2016-07

Design of the Analog Transmitter Module in 130 nm CMOS technology

Núñez-Corona, Joel A.

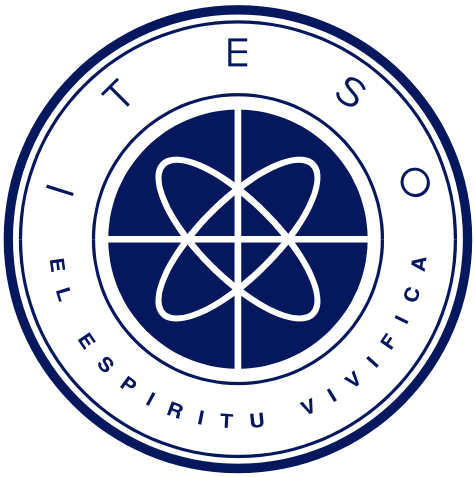
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Especialidad en Diseño de Sistemas en Chip

Reconocimiento de Validez Oficial de Estudios de nivel superior según Acuerdo Secretarial 15018,

publicado en el *Diario Oficial de la Federación* el 29 de noviembre de 1976

DEPARTAMENTO DE ELECTRÓNICA, SISTEMAS E INFORMÁTICA



Design of the Analog Transmitter Module in 130 nm CMOS technology

Tesina para obtener el grado de:

Especialista en diseño de sistemas en chip

Autor: Joel Antonio Nuñez Corona

Directores: Federico Lobato López y Esteban Martínez Guerrero

Guadalajara, Jalisco, Julio 2016

ACKNOWLEDGMENTS

The author of this thesis thanks first to God for giving me the opportunity to get this new personal achievement and have reached a new goal in my life.
To my parents, especially to my mother Maria Guadalupe Corona, strength and example of life, whose tenacity and perseverance has taught to overcome new challenges.
To my beloved wife Ana Maria Palacios and my dear children Fatima, Paola and Marco; for their support and understanding to complete another step in my academic preparation.
To my brothers: Maritza, Carolina, Saul and Ivan; whose affection and love over time has inspired me to face life and overcome every day.
To Dr. Esteban Martinez and Federico Lobato for their support and guidance during this Specialty in Design of Systems on Chip.
To my partner and friend Aaron Désiga for his help to complete the designs of the Verilog-A codes.
To all teachers whose classes helped me acquire knowledge to complete this specialty.
To CONACYT, for its financial support was key to complete this specialty.
To ITESO, by providing the opportunity to complete this stage of my life in this great university.

ABSTRACT

This report documents the design procedure of a SerDes Transmitter module for PCI e-gen1 applications. The Transmitter module has programmable features: impedance, amplitude and pre-emphasis. Design was performed in 130nm CMOS technology (process IBM cmrf8sf). The design of the Transmitter module is divided in three stages as follow: First, a behavioral model of Transmitter module was created in Verilog-A language. This model has 2 inputs and two complementary outputs. This behavioral model was used to verify the proper system-level response of the Transmitter module alone. Simulations based on behavioral model were performed using Spectre simulator in Virtuoso-Cadence. Simulations results show that the three features: impedance, amplitude and pre-emphasis could be properly modulated with the developed Verilog-A code. The behavioral model of Transmitter module was also assembled with other modules of SerDes system in order to perform mixed-signal simulations and validate the correct response of the entire SerDes system for PCI-e gen1 specifications.

As second stage, the Transmitter module was designed at transistor level with two complementary outputs. For this, we used pCells (transistors, resistors and capacitors, and vias for interconnecting devices) from CMRF8SF technology process. Transmitter module is composed by several blocks such as Mux, Decoders, basic digital cells, buffers, and pseudo-analog cells among others. Transistors of digital cells were sized by applying the basic scaling theory while transistors of pseudo-analog cells were sized by analytic calculations in an iterative way. Mux and Decoders were assembled with digital cells. To validate the correct response of each designed internal block, test-benches were created to test each block. Transient simulations were performed using Spectre in Virtuoso Cadence. After verifying the proper response of internal block alone, they were assembled to form more complex blocks such as Decoders, ZAP UNIT and others and they were also tested. Finally, all internal blocks of Transmitter module were assembled and tested at nominal PVT conditions. Critical PVT corners were evaluated and it was verified that response was under tolerance range.

The third stage of design was the layout creation in a custom way. A similar bottom-up methodology followed to design at schematic-level of internal modules was applied to design the layout of each internal block of the Transmitter module. Keeping in mind the integration of internal blocks, the low–level digital cells were designed with the same height. The layout of each internal block was verified for DRC and LVS using Calibre tool and design rules of CMRF8SF process. Extraction of parasitics from layout using Calibre-PEX was done for each internal block. Post-layout response was compared to the pre-layout one; for this, Spectre simulation was performed of each internal blocks using simultaneously the extracted Calibre view and schematic view. Once the correct response was verified, the internal blocks were assembled and routed in order to build the entire layout of Transmitter module. Finally, the layout of Transmitter module was folded in order to generate the two complementary outputs. At this top level, the layout of Transmitter module was also checked for DRC and LVS.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

This technical report presents the design of a Transmitter module for digital signals with differential outputs, Tx and Txbar, where one complements the other. This module is part of a SerDes to be developed by the members of the Specialty in Design of System on Chip, SerDes SoC was designed for serial communication protocol PCI Express 1 at 1.5 GHz. The technology used is CMOS 130nm process IBM cmrf8sf under MOSIS license.

The Behavioral model of the Transmitter module, to tests the design at system level is also presented. These modules are performed in Verilog –A language. These modules are verified using the same testbenches used in transistor level design.

This report is organized as follow: Chapter 1 reviews the SerDes, the considerations necessary for the development of the project as well as the tools used, Chapter 2 shows the development of each module block at transistor level, besides the development of behavioral modules described in Verilog-A language. Chapter 3 deals with the pre-layout tests individually to each of the cells as well as the complete Transmitter module with differential outputs. Test of behavioral models were performed by replacing each block in the Transmitter module at transistor level. PVT simulation test of the Transmitter module were performed. Finally chapter 4 shows the layout design of individual cells as well as the complete Transmitter module with differential outputs, they were verified with DRC and LVS CALIBRE tools and show that each block pass without DRC and LVS errors. Post Layout tests were performed to the Transmitter module, time delays and amplitude reduction in the output signal were observed.

CHAPTER 1: FUNDAMENTALS OF SERDES SYSTEM

1.1 SerDes overview.

A SerDes (serializer- deserializer) is a block of microprocessors and other high-speed devices, responsible of data conversion serial to parallel and parallel- serial for the transmission and reception data. Its operation is as described below: the parallel data is stored in parallel registers by synchronizing these with the rising or falling edge of the reference clock. Once stored, the data is encoded in some standard, for SerDes the 8B/10B encoding is used, this encoding is used to transmit data with a minimum of loss. The circuit clock and data recovery (CDR) of the SerDes must have a certain density level to prevent information loss.

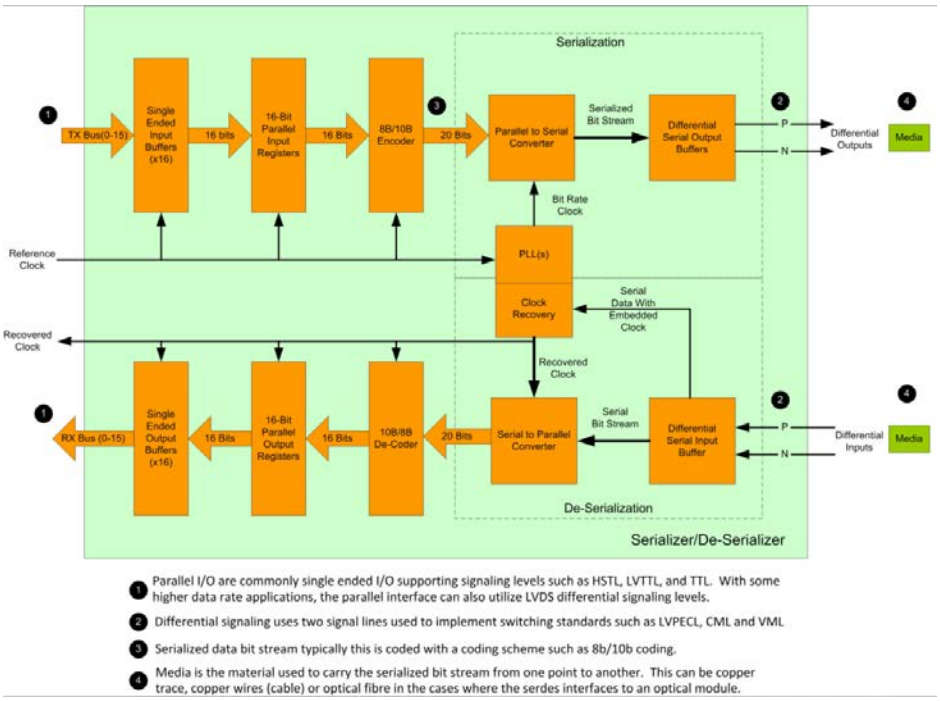


Fig. 1: High-level SerDes IC block diagram [1].

The de-serialization is the process where the serial data is converted to parallel data to form the original data that was transmitted. This process depends on the CDR circuit providing a clock recovered, which controls the time for the shift registers, used to form the encoded data. The data once again become parallel, is decoded to the original data. The data are fed to output registers and are synchronously sent by the parallel output buffers. Typically, these buffers are single-ended signal buffers. The clock frequency is aligned to the input data rate. Often the transmission and reception in the SerDes is done at the same time.

The phase-locked loop (PLL) is an important part of SerDes, this produces the high-speed clock used to drive the serial of the transmitter in addition to receiving the path of the device. According to its architecture may have one, to function as a transceiver or two PLL one for transmission and one for reception. The reference clock is a clock input that is used to drive the PLL. Usually it has a relationship with the serial data rate which must operate the SerDes. For example, a SerDes with parallel 10-bit interface could use a reference clock of 125 MHz, with which the SerDes operates at a speed of 1.25 Gbps [1].

1.2 PCI Express Protocol

<<The PCI Express architecture is specified in layers, as shown in Figure 2. Compatibility with the PCI addressing model (load-store architecture with a flat address space) is maintained to ensure that all existing applications and drivers operate unchanged. The PCI Express configuration uses standard mechanisms defined in the PCI plug-and-play specification. The software layers generate read and write requests that are transported by the transaction layer to the I/O devices using a packet-based, split-transaction protocol. The link layer adds sequence numbers and Cyclic Redundancy Check (CRC) to these packets to create a highly reliable data transfer mechanism. The basic physical layer consists of a dual simplex channel implemented as a transmit pair and a receive pair. The transmit and receive pair together are called a lane. The initial speed of 2.5 Gb/s provides a nominal bandwidth of about 250 MB/s in each direction per PCI Express lane. Once overhead is taken into account, about 200 MB/s of this is usable by the device for data movement. This rate represents a twofold to fourfold increase over most classic PCI boards. And unlike PCI, where the bus bandwidth was shared among devices, this bandwidth is provided to each device>> [2].

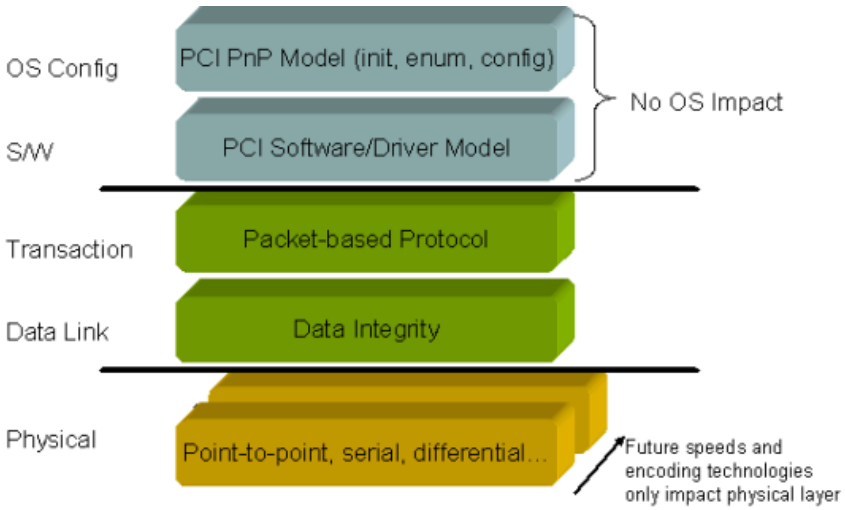


Fig. 2: PCI Express Layered Architecture [2].

The PCI Express architecture is composed of two differential pairs of low voltage AC-coupled (one pair for transmission and one pair for reception). Part of physical link uses a de-emphasis scheme to reduce interference, improving data integrity. It has a data clock embedded in the code 8b / 10b to achieve high-speed data transmission. The initial frequency is 2.5 Gb/s (Generation 1) which can reach up to 10 Gb/s (the maximum using signals in copper). The bandwidth PCI Express Connection can be scaled linearly by simply adding signal pairs to form multiple channels. The physical layer provides from 1X to 32X lane widths, which in theory divides the input data in each of the lanes. The encoded data are transmitted through the rails. This function does not affect the other layers. The way it should work is done between agents at each end of the link, no firmware or operative system (OS) is involved in this process. PCI Express architecture provides future improvements in speed and encoding techniques, which only affect the physical layer [2].

1.2.1 8b/10b encoding

The code defines the mapping from a 8bit byte (256 unique data words) and an additional 12 special (or K) characters into a 10bit symbol, hence the name 8b/10b encoding. It has been widely used in high speed serial communication standards that need a run length limited, charge balanced data stream for reliable data transmission and clock recovery. Because of its many feature, the code has been used in the physical layer (PHY) of a number of current and emerging standards, including Fiber Channel, Gigabit Ethernet, and Rapid I/O, to name a few.

1.3 SerDes Blocks to be designed

1.3.1 Digital Des-Serializer (SIPO Serial Input Parallel Output)

This block is responsible for receiving the pattern data and synchronizes with the reference clock for this purpose a CDR is used. A Decoder is needed because the data is usually encrypted. This is an internal module of the Des-Serializer.

