 21(7):421–428, 1998.
- [8] Mark E Dean, Ted E Williams, and David L Dill. Efficient self-timing with level-encoded 2-phase dual-rail (ledr). In *Proceedings of the 1991 University of California/Santa Cruz conference on Advanced research in VLSI*, pages 55–70. MIT Press, 1991.
- [9] R Dobkin, Ran Ginosar, and Avinoam Kolodny. Fast asynchronous shift register for bit-serial communication. In *Asynchronous Circuits and Systems, 2006. 12th IEEE International Symposium on*, pages 10–pp. IEEE, 2006.
- [10] R Dobkin, Yevgeny Perelman, Tuvia Liran, Ran Ginosar, and Avinoam Kolodny. High rate wave-pipelined asynchronous on-chip bit-serial data link. In *Asynchronous Circuits and Systems, 2007. ASYNC 2007. 13th IEEE International Symposium on*, pages 3–14. IEEE, 2007.

- [11] Rostislav Dobkin, Michael Moyal, Avinoam Kolodny, and Ran Ginosar. Asynchronous current mode serial communication. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 18(7):1107–1117, 2010.
- [12] Rostislav Reuven Dobkin, Arkadiy Morgenshtein, Avinoam Kolodny, and Ran Ginosar. Parallel vs. serial on-chip communication. In *Proceedings of the 2008 international workshop on System level interconnect prediction*, pages 43–50. ACM, 2008.
- [13] Dmitry Ischenko. Tsp backend flow, Technion Project report in VLSI Lab, 2011.
- [14] MJE Lee. *An Efficient I/O and Clock Recovery for TERABIT Integrated Circuits Design*. PhD thesis, PhD Thesis, Stanford Univ, 2001.
- [15] Se-Joong Lee, Kwanho Kim, Hyejung Kim, Namjun Cho, and Hoi-Jun Yoo. Adaptive network-on-chip with wave-front train serialization scheme. In *VLSI Circuits, 2005. Digest of Technical Papers. 2005 Symposium on*, pages 104–107. IEEE, 2005.
- [16] Thomas H Lee. *The design of CMOS radio-frequency integrated circuits, chapter 6*. Cambridge university press, 2004.
- [17] Atul Maheshwari and Wayne Burleson. Current sensing techniques for global interconnects in very deep submicron (vds) cmos. In *VLSI, 2001. Proceedings. IEEE Computer Society Workshop on*, pages 66–70. IEEE, 2001.
- [18] Danniell Nahmann. High-speed, current mode, serial link communication, Technion MsC. dissertation, 2013.
- [19] Ethiopia Nigussie, Juha Plosila, and Jouni Isoaho. Current mode on-chip interconnect using level-encoded two-phase dual-rail encoding. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 649–652. IEEE, 2007.
- [20] Naoya Onizawa, Atsushi Matsumoto, and Takahiro Hanyu. Long-range asynchronous on-chip link based on multiple-valued single-track signaling. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, 95(6):1018–1029, 2012.
- [21] Samuel Palermo. *Design of high-speed optical interconnect transceivers*. PhD thesis, Stanford University, 2007.
- [22] S-M Park and C Toumazou. Gigahertz low noise cmos transimpedance amplifier. In *Circuits and Systems, 1997. ISCAS'97., Proceedings of 1997 IEEE International Symposium on*, volume 1, pages 209–212. IEEE, 1997.

