FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

U. PORTO

FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

10Gb/s Programmable Driver With Active Termination

Nuno Alves Pereira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Cândido Duarte

Co-Supervisor: Joaquim Machado

February, 2017

Abstract

As today's signaling and speeds of analog circuits reaches Giga Hertz (GHz), systems need to be capable of handling high rates and the supporting analog components have to generate and amplify high-frequency signals. To maximize the signal integrity of the digital outputs, high-speed ADCs use high speed interfaces and differential signaling. This is achieved through the use of 2 wires for each discrete signal that is to be carried across a circuit board or cable. The voltages on each of these conductors swing in opposite directions and also have a very small amplitude when compared to single-ended signaling such as CMOS or TTL. It is because of the inherent noise immunity of the differential circuit that low voltage swings can be used. This in turn means that the signal frequency can be faster as the rise time is shorter.

In this dissertation it is proposed a fully functional output Driver, capable of working with a single differential pair at speeds up to 10 Gbp/s, overcoming the issues mentioned above. This Driver should be programmable in a way, that if desired it's output termination impedance can be chosen from a range of values around 100 Ω for a better match with the line and to compensate for Process, Voltage and Temperature variations. The Driver has also a programmable output swing voltage that can be defined for high voltage transmission 1.6V or low power transmission 1V by activating different paths within the circuit. The circuit must too perform in a satisfactory way for PVT variations.

The results obtained are promising and demonstrated that it's possible to build a *Driver* with this characteristics and still keep good integrity of the transmitted signal. Thus, it is necessary more investigation work concerning the accuracy of the design.

This document describes not only the proposed architecture and full schematic detail explanation, but also an evaluation and analysis of the results for a 10GHz transmission. Possible future work regarding improvements and demanding challenges that were not overcome are also discussed.

Acknowledgements

First, I would like to offer my sincerest gratitude to my supervisors, Professor Cândido Duarte from FEUP and Eng. Joaquim Machado from SYNOPSYS, who have supported me throughout my thesis with their patience, knowledge and "getting their hands dirty" when most don't. The level of my thesis dissertation is attributed to them for their encouragement and support and without them, it would have not been completed or written. I could not have thought of more helpful supervisors than them.

I would also like to thank Synopsys for providing the tools and support to the achievements on this work. Also, I would like to take this opportunity to highlight how friendly the environment in this company was, which resulted in making all the hours I spent there working much easier.

I would also like to thank all the friends for the company through this phase.

To my parents, brother and close family, I'm most grateful for providing me with literally everything I needed for this thesis to be done. From all the rides my father gave me to and from SYNOPSYS until the delicious food from my mother, thank you with all my heart.

Finally and most importantly, I would like and absolutely need to emphasize my deepest thanks to my dear girlfriend for all the support and help in everything that concerns my well being. I love you the most, you're the best, Jessica.

Nuno Pereira

“You should be glad that bridge fell down.
I was planning to build thirteen more to that same design”

Isambard Kingdom Brunel

Contents

List of Figures

List of Tables

Abreviaturas e Símbolos

CMOS	complementary metal-oxide-semiconductor
CML	Current mode logic
HVP	High Voltage Path
HV	High Voltage
LV	Low Voltage
LVP	Low Voltage Path
OD	Output Driver
TL	Transmission Line
VML	Voltage mode logic
vlsi	Very large scale integration
WWW	*World Wide Web*

Chapter 1

Introduction

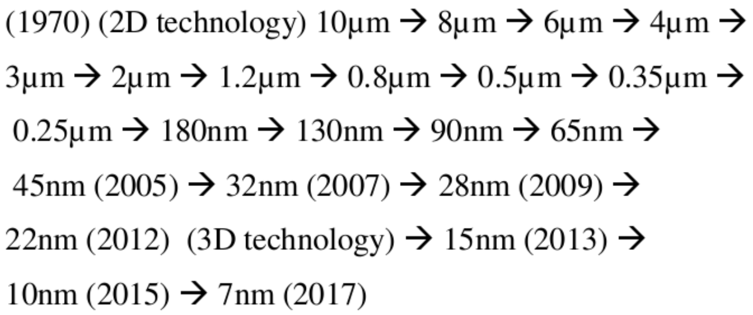
This document reflects the work performed in the final curricular unit of Integrated Master in Electrical and Computer Engineering, telecommunications major in the current year of 2016/2017. This dissertation was done in a professional environment in Synopsys Portugal. My work was supervised by professor Cândido of Faculty of Engineering of Porto and Co-supervised by Engineer Joaquim Machado of Synopsys Portugal.

1.1 Context

Computing devices, such as computer servers, workstations, personal computers, game consoles, and smart phones, have become increasingly more powerful with each new generation of semiconductor process. To keep up with the increase in performance, the data communication speed between the components of the computing device has also been increasing, rising from a few hundred Mb/s in the early 1990s to several Gb/s nowadays.

CMOS is a combination of N-type and P-type MOSFET (Metal-Oxide- Semiconductor Field-Effect Transistor). CMOS technology is used for constructing integrated circuits, microprocessors, microcontrollers, sensors, RAM (Random Access Memory) and many more digital circuits. Gordons Moore observed that number of transistor doubles after every 18 months in an integrated circuit [1] . This computerized electronics world demands more and more faster devices. This can be achievable by scaling CMOS technology from fraction of millimeters to few of nanometers in today technologies.[2]

After bipolar junction transistor MOSFET(Metal-Oxide-Semiconductor Field Effect Transistor) comes with very interesting feature like: low power consumption, low operating voltage, higher speed etc. which make MOSFET useful in electronics design. Two types of MOS transistor PMOS and NMOS are invented and used for designing integrated circuits. Both types have very high static power consumption. This problem is solved if and only a logic designed in such a way that it consumes no power in static state. After decades Frank Wanlass introduces a new logic designed using two complementary p-type and n-type MOSFETs. Two main advantages of CMOS technology is high noise immunity and very low static power consumption [7]. The last several decades have seen innovation of new CMOS technologies with excellent features. The trends of MOS integrated circuits downsizing is shown in figure 1.1



(1970) (2D technology) 10µm → 8µm → 6µm → 4µm →
3µm → 2µm → 1.2µm → 0.8µm → 0.5µm → 0.35µm →
0.25µm → 180nm → 130nm → 90nm → 65nm →
45nm (2005) → 32nm (2007) → 28nm (2009) →
22nm (2012) (3D technology) → 15nm (2013) →
10nm (2015) → 7nm (2017)

Figure 1.1: Trend of downsizing for MOS integrated circuits [3, 4, 5, 6]

In this work, the technology used in the design KIT was 28nm.

1.2 Motivation

As data communication reaches multi-gigabit/sec rates, the task of ensuring good signal integrity, both on-chip and off-chip, becomes increasingly important. Understanding the high-frequency physical effects introduced by the wire or interconnect is as important as the silicon design itself. Moreover, device jitter (generated by on-chip circuitry) now becomes a signal-integrity (SI) problem, because system-level behavior (such as jitter amplification and cancellation) must be modeled. The time when signal integrity was considered, only after the silicon was built, has passed [7]. Signal-integrity design considerations must be considered upfront to ensure the robust operation of modern high-speed systems. New design methodologies must be introduced and employed to account for the physical effects that could be ignored at lower data rates but not at higher-data rates.

Before building any hardware or system, worst-case design parameters and interconnect electrical behavior must be evaluated and analyzed. A detailed and accurate understanding of the electrical behavior of interconnect, advanced signaling, and circuit techniques can be used to overcome the non-ideal effects.

In this dissertation it is proposed a fully functional output Driver, capable of working with a single differential pair at speeds up to 10 Gbp/s, overcoming the issues mentioned above. This Driver should be programmable in a way, that if desired it's output termination impedance can be chosen from a range of values around 100 Ω for a better match with the line and to compensate for Process, Voltage and Temperature variations. The Driver has also a programmable output swing voltage that can be defined for high voltage transmission ($1.6V$-$1.V$ or low power transmission ($800\ mV$) by activating different paths within the circuit. The circuit must too perform in a satisfactory way for PVT variations.

1.3 Dissertation Structure

In addiction to this introduction, this dissertation has 4 more chapters.

In chapter 2, fundamental knowledge about this dissertation is described. In chapter 3, the implementation and explanation about every part of the design is presented. In chapter 4, the results are presented. In chapter 5, conclusions and future work that can be developed.

Chapter 2

Fundamental concepts

This chapter covers the fundamental concepts of this dissertation. Section 2.1 introduces the scope of the work for the reader to understand what applications is the driver being design for. Next, section 2.2 analyses the transmission line and compares serial transmission with parallel transmission. Section 2.3 discusses the technology chosen. In 2.4 explanation about the transmitter is given with section 2.5 focusing on the driver presented in the transmitter and different topologies for it's design. An important part of the driver developed in this work is the voltage regulator and so, section 2.6 will approaches that subject. Finally, in 2.7 related work is presented and was were all the theory was studied in order to be able to complete this work.

2.1 High-Speed Communications

High-speed interfaces are used for fast data transfers in data communications hubs, wireless base stations, flat-panel displays, servers, and peripherals like printers and digital copy machines. Typical distances reach from a few inches (between ICs or from board to board) up to several meters (between systems). Due to the short data link distance, ground potential differences between driver and receiver are assumed to be small, and the required common-mode input voltage range of a receiver is commonly limited to a few volts.

Point to point data transfer systems consists of a transmit driver, a receiver and the transmission line as shown in 2.1.

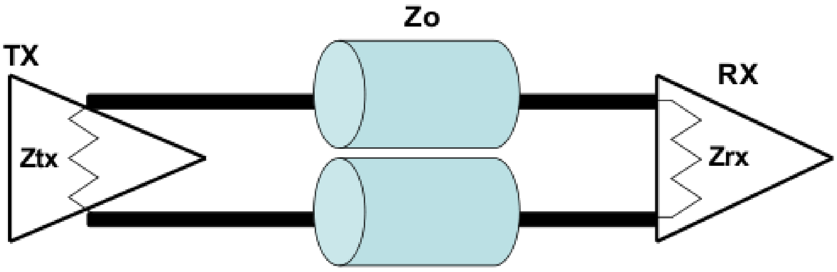


Figure 2.1: Point to point data transfer system

Achieving low power operation and maintaining signal integrity are two key challenges in transmit driver design. To maintain minimal return loss, good signal integrity and minimize reflections, the driver needs to provide internal impedance that matches the transmission line impedance ’Zo’. Ideally, both the transmit driver and the receiver should have differential internal termination = 2*Zo and in this work it’s value is 100 Ohms.

2.2 Transmission Line

When considering a high speed communication the non-dispersive or ideal model isn’t an accurate representation of the reality and so it shouldn’t be considered in most cases. Real channels often show a behavior close to a "pass-band" and usually respond differently to inputs with different frequency components, meaning they are dispersive.

With this said, it is necessary to redefine the simple non-dispersive model (AWGN) to represent this type of channels. An often used model is the disperive channel model:

$$r(t) = u(t) * hc(t) + n(t)$$

Where, u(t) is the signal to be transmitted, h(t) is the step response from the channel and n(t) is white Gaussian noise. In essence, the channel model follow the characteristics of hc(t) filter. Considering the most simple dispersive channel, hc(t) can me modeled as a simple low-pass filter.

In 2.2, the model for a dispersive channel is shown, Hc(f) represents the transfer function of the channel, typically a low pass behaviour, and AWGN is white noise introduced in the channel.

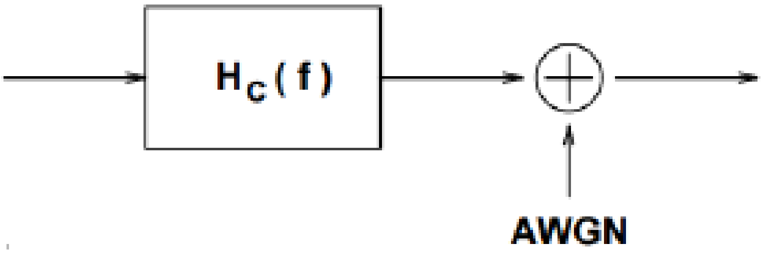


Figure 2.2: Dispersive channel

2.2.1 Transmission Type - Parallel Link

In data transmission, parallel communication is a method of conveying multiple binary digits (bits) simultaneously. It contrasts with serial communication, which conveys only a single bit at a time; this distinction is one way of characterizing a communications link.

The basic difference between a parallel and a serial communication channel is the number of electrical conductors used at the physical layer to convey bits. Parallel communication implies more than one such conductor. For example, an 8-bit parallel channel will convey eight bits (or a byte) simultaneously, whereas a serial channel would convey those same bits sequentially, one at a time. If both channels operated at the same clock speed, the parallel channel would be eight times faster. A parallel channel may have additional conductors for other signals, such as a clock signal to pace the flow of data, a signal to control the direction of data flow, and handshaking signals.

One problem with parallel communication is that we cannot increase the signal frequency for a parallel transmission without limit, because, by design, all signals from the transmitter need to arrive at the receiver at the same time. This cannot be guaranteed for high frequencies, as you cannot guarantee that the signal transit time is equal for all signal lines (think of different paths on the mainboard). The higher the frequency, the more tiny differences matter. Hence the receiver has to wait until all signal lines are settled and that waiting lowers the transfer rate.

Another obstacle with parallel transmission is when crosstalk occurs. The higher the frequency, the more pronounced crosstalk gets and with it the higher the probability of a corrupted word and the need to retransmit it. Parallel communication is and always has been widely used within integrated circuits, in peripheral buses, and in memory devices such as RAM. Computer system buses, on the other hand, have evolved over time: parallel communication was commonly used in earlier system buses, whereas serial communications are prevalent in modern computers.

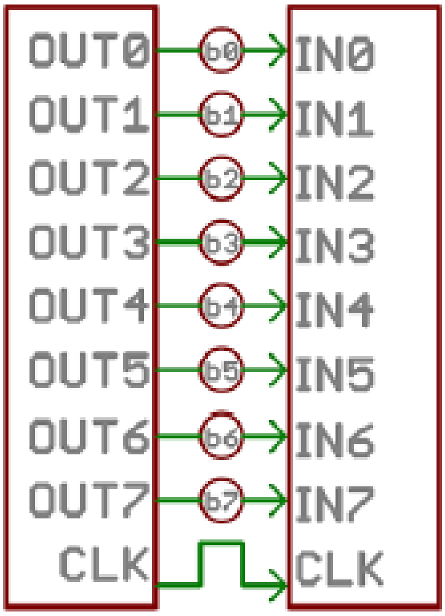


Figure 2.3: Parallel Link Communication

2.2.2 Transmission Type - Serial Link

Serial link communications have either a single data wire, or a single differential pair, and the remaining of the wires are either ground or control signals. The intended transmitting bits are sent sequential by the same channel (wire). This results in a lower cost of transmission but a delay in the transfer rate.

In this type, the transmission can be asynchronous or synchronous. In the asynchronous case, group of bits are sent independently with the help of a signaling system. In the synchronous transmission, bits are aggregated in a bus to be sent continually.

Serial interfaces are becoming more prevalent than parallel interfaces given their ability to deliver high speed operation using less overall power consumption and input/output counts. They not only reduce signal interference, noise and crosstalk, but also eliminate the need for multiple line drivers and buffers. [8]. In this work the driver developed will account for a serial link transmission, as the parallel type limits the speed of the communication in a way, 10GHz would be impossible to achieve. A single differential pair wire with 100 Ohms characteristic impedance will be considered.

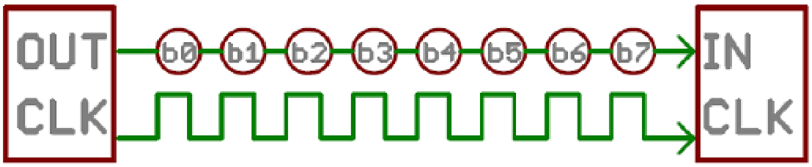


Figure 2.4: Serial Link Communication

2.3 Technology

Today's integrated digital logic devices trace their roots back to RTL (Resistor-Transistor Logic) components which were pioneered in the 1960s. RTL eventually evolved into DTL (DiodeTransistor Logic) which in turn was followed by TTL. With TTL (Transistor-Transistor Logic) and then ECL (Emitter Coupled Logic), designers found components that were orders of magnitude better than the first RTL devices, providing acceptable noise immunity, the ability to fanout to more than one device, and usable (25 MHz) propagation speeds. At the time, this logic provided engineers with a technology that enabled them to increase the density of their systems by providing prepackaged functions for "drop in" design use. However, these early devices consumed a significant amount of power (typically 30 to 40 mW per gate at 1 MHz) which catalyzed a demand for lower powered devices. In response to the demand for lower power, RCA developed the first MOSFET based, mass production logic devices in the early 1970s. This family provided devices that accommodated a large range of supply voltage and had zero quiescent (excluding leakage) currents. The outputs would swing from rail-to-rail and the inputs were high impedance, so input current was minimal. Drawbacks to this early CMOS offering included susceptibility to static induced latchup, poor output drive.

2.3.1 CMOS vs BIPOLAR Transistors

Both bipolar and CMOS technologies have advanced significantly along their respective evolutionary paths, with CMOS having merits over Bipolar in areas of low power dissipation, large noise margins and greater packing densities. Bipolar has merits over CMOS in areas of faster switching speed and large current capabilities being many times preferred for that reason.

A new family called BiCMOS has emerged which combines both Bipolar and CMOS technologies in single IC. BiCMOS utilizes benefits of both Bipolar and CMOS technologies. It presents high input impedance, low quiescent power, and strong output drive characteristics. However, BiCMOS devices consume more power than pure CMOS ICs, and require approximately 30% more die space to implement the same function. Manufacturers of semiconductors are continually seeking opportunities to decrease cost from their processes and products, which has resulted in widespread use of inexpensive plastic packages. These plastic packages have reduced thermal capabilities.

The low power dissipation of the technology is one of the reasons why CMOS dominates the VLSI market. [9]. Programmable logic device manufacturers have benefited from the CMOS technology, and are aggressively pursuing smaller pitch processes to gain advantages in speed, power, and cost.[5, 6].

Various design techniques also exist to decrease die size, and thereby decreasing cost, at both the architecture and implementation level. One method of decreasing die size is to implement the output buffer in an NMOS fashion. While saving space, this method diverges from the complementary output structure resulting in a loss of some benefits. Since most programmable logic devices utilize a form of I/O structure where a pin may operate as an input or an output, the impact of modifying an "output" buffer also has "input" ramifications. A true CMOS (complementary) buffer provides desirable characteristics such as rail-to-rail output swings and overvoltage protected inputs. [9].

2.4 Transmitter

When designing a transmitter for a high speed communication application, some features need to be defined in order to choose the desired technology and architecture. For the desired application we need to know the specs we are trying to achieve in:

- Required bandwidth
- Termination
- Output Swing
- Power consumption
- Delay

As the speed increases and higher frequencies are achieved, more obstacles to the design start to appear. For high speed like 10Gbp/s, the size of the MOSFETs at the output stage need to be kept small in order for their switching speed stay high. This restriction limits other important parameters like the current flowing through the driver branches or if the MOSFETs are small, their equivalent resistance becomes high and that needs to be accounted for. When trying to achieve optimum performance, the characteristic impedance of the transmitter must be consistent and equal to the load termination Z0 = 100 Ohms. Care must be taken to avoid impedance discontinuities and an appropriate termination network is essential. Though the overall intent of the termination network may always be the same, a considerable number of variables must be considered when deciding on the appropriate termination scheme. This scheme can be done in series termination or parallel termination.

2.5 Driver topologies

In this section it will be described the difference between current mode logic, voltage mode logic and the hybrid driver where both of the previous logics are used.

2.5.1 Current Mode Logic

What makes current mode logic driver a good candidate for modern CMOS technologies is that it can operate at relatively low power supply, has tolerance to common mode noise, and is able to work at very high frequency.

Current mode drivers switch a constant current source of several mA to produce an output differential voltage over a termination resistor (see Fig.2.5). Implementing a switching mechanism to disable these large currents dedicated circuit techniques that complicate hardware and increase power consumption.

For a CML driver, the differential pair can be made of nMOS transistors with a pMOS current mirror as current source or two pMOS transistors with a nMOS current mirror. In the first case, like in the 2.5 nMOS transistors control the current flow of each side of the differential pair in response to differential input. The size of the nMOS transistors is determined by optimization of CML driver operation. The pull-up circuit, pMOS transistors, is tuned to the characteristic impedance of transmission line to achieve impedance matching.