1.3.2 Digital Serializer (PISO Parallel Input Serial Output)

This block takes the input signal and converts parallel to serial data, an encoder is required using the 8b/10b code. The parallel data is stored and synchronized with the transmission clock to be converted to serial data. This data must provide to the stage of analog transmission to send the data out of the chip.

1.3.3 Analog Receiver

This block compensates for the attenuation experienced by the serial data to be transmitted via the communication bus. This stage receives a differential digital signal and must deliver a CMOS with the lowest possible noise and jitter for use by digital signal stages.

1.3.4 Analog Transmitter

This block couples the differential digital signal received to eliminate or reduce the noise thereof. With this digital block can properly process the digital signal. This

stage receives a signal single-ended and sends it to the outside of the integrated circuit differentially. Also has an output impedance matching.

1.4 SerDes design specifications

Specifications	
Bus	1 bus TX/RX
Speed Transmission	1.5 Gbits/s
Technology	CMOS 130 nm
Dimensions	1.5mm x 1.5 mm
Voltage	1.2 V
Package	40 pins package

Table 1: SerDes Specification [3].

1.5 Project scope

The scope of the project is to design and validate the blocks of a SerDes system targeted to work at 2.5 Gbps using the PCI- Express gen 1 protocol.

1.6 Considerations

To complete this work it will be on multidisciplinary team, each member shall deliver a part of the SerDes. The information of the first generation of the Specialty in Design of System on Chip will be used to complete the full design, due MOSIS constrains the technology will be migrated from 180 nm to 130 nm. The technical knowledge to fulfill the design of each block will be provided by ITESO during the courses of Specialty in Design of System on Chip. All the blocks will be done on time to the final integration; validations and test benches of each block will be done and also the clock reference will be external.

1.7 Constrains

Working Deadline to get the full design is July 30, 2016. The design will be implemented using cmrf_8sf process from IBM's 130nm technology, all the design tools (Cadence) are only available at ITESO campus but we have 24/7 access. The chip will be built in a 40 pin Package with a maximum chip area 1.5x1.5 mm this is a restriction for MOSIS academic projects.

CHAPTER 2: DESIGN OF TRANSMITTER MODULE

2.1 Behavioral model of Transmitter module

The behavioral model of the Transmitter module was created in order to model in an easy and fast way the system; according to basic design flow for analog blocks as shown in Fig. 5. We used Verilog-A language to create behavioral models of each block of Transmitter module. The behavioral model also called macro-model is very important in situations where changes in design specifications are done, in which we could have the following options:

1. Change system architecture and keep technology.
2. Change technology and keep system architecture.
3. Change system architecture and technology.

Option number three is the riskier in businesses because it represents a greater effort and design time (money, time and technical resources). In this approach, all has to be verified and validated again: functionality of the architecture and transistor technology should meet the goals of design. However, if the architecture remains the same, and if transistor technology is changed, modeling can help us to accelerate models validation, and this will save time and money [4].

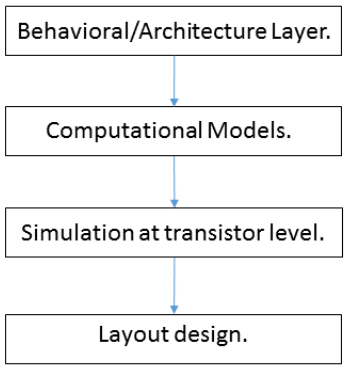


Fig. 3: Basic analog design flow [4].

First, we describe the behavior models of each block composing the Transmitter module: ZAP UNIT, Multiplexer 2-1, Decoder 4-12 and other internals blocks. Then the internal blocks are assembled in a single module called to form the Transmitter module.

The ZAP UNIT is composed by the following cells (Fig. 6).

- Buffers.
- Multiplexers.
- Basic cells.
- Tied_up and tie_down circuit.

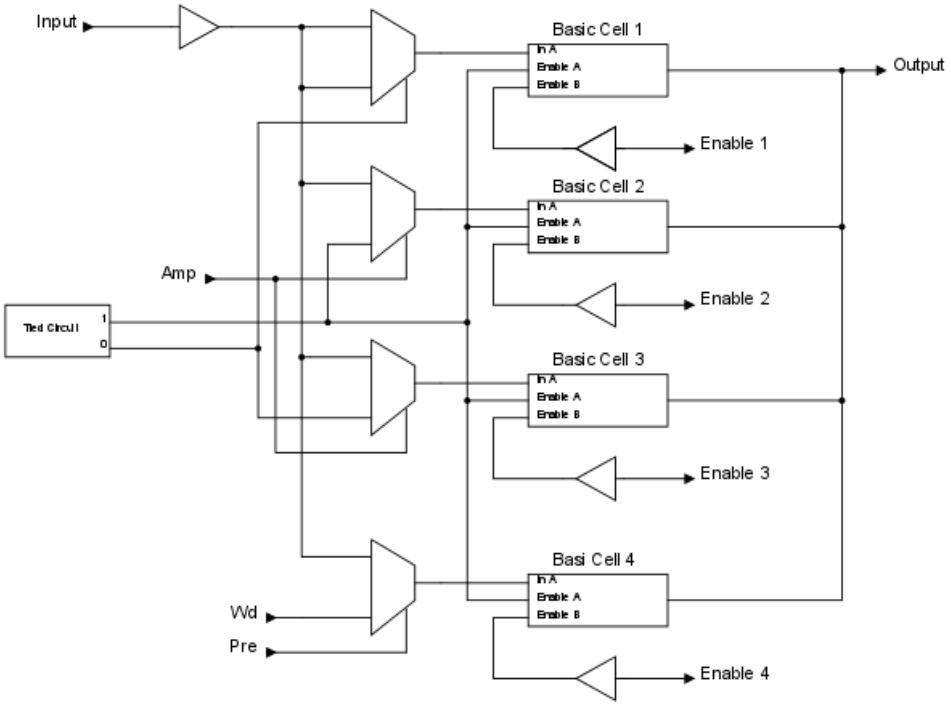


Fig. 4: Schematic of ZAP UNIT.

There are two options to make the behavioral model of the ZAP UNIT, one is describing the module as a whole entity and the other is describing the behavior of each block containing ZAP UNIT or another complex block. The main advantage of using the second option is that we can test cells individually. For this reason, we have created internal cells of ZAP UNIT, and then behavioral model of ZAP UNIT was created by integrating internal cells.

The behavioral model of Multiplexer 2-1 was created using an *if* condition [4], whose functionality principle is as follow: if enable input is equal to zero the output will be the signal in input 1, by contrary, if enable is equal to one the output will be the signal in input 2. The Verilog-A code of Multiplexer 2-1 is presented in the Appendix a. Fig. 7 shows the response of the behavioral model of Multiplexer 2-1.

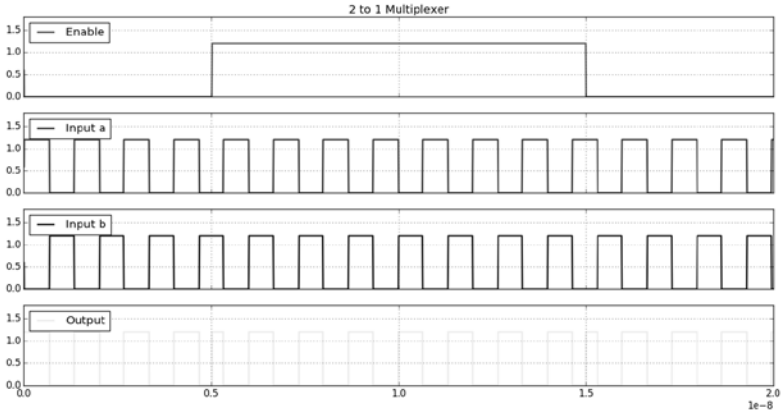


Fig. 5: Multiplexer 2-1 behavior transient response.

Fig. 8 shows the electric diagram of Basic cell. The behavioral model of this cell was emulated as a voltage divider [4]. The impedance values R_x can be calculated using the voltage divider formula (Eq.3):

$$V_{R_x} = \frac{R_x}{R_x + R_{bc}} V_{in} \qquad \text{(Eq. 1)}$$

Where R_{bc} is the impedance in the basic cell, R_x is the impedance of the transmission line (100 Ω), V_{in} is the input voltage and V_{R_x} is the voltage at R_x. In this way, the total resistance depends on the values of Enable A and Enable B inputs; for instance, if Enable A is *On* and Enable B is *Off*, the total resistance at output node is the sum of R1 and R2 while if Enable A and Enable B are *On* the total resistance at output is R2. The Verilog-A code of Basic cell is presented in the Appendix b.

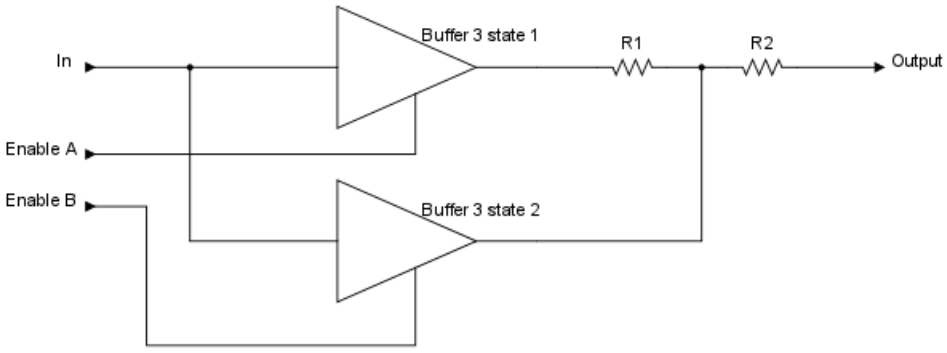


Fig. 6: Basic Cell Diagram [5].

The way to prove that the model is working correctly is to make a simple test. In the Transmitter module, the values of R1 and R2 are equal (820 Ω). Then, if we add a resistor at the output of the basic cell with a value of 820 Ω too, the corresponding voltage value at the output when Enable A is “on” should be one third of the input value (Fig.9). On the other hand if enable A and Enable B are “on” the voltage value at the output will be at half of the value of the input voltage (Fig. 10).

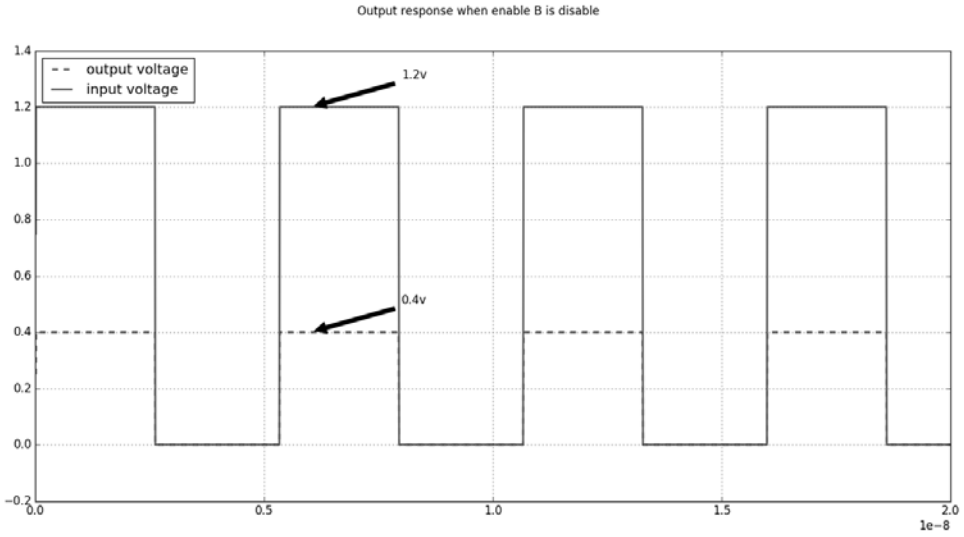


Fig. 7: Output voltage when enable B is disabled.

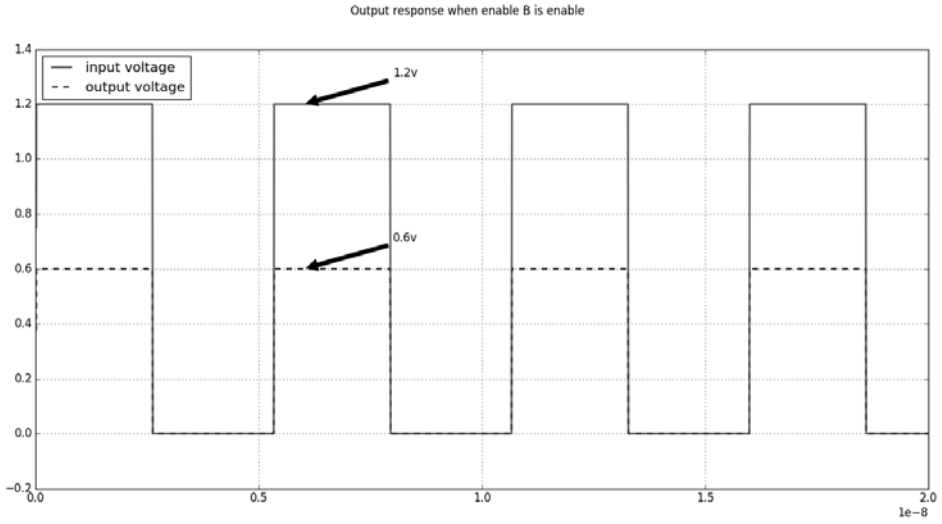


Fig. 8: Output voltage when enable A and enable B are both enable.

The only purpose of buffers (Fig. 6) in the Transmitter module is to give strength to the signals. So we decided do not create the behavioral model for the buffer cell because it just send a signal to enable the second 3– state buffer in the basic cell (Fig. 8), and at least we want to test a high delay or low level voltage in the “high state” in which we desire modeling.