- [23] S-M Park and C Toumazou. Low noise current-mode cmos transimpedance amplifier for giga-bit optical communication. In *Circuits and Systems, 1998. ISCAS'98. Proceedings of the 1998 IEEE International Symposium on*, volume 1, pages 293–296. IEEE, 1998.
- [24] Sung Min Park and Hoi-Jun Yoo. 1.25-gb/s regulated cascode cmos transimpedance amplifier for gigabit ethernet applications. *Solid-State Circuits, IEEE Journal of*, 39(1):112–121, 2004.
- [25] Sasa Radovanovic, A-J Annema, and Bram Nauta. A 3-gb/s optical detector in standard cmos for 850-nm optical communication. *Solid-State Circuits, IEEE Journal of*, 40(8):1706–1717, 2005.
- [26] Ilkka Saastamoinen, Teemu Suutari, Jouni Isoaho, and Jari Nurmi. Interconnect ip for gigascale system-on-chip. In *Proc. of Euro. Conf. Circuit Theory and Design (ECCTD)*, pages 281–284, 2001.
- [27] Michele Stucchi, Stefan Cosemans, Joris Van Campenhout, Z Tokei, and Gerald Beyer. Benchmarking on-chip optical against electrical interconnect for high-performance applications. In *Interconnect Technology Conference and 2011 Materials for Advanced Metallization (IITC/MAM), 2011 IEEE International*, pages 1–3. IEEE, 2011.
- [28] C Sung Min Park Toumazou. a packaged low-noise high-speed regulated cascode transimpedance amplifier using a $0.6\mu\text{m}$ n-well cmos technology. *Solid-State Circuits*, 2000.
- [29] Teemu Suutari, Jouni Isoaho, and H Tenhumen. High-speed serial communication with error correction using $0.25 \mu\text{m}$ cmos technology. In *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, volume 4, pages 618–621. IEEE, 2001.
- [30] Christer Svensson and Jiren Yuan. A 3-level asynchronous protocol for a differential two-wire communication link. *Solid-State Circuits, IEEE Journal of*, 29(9):1129–1132, 1994.
- [31] Vishak Venkatraman and W Burleson. An energy-efficient multi-bit quaternary current-mode signaling for on-chip interconnects. In *Custom Integrated Circuits Conference, 2007. CICC'07. IEEE*, pages 301–304. IEEE, 2007.
- [32] Vishak Venkatraman and Wayne Burleson. Robust multi-level current-mode on-chip interconnect signaling in the presence of process variations. In *Quality of Electronic Design, 2005. ISQED 2005. Sixth International Symposium on*, pages 522–527. IEEE, 2005.

- [33] Meng Xiongfei, Resve Saleh, and Karim Arabi. Layout of decoupling capacitors in ip blocks for 90-nm cmos. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(11):1581–1588, 2008.
- [34] Jiang Xu and W Wayne. A wave-pipelined on-chip interconnect structure for networks-on-chips. In *High Performance Interconnects, 2003. Proceedings. 11th Symposium on*, pages 10–14. IEEE, 2003.
- [35] Jiang Xu and Wayne Wolf. Wave pipelining for application-specific networks-on-chips. In *Proceedings of the 2002 international conference on Compilers, architecture, and synthesis for embedded systems*, pages 198–201. ACM, 2002.
- [36] Sheng Xu, Vishak Venkatraman, and Wayne Burleson. Energy-aware differential current sensing for global on-chip interconnects. In *Circuits and Systems, 2006. MWSCAS'06. 49th IEEE International Midwest Symposium on*, volume 1, pages 718–722. IEEE, 2006.
- [37] Yongxin Zhang. Drclvs for digital block, Technion working manual in VLSI Lab, 2014.
- [38] Yongxin Zhang. Mixed signal integration, Technion working manual in VLSI Lab, 2014.

שהסרת המשוב משפרת את תגובת התדר של המעלג. בשל דרישת מיקסום התדר, אנו מגדילים את אימפדנס הכניסה לדרגת הכניסה של RGC. אבל מסיבות אלו, במיוחד אימפדנס הכניסה, מעגל זה כבר אינו מתאים לעובדה באופן-זרם.

לאור הלקחים שנלמדו בחקירת מעגל TIA הנייל, אנו מציעים מעגל פשוט מאוד להשתתת תדר פעולה מקסימלי. מעגל זה פועל לכארה באופן מתח (VM) במקומות באופן זרם. במעגל המוצע יש רק שני שלבי מהפץ במסדר, ורק שני שלבי מהפץ במקלט. בגיןוד למקובל באופן מתח, אין במקלט קו התמסורת שום רפיורים ואין הגברת הסיגנל בדרך. תצורה זו מושגנה תדר גבוה יותר וצריכת הספק קטנה יותר. בטבלת השוואה בין כל התצורות ניתן לראות כי יחסית לתצורת TIA, התצורה המוצעת של שני מהפכים במסדר ושני מהפכים במקלט מושגנה תוצאות טובות יותר גם בתדר וגם בהספק.

השבד FOX2 כולל 30 ערכויים תקשורתיים בעלי אורךים שונים של קו התמסורת (הארוך ביותר מגע ל- 7 מ"מ) לצורות מעגלים שונות ואופני פעולה שונים. בשל מגבלת שכבות מתכת, רק שכבה 6 משמשת כקו תמסורת.

אופיו של קו התמסורת משפיע מאוד על ביצועי המסדר והמקלט. על מנת להבטיח הערכה אמינה של ביצועי העroz בסימולציה, השתמשנו במודול RLC לקו התמסורת (הכולל הערכות של ערכים מפולגים של התנדבות, קיבול והשראות). השתמשנו בתכנון התקן האלקטרומגנטי HFSS על מנת לקבל מודלים של התנדבות, השראות וקיבול של קווי התמסורת בכל תדר רצוי, באמצעות חישוב האימפדנס לאותו תדר. למען הגדלת הדיווק, ניתן להعبر את העריכה המדוייקת של קווי התמסורת מתוכנת קידנס לתוכנת HFSS לקבלת פרמטרי S טובים ביותר. פרמטרים אלו יכולים להיות מוחזרים לתוכנת קידנס ולמדל את הקו באופן מיטבי. אבל בשל מגבלות הזמן, התקן שמהפרט בתיזה ומשמש לייצור עדין מבוסס על מודלי RLC.