When all the bias current flows through either nMOS transistor, the output voltage swing reaches $RD*Ibias$. So it is seen that the maximum output differential voltage swing in only a function of the drain resistor and the bias current, providing that the full current switching takes place. From the relationship between the bias current $Ibias$ and the minimum input voltage $\Delta V^2 in, min$ is expressed in 2.1:

$$Ibias = 1/2 * Kn * W/L * \Delta V^2 in, min \tag{2.1}$$

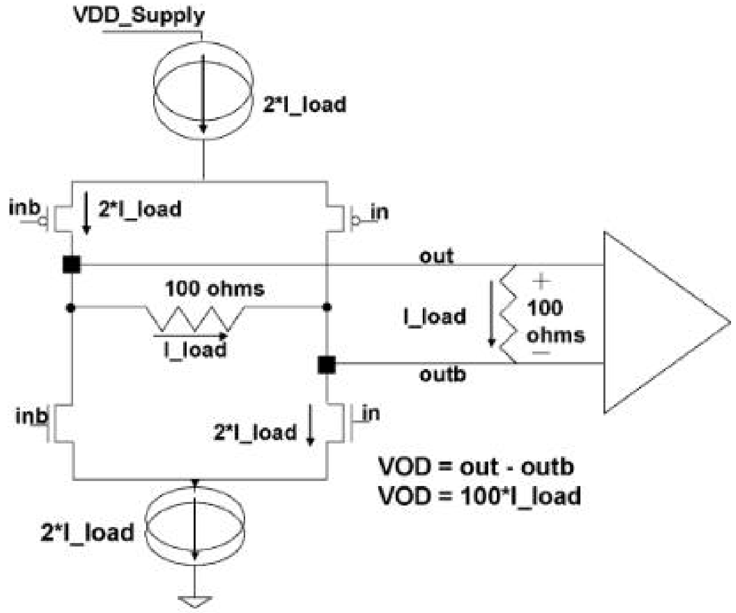


Figure 2.5: Current Mode Logic Output Driver

,where Kn is a process variable related to mobility and gate capacitance per unit area, L and W are length and width respectively. L should be minimum value, allowed by the chosen technology, so the only freedom for optimization is in the nMOS transistor gate width W.

2.5.2 Voltage Mode Logic

Currently, a big competitor for the CML driver is the Voltage-Mode-Driver (VML). The CML driver requires static current flow and its power dissipation is larger than in VML, which gently overcomes these disadvantages. This mode, consumes four times less power than CML drivers and in addition it supports-high swing termination voltage. The basic topology of VML transmitter is shown in 2.6

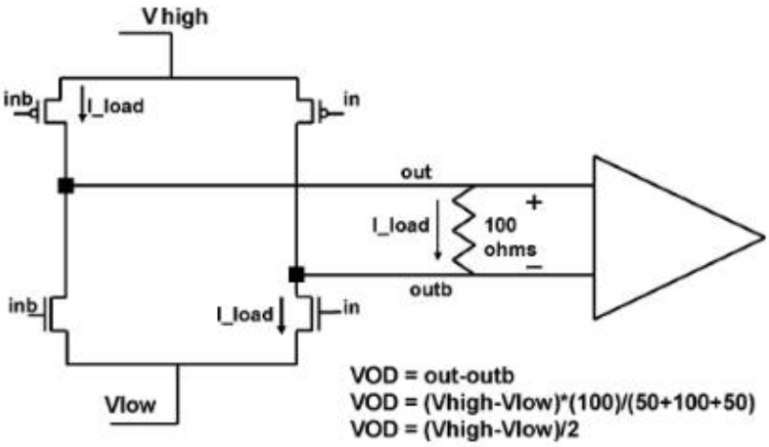


Figure 2.6: Voltage Mode Logic Output Driver

The driver is subdivided into two branches: the pull-up and the pull-down. Each branch is implemented as nMOS or pMOS switch transistor followed by a series termination resistor. Either of branches must be impedance-matched to the transmission line. This means that the sum of the resistance of the transistor and resistor should be equal to transmission line impedance. The problem of impedance matching is still more difficult here than on CMLogic but it's possible to achieve good matching by focusing on appropriate ratio between the transistor equivalent resistance and the resistor resistance.

In this topology the output swing will be controlled by the supply voltages (Vhigh and Vlow), and for a balanced output, the equivalent resistance of the pull-up branch should be equal to the pull-down branch.

2.5.3 Hybrid Mode Logic

The Hybrid Mode Logic, tries to overcome the shortcomings of CML and VML about meeting speed as well power specifications in deep sub-micron technologies. Voltage mode signaling is slow while current mode signaling suffers from serious static power dissipation problem.

Hybrid Mode Logic operates in current mode for high input data rates and for lower input data rates, it switches to a voltage mode operation were the static power dissipation is much reduced. The switching in the mode of operation of the Hybrid Driver is controlled by for example a Schmitt trigger based control circuit.

A possible implementation of this topologie is shown in 2.7.

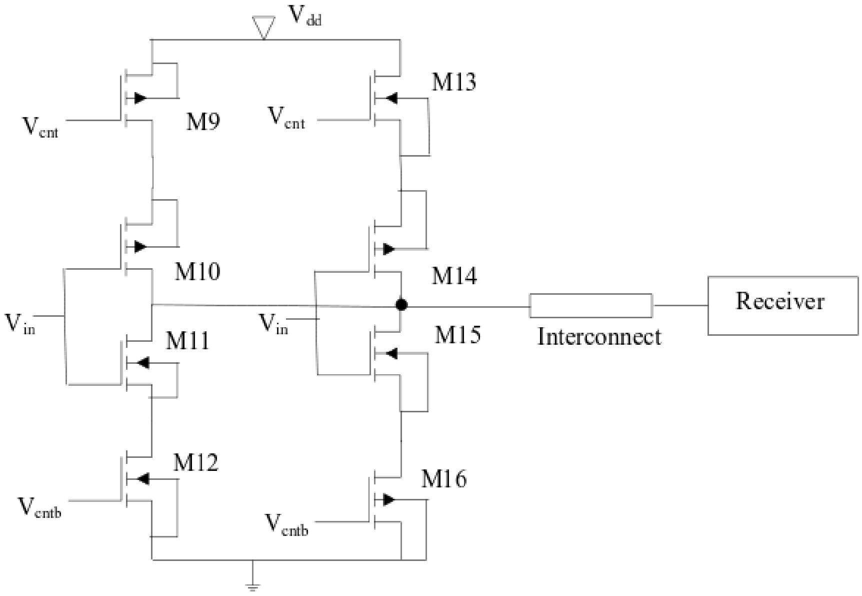


Figure 2.7: Hybrid Mode Logic Output Driver

2.6 Voltage Regulator

In analog design circuits, when we want to supply a circuit, it isn't always possible to use an external voltage or current source since this sources can have 10% error over its nominal value and so, and that can be undesired. To overcome this uncertainty and to be able to deliver a robuster driver, the use of a voltage regulator is a possible solution. A well designed regulator achieves 4-5% error only(more than 50% improvement over a external source). A voltage regulator is a block that generates a fixed output voltage of a pre-set magnitude that remains constant regardless of changes to its input voltage or load conditions. There are two types of voltage regulators: linear and switching.

A switching regulator converts the dc input voltage to a switched voltage applied to a power MOSFET or BJT switch. The filtered power switch output voltage is fed back to a circuit that controls the power switch on and off times so that the output voltage remains constant regardless of input voltage or load current changes.

A linear regulator employs an active (BJT or MOSFET) pass device (series or shunt) controlled by a high gain differential amplifier. It compares the output voltage with a precise reference voltage and adjusts the pass device to maintain a constant output voltage.[10]. We know that Ohm's law states

$$V = I * R \tag{2.2}$$

If a linear regulator maintains a constant output voltage (V) over varying input voltage and output current into the load, it follows that R is what is being controlled by the regulator. 2.8 shows a typical LDO regulator topology:

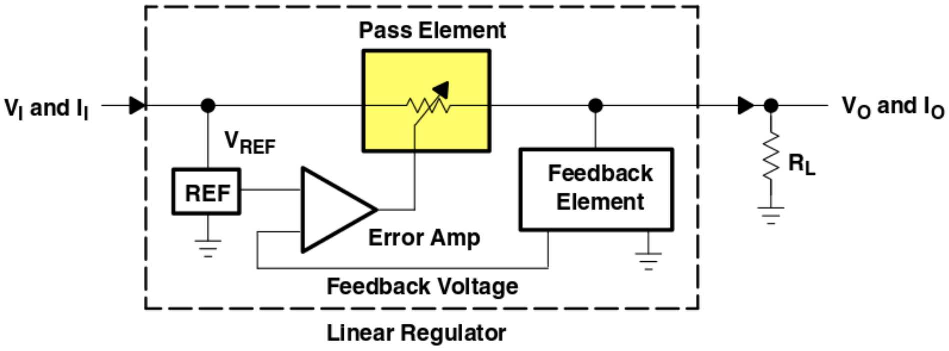


Figure 2.8: Low Drop Out Regulator

While regulating, the pass element (show in 2.8 in yellow color) is always on in a linear regulator. Using the potentiometer model as a guide and remembering Kirchoff's current law, we can see how the input current must be equal to the output current (assuming there isn't quiescent current). The quantity of dissipated power (PD) can be extracted by the following equation:

$$PD = (VI - VO) * PO \tag{2.3}$$

with PI being the power placed into our system and PO being the power delivered to the output. And so, with VI, VO and IO(max) we can calculate the maximum power dissipation by our regulator:

$$PD = (VI - VO) * PO \tag{2.4}$$

In this dissertation it will be used a linear regulator (Low drop) over a switching one, in a way of not wanting to use inductive type components that for example buck or boost converters introduce, due to its obstacles in the integration. Another reason is that the noise output from a linear regulator is much lower than a switching regulator with the same output voltage and current requirements. However, the linear regulator's power dissipation is directly proportional to its output current for a given input and output voltage 2.4, so typical efficiencies can be 50% or even lower. Switching regulators can achieve higher values of efficiency.[10]

2.7 Related Work

As described in the previous chapter 1, the scope of this work is to develop a high speed output driver for communications up to 10GHz with a programmable output swing for high and low power transmissions. The driver termination impedance has to match the line at 100 Ω while enduring PVT variations.

Knowing that process, voltage and temperature (PVT) variations dramatically affect the driver output resistance [11] , [12], components like the output resistors can be affected by as much as 10% or 20% error which would result in a unacceptable mismatch to the TL impedance and so, reflections [11]. Two types of terminations exist; parallel and series terminations. The first eliminates TL reflections, however it consumes more power. The second provides and output driver which absorb incident waves, which effectively damps TL reflections [13].

With the fundamental concepts of chapter 2 learned, research was done about output drivers that could meet this requirements. The following text refers what already existed and what limitations existed in related works.

In paper [14] a design for a 10 Gb/s output driver offering a constant output voltage swing level over process, voltage and temperature variations is presented. This output voltage swing level can be controlled by MOS resistances in the final output driver stage by a feedback technique in a way that compensates PVT variations. This feedback technique works as follows, the circuit generates gate voltages of pMOS and nMOS in the final stage of the output driver so that the voltage swing level stays adjusted by changing the MOS equivalent resistance. This way, the MOS "on" resistance with approximately linear I-V characteristics is obtained by operating the transistors in the triode-region. The disadvantages with this topology is that changes done to the MOS "on" resistance to achieve a more stable output voltage swing, will return a mismatch between the driver termination and the TL.

Another way to achieve termination match with the transmission line, while compensating the PVT variations is to build the output stage with the scheme presented in this papers: [12] , [15]. In order to achieve 50Ω differential termination (100 Ω total), the driver consists in multiple "slices" that contain different number of elements as shown in 2.9:

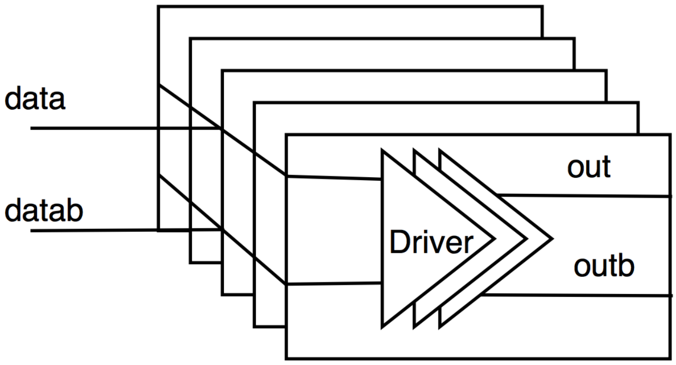


Figure 2.9: Output stage of a VML driver with programmable termination

Each slice/module, has an output driver that can be working alone if only one module is enable or in parallel if multiple modules are enabled. Signal "data" and "datab" carry the differential information to be transmitted and connect to each module, the module can be conducting or not. The "out" and "outb" signal are transported in output wires that are connected to all modules too. By giving a impedance value to each module of 500Ω for example, we can build a five module driver like the one in 2.9 and if by PVT variations the effective impedance of the driver isn’t close to the desired value, we can enable and disable as many slices as we desire to achieve good match with the TL impedance.

Chapter 3

Design Architecture

In this chapter the overall design and implementation of the proposed driver will be explained in detail. This work was fully developed in Synopsys with Synopsys Sofware, being the design KIT provided by the company.
This chapter explains the existing blocks of the architecture and all the component function and sizing are detailed. Theoretical concepts and application will be described as they were used in the development of this work.

The objective of this work was to develop a fully functional output *Driver*, capable of working with a single differential pair at speeds up to 10 Gbp/s. This Driver had to match its output impedance with the impedance of a line with 100Ω and it should be programmable in matters of output swing. The driver should have desired behavior when regarding the model - typical conditions and with PVT variation - not typical conditions.

Too achieve this objectives it's presented in 3.1 the block diagram that will serve to explain the architecture behind this work. For simplicity reasons, many nets and signals were omitted in this diagram but will be shown and explained further on.

In 3.1 it's possible to observe the Block diagram of the Driver. The main blocks that consitute this work are:

- *LDO Regulator*;
- *Reference Circuit*;
- *Regulator Control*;
- *Data Generator and Control Block*;
- *Slices Control Unit*;
- *Driver*;

Firstly, an explanation about each block function is presented. The *LDO Regulator* block is responsible for stabilizing and fixing the voltage in order to be a voltage supply with less tolerance than if we used an external one. In order for the *LDO Regulator* to work, it needs a reference

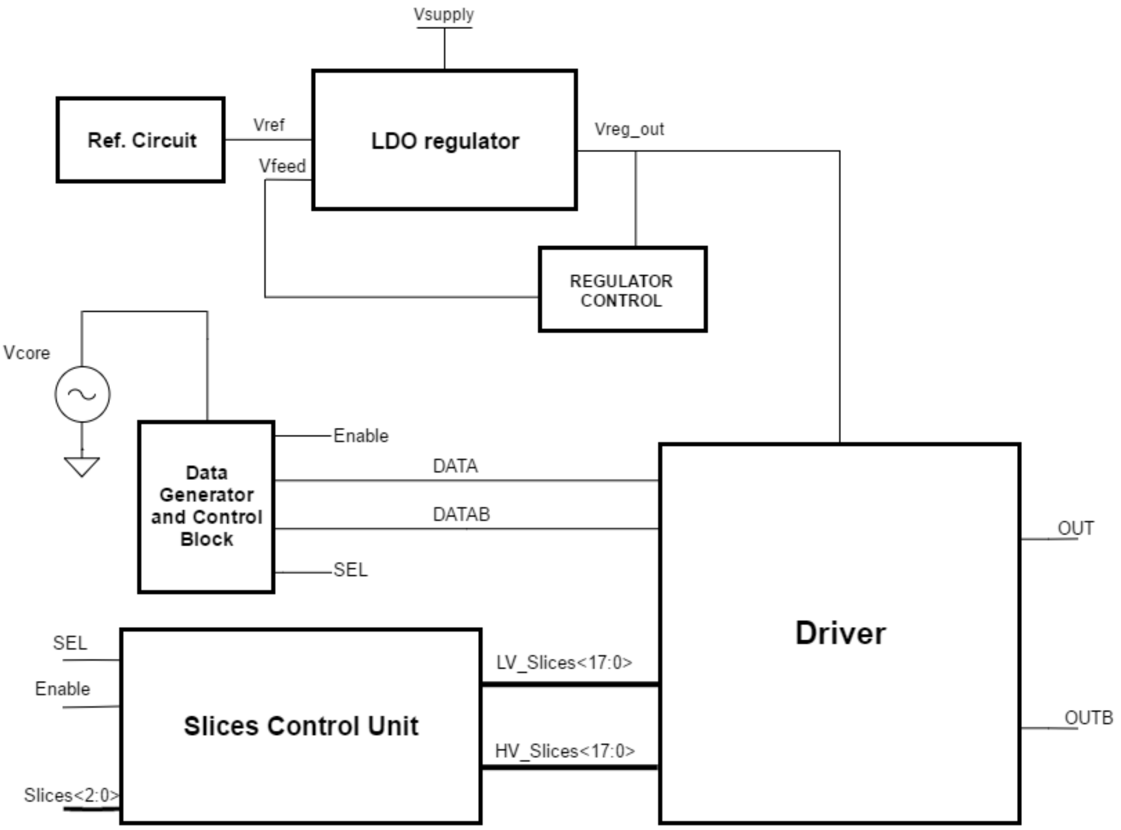


Figure 3.1: Full Block Diagram

voltage to compare with, and so the *Reference Circuit* function is to provide a fixed voltage for the Regulator. The regulator output is controlled by a net of resistors that with different ratios act as a voltage divider, pushing the output of the Regulator to $1V$ or to $1.6V$.

When developing this work, it's necessary to consider that the DATA we want to transmit isn't ready for transmission. The *Data Generator and Control Block* aggregates the circuitry necessary to treat the digital DATA signals and shift them to the necessary voltage levels. This block has still another function that is to generate the control signals *Enable* and *SEL*. This two signals are behind all the logic presented in this work, being the first responsible for turning ON or OFF the Driver and the other to chose the mode of operation (High Voltage or Low Voltage).

The *Slice Control Unit* is the block responsible for creating the BUS signals that control the number of slices active for a perfect impedance match to the line. This block outputs a common signal for the low voltage path and for the high voltage. Further explanation about this work will be given in 3.0.6.

For last, the *Driver Block* is the driving block that will be responsible for the impedance match to the line. This block is constituted by some switches, a *Pre-Driver* circuit and a *Output Driver*

Circuit.

Now that a general explanation about the blocks was given, we will descend in the hierarchy of the project and look deeper into the design. The following image 3.2 adds some nets and shows the logic behind the architecture for the regulator and the driver blocks:

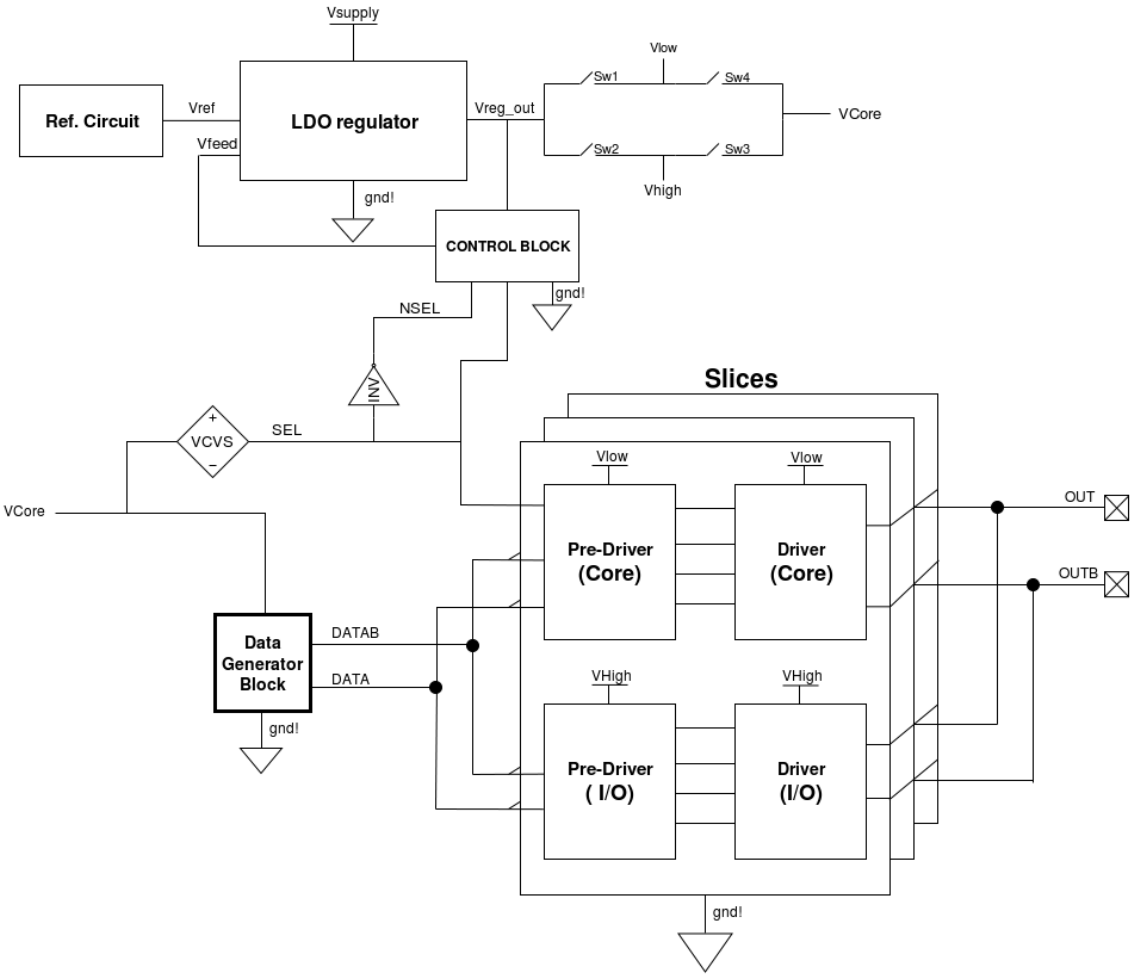


Figure 3.2: Top half architecture of the Driver

In 3.2 it's possible to observe:

- The regulator group where the *LDO Regulator*,*Reference Circuit* and the *Regulator Control* blocks are included.
- The *Driver* block content.
- And the *Data Generator Block* which is constituted by *Level shifters* and *Inverter* components.