For the Decoder 4-12 its Verilog-A code [4] was created using a truth table, such table is adapted from [5]. This kind of Decoder is called thermometer type (Fig. 11). The Verilog- A code of Decoder 4-12 is shown in the Appendix c. The functionality of this block was also verified by Spectre simulation.

I3	I2	I1	I0	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	1	0	1	0	0	0	0	0	0	0	1	1	1	1	1
0	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1
0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1
1	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 9: Truth table Decoder 4-12 [5].

The Delay inverter is another important key in the pre – emphasis function; this block will delay and invert the original signal to create the required pre – emphasis. The code for this block was adapted from reference [6]. This code has useful instructions that were used in other modules. The Verilog-A code of Delay inverter is shown in the Appendix d. The value of the time delay is 57 pS, which represents 10% delay of the input signal.

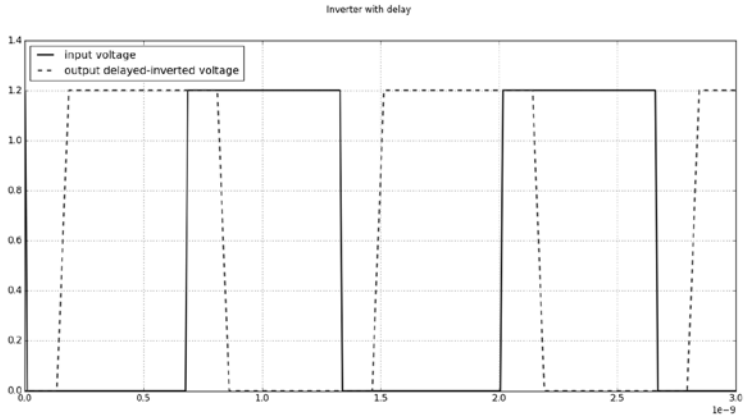


Fig. 10: Time - response of an inverter with delay.

The Decoder 2-3 as described in [7] controls the amplitude and the pre – emphasis. This Decoder also uses the thermometer type output. The truth table is shown in Fig. 13. The Verilog-A code of Decoder 2-3 [4] is presented in Appendix e.

I1	I0	S2	S1	S0
0	0	0	0	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

Fig. 11: Truth table for Decoder2-3.

Finally, all the above described behavioral models were integrated to implement the ZAP UNIT module. Fig. 14 presents the transient response for amplitude of ZAP UNIT using behavioral models of its internal cells.

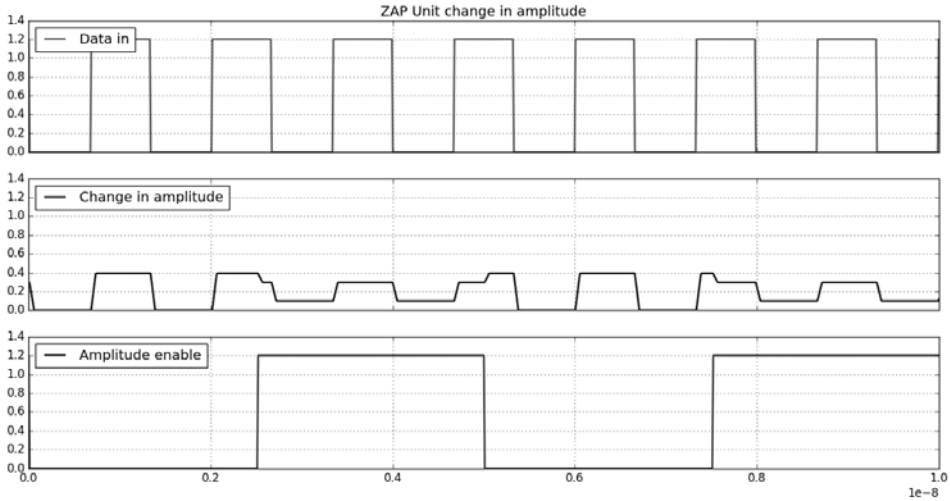


Fig. 12: ZAP UNIT transient response based on its behavioral model.

2.1.1 Behavioral model of Transmitter module with Complementary Output

Fig. 15 presents the block diagram of Transmitter module assembled with Verilog-A modules of its internal blocks. Note however that in Fig. 15, there is not unit ZAP buffers, and not direct-out buffer because for behavioral modeling there are not needed.

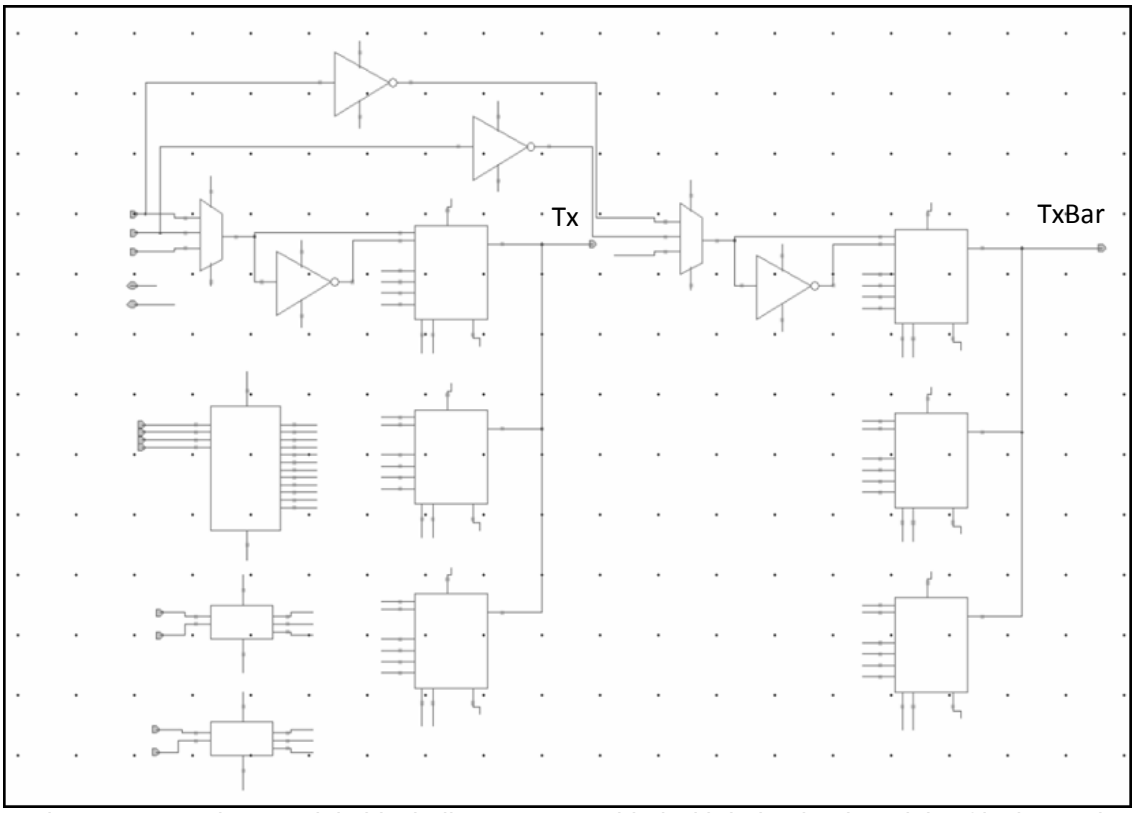


Fig. 13: Transmitter module block diagram assembled with behavioral models of its internal blocks.

2.1.2 Simulation Validation of Behavioral model of Transmitter module with Complementary Output

Fig. 16 presents the testbench to validate the pre-emphasis response of Transmitter module based on its macromodel.

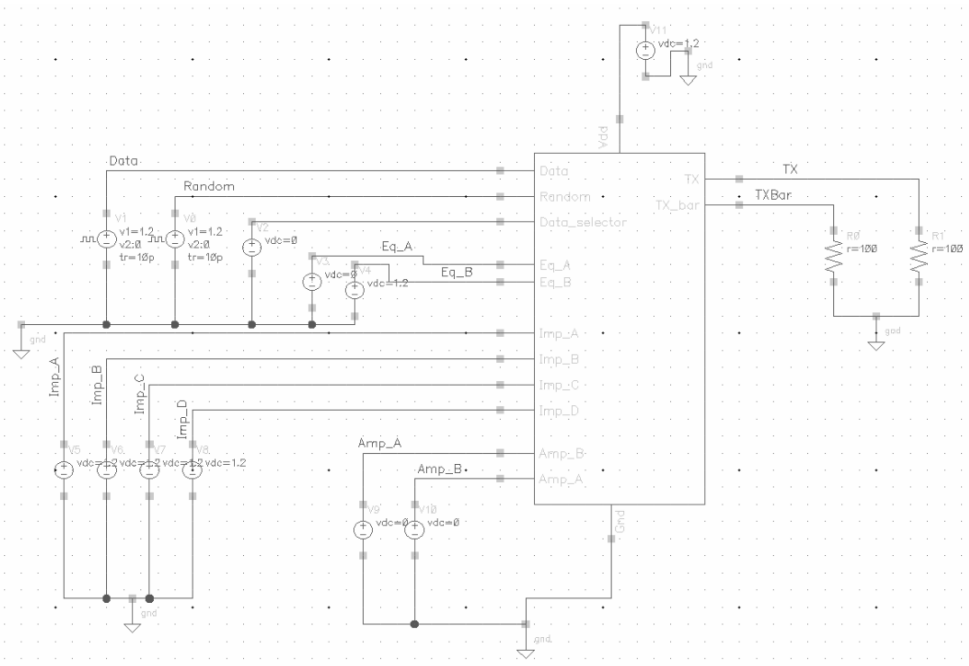


Fig. 14: Testbench for Pre emphasis response validation of Transmitter module based on its macromodel.

Fig. 15: Transient response for Pre emphasis of Transmitter module based on its macromodel.

We note on Fig.17 that Normal (Tx) and Complementary (TxBar) outputs have the same amplitude and period. It is also evident that Complementary output is 180° phase shifted with respect Normal output, which imply that macromodel of Transmitter module generates properly the expected outputs.

2.2 Design of Transmitter module in 130 nm CMOS

Fig. 18 shows the block diagram of the circuit to be designed. More detailed information of this module was presented in [8].

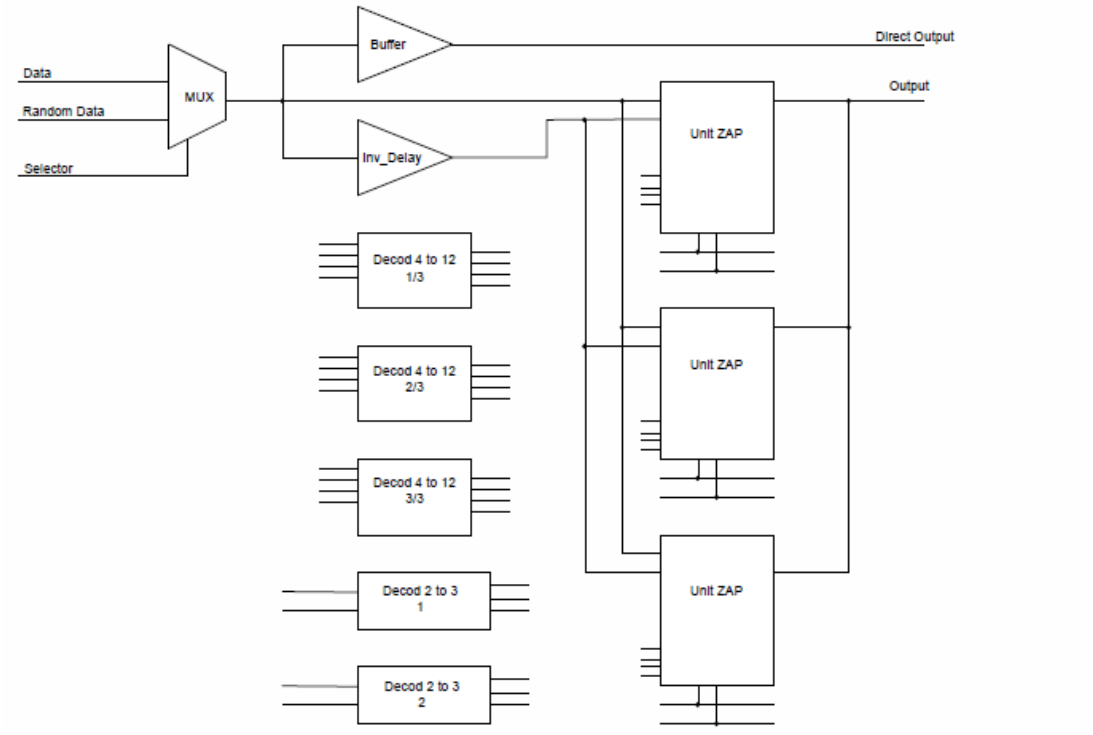


Fig. 16: Transmitter block Diagram [8].

2.2.1 Transistor level design of internal blocks

The design procedure of all of internal blocks of Transmitter module consisted in migrate previous design [8] in 180 nm to 130 nm by applying scaling theory.

$$\frac{W}{L}_{180nm} = x \qquad (Eq.2)$$

$$xL_{130nm} = W_{130nm} \qquad (Eq.3)$$

2.2.2 Inverter cell

The transistors sizes of the Inverter cell are shown in the Table 2 and Fig. 19. The size of all the transistors were obtained from [8] by applying Eq.2 and Eq.3.

Transistor		Value	Unit
T0	L	130	nm
	W	2.16	um
T1	L	130	nm
	W	860n	um

Table 2: Values PMOS and NMOS Inverter cell.

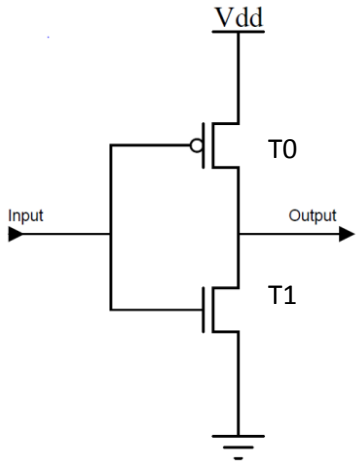


Fig. 17: Schematic at transistor level of Inverter cell.