הסדר והמקלט פועלים בקצב גבוהה בהרבה مما ניתן להعبر דרך החיבורים החיצוניים לשבד. על מנת להעניק את התדר, הוספנו מעגל למדידת קצב שככל הוציא סיגנל היחסני לתדר השעון הפנימי והמייצג אותו, כתגובה לבנייה בקרה מתאימה.. באמצעות סיג널 זה אנו יכולים להעניק את קצב העבודה ולהשוותו לתוצאות הסימולציה. כמו כן, להבטחת מדידה נאותה, שלונו גם לצורך בדיקות מסדר מקלט פשוטים הפועלים בקצב נמוך שמאפשר העברת הסיגנלים על גבי היציאות והכניות של השבד.

בקר ספרתי תוכנן לבקרת פעולות השבד. הבקר המקורי משבד FOX2 שופר וisoner. מערכת תכנון דיגיטלי של סייע לתרגם את הבקר למעגלים ששולבו בשבד. כל האימוטים הספרתיים בוצעו, לאבטחת פעולה נכונה של הבקר הספרתי. כמו כן אומתה פועלה נכונה של הבקר על ידי סימולציה כוללת של כל השבד.

מצאו שמעגלי האספקה כוללים רשות האספקה איננה מהירה דיה על מנת לספק בזמן זרים הנדרשים על ידי המעגלים האנגלזים והמעגלים הספרתיים המהירים ביותר. אי לכך, הוספנו לשבד קבלי צימוד ספקים, במיוחד ליד כל המעגלים המהירים. קבלים אלו מבוססים על טרזוניסטורי NMOS ומיעדים לסייע למעגלים המהירים להתגבר על מגבלות ספק הכוח ורשות האספקה.

תכן העריכה (LAYOUT) של המעגלים המהירים הינו קרייטי, ומאיץ רב הושקע בכיוונו הגדל של כל טרזוניסטורי להשתתת ביצועים מקסימליים, תוך שימוש סימולציות רבות לצורך הוכנות.

כל חלקים השבב חוברו ייחודי במערכת אינטגרציה שאיפשרה אינטראקציית מיטבי למניעת טעויות. לאחר סיום העריכה ביצעה סימולציה על המעלג הסופי, לוידוא פועלה נכונה ולהערכת ביצועים. בעוד שהערכת הביצועים באופן זה איננה שלמה, בדיקת הנכונות טובה למדי ויש סיכוי רב שהשבדים יפעלו נכונה.

במעגל הסופי 30 ערכאים ניסיוניים, כאשר העroz הארוֹץ ביותר מגע ל-7 מ"מ, והמעגלים מכילים תצורות שונות להשוואה.

השבדים שבו מייצור אולם לפני שניתן לבדוק אותם יש לאירוע את השבדים, לייצרلوح בדיקות, להרכיב אותם, ולהוכיח את מערכת הבדיקה.

מעגלי קליטה בתדר גבוה לתקשורת על השבב

תקשורת נתונים למרחק בתוך שבבים דיגיטליים גדולים ומערכות-על-شبב (SOC) נעשית מأتגרת יותר ויוטר מפני שקווי התקשורת שבב משתפרים בקצב איטי יותר מאשר קצב השיפור במעגלי החישוב הודות להתקדמות הטכנולוגית. פתרונות תקשורת על גבי עורקים מקביליים אפשררים העברת מידע בקצב גובה אבל צורכים הספק גובה, תופסים שטח רב, חשופים לרעש ומסקימים על ניתוב חוטים שבב.

ארQUITטורה של קו תקשורת טוית מהירה אסינכרונית, המשלב מעגל לעזר שעון, רגיסטרי הזזה מהירים Level Encoded (LEDR) מסוג (NRZ) (Dual Rail), כולל ממיררים מקבילי לטורי ובוחרת מטורי למקבילי, וכן ערוץ של קו תמסורת דיפרנציאלי המכיל מספר אותות בו זמנית (בעל צינור גלים או wavepipelining) הפעיל באוף זרם (current mode, CM) לבנותה כך שתפעול בקצב המוכתב על ידי השהייה שער בודד. השעון מספק אותות בקרה לריגיסטרי הזזה במשדר, המאפשרים המרה מאות מקבילי לטורי. לאחר קידוד LEDR מושג הקצב המכטימי, ומעגל אנלוגי משדר את אותן על גבי קו התמסורת לכיוון המקלט. מפניהם השעון שוחזרו מפעילים רגיסטרי הזזה במקלט שמmirים את אותן הטרוי בחזרה למקבילי.