In 3.2 it's shown that the *Output Driver* is constituted by slices each one containing a low power path that uses Core Devices and one high voltage path, containing I/O devices. Every slice is equal to the other seventeen. This two technologies, Core and I/O differ in supply voltage, with Core devices working with supplies below 1.1V while I/O being high voltage works with much higher voltages such as 2.5V or 3.3V. This two different paths never are active at the same time, meaning only one path drives the DATA and DATAB signals through the driver to the impedance line (OUT and OUTB pins). The lower voltage path is designed only for transmissions of 1V or less with the Core devices. If any voltage superior to this value is fed to this path, the transistors will burn. The high voltage path is used for transmissions of 1.6V and is designed with I/O . Note that Core devices can't handle higher voltages than 1.1V but it's important to not forget that high voltage devices even if used in low supply can handle high voltage. This will be done later on.

For good signaling like it was spoke in the previous chapters, the line impedance of 100 Ω must match the *Driver* termination impedance. It isn't enough to build a model for typical conditions with perfect impedance match because with PVT variations the components value can change as much as 10% or 20% and the circuit may not work at all. For example, the resistors value after fabrication has 20% tolerance which means that if we want to integrate a resistor with 750 Ω, its value can be anywhere between 600 Ω and 900 Ω. This uncertainty will be fought in this work through the use of multiple slices that we can activate to bring the error down. For a eighteen slice scheme, keeping in mind we want optimal fifteen slices working with 1500 Ω impedance (750 Ω differential), if the resistors used had -20% value meaning it's true value (differential) is 600Ω what we can do is activate less slices than fifteen in order to bring the termination impedance of the driver up to 50 Ω (differential). Meaning for this exact case that we can activate only 12 slices for a match at 50 Ω differential. The same happens when the resistors value is higher than the desired. When the single slice impedance value happens to be close to 900 Ω, we can activate 18 slices for a very good match. This can be done to fight the tolerance because after fabrication the resistors can be anywhere in the range of -20% up to 20% nominal value but all the resistors fabricated together will be equal, meaning if one has +20% error, all will have +20% error.

Both low and high voltage paths are identical in architecture but different in sizing and as explained above, in the technology of the transistors used. The driving path is constituted by a block called *Pre-Driver* and a block called *Driver*. To better understand the nature of these blocks, let's look at what the *DATA* and *DATAB* signals are. These signals are generated from an external voltage source that is usually integrated by the manufacture in the chip through a digital block. We will call it *Vcore*. This block is named *Data Generator Block* in 3.2 and isn't supposed to be designed in this thesis. For this work it will be considered that the starting point is after this, with the differential signals at the input of the driver *DATA and DATAB* and so, for a digital input we have to attack it with logic gates. Rarely a digital signal is attacked by an analog circuit.

Looking at 3.2 it's possible to see that each slice as two paths, and two *Pre-Drivers*, one for each path. The *Pre-Driver* main function is to enable or disable the respective path when the other one is operating. The control signals used for this purpose are provided by the *Slices Control Unit*. In resume, the logic behind this architecture works by these sequence: the *Driver* is controlled by the *Pre-Driver* which is controlled by $LVSlices < 17:0 >$ and $HVSlices < 17:0 >$. This last two signals are the outputs from *Slices Control Unit* that has the *SEL* control signal as input. This *SEL* signal besides being responsible for defining what path will be active it also controls the output of the regulator for a concordant logic.

Further explanation about the design and implementation of the *Pre-Driver* block is given in subsection 3.0.7.

Now, the *Driver* shown in 2.6 is the block that allows good line impedance matching while driving the differential *DATA* and *DATAB* signals. This block, acts like a two branch switch scheme that alternates between one branch and another to conduct current through it. The ohm law says that:

$$R = V * I \tag{3.1}$$

If the Driver impedance is fixed because it needs to match the line impedance, the amount of current through the Driver will define the output swing of the circuit. For different outputs (programmable) changing the supplies (Vhigh and Vlow) of this output stage will vary the current and so the output swing. This let us conclude that the key of achieving different and programmable voltage swing is regulating the supply of this block. Further understanding about the *Driver* block will be given in section 3.0.8.

In chapter 2 it's explained that the use of a regulator it's a matter of some complexity and in 3.0.2 a detailed lecture about the regulator is to be given in a way it's possible for the reader to get a better comprehension about the subject. For now, the reason why a *LDO regulator* was used instead of a switch converter or any other regulation technique is set by the application in this work and the advantages that Linear Regulators bring over switching converters for the desired application. Switching regulators are highly efficient and able to step up (boost), step down (buck), and invert voltages with ease. However, there is a need for an alternative because switching regulators have some weaknesses.First, the complexity of the design which takes a lot of design effort to get a new product working properly. Second, the level of integration of contemporary switching regulators doesn't come cheaply and increases the chip size, as mentioned in 2 some topologies have inductive components. Finally, the high frequency switching brings noise to the output. Voltage and current ripple at the input and output filters, generated by high-frequency operation, can be a major issue for a design using a switching regulator. And while the problems can be tackled, it takes time and design skill to do so. Linear regulators address all the key weaknesses of the switching type.

For once, in this work we want the *Driver* to be able to transmit at 1.6V, 1V and less than 1V, so for our regulator there's only need to be able to step down the voltage from an external source like 2.5V or 3.3V, so no need for a regulator that steps up. The low drop-out that we desired is best suited for a linear regulator where it's efficiency is high if the difference between input and output voltages is small, instead of a switching regulator that when low load currents are used, the switch-mode quiescent current is usually higher than in the first one. The last and maybe the most important reason why a LDO regulator was preferred was due to the ripple/noise being low in contrast to the switching regulator that presents high ripple due switching rate [16]. Further matter about the regulator subject is presented in the next subsection 3.0.2.

3.0.1 Impedance Control Block

The *Impedance Control Block* is controlled by the input control signal "SEL" and it's the block responsible for defining if the regulator is going to output 1.6V or 1V. If the "SEL" signal is high, then that means the regulator needs to output 1.6V. Likewise, if the "SEL" signal is low, the regulator will regulate the voltage to 1V.

Impedance Control Block acts as a voltage divider. It was implemented as three resistors in a way the output voltage of the voltage divider always equals the reference of 0.8V. As shown in 3.3 two switches controlled by the control signal "SEL" will determine what resistor , R1=2.5k Ω or R2=10k Ω is in series with the third resistor R3 = 10kΩ. One path leads to a voltage division of 0.5 and the other of 0.8. The switches in this work were all implemented by a pass gate topology, where PMOS source is connected to NMOS source and the same for both drains. This switches are each controlled by two Bias voltages applied to the PMOS and NMOS gates, when PMOS gate is low and MOS gate is high, the switch is closed (conducting).

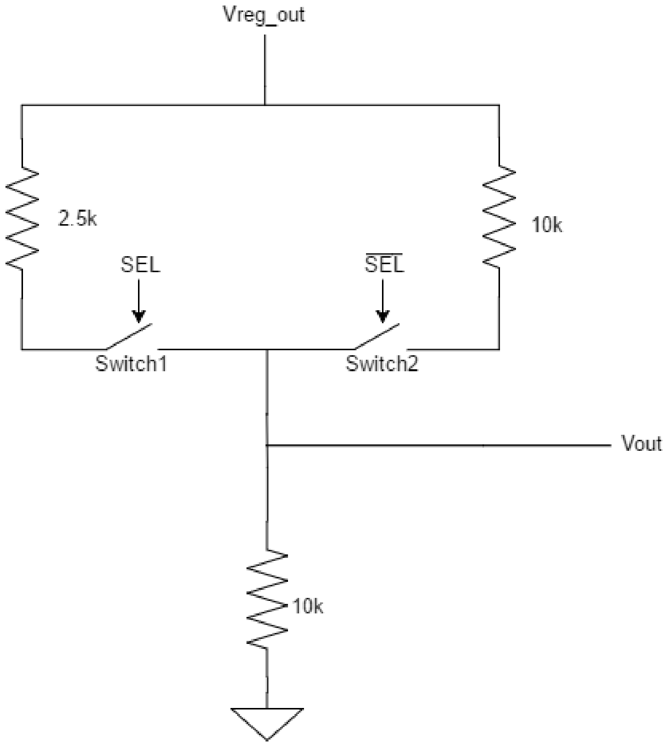


Figure 3.3: Impedance Control Block architecture

This element should have an equivalent resistance big enough in order to limit the quiescent current. And so, for a very small power dissipation the resistors values were chosen using the Ohm law 3.1. The values chosen for this work are presented in 3.1:

Table 3.1: Impedance Control Block resistors and voltage output relation

Voltage(V)	R1	R2
1.6	10k Ω	10k Ω
1	2.5k Ω	10k Ω

3.0.2 Regulator

In this section, a detailed explanation about the implementation of the LDO regulator is given. To better understand the proposed regulator, the following block diagram is shown:

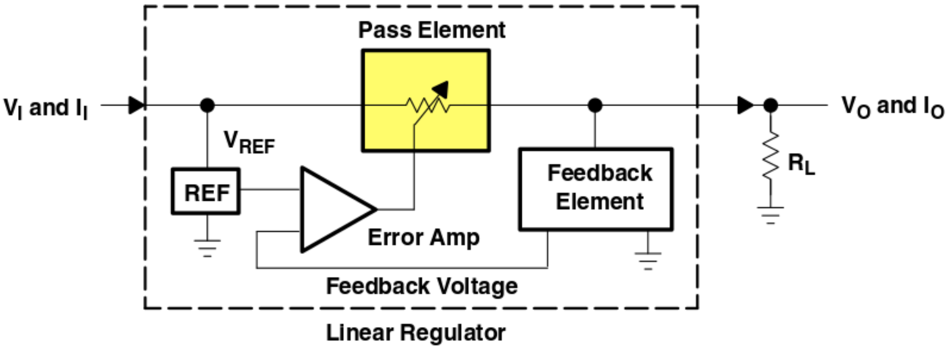


Figure 3.4: Low Drop Out Regulator

In 3.4 we have four main blocks or elements: *Voltage Reference Block*, *Error Amplifier*, *Feedback Element* and *Pass Element*.

3.0.3 Error Amplifier Block

This block is one of the major parts of this thesis, being a subject of deep study in a previous stage of this work, and then implemented in the LDO regulator. The *Error Amplifier* was designed as a two-stage differential amplifier that compares the reference voltage discussed in 3.0.4 to the feedback voltage and if the two voltages differ in value, the error amplifier will try to compensate that difference by changing the voltage at the gate of the *Pass Element*. This *Pass Element* is itself a stage with gain, more specifically a common source stage amplifier.

For the first stage of the amp-op, the differential pair, in CMOS technology there are two possible architectures: The differential NMOS pair and the differential PMOS pair.When the input differential pair is designed with NMOS, the load is designed with PMOS that can assume a diode connected configuration 3.6 or be designed to act as load and work in the saturation region having a voltage bias 3.5 or even be design like an active-load by using a current mirror configuration3.7. When the input differential pair is designed with PMOS and not NMOS, the NMOS load can assume all the the configurations PMOS did in the previous case. All topologies are used in different applications, having different gains for the stage with different PMOS or NMOS load configurations. We'll discuss briefly each one and what configuration is best suited for this work after analyzing how much gain is needed for our error amplifier.

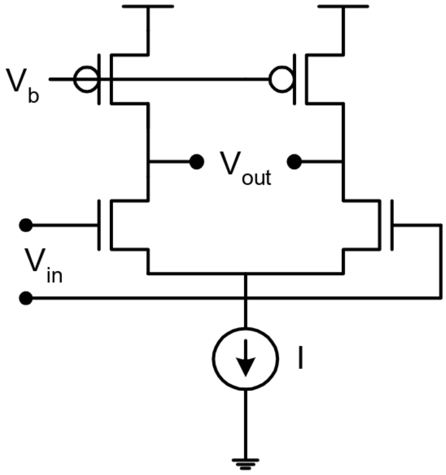


Figure 3.5: NMOS Differential Pair with PMOS load

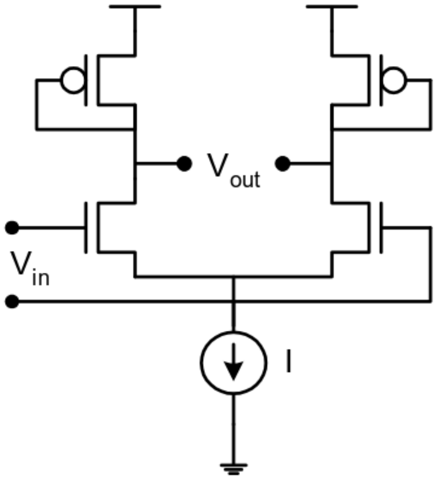


Figure 3.6: NMOS Differential Pair with PMOS diode connected

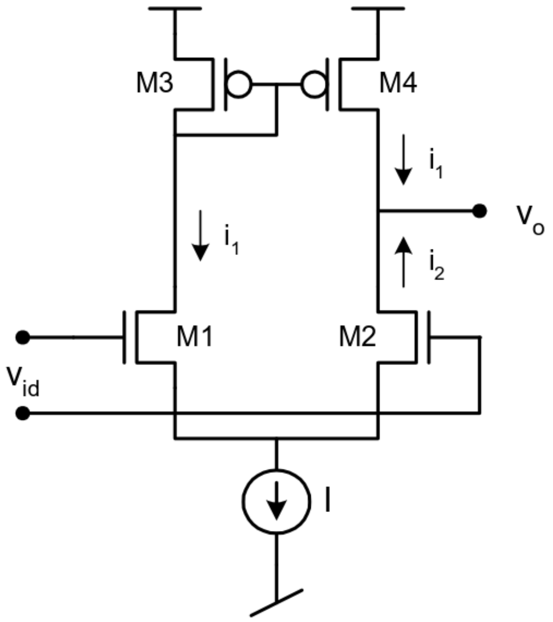


Figure 3.7: NMOS Differential Pair with PMOS current mirror

So, first we need to know how many stages are needed for the amplifier meaning how much open-loop gain is desired for the amplifier to have. For this, the desired error of the amplifier needs to be analysed. For a tolerance of 2% and with the regulator set to 1.6V, it can output aa value in the range of 1.568V up to 1.632V which results in the feedback voltage ranging from 0.784V to 0.816V. If the reference is said to be 0.8V , the error amplifier will sense a 0.016V difference at the inputs. That is how much we want the amplifier to be able to sense and to correct. Like it was said in this section, the output of the error amplifier will control the voltage of the *Pass Element* and as so, for an external supply of 2.5V, for the PMOS to work it has to have at least 0.5V of VGS. Then the output of the error amplifier must be able to reach 2V:

$$V_{out} = 2V \tag{3.2}$$

And so, if the output of an amplifier is it's input multiplied by a GAIN:

$$V_{out} = A_V * V_{in} \tag{3.3}$$

,with Vin being the voltage difference of 0.016V, Av the open loop gain of the error amplifier and Vo = 2V ,

$$Av = 2/0.016 = 125 = 41.9dB \tag{3.4}$$

For 41.9 dB open loop gain, one stage only isn't enough to reach this high gain because the stage of the differential stage (Av1) is equal to:

$$Av1 = gm2 * (ro2//ro4) \tag{3.5}$$

,with gm2 being equal to twice the current through the transistor to be divided by *Veff*

$$\begin{aligned} gm2 &= \frac{2*ID}{V_{eff}} \\ &= \frac{2*ID}{(V_{gs} - Vt)} \end{aligned} \tag{3.6}$$

From 3.6 we can observe that for our low power regulator that will operate with currents close to 100 μ or 200 μ and with V_{eff} close to 100mV, what happens is that the total gain of one stage is somewhere near:

$$gm2 = 2 * 200/0.1 = 2mS \tag{3.7}$$

$$Av1 = 2m * 10K = 20 = 26dB \tag{3.8}$$

, ro4 and ro2 values were determined in the range of 6kΩ 10kΩ by size limit of these transistors that can't be huge because both low power operation is desired and for high speeds, if these transistors were huge, the switching speed would decay. When it comes to Mosfet transistors, NMOS and PMOS have some different characteristics that are important to understand to deal with these non-ideal devices and electrical effects. For once, the threshold voltage which is the gate voltage at which conduction (current flows through transistor) takes place is different for NMOS and PMOS, and even less similar when we are testing the circuit for not typical conditions (fast-fast,slow-slow,fast-slow and slow-fast) for validation of the design. So, we can conclude that a second stage is needed to boost the overall GAIN (A_V). The pass element in the design of the regulator has approximately one voltage gain and so it won't help us at improving the amplifier overall gain. So the architecture of the amplifier needs to be a two-stage amplifier. The open loop gain of the regulator (*GAIN*) is the gain of the first stage (A_{v1}) multiplied by the second stage gain (A_{v2})* third stage gain 1*V/V*:

$$\begin{aligned} Vout &= A_V * V_{in} \\ &= A_{v1} * A_{v1} * V_{in} \end{aligned} \tag{3.9}$$

From the above conclusion, we know now that we want a differential input differential output configuration for the first stage and a differential input single ended-output for the second stage because the output of the error amplifier will be single ended (controls the gate voltage of the pass element). Between the architectures 3.5 and 3.6 the gain calculus changes. For a diode connected PMOS:

$$\begin{aligned} Av1 &= -g_{mN}(\frac{1}{g_{mP}}||r_{oN}||r_{op}) \\ &= -\frac{g_{mN}}{g_{mP}} \end{aligned} \tag{3.10}$$

This means we adjust the gain of this first stage for this first configuration by the relation of transconductances of NMOS and PMOS. Another solution is to use the configuration in 3.5 and so, the gain changes to the relation of the equivalent resistances of the MOSFETs:

$$Av_1 = -g_{mN}(r_{oN}||r_{oP}) \tag{3.11}$$

Now, the only thing that is left is to define if the differential pair is designed with NMOS or PMOS. During the development of this work, the amplifier changed to a PMOS input with NMOS load, because the margins were too short for all the transistors to be in the saturation area. To understand better this change, let's first discuss the architecture of the second stage of this amplifier.

Like it was discussed previously, the overall gain (AV) is the combination of A_{v1} and A_{v2} and so, it's not difficult to achieve the desired gain of 42dB since the stages gain now multiply. For this second stage it's also desired to have a differential input and a single ended output that will attack the last stage (Pass element) that in fact is a common source stage. This means, this common source stage will represent an infinite load for our second stage and so, the best solution found for our application was the use of a folded cascode current mirror for it's advantage in gain when the load he sees is high. The folded cascode configuration solves the problem with margins/low voltage headroom the cascode mirror has and this architecture has simple biasing too, only being needed to generate some simple references to keep the transistors in saturation.

Following this second stage, the third stage is constituted by the pass device, a PMOS transistor. Since the PMOS pass element is a voltage-driven device as opposed to a current-driven device (like a PNP transistor), the quiescent current is very low and remains constant and independent of output loading over the entire range of output load current. By varying the gate voltage of this device, its output resistance R_o changes and so, the element acts as a variable resistance for stability. The size of this device depends on the load current the amplifier needs to provide.
This stage voltage gain formula is the following:

$$\begin{aligned} Av_3 &= \frac{g_m R_S}{g_m R_S + 1} \\ &= 1\frac{V}{V} \end{aligned} \tag{3.12}$$

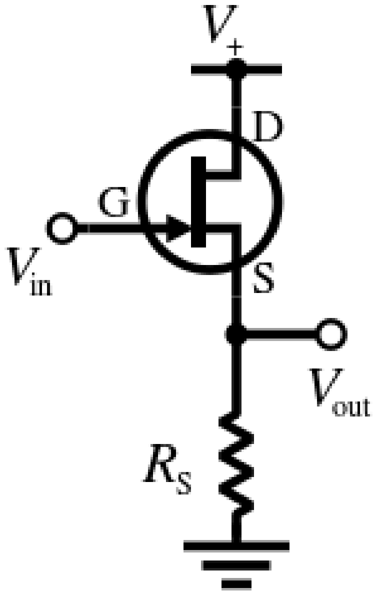


Figure 3.8: Pass Element Stage of LDO Regulator

In the next chapter 4 all the currents for the *Driver* are presented but for this work, the load current of the regulator is maximum at 100mA and so, the following calculus were done for the size of the transistor:

$$I_D = k * \frac{W}{L}((V_{GS} - V_T)V_{ds} - 0.5 * V_{ds}^2) \tag{3.13}$$

Having the equation for the current through the MOSFET in 3.13 we need to do some constraints in order to extract the $\frac{W}{L}$. First, we can understand that for a fixed I_D maximum and setting L parameter to the minimum possible which is 30nm for high voltage devices in this technology, the worst case scenario, where W maximum size is calculated is when $V_{GS} - V_T$ is minimum. This happens for a V_{GS} of 0.5V where the transistor almost falls off the saturation. L parameter is set to minimum because it isn't desired to increase this transistor threshold voltage which is set closely to 0.45V for the described minimum L.