2.2.3 NOR2 cell

The transistors' sizes of the NOR2 cell are shown in the Table 3 and Fig. 20. Eq.2 and Eq.3 were used to obtain the size of all the transistors taken from [8].

Transistor		**Value**	**Unit**
T0	L	130	nm
	W	2.6	um
T1	L	130	nm
	W	2.6	um
T2	L	130	nm
	W	430	nm
T3	L	130	nm
	W	430	nm

Table 3: Values PMOS and NMOS NOR2 cell.

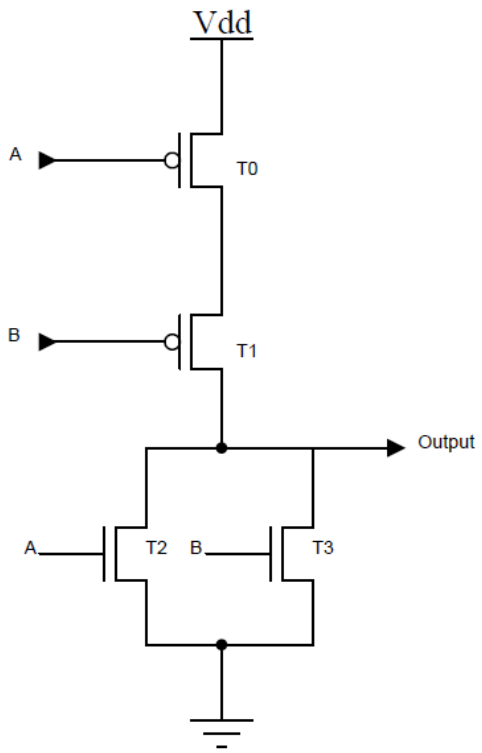


Fig. 18: Schematic at transistor level of NOR2 cell

2.2.4 NAND2 cell

The transistors' sizes of the NAND2 were calculated by applying Eq.2 and Eq. 3 to transistor' sizes from [8], final results are shown in Table 4. Fig. 21 shows the schematic at transistor level of NAN2 cell.

Transistor		Value	Unit
T0	L	130	nm
	W	2.6	um
T1	L	130	nm
	W	2.6	um
T2	L	130	nm
	W	430	nm
T3	L	130	nm
	W	430	nm

Table 4: Values PMOS and NMOS NAND2 cell.

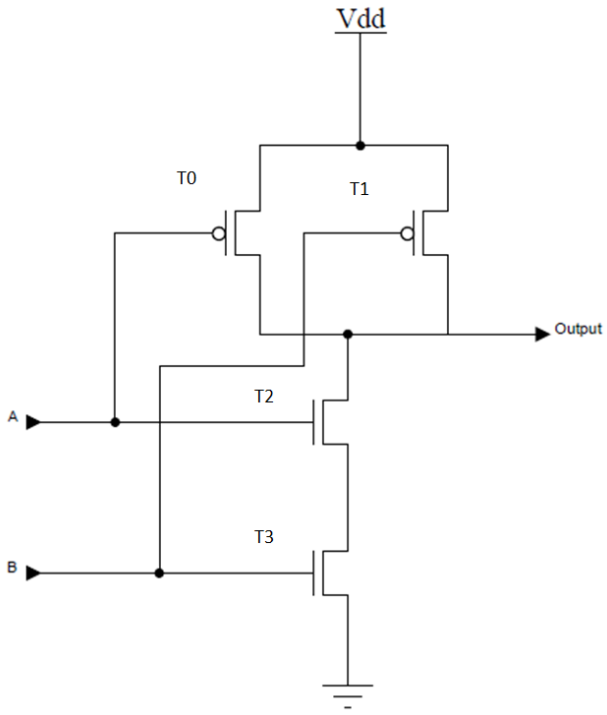


Fig. 19: Schematic at transistor level of NAND2 cell.

2.2.5 NOR3 cell

The transistors' sizes of the NOR3 cell are shown in the Table 5 and Fig. 22. The size of all the transistors were obtained from [8] by applying Eq.2 and Eq.3.

Transistor		Value	Unit
T0	L	130	nm
	W	2.6	um
T1	L	130	nm
	W	2.6	um
T2	L	130	nm
	W	2.6	um
T3	L	130	nm
	W	430	nm
T4	L	130	nm
	W	430	nm
T5	L	130	nm
	W	430	nm

Table 5: Values PMOS and NMOS NOR3 cell.

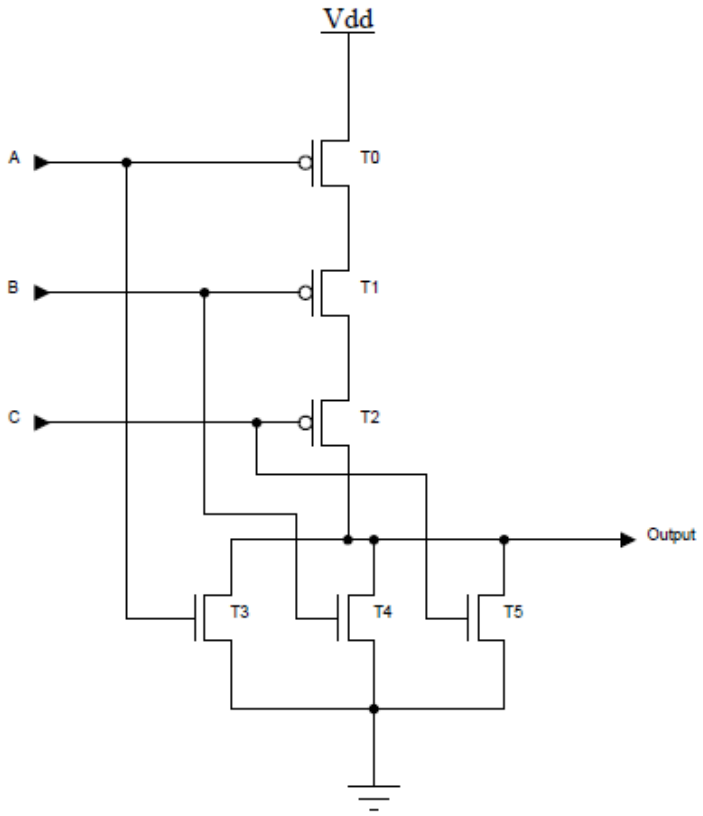


Fig. 20: Schematic at transistor level of NOR3 cell.

2.2.6 NAND3 cell

The transistors' sizes of the NAND3 are shown in the Table 6 and Fig. 23. The size of all the transistors were obtained from [8] by applying Eq.2 and Eq.3.

Transistor		Value	Unit
T0	L	130	nm
	W	2.6	um
T1	L	130	nm
	W	2.6	um
T2	L	130	nm
	W	2.6	um
T3	L	130	nm
	W	430	nm
T4	L	130	nm
	W	430	nm
T5	L	130	nm
	W	430	nm

Table 6: Values PMOS and NMOS NAND3 cell.

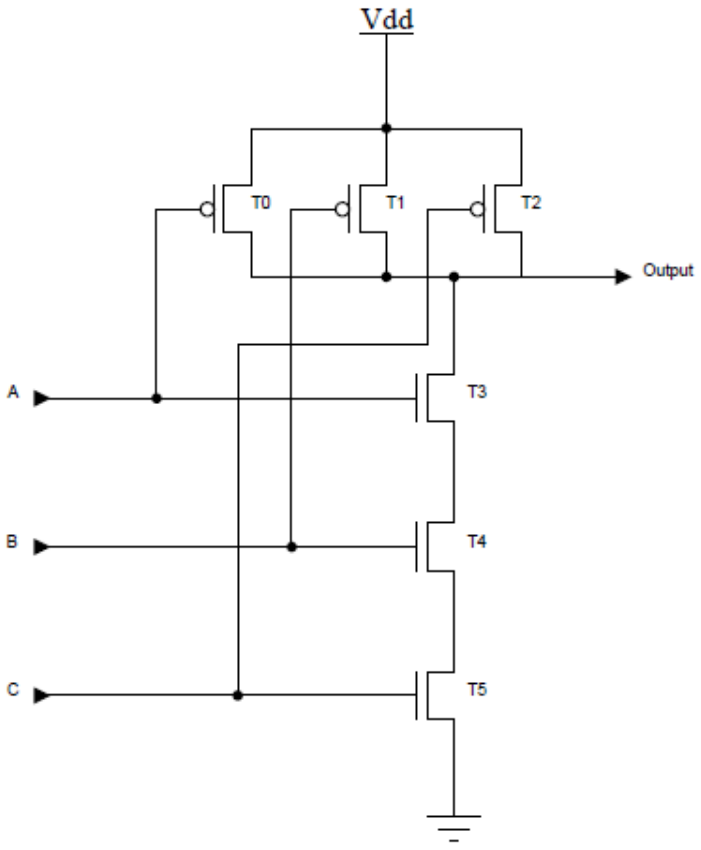


Fig. 21: Schematic at transistor level of NAND3 cell.

2.2.7 Buffer cell

The transistors' sizes of the Buffer cell were obtained by applying Eq.2 and Eq.32 to transistors' size from [8], they are shown in the Table 7 and Fig. 24.

Transistor		Value	Unit
T0	L	4	um
4 fingers	W	150	nm
T1	L	2.24	um
4 fingers	W	150	nm
T2	L	560	nm
	W	150	um
T3	L	1	um
	W	150	nm

Table 7: Values PMOS and NMOS Buffer cell.

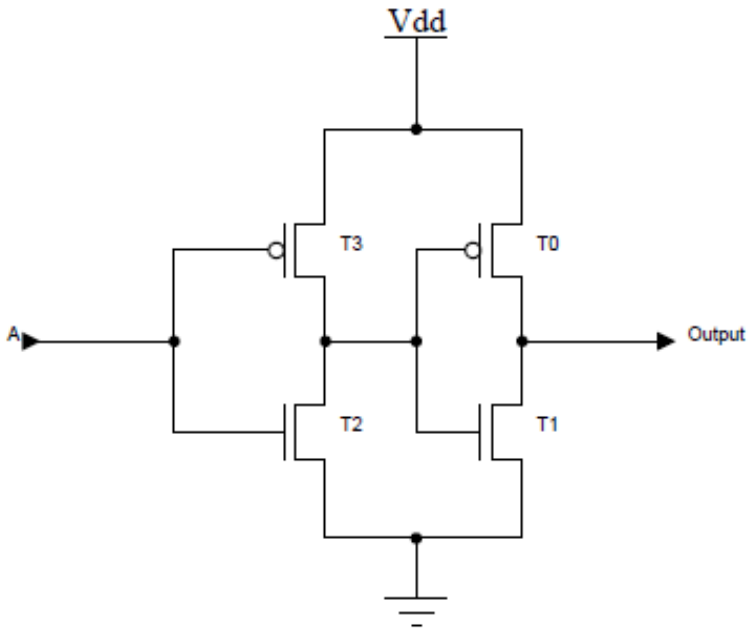


Fig. 22: Schematic at transistor level of Buffer cell.

2.2.8 3-State Buffer cell

The transistors' sizes of the 3-state Buffer are shown in the table 8 and Fig. 25. They were obtained from [8] by applying Eq. 2 and Eq. 3.

Transistor		Value	Unit
T0	L	130	nm
fingers 6	W	17.4	um
T1	L	130	nm
fingers 6	W	17.4	um
T2	L	560	nm
fingers 6	W	19.08	um
T3	L	130	nm
fingers 2	W	6.5	um
T4	L	130	nm
fingers 2	W	6.5	um
T5	L	130	nm
fingers 2	W	5.92	um
T6	L	730	nm
fingers 2	W	5.92	um

Table 8: Values of PMOS and NMOS 3-State Buffer cell.

To develop pseudo analog cells such as Multiplexer 2-1, 3-State Buffer, Delay inverter, we applied the same consideration for digital gates; we take in to account the transistors' sizes of such cells presented in work [5] and applying Eq.1 and Eq.2.

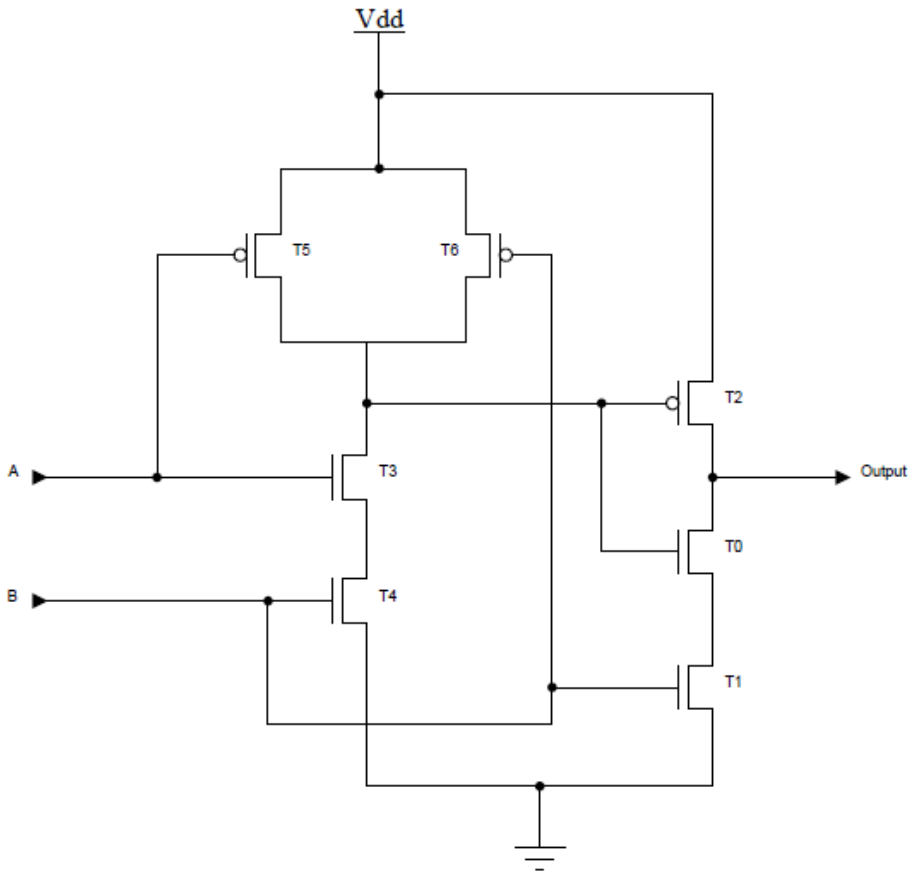


Fig. 23: Transistor-level schematic of 3-State Buffer cell.