במחקר קודם תוכנן ויוצר שבב ניסוי בשם FOX1 שהכיל קווי תקשורת מהירים ומעגלים ניסיוניים רבים וממוש בטכנולוגיה 65 ננומטר בחברת יbam, להדגמת קווי התקשורת המהירים. המטרה הייתה להשיג מיתוג בזמן סיבית של השהייה שער בודד, המביא לקצב שידור מידע של 67 גיגה-ביט לשנייה. אבל החלקים האנלוגיים של המשדר והמקלט לא היו מהירים דיים כי המשדר האנלוגי היה מסובך מידי והמקלט, שהתבסס על התאמה של מקלט זרם עם מסווב. כתוצאה לכך, המהירות המקטימלית שאותו שבב ניסוי היה אמרור להשיג הייתה מוגבלת ל- 28 גיגה-ביט לשנייה בלבד. כמו כן, בשלTeVות בתהליכי התכנון של מעגלי הבדיקה הספרתיים שבגינה רשות הפעלת השעון לא הייתה בעלת כח דחיפה מספיק, השבב לא פעל כלל.

במסגרת המחקר הנוכחי, תוכנן שבב ניסוי שני, FOX2, והוא מיוצר בחברת טואור בישראל בטכנולוגיה של 180 ננומטר. חלק מן המעגלים הספרתיים הקודמים משמשים שוב שבב החדש, תוך בדיקה מחדש של התכנון בرمת RTL ותוכנו מחודש של הרמה הפיזיקלית כולל השמה, חיוט, הוספת שעון, בדיקה מעמיקה של כל התזומות ובדיקות של עמידות הספקים, יציבות הסיגנלים וככוננות המימוש. במסגרת התכנון נבנה הליך תכנון (design flow) מלא ומפורט הכול גם תכנון ספרתי וגם תכנון של אות מעורב (mixed signal design). עבודה זו מתרכזת בתכנון של המעגלים האנלוגיים המהירים על מנת להשיג את מטרות השבב. כמו כן, מוצע מודול חדש של קו התמסורת על מנת למדל אותו במסגרת התכנון.

משדר ומקלט הפעלים באוף-זרם (CM) יכולים לאפשר תקשורת מהירה מאוד בתנופת מתח נמוכה לאורך חוטים ארוכים שיש להם קיבול גובה. בהשוואה לפעולה באוף-מתוח (VM), פועלות אופן זרם אמורה לאפשר תנופה נמוכה יותר, צירכית הספק דינמי נמוכה יותר, מרחק רב יותר ופועלה מהירה יותר. בבדיקה משתמשים במעגל טיפוסי במשדר וגם משווים אותו למעגל משופר שתוכן במירוח לפעולה אסינכרונית. בשל מגבלות רוחב סרט וקלוקול אותן הנובעים מהקיבול הגבוה של קו התמסורת,anno משתמשים בסוג מיוחד של מקלט אוף-זרם.

תצורת שער-משותף של trans-impedance משמשת בדרך כלל ליישומים רחבי סרט חזותות לכך שקיבול הכניסה הגבוה משפייע חיובית על רוחב הסרט. אבל לעיתים הביצועים של תצורה זו אינם מספיקים. בדקנו תצורה משופרת הנהוגה ביישומים אלקטרו-אופטיים והמובוסת על regulated-cascoded trans-impedance (RGC TIA). לאחר שהיישום שלנו שונה מאלקטרו-אופטיקה (בעיקר מושם שהסיגנל המתקבל גדול יותר ומשום שהוא מעדיפים תדר עבודה גבוהה על פני רוחב סרט והגב), שיפרנו את המעגל כי מצאנו

המחקר נערך בהנחיית פרופ' רן גינוסר ודר' אהרון אוניקובסקי
בפקולטה להנדסת חשמל.

אני מודה לטכניון על התמיכה הכספית הנדיבת בהשתלמותי.

מעגלי קליטה בתרד גבוח לתקשרות על השבב

**חיבור על מחקר לשם מילוי תפקידי של הדרישות לקבלת התואר
מגיסטר למדעים בהנדסת חשמל**

יונגשין זאנג

הוגש לסנט הטכניון – מכון טכנולוגי לישראל

תשבי תשע"ה חיפה אוקטובר 2014

מעגלי קליטה בתדר גבוה לתקשורת על השכבה

יונגשיין זאנג