3.0.4 Reference Circuit

The reference circuit was not subject of implementation in this dissertation, instead it is assumed that a high precision voltage reference is available for use for the *Error Amplifier Block* with the value of 0.8V. However, a brief explanation about how this reference voltage could be generated is to be done and the basics of selecting a reference. Typically what is desired from a reference is that it's output is independent of process parameters and loading in a way we can always assume that it's value is close to ideal. A good candidate that can output a value in this conditions is a full CMOS process *Bandgap reference circuit* [17] as shown in fig 3.9:

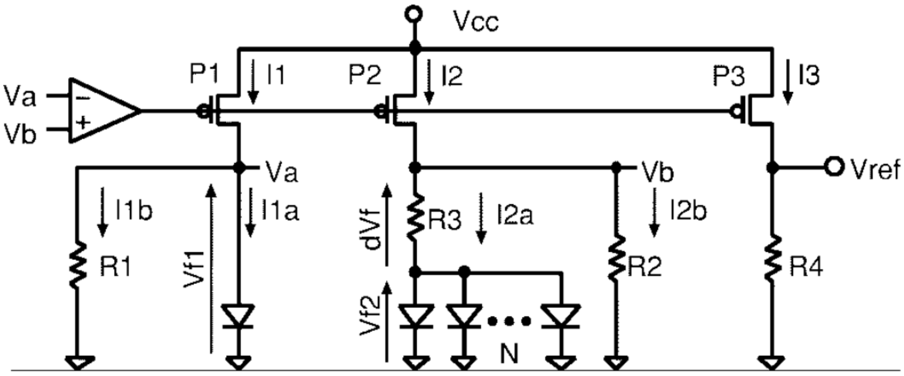


Figure 3.9: Bandgap Reference Circuit

In figure 3.9 a possible solution for the *REF* block is presented. This circuit represents a possible implementation of a bandgap voltage reference in which the temperature, power supply variation and circuit loading don't interfere (or little) with the output voltage of this circuit, giving us a reliable reference for our LDO regulator.

The concept of the proposed BGR is that two currents, which are proportional to *Vf* and *Vt*, are generated by only one feedback loop. PMOS transistor dimensions of *p1*, *p2*, and *p3* are the same, and the resistance of *R1* and *R2* is the same. By implementing this circuit, *Vref* for the proposed solution is determined by the resistance ratio of *R2*, *R3* and *R4* and not so much by the absolute value of the resistance. Which as desired, gives us a stable reference to use.

3.0.5 Level Shifters

The digital signals *DATA* and *DATAB* are usually generated from the *VCore* source. This digital signals have two-states, high meaning 1V and low equals 0V. This digital signals need to be attacked by digital ports that transform them into analog signals and may or not need to be shifted to another voltage domain meaning, higher or lower amplitude than 1V may be needed. This is something that happens in this work, since we want to transmit not only at 1V but too at 1.6V. For this, the information signals are attacked by level shifters in order to transform them into the inputs that we need. A level shifter cell is used to shift a signal voltage range from one voltage domain to another. For the low voltage path, no level shifting needs to be done and so a simple inverter is sufficient. Usually, level shifting from a high input voltage to a lower output voltage doesn't bring any problems, the difficulty is to shift the output signal of the level shifter to a higher value than the input voltage. For this, a circuit likewise the one on 3.10 can be used:

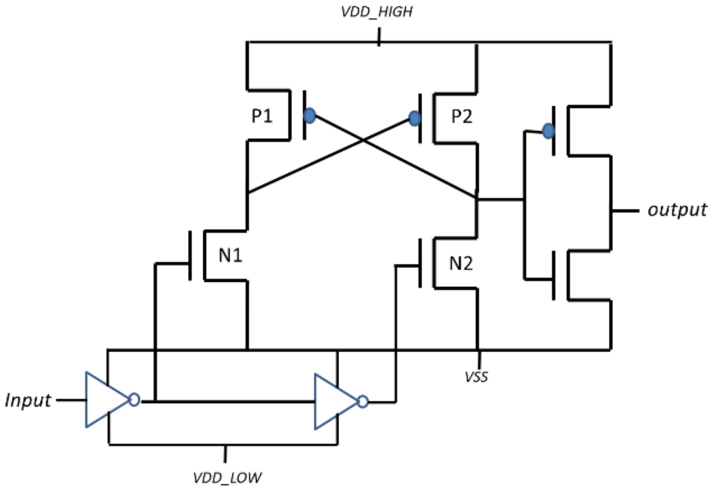


Figure 3.10: Level Shifter Low-to-High

In this topology a cross coupled transistor amplifier architecture is used to amplify the low voltage signal. The size of the four transistors are chosen such that the Level Shifter circuit operates reliably for the desired input and output voltage levels. Disadvantages of this circuit are related to power consumption and delay increase. For this dissertation, because the focus of the work was to develop a *Driver* and not to build a voltage source or the information to transmit, voltage controlled voltage sources were used provided by a library from *Synopsys*. This VCVS had mathematical expressions on the test bench for its gain, depending on what transmission was selected (SEL signal high or low).

3.0.6 Slices Control Unit

Slices Control Unit outputs a common signal for both high and low voltage paths that controls the amount of slices active in the *Driver*. If $LVSlices < 17:0 >$ is high for 15 slices for example, $HVSlices < 17:0 >$ needs to be low. Only one path can be active at the same time for the circuit to work. This two signals $LVSlices < 17:0 >$ and $HVSlices < 17:0 >$ are dependent of its inputs: $Slices < 2:0 >$, *Enable* and *SEL*. This inputs functions are as follows: the $Slices < 2:0 >$ defines how many slices are active between twelve slices (minimum) and eighteen slices (maximum). *SEL* defines which output is high and what output is low by controlling the supply of the logic gates in the *Slices Control Unit*. It also controls the supply of the *Pre-Driver* and *Driver* for each path. Finally, when signal *Enable* is high the circuit works (Driver Mode ON) and when is low, it forces the outputs to low (Driver Mode OFF). The following image 3.11 shows the overall logic of this half circuit.

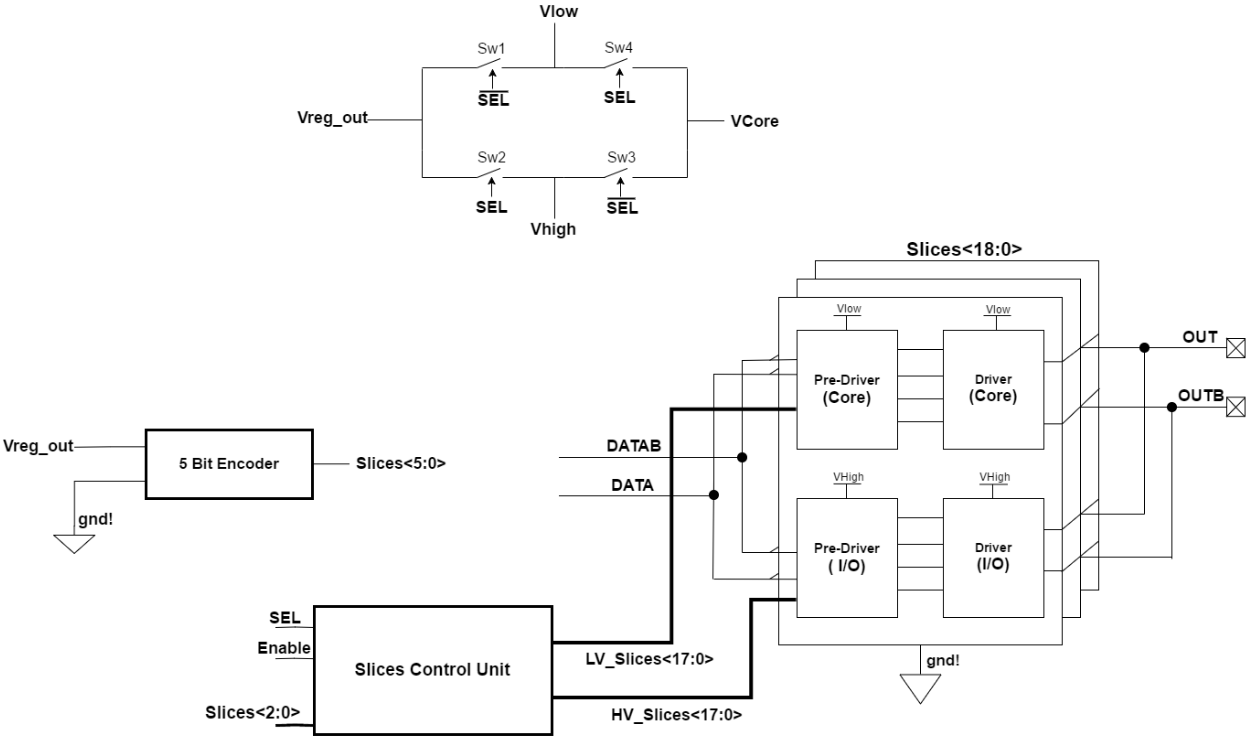


Figure 3.11: Bottom Half Circuit Block Diagram

The output signals $LVSlices < 17:0 >$ and $HVSlices < 17:0 >$ are similar in function and both control eighteen slices each. The number of slices active at one moment, depends of the hexadecimal value of the $Slices < 2:0 >$ input. From the eighteen slices, the minimum number that can be left active is twelve. Particularly, if $Slices < 2:0 >$ value is hexadecimal one, twelve plus one slices will be active. For a hexadecimal three, twelve plus three slices will be active. Following this logic, the following table was built in order to help us build the *Slices Control Unit* block.

Table 3.2: Relation Between $Slices < 2:0 >$ and Number of Slices active

$Slices < 2 >$	$Slices < 1 >$	$Slices < 0 >$	Number of Slices
0	0	0	12
0	0	1	13
0	1	0	14
0	1	1	15
1	0	0	16
1	0	1	17
1	1	0	18

To implement this logic, each signal from the outputs needs to be controlled individually by signal $Slices < 2:0 >$. This control was done using NAND gates with three inputs, NOR gates, OR gates, among others. The following picture shows the design of this circuit.

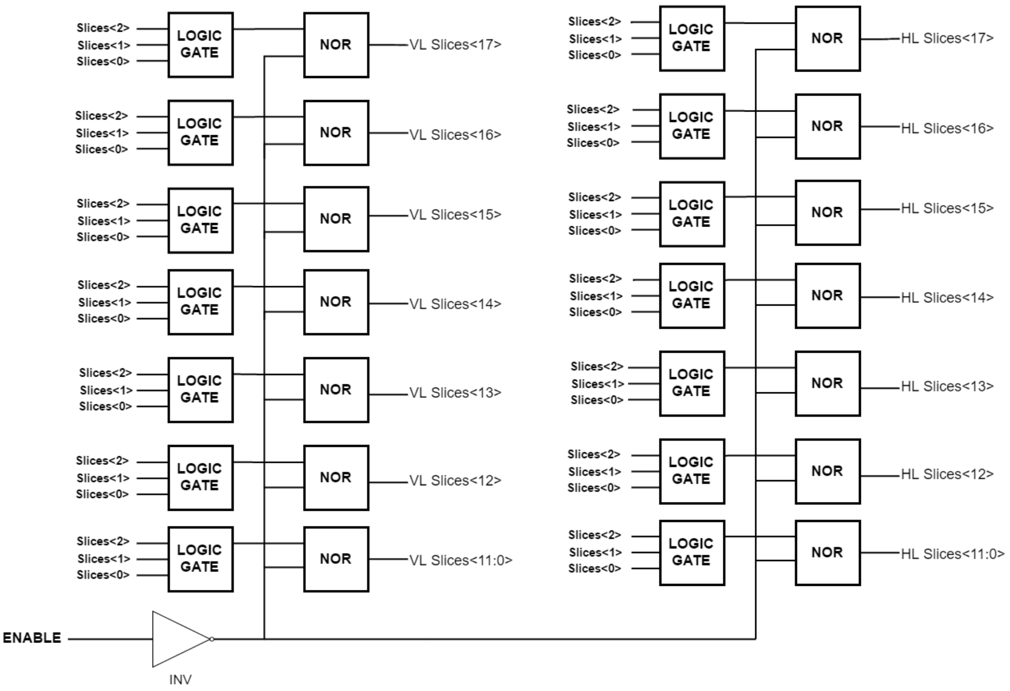


Figure 3.12: Slices Control Unit Architecture

As presented in 3.12, there are two different logic columns, one for the low voltage output and one for the high voltage. The activation of one path over the other is done by signal *SEL*. This signal is connected to the supply of all the *LOGIC GATES* in the high voltage column, and to the supply of all the *NOR* gates of both columns. The *SEL* signal like the *Enable* goes trough an inverter and the result is connected to the low voltage output. Meaning that if *SEL* signal is "low", opposite is high, and the low voltage column is still being supplied in order to work. Continuing with the scenario where the low voltage transmission is desired, we want the output signal $LVSlices < 17:0 >$ to be high according to the $Slices < 2:0 >$ word. If the output is connected to a NOR gate, the inputs of this gate both need to be low. The *Enable* signal will be high, but because the connection to the *NOR* is done after the inverter, this input is assured. The other input will be "low" since the *Logic Gates* are designed to output "low" when the desired $Slices < 2:0 >$ word is selected.

Truth tables were designed to help in the process of designing theses gates and are shown below. *Logic Gate* number one will correspond to the output $LVSlices < 11 : 0 >$,*Logic Gate* number two will correspond to the output $LVSlices < 12 >$ and so on, until *Logic Gate* number seven will correspond to the output $LVSlices < 17 >$.

A minimum of twelve slices are always activated and so it is independent of the control word. So this first Logic gate isn't really a gate but the opposite from *Enable* signal connected two times to the NOR from this output.

Table 3.3: Truth Table For Logic Gate 2

$Slices < 2 >$	$Slices < 1 >$	$Slices < 0 >$	Output
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0

From the above table what we can conclude is that when one of the three signals is high, the logic gate needs to output low. So the first logic gate is an *OR* gate. When we want to activate an extra slice, the control word will be 001.Below there will be a description for each logic gate developed to implement the logic for this block.

Table 3.4: Truth Table For Logic Gate 3

$Slices < 2 >$	$Slices < 1 >$	$Slices < 0 >$	Output
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0

For a control word of 010 or two in hexadecimal, we want this logic gate to be conducting when thirteen or more slices are needed. So, this time, looking to the truth table 3.4 is possible to observe that all the control words above the thirteen slices have $Slices < 2 >$ and $Slices < 1 >$ signals different from zero. This is convenient because we this we just need to build a simple two input NOR that outputs zero only when a high is presented at its input. The third logical gate is then a two input NOR.

Table 3.5: Truth Table For Logic Gate 4

$Slices<2>$	$Slices<1>$	$Slices<0>$	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0

For this next truth table, simple inspection won't let's us conclude about the logic gate needed. So, a Karnaugh map was built and the mathematical expression achieved was :

$$\overline{Slices<2>*Slices<1>+Slices<2>} \tag{3.14}$$

Implementation of this expression results in a AND gate with inputs Slices<2> and Slices<1>, and the result of this AND is the input to a NOR gate with Slices<2> again. In figure 3.13 the previous explanation is shown in a visual manner:

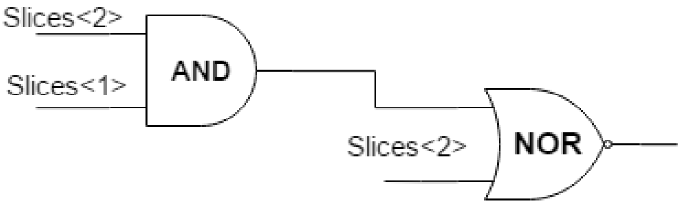


Figure 3.13: Logic Gate 4

The next logic gate is a simple inverter with $Slices<2>$ as input because from the truth table is possible to observe that this slice, the sixteen, will always be active for control words with $Slices<2>$ high and be off for control words with $Slices<2>$ low.

Table 3.6: Truth Table For Logic Gate 5

$Slices<2>$	$Slices<1>$	$Slices<0>$	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0

When the control word is 101 or 110 we want the seventeen slice to be active, so again by the Karnaugh table was possible to come to the conclusion that the logic expression is:

$$\overline{Slices < 2 > *Slices < 0 > +Slices < 2 > *Slices < 1 >} \qquad (3.15)$$

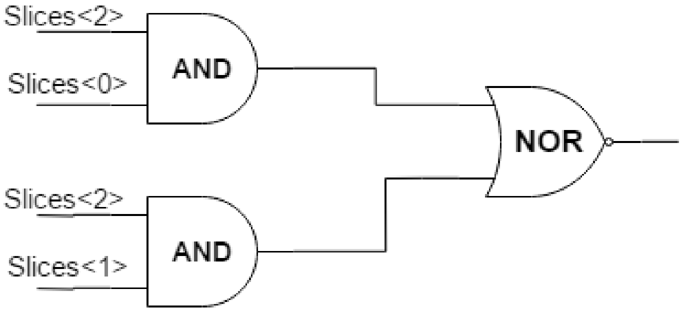


Figure 3.14: Logic Gate 5

For the final logic gate that only activates when the control word is 110 the logic function is show below and is implemented by a three input NAND.

$$\overline{Slices < 2 >} * Slices < 1 > *Slices < 0 > \qquad (3.16)$$

3.0.7 Pre-Driver

In this section the function and architecture of the *pre-driver* will be discussed. In this architecture there are two *pre-driver* blocks, one with high voltage devices and one with low voltage. This blocks are the control blocks that enable or disable the low and high voltage paths. When we want to transmit in low voltage, the pre-driver receives a signal "0" in the control input and disables the high voltage path. The same happens when we want to transmit in high voltage, the low voltage pre-driver will disable the low voltage path. In 3.15 it's possible to see the pre-driver design. It's constituted by four logic gates, each one drives the DATA and DATAB signals to the gates of the PMOS and NMOS of the driver. PMOS gates are driven by *OR* gates and the NMOS gates are driven by *AND* gates.

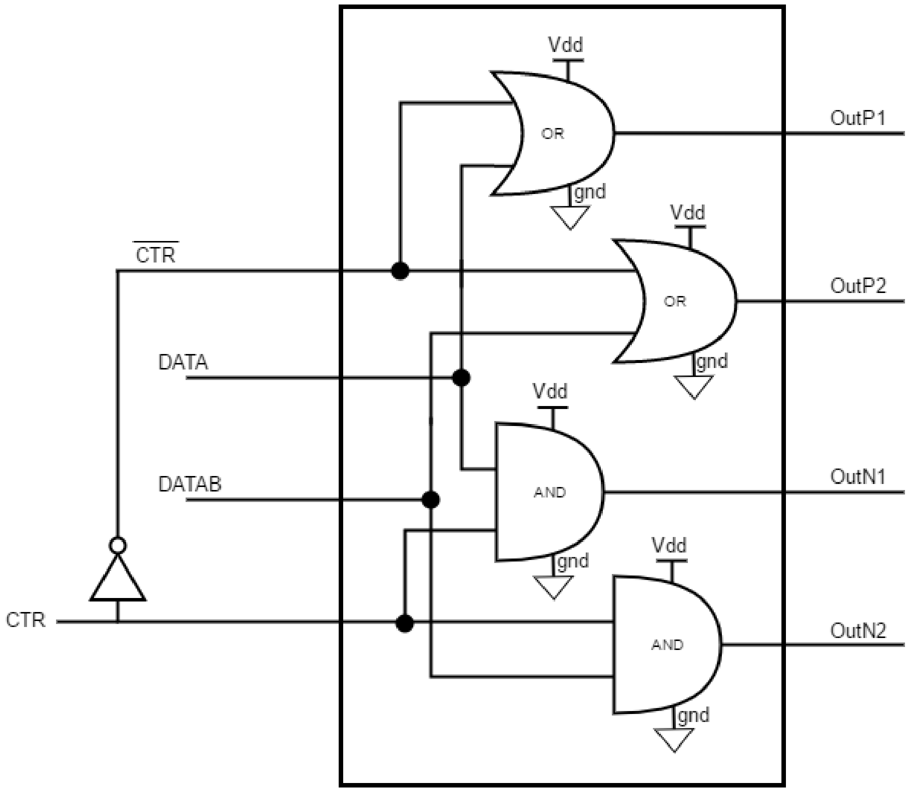


Figure 3.15: Pre-Driver architecture

The *CTR* signal is the control signal which tells the *Pre-driver* if he should drive the *output driver* or not. CTR signal connects to the *AND* gates and $\overline{CTR}$ connects to the *OR* gates. When *CTR* is high, the output of the *AND* gate is equal to the DATA signal, meaning it drives the signal to the *Output Driver*, more specifically the NMOS transistors gates. For the $\overline{CTR}$ signal, keeping in mind that it will drive a PMOS gate that conducts when connected to "low" and cuts when connected to "high", when *CTR* is low $\overline{CTR}$ is high and so the output at the *OR* is high too, cutting the transistor. When *CTR* is high $\overline{CTR}$ is low and so the output at the *OR* is equal to the DATA signal. So, in conclusion, the *Pre-driver* will drive when *CTR* is "high" and will cut the *Driver* when *CTR* is "low".

Table 3.7: Logic Function OR

CTR	$\overline{CTR}$	DATA	OUTPUT	STATE
0	1	0	1	driving
0	1	1	1	driving
1	0	0	0	turned-off
1	0	1	1	turned-off

Table 3.8: Logic Function AND

CTR	$\overline{CTR}$	DATA	OUTPUT	STATE
0	1	0	0	turned-off
0	1	1	0	turned-off
1	0	0	0	driving
1	0	1	1	driving

After being explained what the *Pre-driver* mode of operation, function and architecture should be, it's time to discuss the design at transistor level. OR and AND gates are simple in design but for implementation we first need to determine what capacitance will this block attack. This capacitance, is in fact the parasitic capacitances of the respective MOSFET that the *Pre-Driver* logic gates will be connected to. This parasitic capacitances are unwanted, but still are part of the transistor. Together with the resistances in the circuit, they put an upper limit to the speed of the transistor.