2.2.9 DELAY INVERTER cell

The transistors' sizes of the Delay Inverter Cell were obtained from [8] by applying Eq. 2 and Eq. 3. They are shown in the table 9 and Fig. 26.

Transistor		Value	Unit
T0	L	260	nm
	W	2.6	um
T1	L	260	nm
	W	1.17	um
T2	L	150	nm
	W	450	um
T3	L	150	nm
	W	900	um
T4	L	130	nm
fingers 4	W	11.28	um
T5	L	130	nm
fingers 4	W	5.2	um

Table 9: Values PMOS and NMOS Delay Inverter.

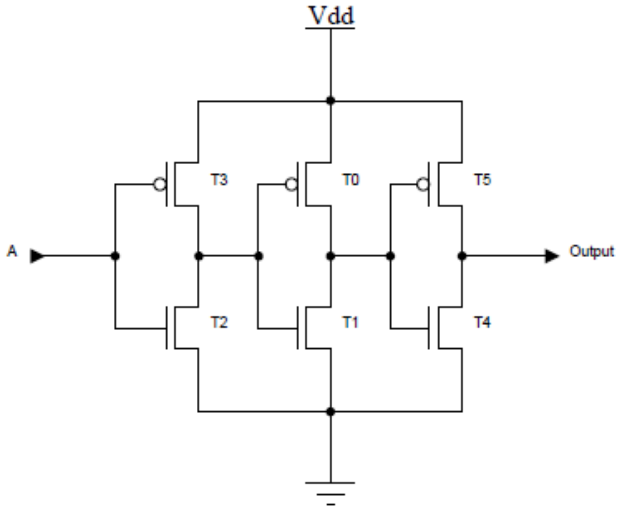


Fig. 24: Schematic at transistor level of DELAY INVERTER cell.

2.2.10 Multiplexer 2-1 Cell

To obtain the transistors' sizes of Multiplexer 2-1 cell we apply Eq. 2 and Eq. 3 to transistors' size from [8], results are shown in the table 10 and Fig. 27.

Transistor		Value	Unit
T0	L	150	nm
fingers 2	W	680	nm
T1	L	150	nm
	W	300	nm
T2	L	150	nm
fingers 2	W	680	nm
T3	L	150	nm
	W	300	nm
T4	L	150	nm
fingers 2	W	680	nm
T5	L	150	nm
fingers 2	W	680	nm
T6	L	150	nm
fingers 3	W	1.2	um
T7	L	150	nm
	W	300	nm
T8	L	150	nm
fingers 3	W	1.2	um
T9	L	150	nm
fingers 3	W	1.2	um
T10	L	150	nm
	W	400	nm
T11	L	150	nm
fingers 3	W	1.2	um

Table 10: Values PMOS and NMOS Multiplexer 2-1.

The schematic of Multiplexer 2-1 is shown in Fig. 27.

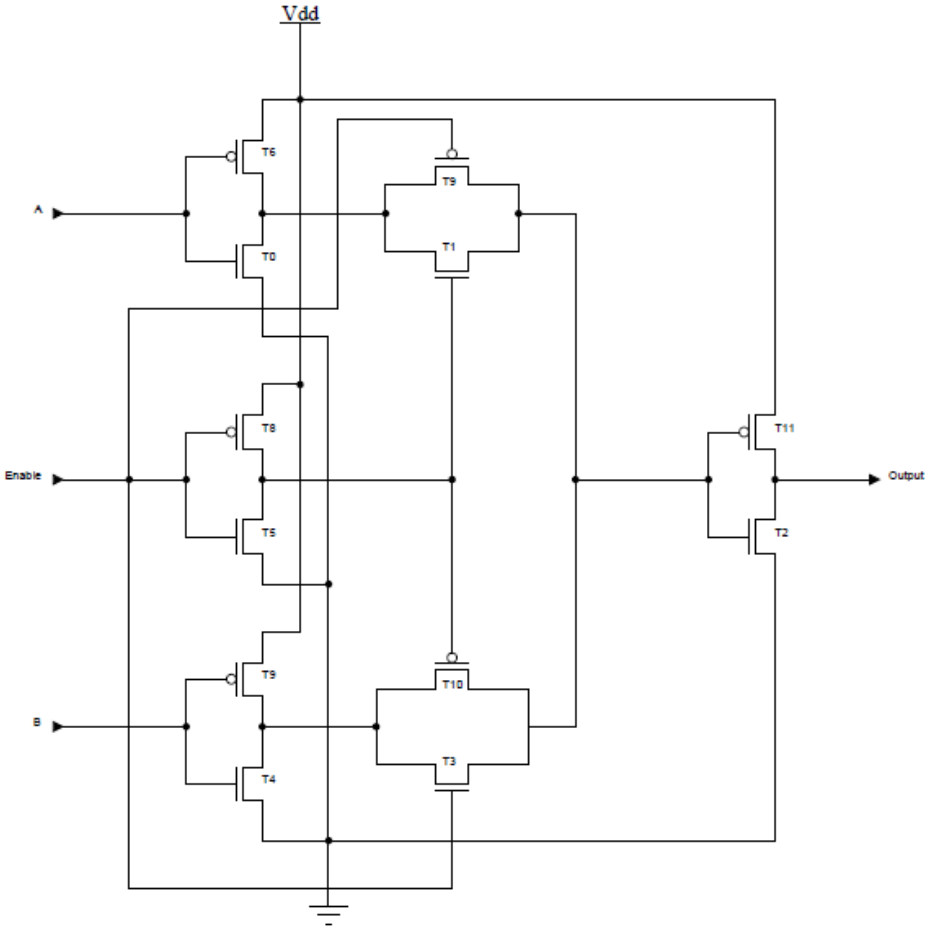


Fig. 25: Schematic at transistor level of Multiplexer 2-1 cell.

2.2.11 Tied Circuit

The transistors' sizes of the Tied circuit cell are shown in the table 11. The size of all the transistors were obtained from [8] by applying Eq.2 and Eq.3.

Transistor		Value	Unit
T0	L	130	nm
	W	390	nm
T1	L	130	nm
	W	390	nm
T2	L	130	nm
	W	390	nm
T3	L	130	nm
	W	700	nm
T4	L	130	nm
	W	700	nm
T5	L	130	nm
	W	700	nm

Table 11: Values of PMOS and NMOS Tied Circuit.

The schematic of Tied circuit is shown in Fig. 28.

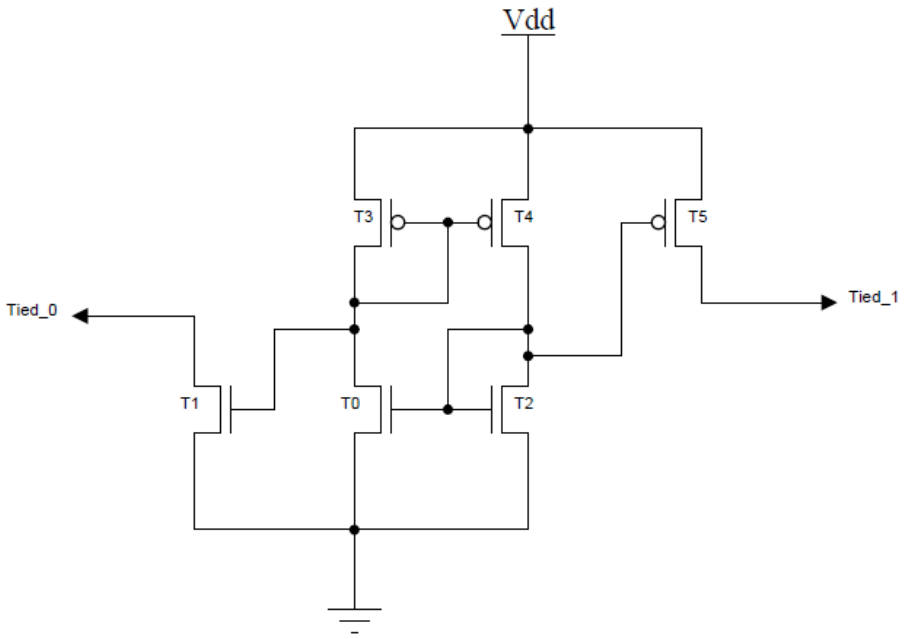


Fig. 26: Schematic at transistor level of Inverter cell Tied Circuit.

2.2.12 Decoder 2-3

We assemble Decoder 2-3 block using low-level digital cells (NOR, NAND, INVERTER), the schematic is shown in Fig. 29.

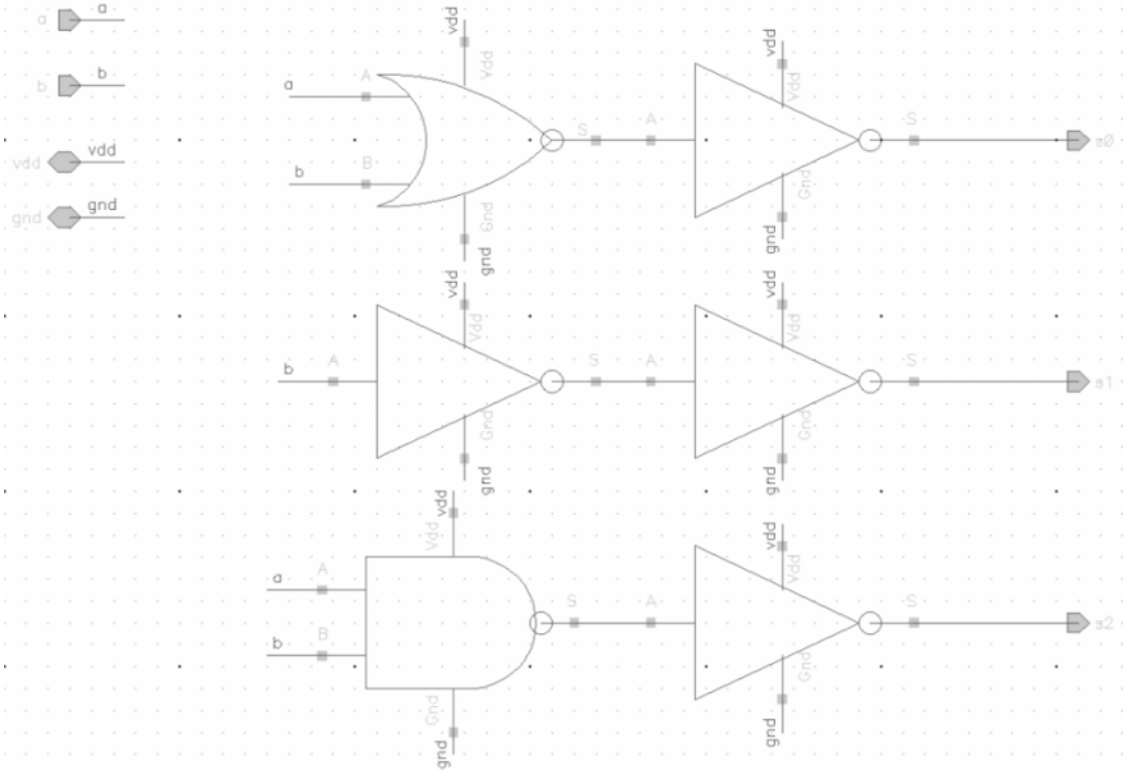


Fig. 27: Schematic of Decoder 2-3.

2.2.13 Decoder 4-12

This block is divided in three parts, the principal reason of doing this is to separate all the outputs (from s0 to s3, s4 to s7 and s8 to s11) to make easy the final layout of the cell and prevent errors during the process of design.

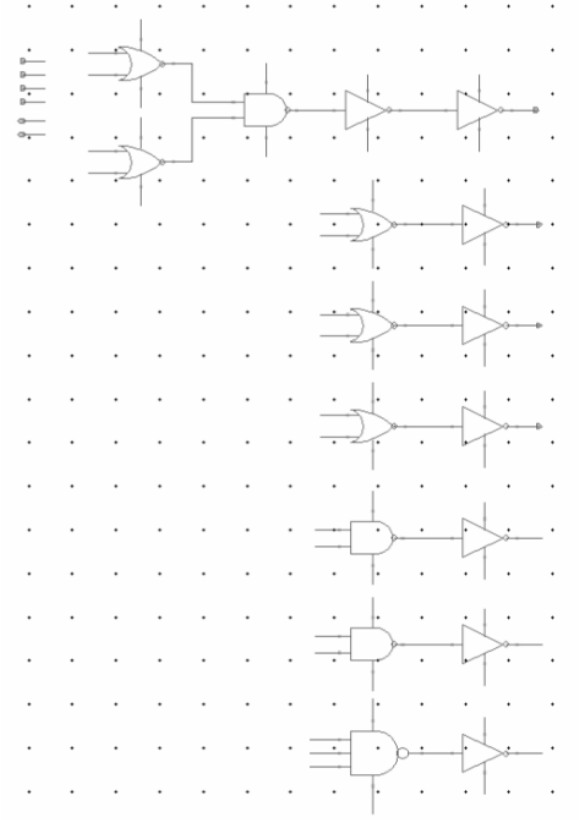


Fig. 28: Block 1 s0 to s3.

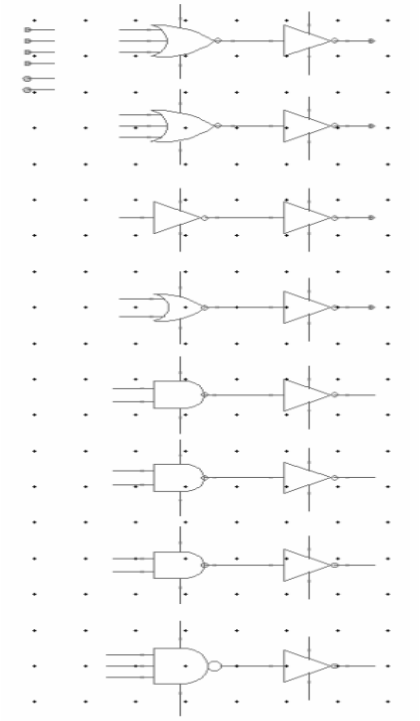


Fig. 29: Block 2 s8 to s11.

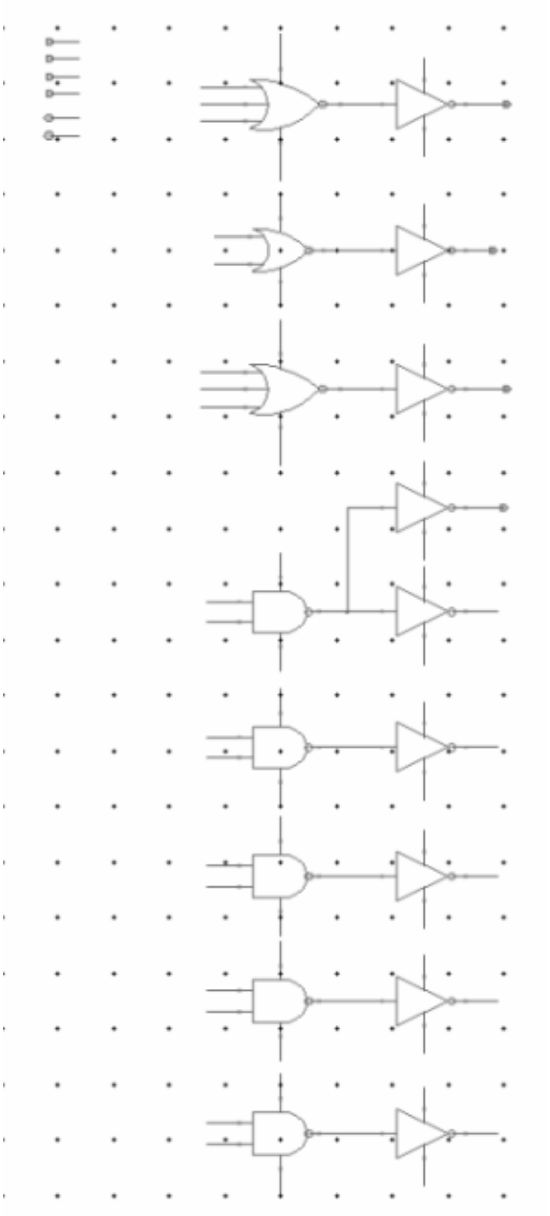


Fig. 30: Block 3 s4 to s7.

2.2.14 ZAP UNIT

As described in [8] this block controls the equalization characteristics of the Transmitter (Amplitude, impedance and Pre-emphasis). The base of this module is a sub module called Basic cell [5], the schematic of the ZAP UNIT is shown in Fig. 33.

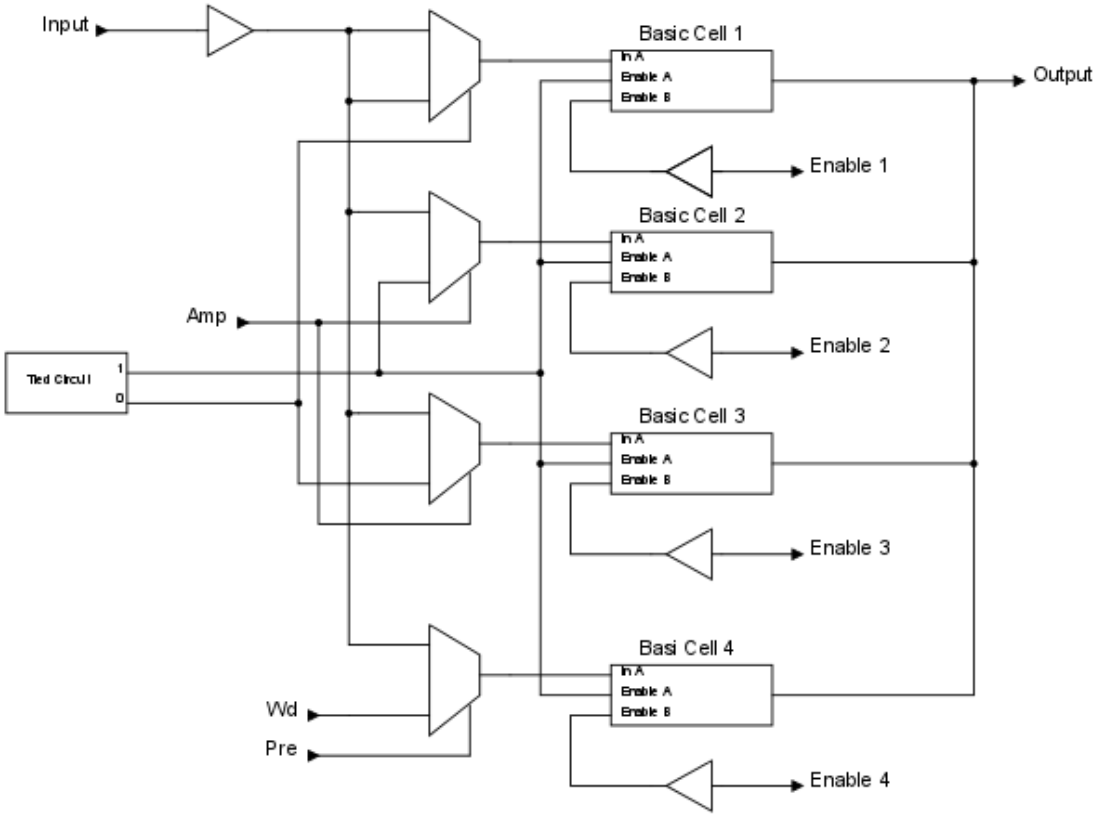


Fig. 31: Schematic of ZAP UNIT.

2.3 Architecture of Transmitter module with Normal and Complementary outputs

In [8], only presents the Transmitter module with a single output, in order to generate both Normal (Tx) and Complementary (TxBar) outputs, we have folded the Transmitter module and two inverters were added at the input of second block to generate the complementary output. In Fig. 34 left part of the circuit generates Tx output while right circuitry generates the TxBar output. The control pins for impedance, pre emphasis and amplitude remains the same for both blocks.

The final architecture of the Transmitter Module with Tx and TxBar outputs designed in 130 nm CMOS technology is shown in Fig. 34.

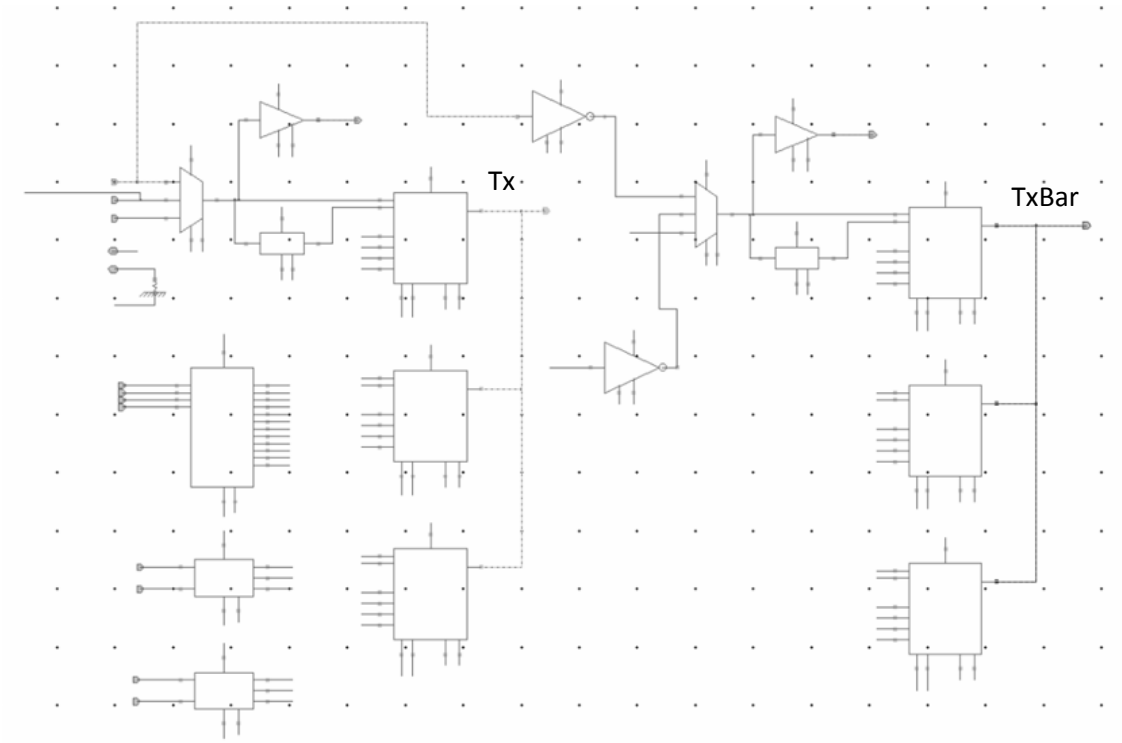


Fig. 32: Schematic of Transmitter module with complementary outputs assembled with internal blocks designed in 130 nm CMOS technology.

CHAPTER 3: PRE-LAYOUT VERIFICATION OF TRANSMITTER MODULE.

3.1 Testbenches of Transmitter modules.

Due to change of technology used in the SerDes design, rise time and fall time were compared against data of work in [8], therefore all the testbenches made in 180nm were used to verify the functionality of each cell designed in 130 nm. We observed a short circuit current in the transient responses, when transistors switch, both nMOS and pMOS networks may be shortly *On* state at same time, this component is generally ignored.

3.1.1 Testbench of Inverter cell

Fig. 35 shows the testbench of Inverter Cell.

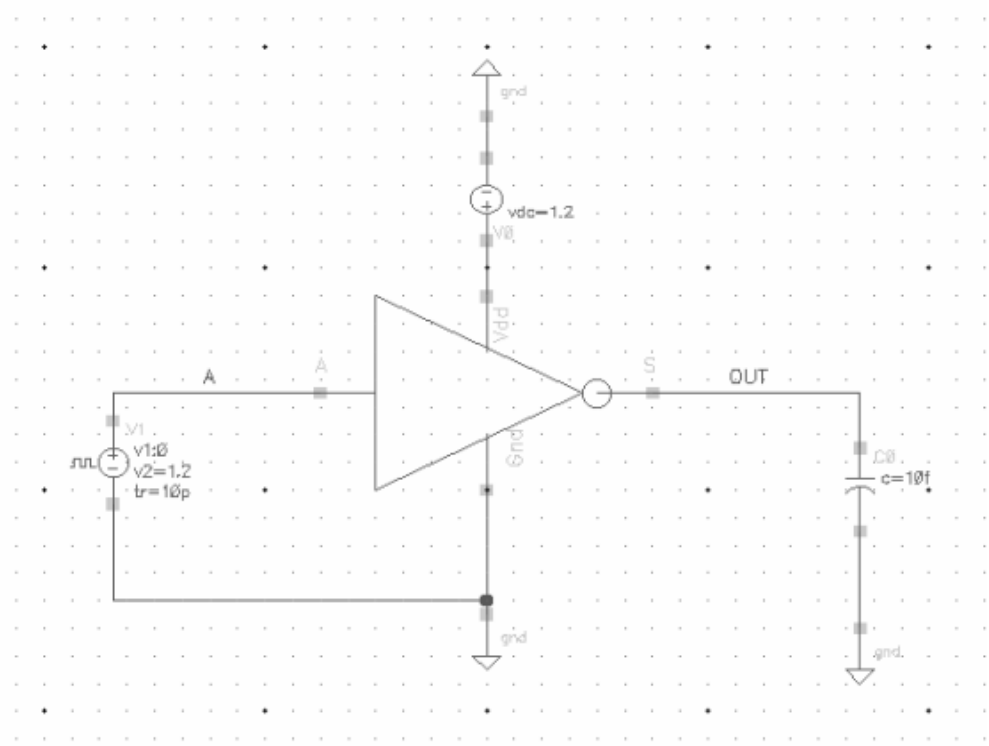


Fig. 33: Testbench of Inverter cell.

The transient response of inverter cell is shown in Fig. 36.

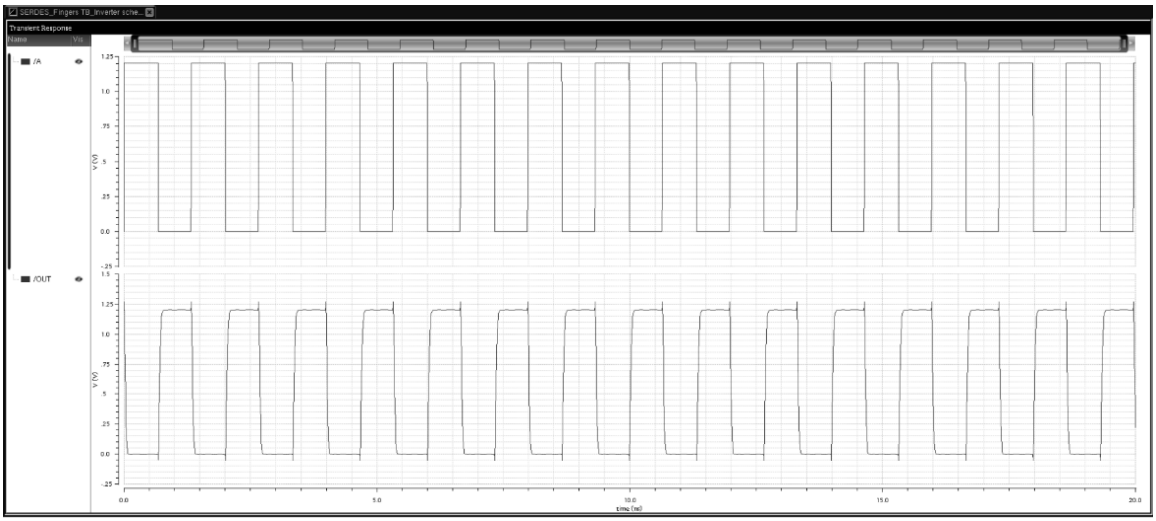


Fig. 34: Transient response of inverter cell.