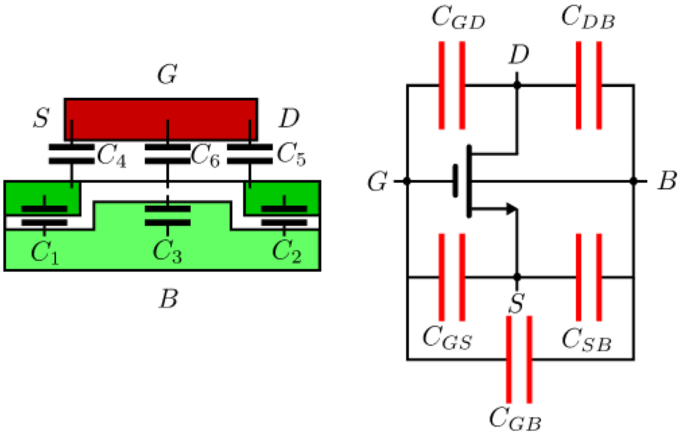


Figure 3.16: Parasitic Capacitances of a Transistor

In 3.16 *C1* and *C2* are capacitances created by the depletion regions between source/drain and bulk. *C3* is the depletion capacitance between the channel and bulk. *C4* and *C5* are capacitances caused by the overlap between the gate and the source/drain diffusions. Finally, *C6* is the oxide capacitance between gate and the channel and is split between drain and source depending on the region of operation of the transistor. What happens is that the capacitance seen by the logic gates in the gate of PMOS and NMOS is called C_G and is split between C_{GD} and C_{GS}.

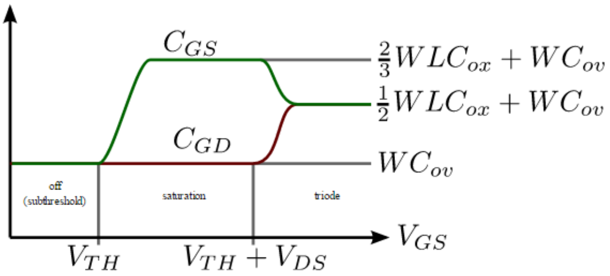


Figure 3.17: Gate Capacitance according to operation region

C_{GD} and C_{GS} have a base value of the overlap capacitance WC_{ov}. To that we add the gate to channel capacitance WLC_{ox} according to the region of operation. In 3.17 a resume about the formulas to calculate the gate capacitance are given. In subthreshold region, there is no gate-channel capacitance because there is no channel. In saturation, the channel is pinched-off and there is no gate-channel capacitance at the drain and only two-thirds go to the source. In triode, the channel is not pinched-off and the gate-channel capacitance is split equally between drain and source. The driver MOSFETs will operate in saturation or in cut-off, as the gate capacitance is bigger in the saturation region, this was the consideration made when sizing the *Pre-driver* because it's the worst case scenario.

The logic gates designed are constituted by two parts: first there's the circuit needed to implement the logic of the gate and second a two-stage inverter (buffer) that was found to be helpful for good signal integrity.

For the *OR* gate, what was done for this work was to use the common *NOR* configuration followed by an inverter to build the desired logic gate. Typically *Q1* and *Q2* are equal and four times bigger than *Q3* and *Q4*, which are equal in size. Further tuning was done to achieve an operation at 10Gbits/s with good signal integrity. The inverter was designed with a 2:1 ratio between PMOS width and NMOS width. In 3.18 the first part of the logic gate is illustrated.

The same concept was applied to the *AND* gate. Here, all the transistors are sized equal 2:2 ratio, because that is known to be the configuration that better balances the driven signal at the output. In 3.19 the first part of the logic gate is illustrated.

If we were to connect two inverter gates together so that the output of one fed into the input of another, the two inversion functions would "cancel" each other out so that there would be no inversion from input to final output. Gate circuits are signal amplifiers, regardless of what logic function they may perform. A weak signal source (one that is not capable of sourcing or sinking very much current to a load) may be boosted by two inverters like the pair shown in the previous illustration. The logic level is unchanged, but the full current-sourcing or sinking capabilities of the final inverter are available to drive a load resistance. This concept was further implemented in

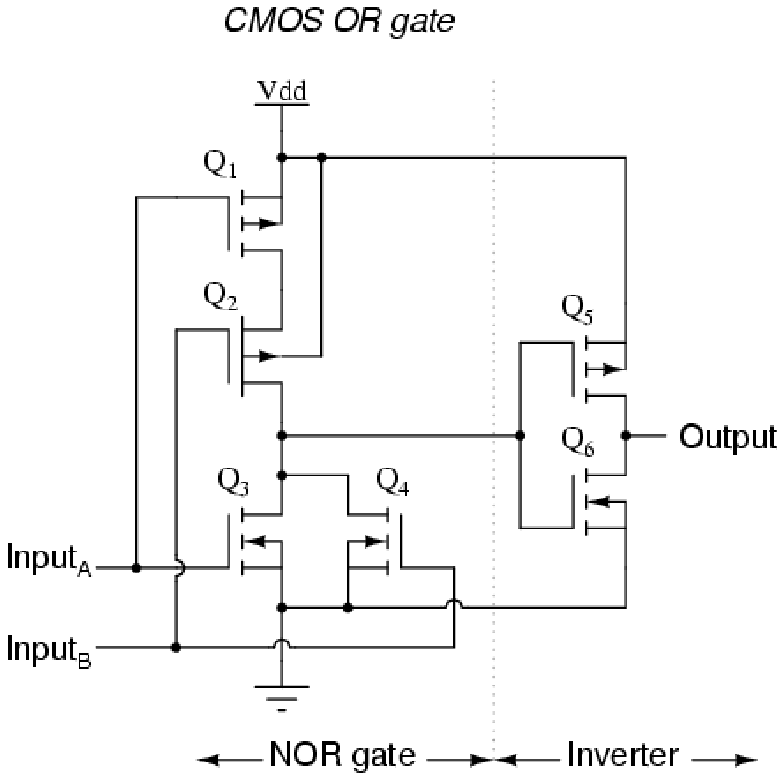


Figure 3.18: Full CMOS OR gate

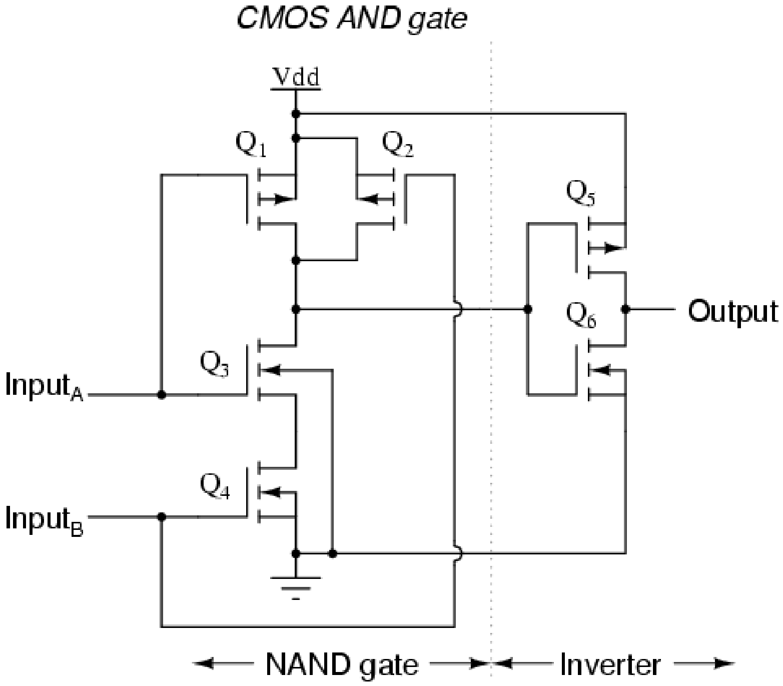


Figure 3.19: Full CMOS AND gate

the high-voltage path than in the low-voltage one, were it was used two-pairs of inverters instead of just one pair to further boost the signaling. What happens is that if we use more inverters we can scale down the size of the devices in each stage, because the previous stage only sees the gate capacitance of the following stage. With this technique, by implementing two pairs of inverters it was possible to achieve small size high voltage transistors that could switch fast enough for our 10Gbit transmission. In 3.20 and 3.21 the architecture of the logic gates of the *Pre-Driver* for the HV path is shown:

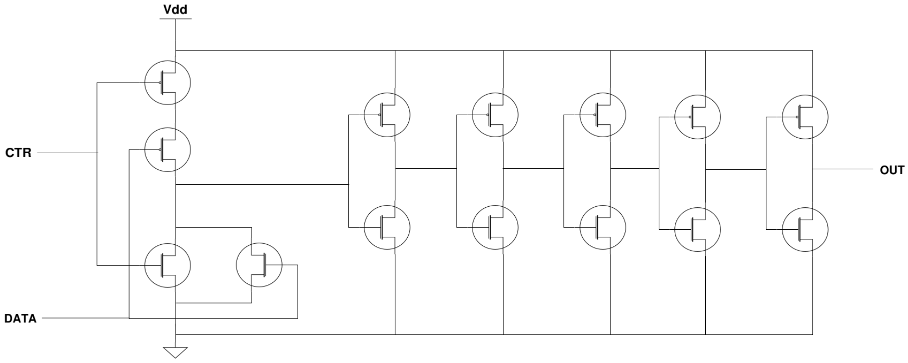


Figure 3.20: OR cell for HV path

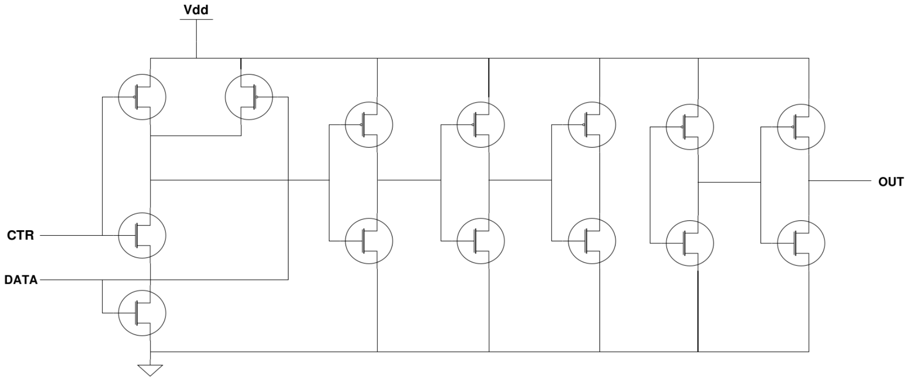


Figure 3.21: AND cell for HV path

3.0.8 Driver

In this section the last block which is the *Output Driver* will be discussed.

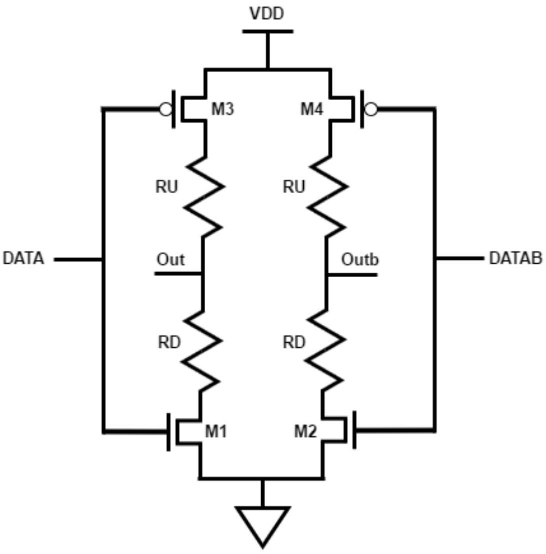


Figure 3.22: Output Stage Driver

The output stage of the driver is designed with two PMOS, two NMOS transistors, two pull up resistors and two pull down resistors. This differential stage alternates the current flow from one branch to the other with the input signals of DATA and DATAB. When DATA signal is "low" and DATAB signal is "high" the current flows through M2 and M3 transistors. When DATA signal is "high "and DATAB signal is "low " the current flows through M1 and M4 transistors.

For the sizing it is desired to keep the transistors size to the minimum since very high frequencies like 10Gbits/s the switching speed of the devices needs to be very high for good signal drive. Even so, the transistors need to have enough size in order to drive the current to the load. Let's start by looking at the expression of the current through the transistor. In saturation the current formula is:

$$I_D = \frac{1}{2} * k * \frac{W}{L}(V_{GS} - V_T)^2 \tag{3.17}$$

and so, by extracting the k and V_T parameters from the design KIT, and setting L to the minimum we can now determine the W for the desired current. For the low voltage path, the Driver will work at most with 1V supply and that means we have to dimension the transistors to be able to handle those currents. For a 1V transmission, the output swing has to be 500 mV. Since the load is 100Ω that means the current will be 5 mA. With 15 slices working, each slice must be able to drive 340 μA.

As we know well, typically for devices of the same technology, NMOS transistors present a higher resistance than it's PMOS counterpart. For this reason, when sizing the output transistors, one must respect the 2:1 ratio for this stage.

With the PMOS and NMOS sizes defined, it's necessary to look at the equivalent resistance of the devices because of the termination impedance of the driver matching the line impedance. For fifteen slices the impedance of each slice needs to be 1500Ω or 750Ω differential. So for a 100Ω equivalent resistance of the transistor, RD and RU will be 650Ω each.

Chapter 4

Results and discussion

In this chapter the results and conclusions of several simulations of the schematic of the *Driver* are explained and shown. This chapter will serve to validate the work developed.

As it was said in the previous chapters, the *Driver* needs to work under typical and not typical conditions specially. Non-typical conditions imply changes in the temperature, process and voltages used. These verifications made in varying conditions are typically named "corners". It is important to test corners, because if a design meets all requirements for all technology corners during simulation stage, the chance of meeting the requirements during chip test increases.

In this chapter, corners testing, results, explanation and conclusions about the *Driver* robustness are presented.

4.1 Circuit Power Consumption

In this section it will be briefly discussed the maximum currents and so the maximum power consumption of the circuit. The power consumption of the circuit is maximum when the high voltage path is ON and the low voltage path is off, the driver in this mode outputs 8mA for a 1.6V operation since the line impedance is 100 Ω. This is the output driver consumption, but we need to look at how much the *Pre-Driver* consumes, the other blocks and how much is the regulator providing. Measurements were made and tables 4.1 and 4.2 shows the power consumption of the circuit for both HV and LV. The full power consumption of the circuit is in the worst case (HV) 225 mW.

Table 4.1: Power Consumption of Driver for HV

Block	Current	Voltage	Power Consumption
Driver	8.6mA	2.5V	21.5 mW
Pre-Driver	80mA	2.5V	200 mW
Regulator	90mA	2.5V	225 mW

Table 4.2: Power Consumption of Driver for LV

Block	Current	Voltage	Power Consumption
Driver	5.3mA	2.5V	13 mW
Pre-Driver	10mA	2.5V	25 mW
Regulator	16mA	2.5V	40 mW

4.2 Data Transmission Results - Driver results

In this section, the *Driver* functionality will be tested under varying conditions such as temperature, voltage supplies and process variation. First, all integrated circuits respond differently in different temperature conditions. Some parameters are dependent on the temperature such as the Threshold Voltage V_{TH} or the mobility of electrons through the transistor, so the circuit performance will inevitable degrade for certain temperature ranges. Simulations were done for a scope of 0°C 50°C and high temperature 125°C and the circuit output in typical conditions is shown in 4.1. With the temperature increase, was possible to observe that the output degrades slightly.

In 4.1 the plot of the output in typical conditions is shown for the low voltage path and in 4.2 the eye diagram of both single ended outputs is shown. As it is possible to observe, the crossing of the signals isn't perfect but it's close to Vdd/2. The desired value for the V_{CM} was half the supply, 0.5V for this path, which wasn't possible to achieve by only optimizing the components size and parameters. Figure 4.1 let us conclude that for the presented architecture, a stable V_{CM} is only possible to achieve by using an additional circuit that employs negative feedback technique for V_{CM} compensation. Besides this, the *Driver* differential output achieves fast rising and falling edges even at 10Gbits, with the pretended value of swing $500mV$. The eye diagram shows very good opening with $520mV$ height and very good width with 49.7ps. This simulations were conducted for a typical scenario, but in the following sections the other corners are presented.

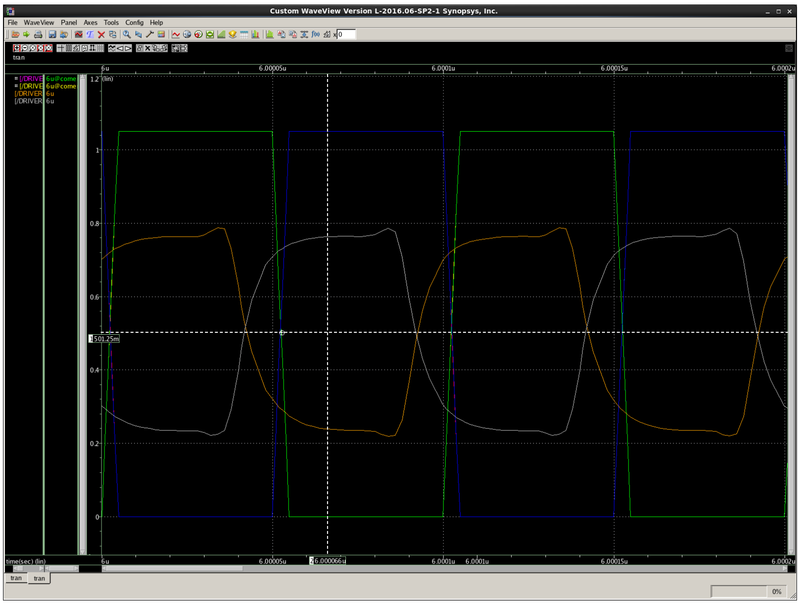


Figure 4.1: Low Voltage Typical Transmission at 10Gbits/s

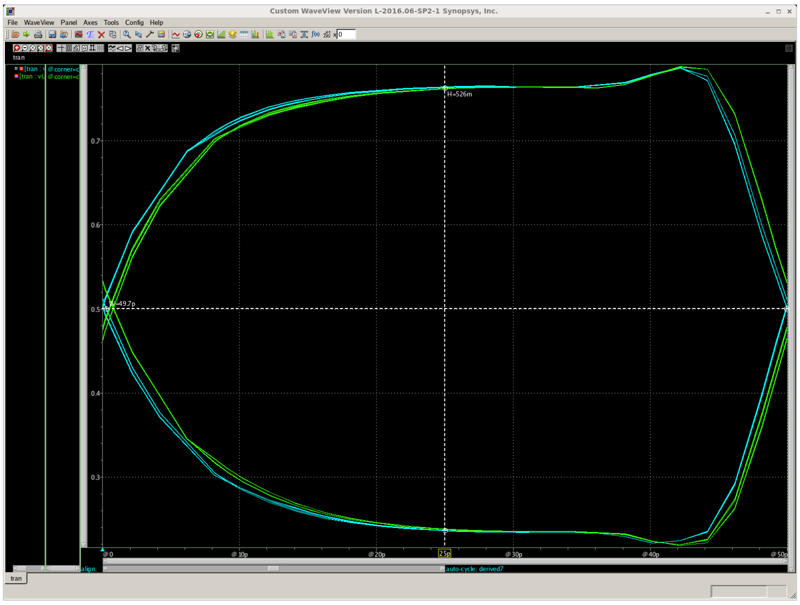


Figure 4.2: Low Voltage Typical Transmission at 10Gbits/s

4.2.1 Influence of Voltage variation

In this section the influence to the circuit of the variation of the external supply voltages is tested. A tolerance of 10% is considered for this corners meaning the supply voltage for the regulator can be between 2.25V and 2.75V and the chip source *VCore* can be between 0.945V and 1.155V. This voltage supply range was tested for all process and temperature corners but only the significant simulations are shown. The tested corners for this section are presented in table 4.3. First, for typical device process, and varying voltages the circuit was simulated with results being shown in 4.3.

If no LDO regulator was used, the *Driver* output swing would be controlled by the external voltage source meaning that for a variation of 10% in the supply would result in a a variation of 10% at the output . The regulator helps to reduce this variation influence and the following images of the simulations show that independently of the supply, the output swing for 1V transmission is kept close to 500 mV and for 1.6V transmission the output is kept at 800 mV. With this, we can say that variation over the external voltage have no effect (neglectable) over the circuit function. The supply variation was always tested with all the other parameters variations, never having significant affect over the operation of the *Driver*, but to avoid an exhaustive and heavy reading it will sometimes be omitted in the following sections. Below the eye diagrams of the Driver output for device corners SS-SF-FS-FF and supply variation for both LV and HV are presented to show that the regulator is a plus in this work robustness.