Rise time and fall time obtained in this work are similar to the obtained in [8], just the amplitude of output signal is different due to the technology used (130nm).

3.1.2 Testbench of NOR2 cell

Fig. 37 shows the testbench of NOR2 Cell.

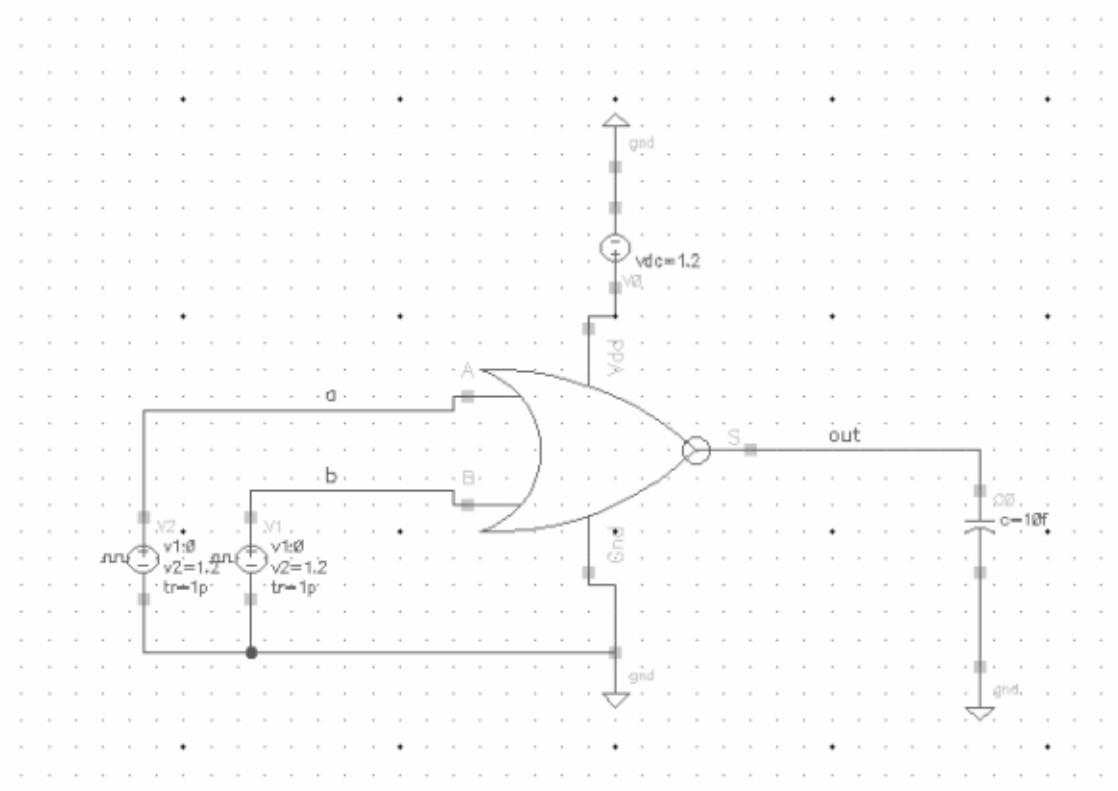


Fig. 35: Testbench of NOR2 cell.

The transient response is shown in Fig. 38.

Fig. 36: Transient response of NOR2.

The transient response (true table) was compared with the obtained one in 180nm [8], we observed that both responses were similar.

3.1.3 Testbench of NAND2 cell.

Fig. 39 shows the testbench of NAND2 Cell.

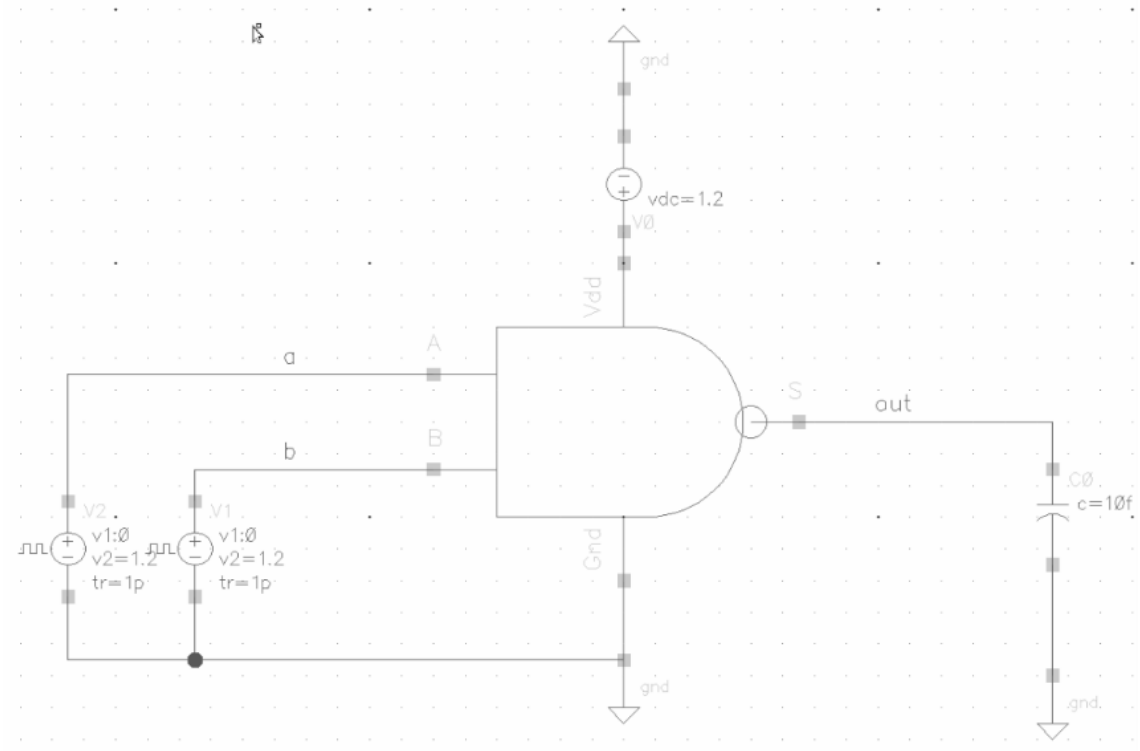


Fig. 37: Testbench of NAND2 cell.

The Transient response is shown in Fig 40.

Fig. 38: Transient response of NAND2.

This transient response shows the output of the NAND2 cell, it was compared with that obtained in [8] using the same configuration and the result is similar.

3.1.4 Testbench of NOR3 cell.

Fig. 41 shows the testbench of NOR3 Cell.

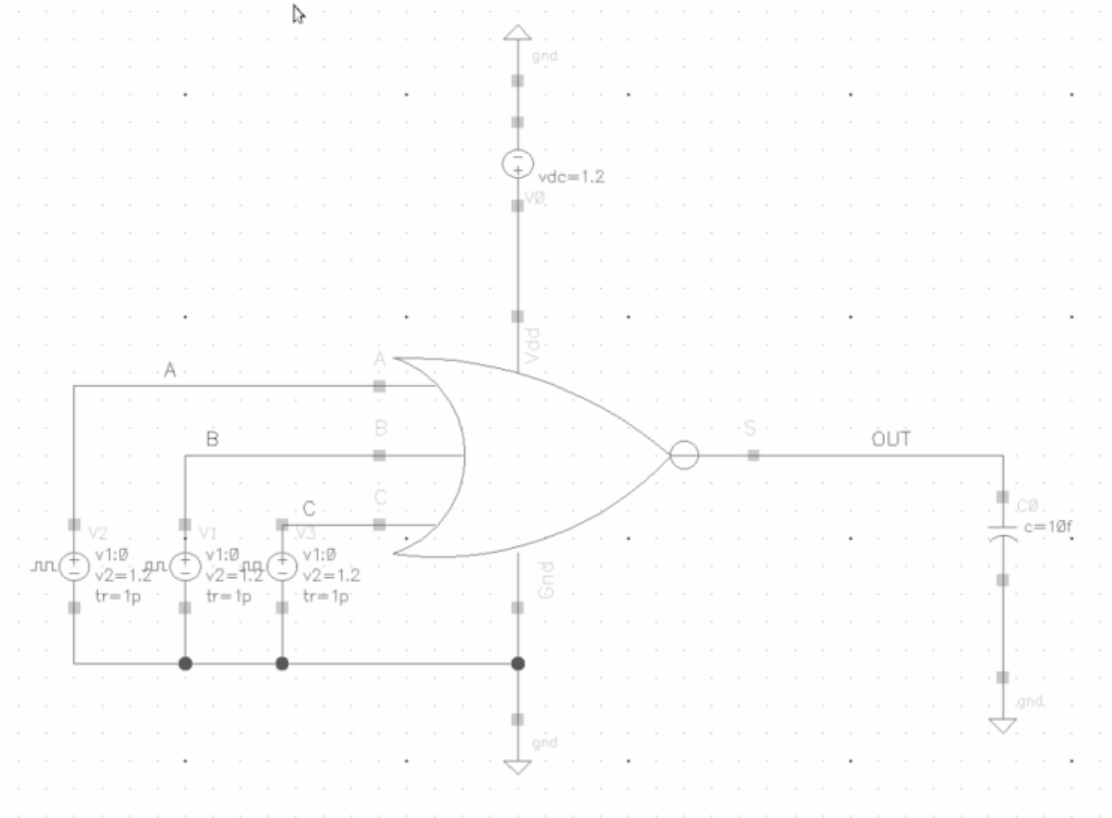


Fig. 39: Testbench of NOR3 cell

The transient response is shown in Fig. 42.

Fig. 40: Transient response NOR3.

In this transient response we can probe the true table of the NOR3.

3.1.5 Testbench of NAND3 cell.

Fig. 43 shows the testbench of NAND2 Cell.

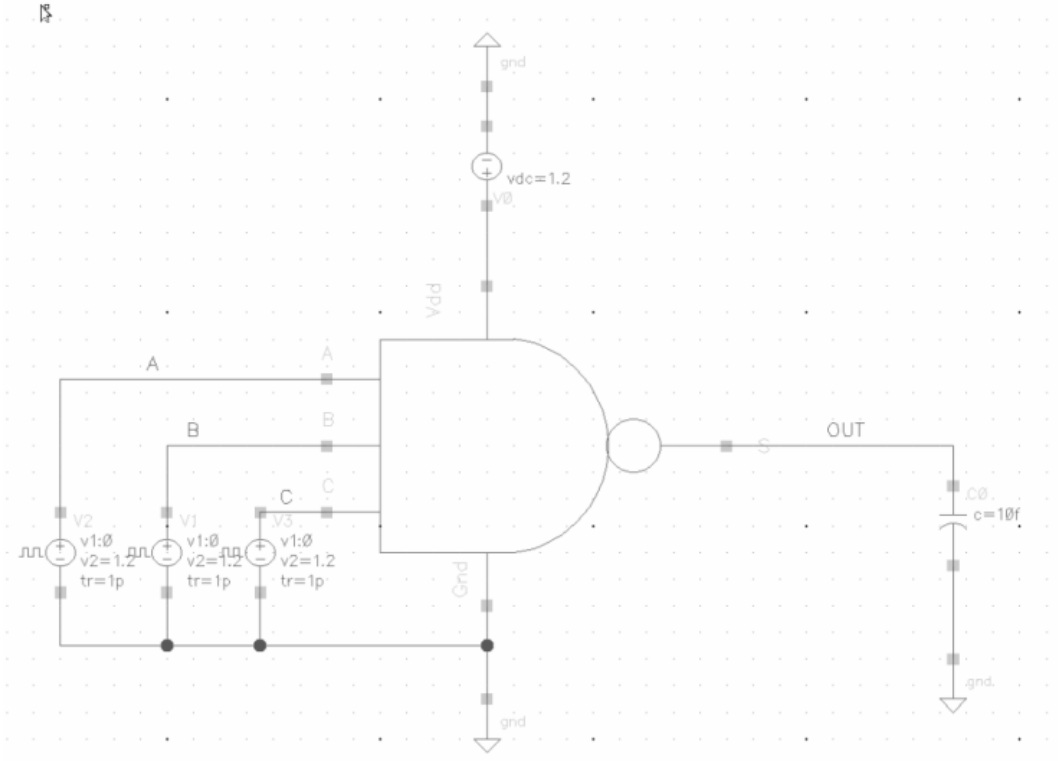


Fig. 41: Testbench of NAND3 cell.

The transient response is shown in Fig. 44.

Fig. 42: Transient response of NAND3 cell.

The transient response was compared with the obtained one in 180nm [8], we observed that both responses were similar.

3.1.6 Testbench of Buffer cell.

Fig. 45 shows the testbench of Buffer Cell.