In 4.3 and 4.4 the waveforms for the outputs for both paths with variable supply and typical process corners are shown. It is possible to extract from the eye diagrams that the desired output voltage swing was achieved, 500mV for LV and 800mV for HV and little jitter was measured. Finally to understand the output voltage sources variation influence over the output, table 4.3

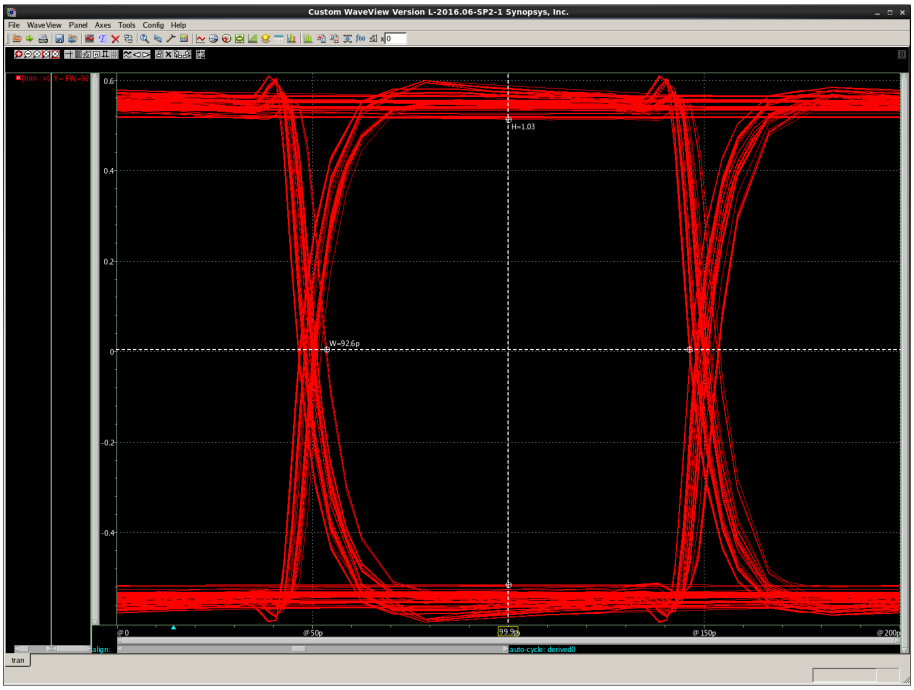


Figure 4.3: Output LV for a supply voltage variation at 10Gbits/s

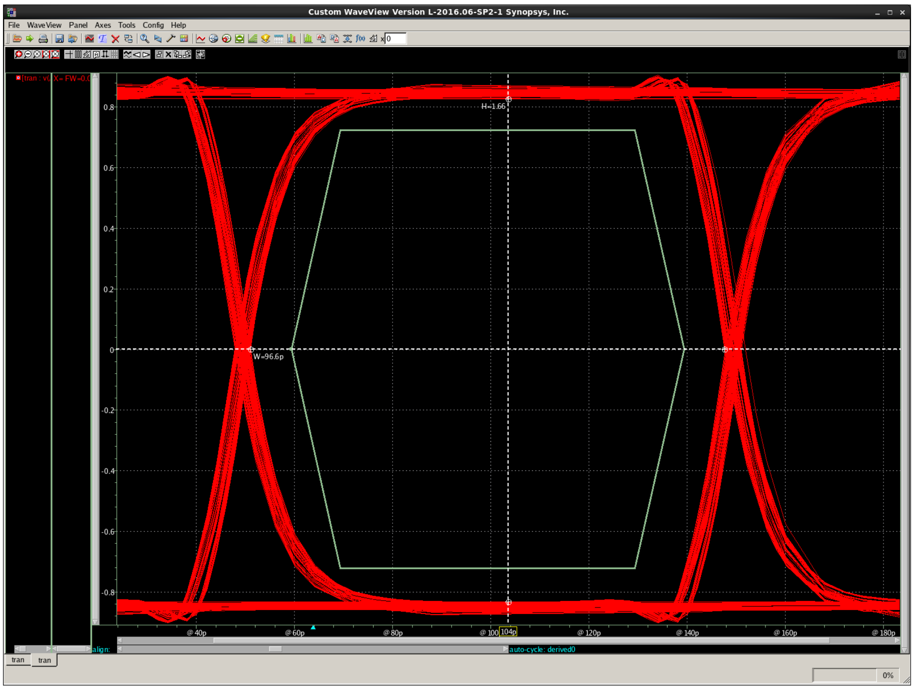


Figure 4.4: Typical Transmission High Voltage at 10Gbits/s

resumes the information for different simulations conducted.

In this work, the *DATA* signals that are meant to be transmitted through the *Driver* are originated from an external data generator that outputs random bit sequences. This signals are then

Table 4.3: Corners for Low and High Voltage Transmission

Corner	Input	Supply	NMOS	PMOS	Voltage Swing
1.1	Typical	$2.25V$	TT	TT	500 mV
1.2	Typical	$2.5V$	TT	TT	500 mV
1.3	Typical	$2.75V$	TT	TT	500 mV
2.1	Typical	$2.25V$	SS	SS	505 mV
2.2	Typical	$2.5V$	SS	SS	505 mV
2.3	Typical	$2.75V$	SS	SS	505 mV
3.1	Typical	$2.25V$	SF	SF	500 mV
3.2	Typical	$2.5V$	SF	SF	500 mV
3.3	Typical	$2.75V$	SF	SF	500 mV
4.1	Typical	$2.25V$	FS	FS	530 mV
4.2	Typical	$2.5V$	FS	FS	530 mV
4.3	Typical	$2.75V$	FS	FS	530 mV
5.1	Typical	$2.25V$	FF	FF	523 mV
5.2	Typical	$2.5V$	FF	FF	523 mV
5.3	Typical	$2.75V$	FF	FF	523 mV

shifted to the same voltage as the desired transmission voltage, $1V$ or $1.6V$. This level shifters were implemented by voltage controlled voltage sources with specific gain dependent on the chosen mode and so, this VCVSs reject the variation in the input, stabilizing the *DATA* at the desired voltages. The output of the *Driver* doesn't change with this variation at all.

4.2.2 Process variation of the resistors

Variations in process affect the devices properties and functionality and for resistors it affects there nominal values meaning resistors can vary by 20% over its value. As explained in 2 and 3 to fight this, the *Driver* was built in slices which could be activated or deactivated for a better tuning in the output termination.

The circuit was tested for a "slow" resistance condition meaning the resistors in the circuit would have a positive 20% mismatch and for a "fast" resistance condition were the resistors would have a negative 20% tolerance. For the "slow" condition, 18 slices were active and for the "high" condition, 12 slices were active.

From this simulations was to possible to extract that no significant change in the values at the output occurred when comparing with typical resistance conditions. The output swing was kept stable by activating more slices when the resistance was 20% higher, and deactivating slices when the resistance was minus 20%. In reality, after fabrication some resistances can be anywhere from - 10% nominal value until + 10% nominal value, but for validation of the circuit only the higher and lower extremes were tested.

4.2.3 MOSFETS Process Variations

In this section process variations over the NMOS and PMOS transistors will be analyze and circuit validation will occur for all corners. Devices and interconnects have parameters that vary because of different film thickness, lateral dimensions and doping concentrations after fabrication. The more significant variations occur in channel length (L), oxide thickness t_ox and threshold voltage V_T. This last variation over V_T is due to different doping concentrations and annealing effects, mobile charge in the gate oxide and discrete dopant variations.

Simulation in this chapter that take these differences into consideration will differ one from another as expected but the important part is the circuit maintaining its functionality and behavior across all corners, meaning that it keeps working and its *Output* is predictable and acceptable. This can be seen in the eye diagrams by comparing them and not seeing a clear difference in the plots.

Following four corners : SS (slow-slow), FS (fast-slow) SF (slow-fast) FF (fast-fast) are compared to the typical case 'TT' for a more precise look at the *Driver* outputs. First, testing slow switching of both NMOS and PMOS transistors which means that both transistors will have higher threshold voltage meaning they will switch only when higher voltages are applied to the device than if we were considering typical process. As it was referred previously, in this section we won't consider the input variation because as seen with different number of slices activated we can overcome this corner and this problem becomes insignificant.

The *Driver* output for *slow-slow* conditions are presented in plot 4.5, and 4.6 is the respective eye diagram for measurements. Input voltage variation as discussed previously doesn't affect the *Driver* and regulator supply tolerance is considered to show that the regulator works in different process corners.

In 4.5 it's possible to observe that for the $1V$ transmission the variation in the voltage output swing is low which is good, meaning this corners affects very little the *Driver* output. The difference we see when plotting the typical output with the SS corner is caused by a difference in the delay of the *Driver* for different device conditions and is not a real distortion of the output signal. The crossing of the edges or the voltage common mode of the signal, is lower than the desired value of $0.5V$ reaching approximately $0.45V$. For this *Slow-Slow* corner the eye diagram keeps showing very good opening and width but the common mode voltage is still a problem as it can be seen in 4.7.

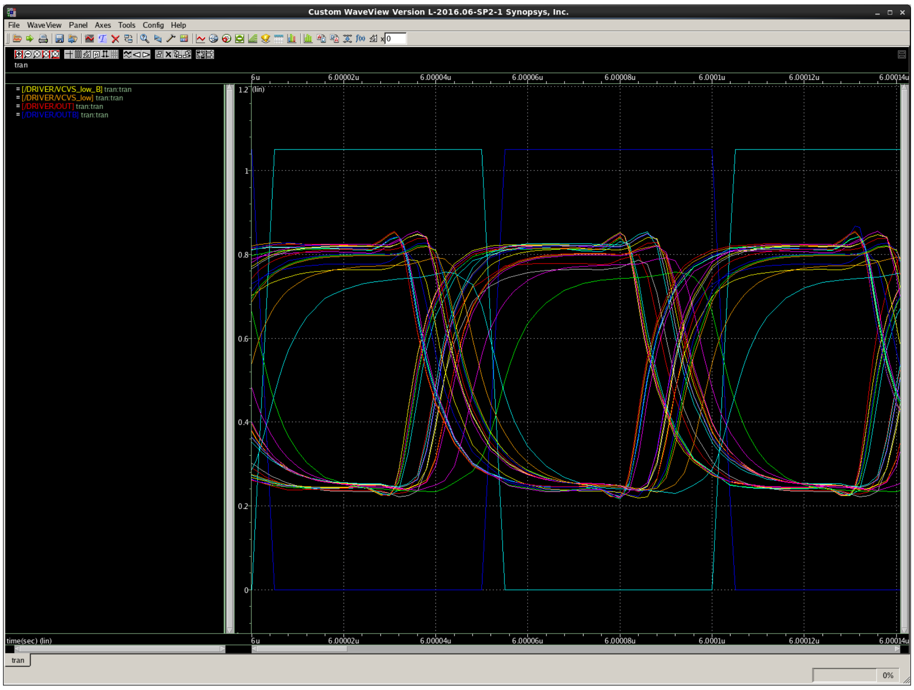

Custom WaveView Version L-2016.06-SP2-1 Synopsys, Inc.
File WaveView Panel Axes Tools Config Help
tran
[/DRIVER/VCVS_low_B] tran:tran
[/DRIVER/VCVS_low] tran:tran
[/DRIVER/OUT] tran:tran
[/DRIVER/OUTB] tran:tran
1.2 (lin)
1
0.8
0.6
0.4
0.2
0
6u
6.00002u
6.00004u
6.00006u
6.00008u
6.0001u
6.00012u
6.00014u
time(sec) (lin)
tran
0%

Figure 4.5: Slow-Slow and Typical outputs at 10Gbits/s

Now, slow switching of both NMOS and fast switching of PMOS transistors is considered in the *Slow-Fast* corner. This means that both transistors will have different threshold voltage from the typical case and the deviation between the two is the highest in this case. As it was referred previously, in this section we won't consider the resistance corners because as seen, with different number of slices activated we can overcome this corner and this problem becomes insignificant.

Simulations for *slow-fast* conditions with typical resistance , typical input and variable supply are presented below. Input voltage variation isn't considered since it has already been discussed that it doesn't interfere with the output. Supply tolerance is considered to show that the regulator works in different process corners.

In 4.8 it's possible to observe that for the $1V$ transmission the voltage output swing varies little, plus the signal keeps good integrity showing little distortion. The crossing of the signals or the voltage common mode of the signal, eye opening and width can be seen in 4.9 and comparison with the SS corner can be seen in 4.11 and from there we can immediately see that process variation isn't affecting the driver output significantly witch validates the desired robustness.

In 4.10 common mode was measured to be close to $0.675mV$ with eye opening slightly above $1V$ with $1.08V$ and width is kept at 49ps.

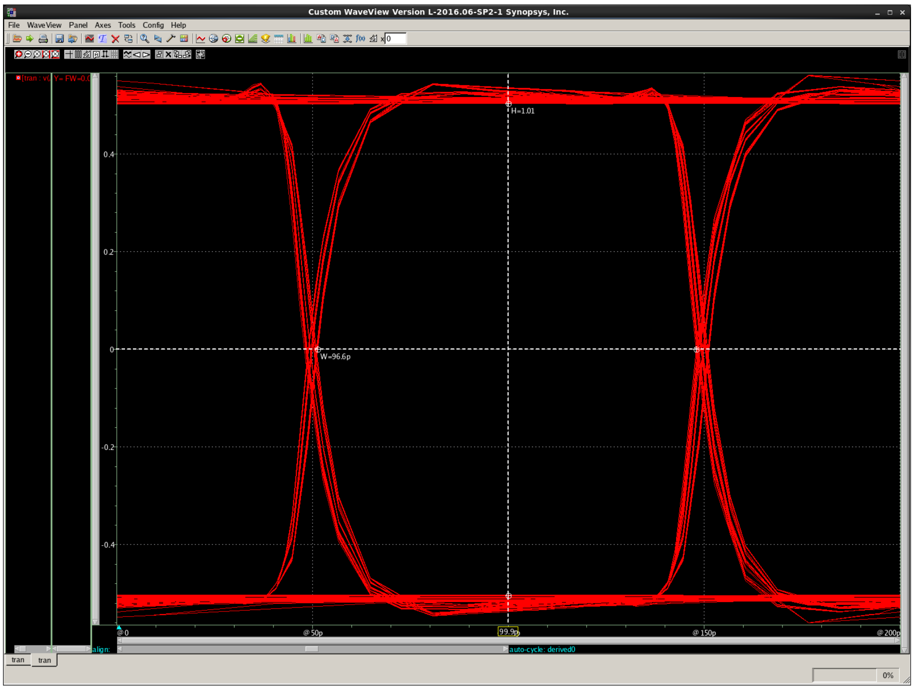


Figure 4.6: Slow-Slow Eye Diagram at 10Gbit/s

In this part the corners tested are the contrast of the previous ones, meaning the NMOS transistor will be tested with fast switching and PMOS with slow. This means that both transistors will have different threshold voltage from the typical case and the deviation between the two is the highest in this case. As it was referred previously, now we won't consider the resistance and input voltage corners.

Simulations for *fast-slow* conditions with typical resistance and variable supplies are presented below.

In 4.12 it's possible to observe that for the $1V$ transmission the variation in the voltage output swing is very little too for all plots. About the crossing of the signals, is in this case closer to the desired value of $0.5V$ but the common mode voltage not having any control is well shown in 4.13. The output of this corner is compared to the previous corner *Slow-Fast* to show that signal integrity, eye opening and width are kept, and the *Driver* works as desired and expected.

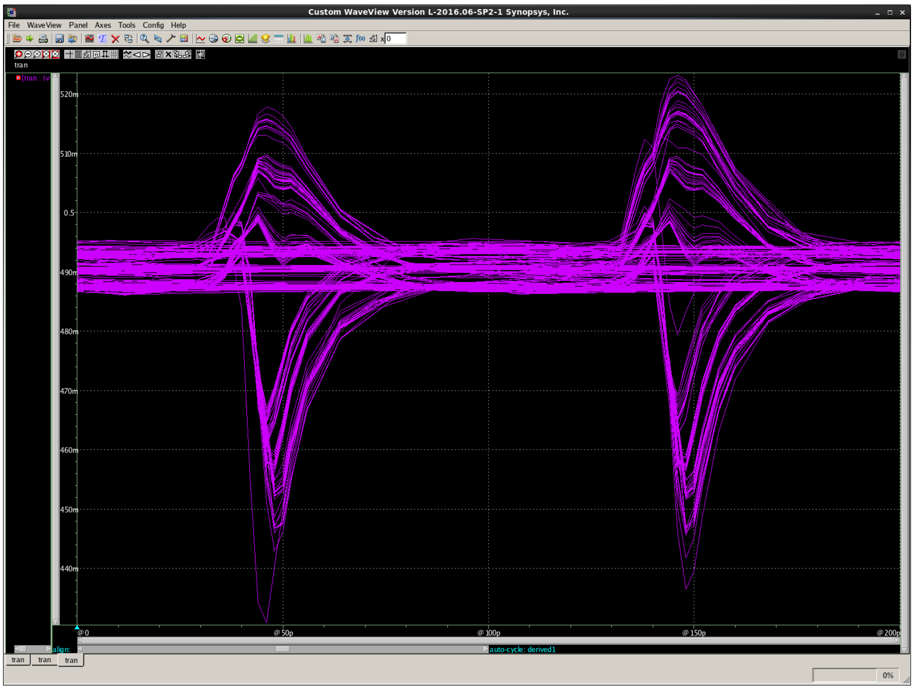

Custom WaveView Version L-2016.06-SP2-1 Synopsys, Inc.
File WaveView Panel Axes Tools Config Help
tran
520m
510m
0.5
490m
480m
470m
460m
450m
440m
@0
@50p
@100p
@150p
@200p
auto-cycle: derived1
0%

Figure 4.7: Slow-Slow Common Mode at 10Gbit/s

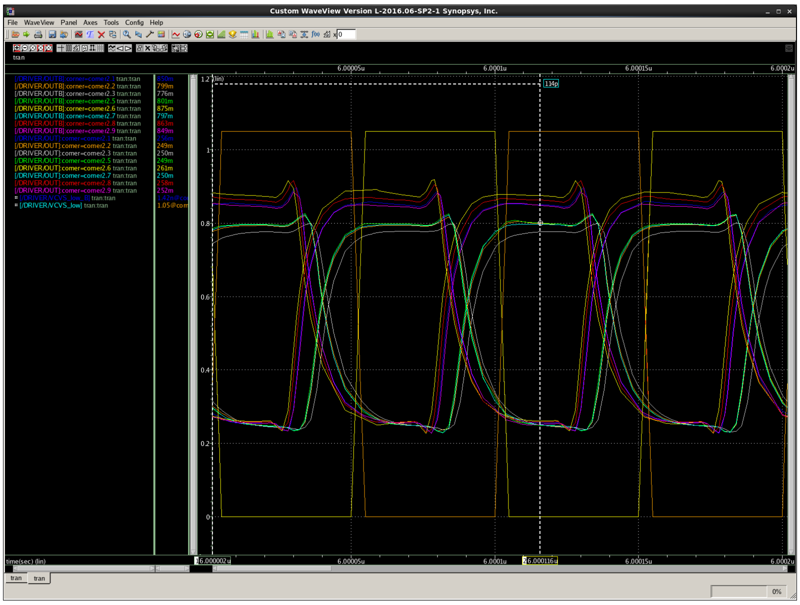

Custom WaveView Version L-2016.06-SP2-1 Synopsys, Inc.
File WaveView Panel Axes Tools Config Help
tran
[/DRIVER/OUTB]:corner=corner2.1 tran:tran 850m
[/DRIVER/OUTB]:corner=corner2.2 tran:tran 799m
[/DRIVER/OUTB]:corner=corner2.3 tran:tran 776m
[/DRIVER/OUTB]:corner=corner2.5 tran:tran 801m
[/DRIVER/OUTB]:corner=corner2.6 tran:tran 875m
[/DRIVER/OUTB]:corner=corner2.7 tran:tran 797m
[/DRIVER/OUTB]:corner=corner2.8 tran:tran 863m
[/DRIVER/OUTB]:corner=corner2.9 tran:tran 849m
[/DRIVER/OUT]:corner=corner2.1 tran:tran 256m
[/DRIVER/OUT]:corner=corner2.2 tran:tran 249m
[/DRIVER/OUT]:corner=corner2.3 tran:tran 250m
[/DRIVER/OUT]:corner=corner2.5 tran:tran 249m
[/DRIVER/OUT]:corner=corner2.6 tran:tran 261m
[/DRIVER/OUT]:corner=corner2.7 tran:tran 250m
[/DRIVER/OUT]:corner=corner2.8 tran:tran 258m
[/DRIVER/OUT]:corner=corner2.9 tran:tran 252m
[/DRIVER/VCVS_low_B] tran:tran 1.42n@co
[/DRIVER/VCVS_low] tran:tran 1.05@corn
1.2 (lin)
1
0.8
0.6
0.4
0.2
0
6.00005u
6.0001u
6.00015u
6.0002u
114p
time(sec) (lin)
6.000002u
6.000116u
0%

Figure 4.8: Driver Output for Slow-Fast corner at 10Gbits/s

The last process corner for the transistors is the *Fast-Fast* corner were both NMOS and PMOS transistors have fast switching. This means that both transistors will have lower threshold voltage meaning they will switch sooner with lower voltages.

Simulations for *fast-fast* conditions with typical resistance and variable supplies are presented below. Like in the previous corners this one too shows good signal integrity and measurements

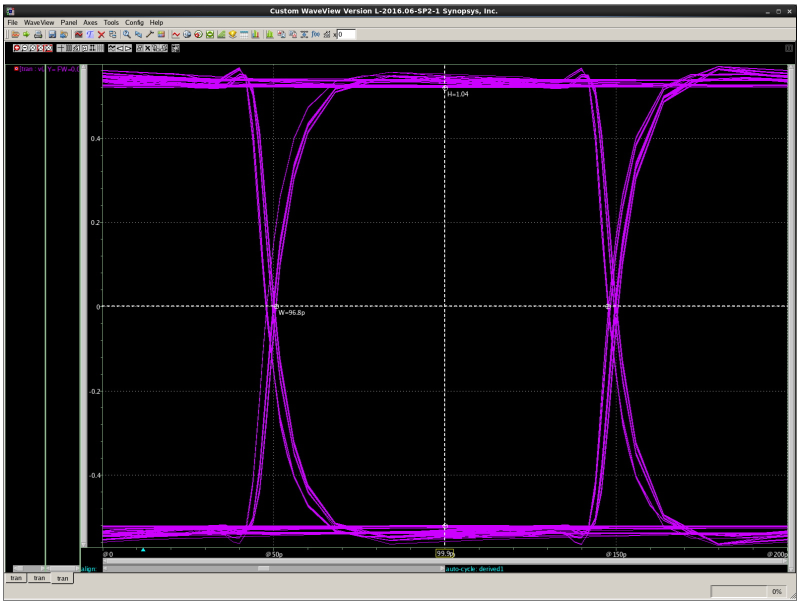

Custom WaveView Version L-2016.06-SP2-1 Synopsys, Inc.
File WaveView Panel Axes Tools Config Help
H=1.04
W=96.8p
0.4
0.2
0
-0.2
-0.4
@0
@50p
99.9p
@150p
@200p
tran
tran
tran
align
auto-cycle derived1
0%

Figure 4.9: Slow-Fast Eye Diagram at 10Gbit/s

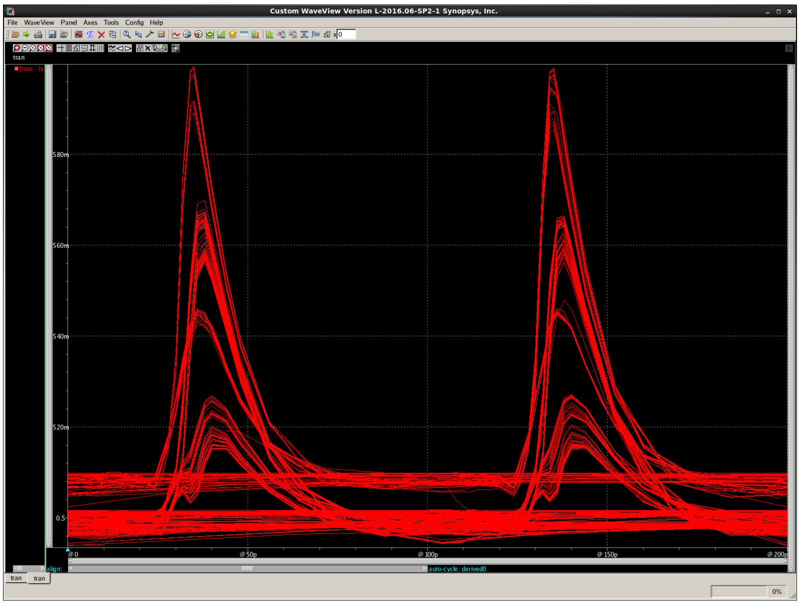

Custom WaveView Version L-2016.06-SP2-1 Synopsys, Inc.
File WaveView Panel Axes Tools Config Help
tran
580m
560m
540m
520m
0.5
@0
@50p
@100p
@150p
@200p
tran
tran
align
auto-cycle derived0
0%

Figure 4.10: Slow-Fast common mode at 10Gbit/s

which makes us conclude that the low voltage path is validated with success for all PVT corners.

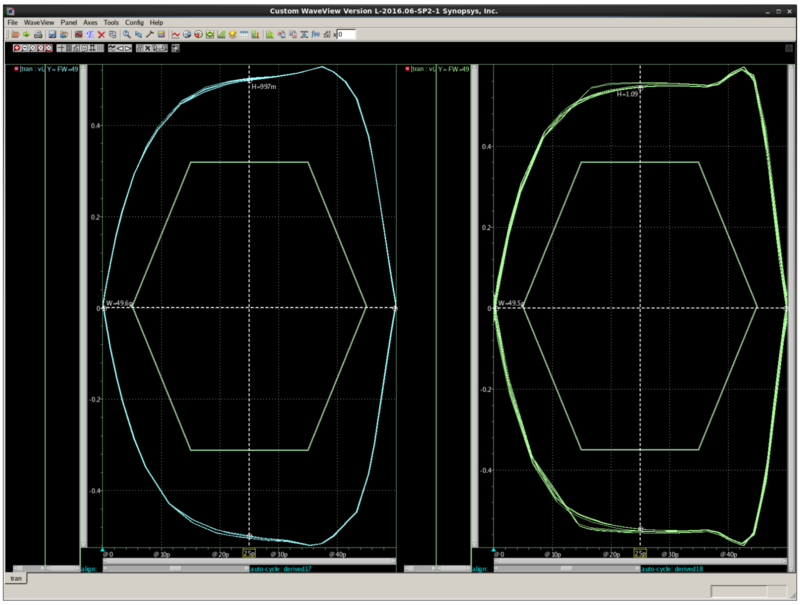


Figure 4.11: Slow-Slow vs Slow-Fast Eye Diagrams at 10Gbit/s

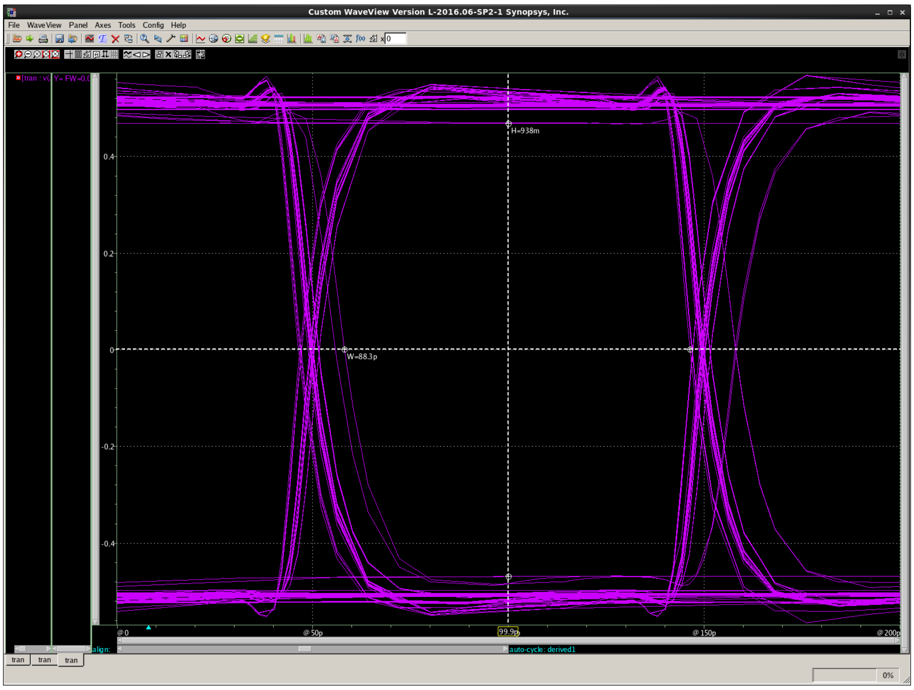


Figure 4.12: Fast-Slow eye diagram at 10Gbits/s

4.2.4 High Voltage Path

For the High voltage path, considering the signal level of 1.6V and the size of this transistors, the transmission at 10Ghz had a lot of distortion. This high voltage transistors are heavy, and don't switch as fast as the low voltage path. Additionally, we are trying to switch them to 1.6V which as

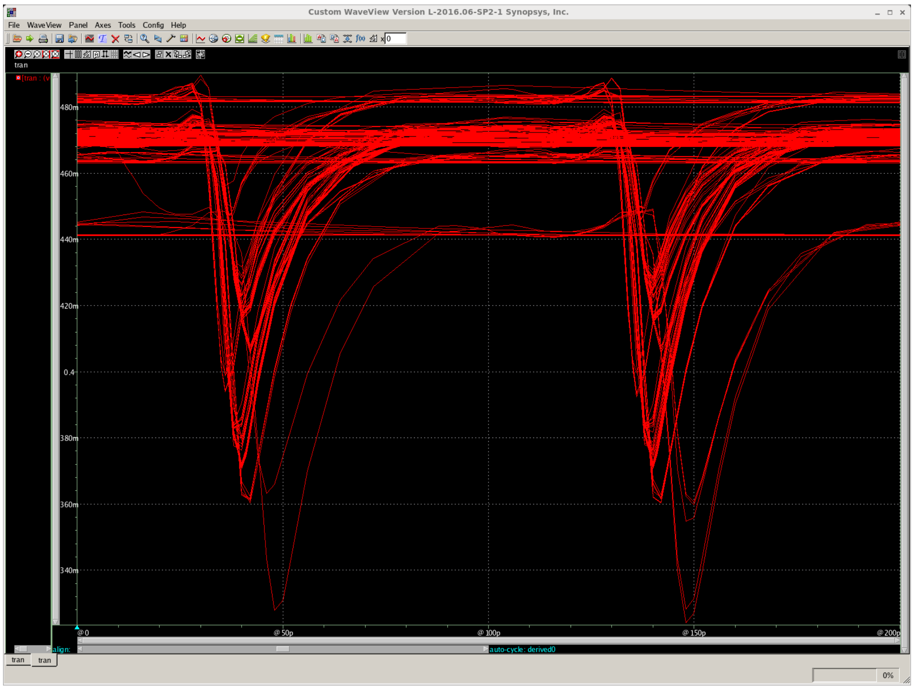

Custom WaveView Version L-2016.06-SP2-1 Synopsys, Inc.
File WaveView Panel Axes Tools Config Help
tran
480m
460m
440m
420m
0.4
380m
360m
340m
@0
@50p
@100p
@150p
@200p
align
auto-cycle : derived0
tran
tran
0%

Figure 4.13: Fast-Slow Common mode at 10Gbits/s

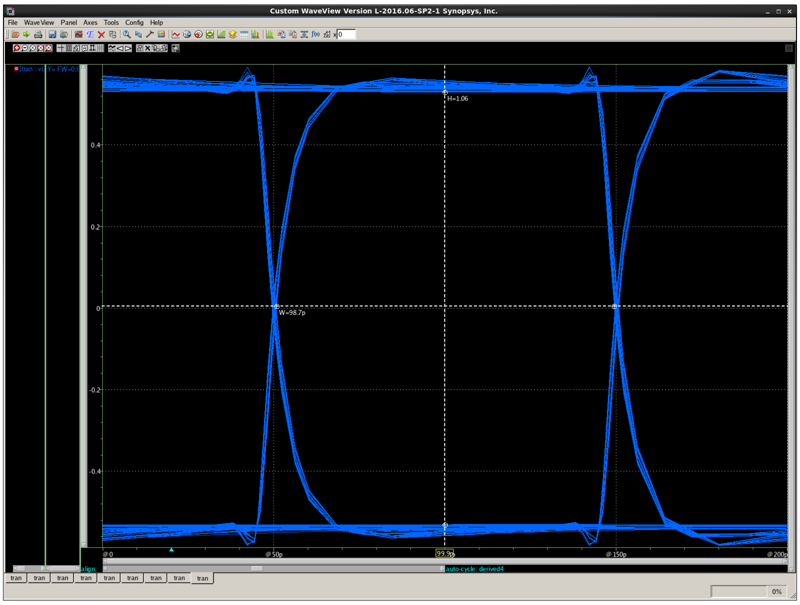

Custom WaveView Version L-2016.06-SP2-1 Synopsys, Inc.
File WaveView Panel Axes Tools Config Help
H=1.06
0.4
0.2
0
W=98.7p
-0.2
-0.4
@0
@50p
99.3p
@150p
@200p
align
auto-cycle : derived4
tran
0%

Figure 4.14: Fast-Fast Output at 10Gbits/s

been found to be very difficult. This problem reveals itself on the simulation in signals between the *Pre-Driver* stage and the *Driver*. What happens can be explained starting by the end of the circuit and coming back until the logic circuit *Slices Control Unit*. For start, the output driver needs to have some size for its equivalent resistance to be small and don't overrun the resistors in this stage, this difficulties the *Pre-Driver* job of signaling the inputs to the PMOS and NMOS gates of OD.

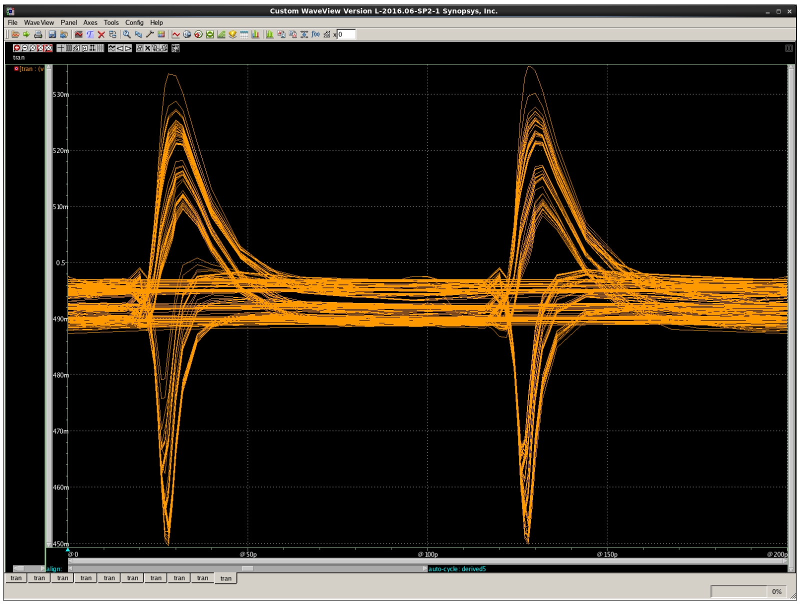


Figure 4.15: Fast-Slow Common Mode at 10Gbits

Architecture of the *Pre-Driver* was explained in the third chapter but was slightly changed in the hopes of achieving better signaling for the High Voltage Path at 10 Gb/s. The changes made were to add another buffer, for a total of two buffers or 4 inverters, after the logic gates in order to give more power to the *Pre-Driver* to be able to have enough slew-rate to switch to 1.6V. This was successful to the point of the plot in 4.16 where some distortion can be seen but distinct levels from high and low are achieved. Below are presented the simulations for a total of 135 corners at 10Gb/s. This 135 corners represent variation at the temperature, input voltages, supply voltages and process variation in the resistors and in the devices (TT-SS-SF-FS-FF).

In the previous discussed approach of sizing the *Pre-Driver* a disadvantage of this technique was possible to be observed, which was the logic gates that were before the *Pre-Driver* were starting to be affected by the sum of the size of all the slices connected. This logic gates generate the control signals $HVSlices < 17 : 0 >$ and SEL and this signals stopped being stable and started to respond and show some variation because of being connected to the 36 slices of both *Pre-Drivers* (High Voltage and Low Voltage). A trade-off here exists, when we increased the size of the *Driver* to compensate for high speeds, the inputs started to degrade, but for a low size *Driver* there wasn't driving capability enough to reach high speeds. This were the best results possible for this topology.

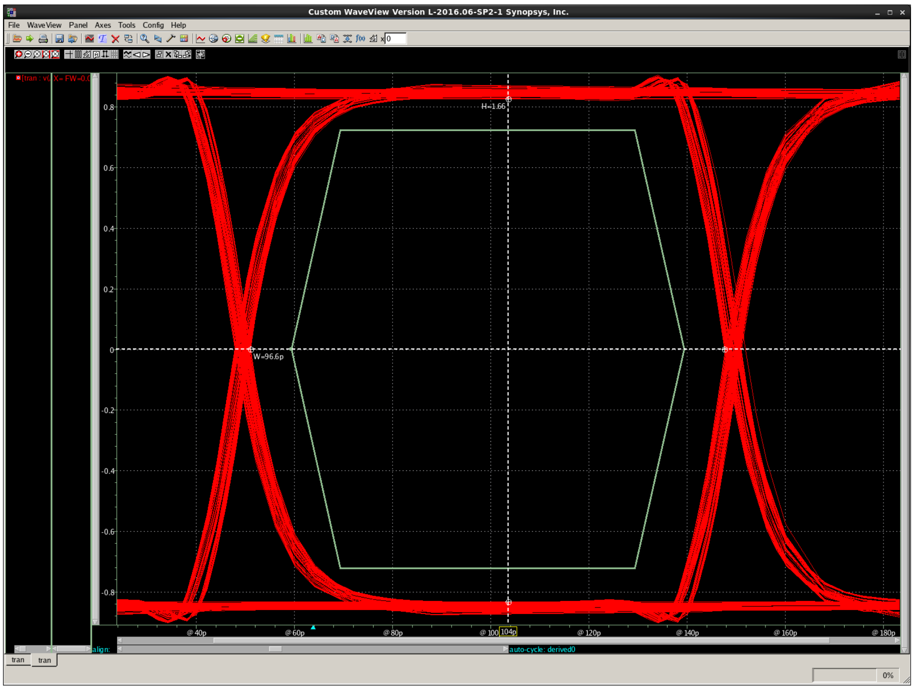


Figure 4.16: Typical-Typical Eye Diagram

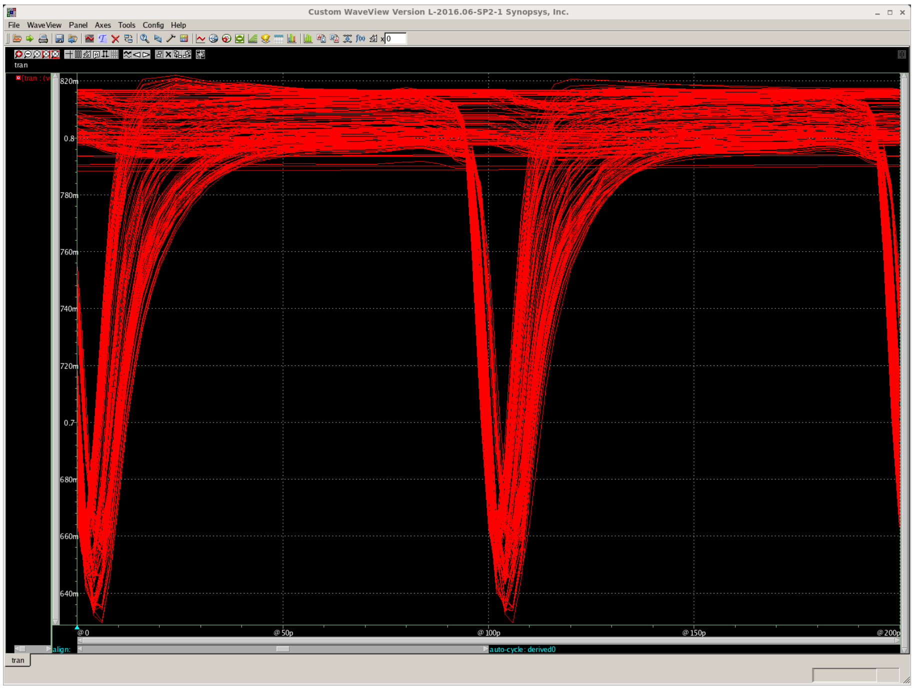


Figure 4.17: Typical-Typical Common Mode

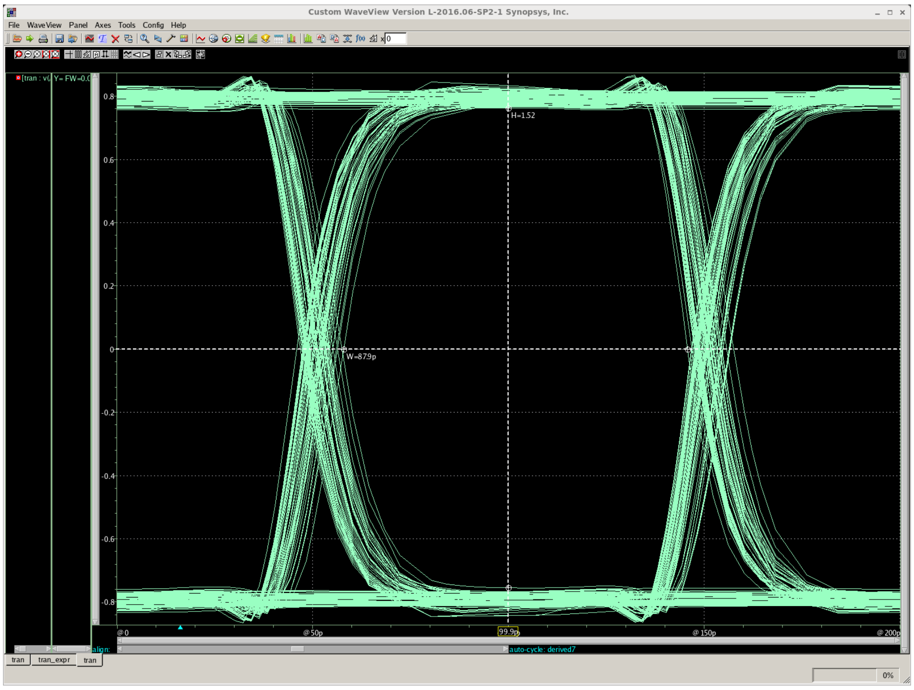


Figure 4.18: Slow-Slow Eye Diagram

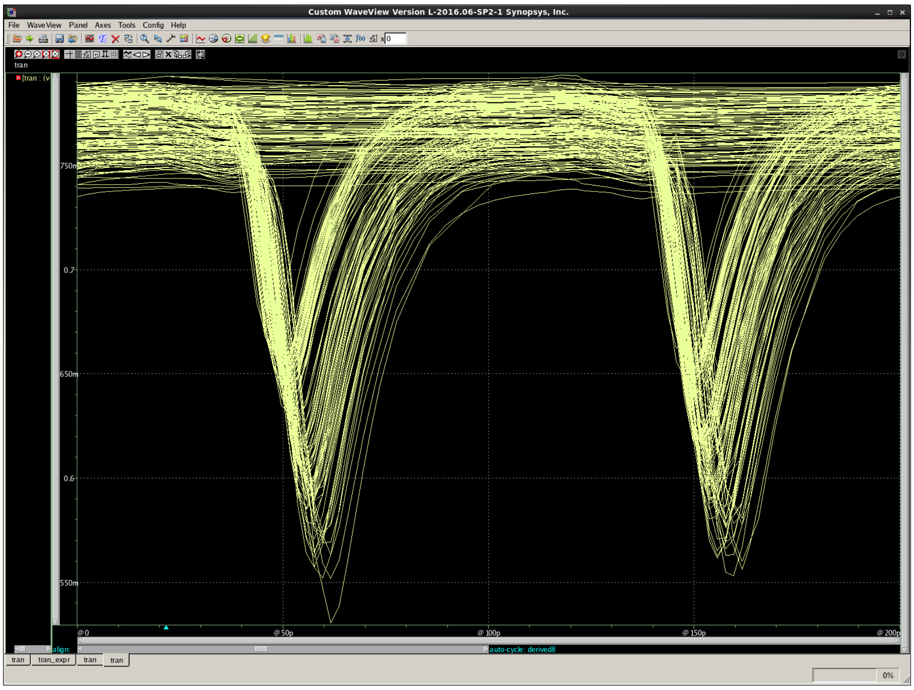


Figure 4.19: Slow-SLow Common Mode

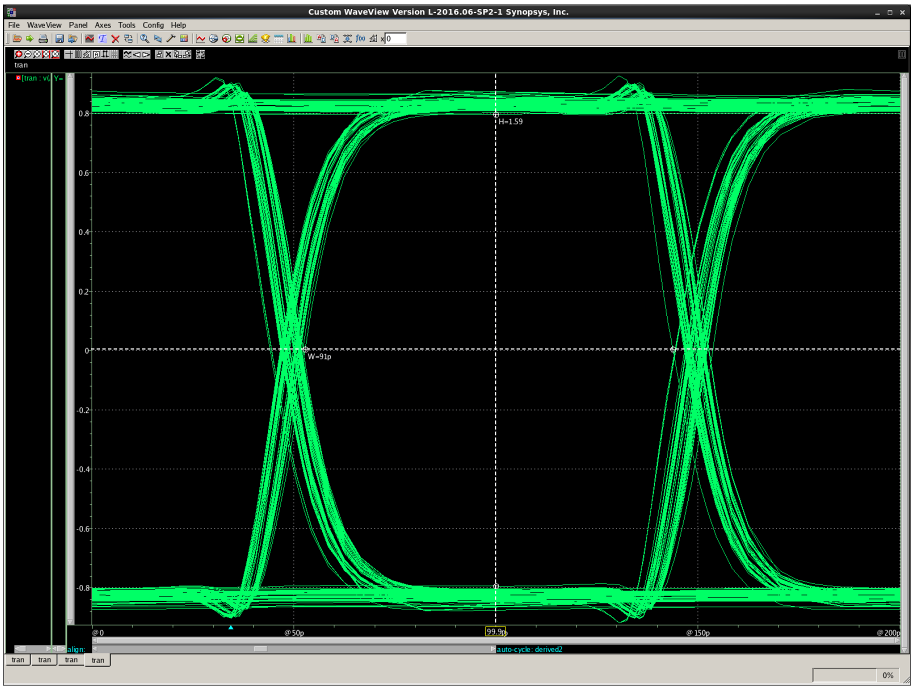


Figure 4.20: Slow-Fast Eye Diagram

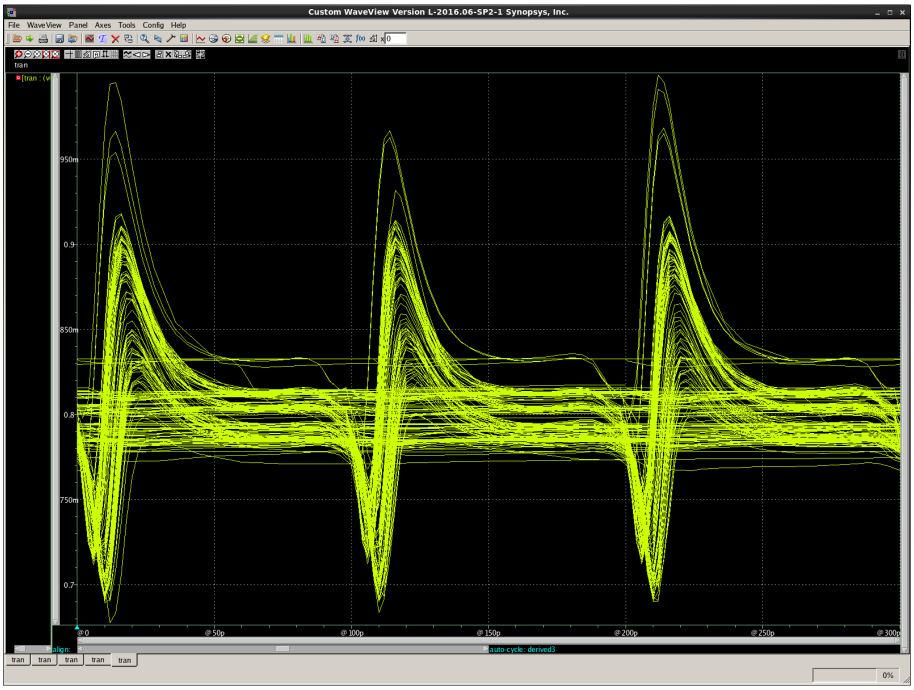


Figure 4.21: Slow-Fast Common Mode

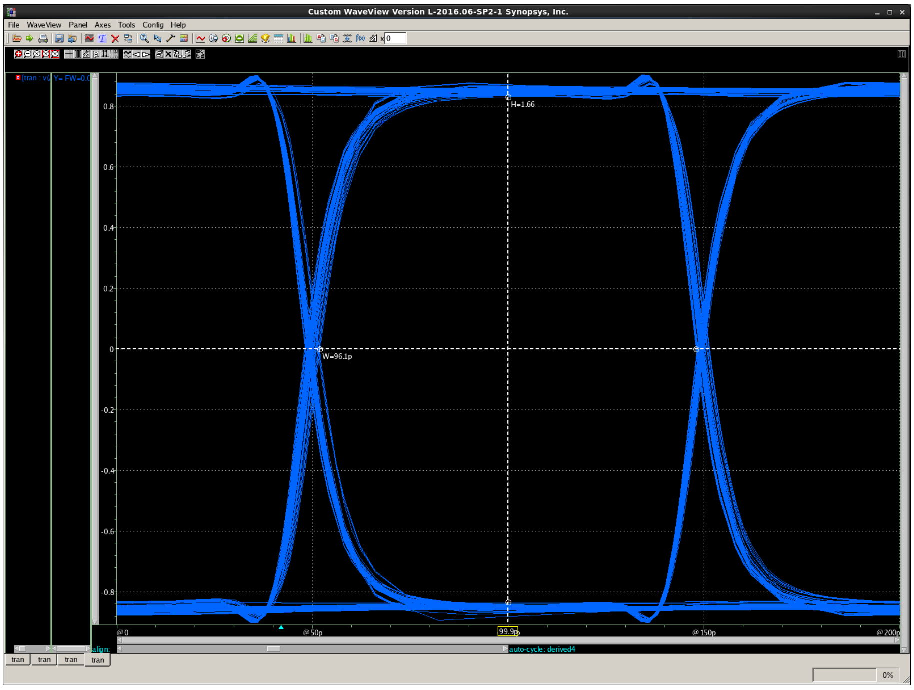


Figure 4.22: Fast-Slow Eye Diagram

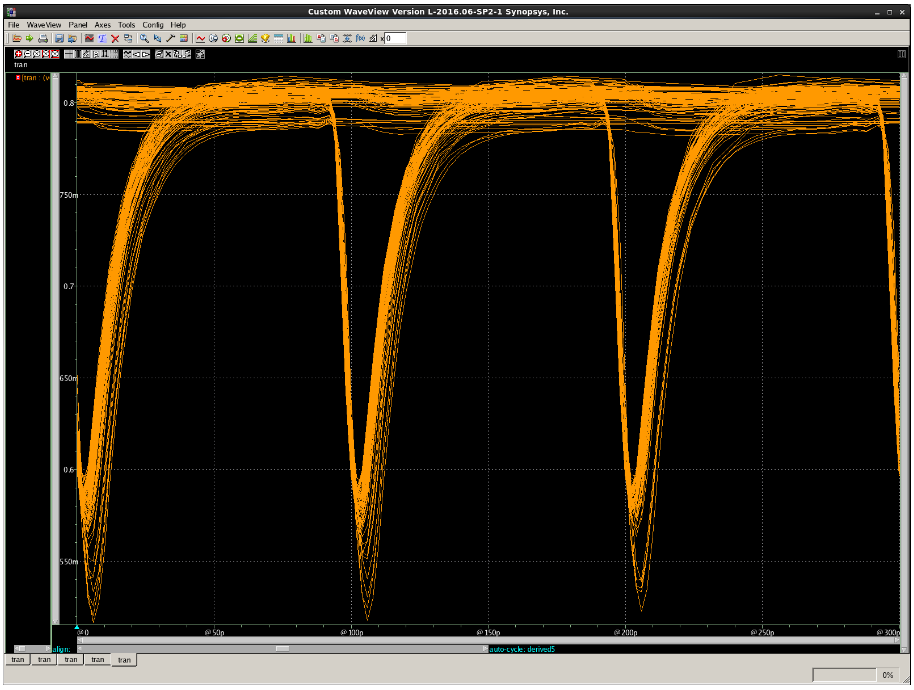


Figure 4.23: Fast-Slow Common mode

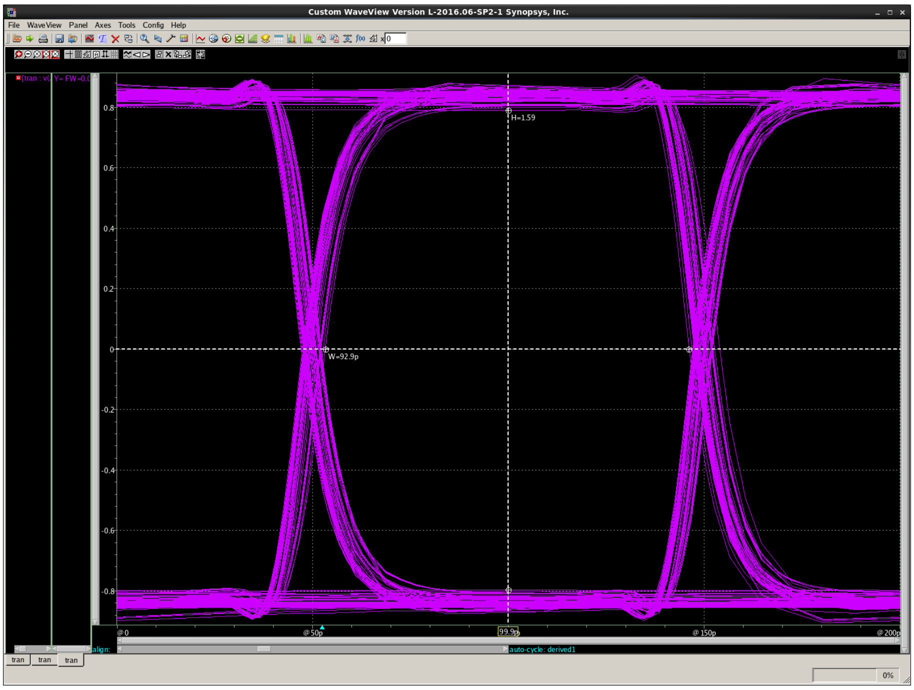


Figure 4.24: Fast-Fast Eye Diagram

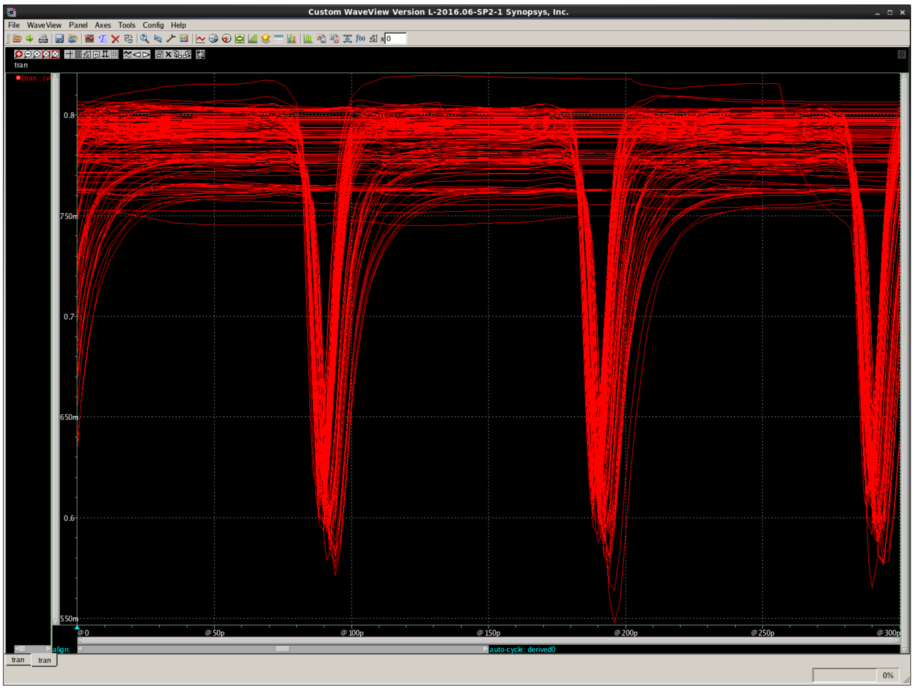

Custom WaveView Version L-2016.06-SP2-1 Synopsys, Inc.
File WaveView Panel Axes Tools Config Help
tran
0.8
750m
0.7
650m
0.6
550m
@0
@50p
@100p
@150p
@200p
@250p
@300p
align:
auto-cycle: derived0
tran
tran
0%

Figure 4.25: Fast-Fast Common Mode

Chapter 5

Conclusion

In this chapter a critic discussion about the work developed and more specifically what was implemented with success and what wasn't is presented in this chapter. Also, it will be highlighted the strong points and weakness for the work developed.
In this thesis, the challenge answered was to develop a transmitter to work at 10Gbps, with fixed termination impedance of 100 Ω. The *Driver* specs such as jitter, voltage output swing, power consumption, area, etc. had a certain level of freedom, meaning the specs weren't limited or fixed. Thus, the analyse of the results and extraction of this values is what defines if this work was successful in finding a robust solution or not.

Overall, this thesis responds to the problem with an architecture that most importantly works and the transmitter can be used. The results presented in the previous chapter 4, shows for the differential outputs of the *Driver* well defined waveforms, with two logical states. The rising and falling edges or slew rate of the circuit was maximized as much as possible for a 10 Gigabits transmission and the results show good response for the circuit specially for low voltage. When using devices with Core technology, the transistors are lighter and faster which results in a better signal integrity that when we use high voltage devices which are bigger and slower introducing some problems for very high speeds. This was well known from the start, but having a higher voltage path with voltages that only I/O devices could handle was a challenge but one that was interesting to explore and see the limitations in the technology. Turning to Core devices again, the transistors could handle a maximum supply of 1.1V and this work presents a low voltage mode of operation at 1V which leaves low margin. With more time, it would have been interesting to develop this work only for Core Devices, having now a higher voltage path at 1V transmission and a lower voltage path at 800 mV or 700 mV. But once again, being this work developed this way, it was possible to comprehend better the limitations of the technology and all the logic behind the level shifting of the control signals in order for the circuit to work properly and to be able to change from one mode of operation to the other was much more complex.

One important factor for the robustness of the *Driver* that wasn't approached in this thesis was the control and compensation of the common mode voltage. At first, a very good sizing and tune of the circuit and optimization of the devices and components size through simulation was meant

to result in a fixed common mode voltage (0.5V for low voltage and 0.8V for high voltage) but when simulating and extracting the results it was possible to comprehend that a control circuit was needed for this purpose. Going from the typical model to PVT variations this was even more obvious. The difficulty here was the speed of the signaling, even with the circuit being well designed, for lower speeds such as 5 or 6Gigabits small devices with less current capability had time to switch completely and the waveforms were good. For 10Gigabits, the slew rate had to increase and the transistors had to be bigger but this meant that the devices would switch slower too. It just wasn't possible without additional circuitry to have a fixed V_{CM} of $V_{dd}/2$
The success in the developed *Driver* lies in its stable output swing for different resistance corners which validates the use of the slices to program and adjust the output impedance. Additionally, the purpose of the regulator was fully applied with success too and the circuit works equally for variations in the supply voltages or other external voltages. Besides some transistor corners being hard on the *Driver*, the signaling still showed fast switching and achieved the desired voltage levels with an output swing of $500mV$ for low voltage transmission and $800mV$ for high voltage.
Power consumption was a concern in the last weeks and all the transistors and elements size were dimensioned to the minimum size possible for the *Driver* to still work, but with a high voltage path of 1.6V operation, the power consumptions measured were still high. In the future, revision of the architecture or other techniques should be done in order to drop the power consumption from it's actual value.

5.1 Future Work

This dissertation offers a detailed design and implementation process of a high speed *Driver* for data communications and is the schematic part for the fabrication of the same. It can be used as a basis for the future projects towards applications in very high speed signaling. The work could be extended further to the following aspects:

- Circuit modules can be added to this design to improve some specs of the *Driver* such as the V_{CM} that needs to be fixed and equal to $V_{dd}/2$. Other spec that should be improved in case the high voltage path is pretended to remain at $1.6V$ is the slew-rate of the HV transistors. A slew-rate control circuit could also be developed to complement this thesis.

- One way to explore this work and give it more usability is to convert the high voltage path to $1V$ and lower the low voltage path to $700mV$ for example. This would give this *Driver* more possibility of being used in real high speed applications. Turning it completely low-power.

- Consumption of power is one of the most important specs and the *Driver* should be analyzed and techniques to lower it's consumption levels should be implemented.

- After the above work is done, the *Driver* should continue the process until fabrication by first designing and optimizing the layout.

References

[1] G. E. Moore. "cramming more components into integrated circuits". *Electronics Magazine*, pages 103–114, 1965.

[2] A.W. Topol et al. "3d fabrication options for high-performance cmos technology". *Wafer-Level 3D ICs Process Technology*, pages pp.197–198, 2008.

[3] Suman Datta. *"Recent Advances in High Performance CMOS Transistors From Planar to Non-Planar"*. 2013.

[4] Asselberghs et al Klekachev, Nourbakhsh. *"Graphene Transistors and photodectors"*. 2013.

[5] Science C.Y Sung, IBM Research and Technology. *"BM Graphene Nanoelectronics Technologies"*. 2011.

[6] IEEE Hiroshi Iwai, Fellow. *"Future of nano CMOS technology"*. 2013.

[7] Xing Chao (Chuck) Yuan Kyung Suk (Dan) Oh. *"High-Speed Signaling: Jitter Modeling, Analysis, and Budgeting"*. Prentice Hall., Oct 6, 2011.

[8] 2004 International Symposium on SoC A.Morgenshtein, I.Cidon. "comparative analysis of serial vs parallel link in noc", 2004. pp. 185-188.

[9] XILINX. "understanding true cmos outputs", October 2000. https://www.xilinx.com/support/documentation/application_notes/xapp329.pdf.

[10] Texas Instruments Bruce Hunter and Patrick Rowland. "linear regulator design guide for ldos". Technical report, Texas Instruments, June 2008.

[11] Hani F.Ragaie 2007 Internatonal Conference on Microelectronics Diana heidar, Mohamed Dessouky. "comparison of output drivers for high-speed serial links", 2007. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4489831.

[12] 4th International Conference on Intelligent and Advanced Systems. "configurable output driver with programmable on-chip impedance supporting wide range data rates", 2012.

[13] Internatonal Conference on Microelectronics Dina heidar Mohamed Dessouky Hani F.Ragaie. "comparison of output drivers for high-speed serial links", 2007. http://www.dbpia.co.kr/Journal/ArticleDetail/NODE02049646.

[14] 2012 International SoC Design Conference (ISOCC) Taeho Kim, Deog-Kyoon Jeong. "a 10 gbp/s voltage swing level controlled output driver in 65-nm cmos technology", 2012. http://www.dbpia.co.kr/Journal/ArticleDetail/NODE02049646.

[15] IEEE INTERNATIONAL SOI CONFERENCE. "enhanced performance of serdes current-mode output driver using 0.13 um pd soi cmos", 2009. `http://ieeexplore.ieee.org/document/5318758/`.

[16] Maxim. "linear regulators in portable applications". *Application Note 751*, 2012.

[17] Akira Umezawa Takeshi Miyaba Toru Tanzawa Shigeru Atsumi Hironori Banba, Hitoshi Shiga and Koji Sakui. "a cmos bandgap reference circuit with sub-1-v operation". *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, page Vol.34 No.5, 1999.