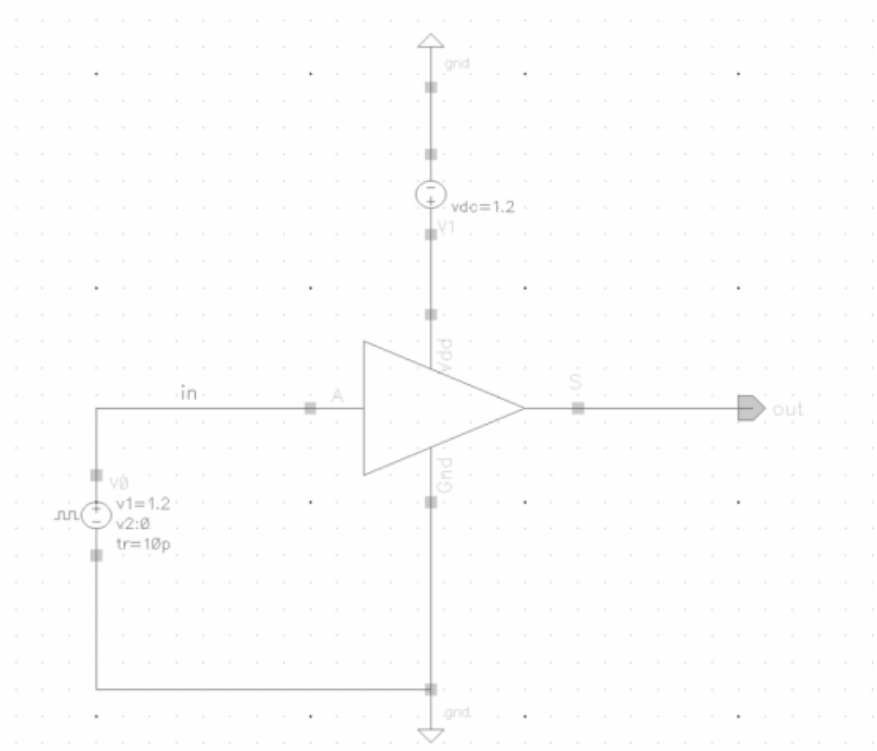


Fig. 43: Testbench of Buffer cell.

The transient response is shown in Fig. 46.

Fig. 44: Transient response Buffer cell.

The transient response shows the result expected, this cell gives strength to the signal to be processed.

3.1.7 Testbench of 3-State Buffer Cell.

Fig. 47 shows the testbench of 3-State Buffer Cell.

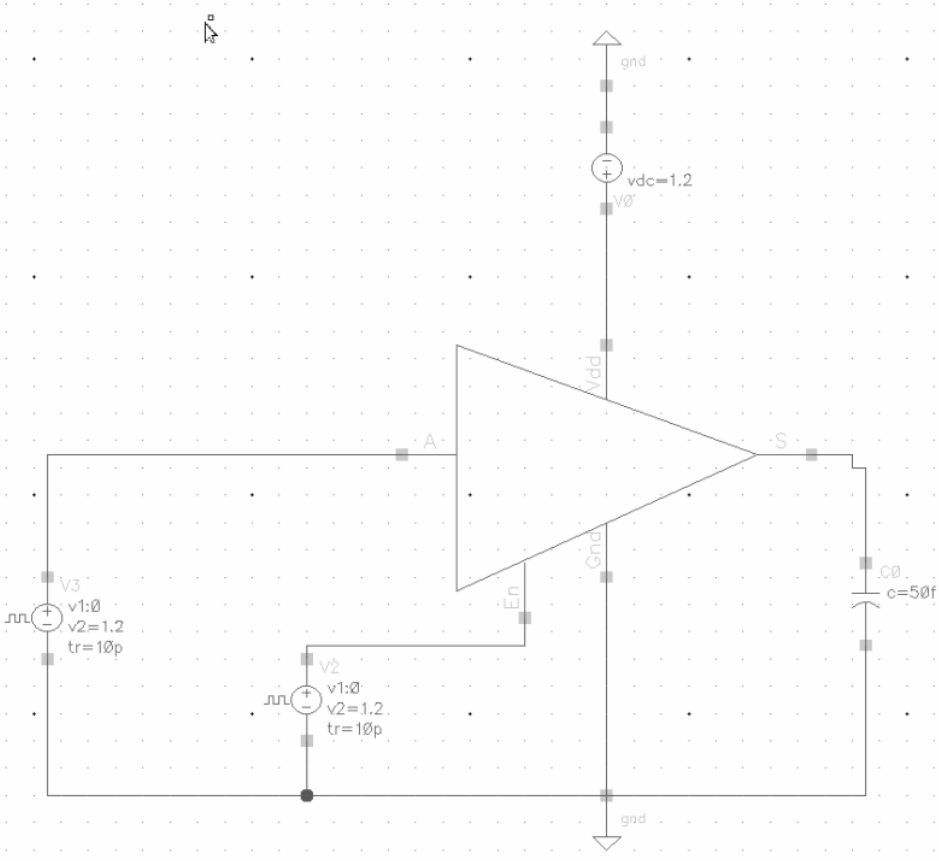


Fig. 45: Test bench 3-state Buffer cell.

The transient response of 3-state buffer is shown in Fig. 48.

Fig. 46: Transient response of 3-state Buffer.

We can observe how the buffer keeps in third state when is not enable. When enable switch was activated, the output changes and let pass the input signal. After turn off the enable switch, it backs to the third state.

3.1.8 Testbench of DELAY INVERTER cell.

Fig. 49 shows the testbench of Delay Inverter Cell

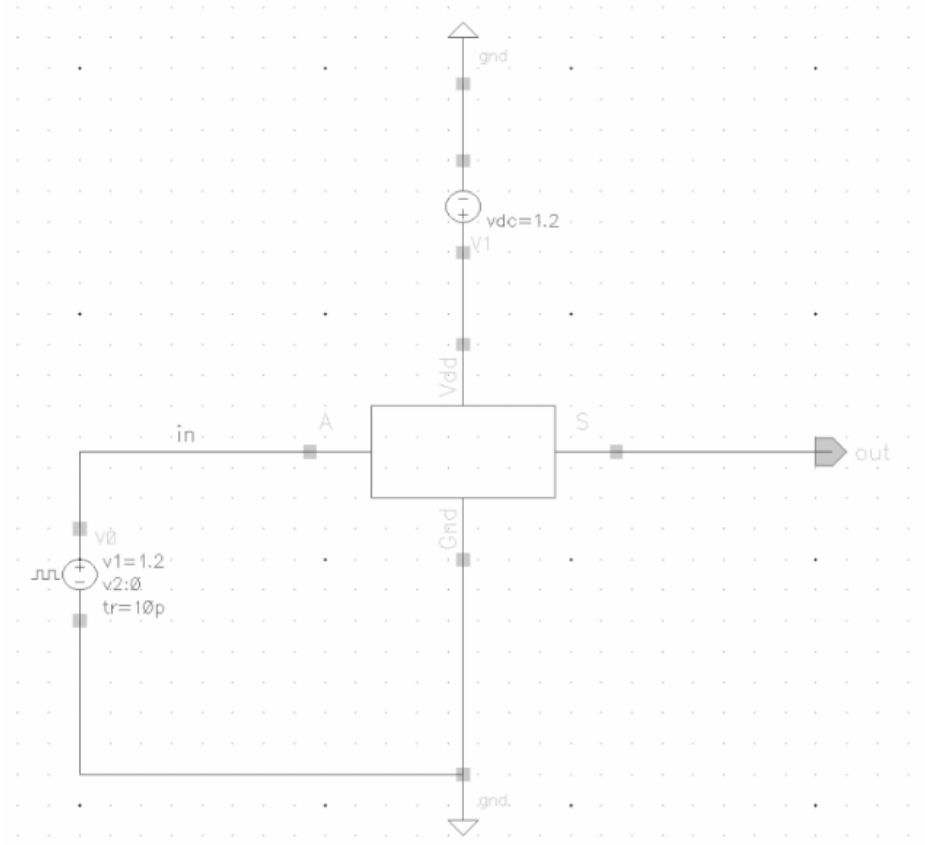


Fig. 47: Testbench of Delay Inverter cell.

The transient response of Delay Inverter is shown in Fig. 50.

Fig. 48: Transient response of Delay Inverter.

The transient response is inverted and has a delay of 10% from the input signal [8] necessary to obtain the characteristics of pre emphasis, amplitude and impedance of the Transmitter module.

3.1.9 Testbench of Multiplexer 2-1 cell.

The test bench is shown in Fig. 51. The transient response is shown in Fig. 52.

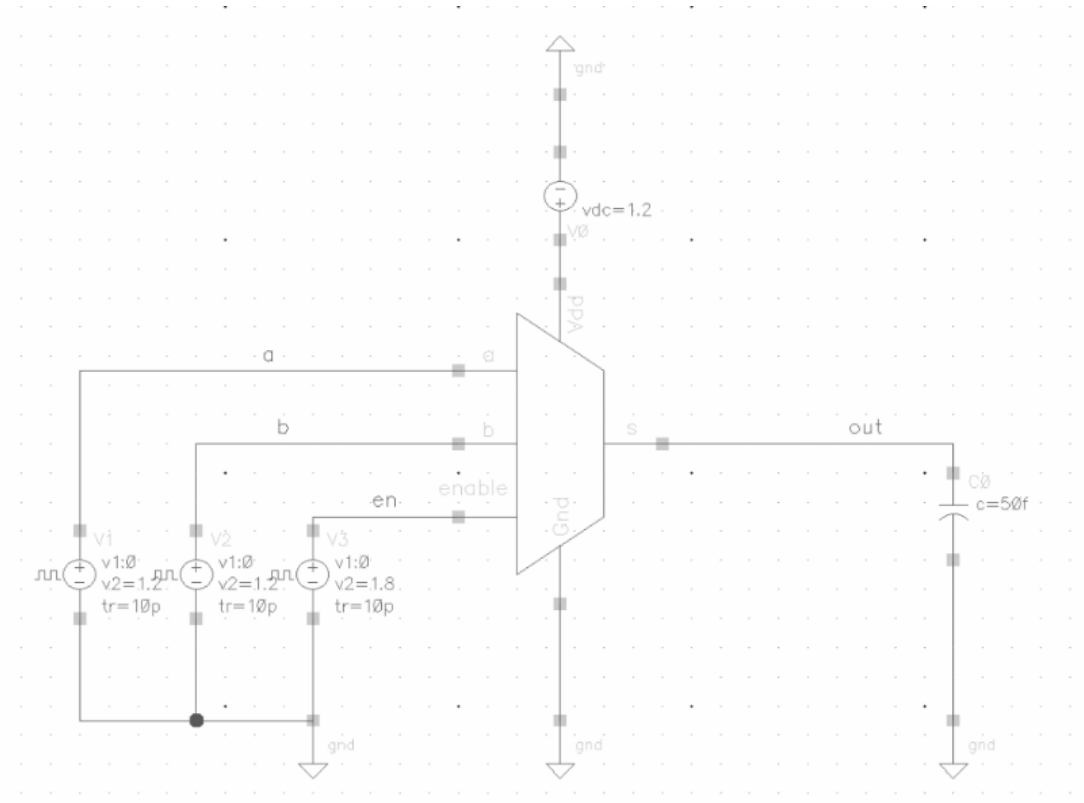


Fig. 49: Test Bench of Multiplexer 2-1.

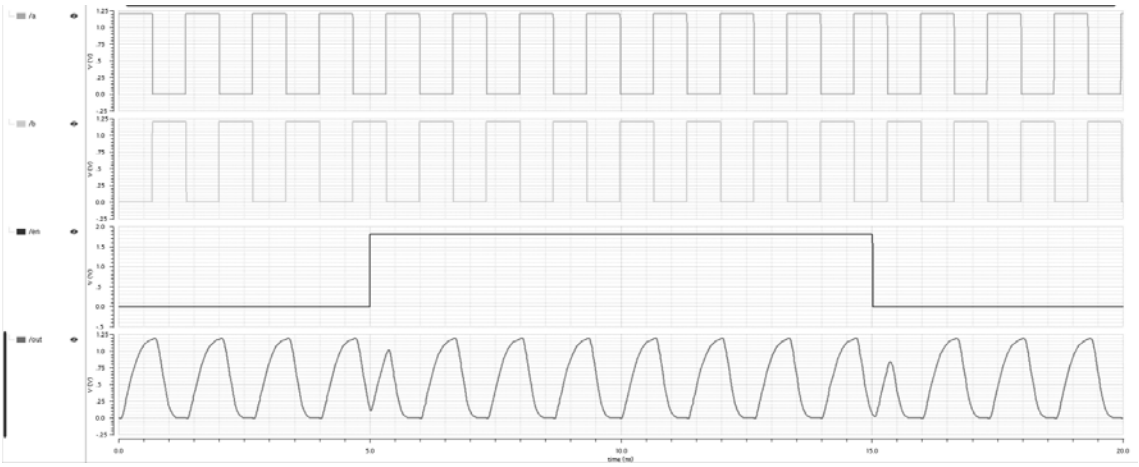


Fig. 50: Transient Analysis of Multiplexer 2-1.

The transient response was compared with the obtained one in 180nm [8], we observed that both responses were similar. The response is not fast enough to complete the transition so a faster design of the Multiplexer should be done in a future work.

3.1.10 Testbench of Tied Circuit.

Fig. 53 shows the testbench of Tied Circuit.

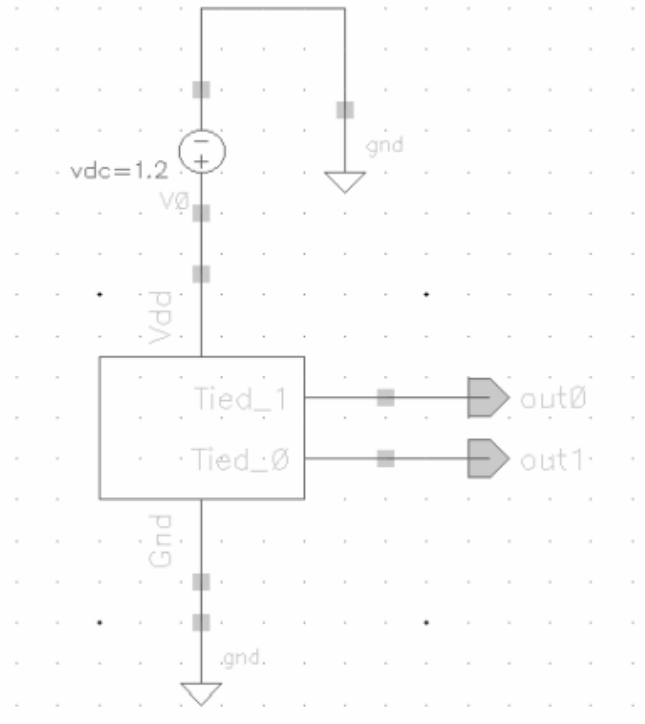


Fig. 51: Testbench of Tied circuit.

The transient response is shown in Fig. 54.

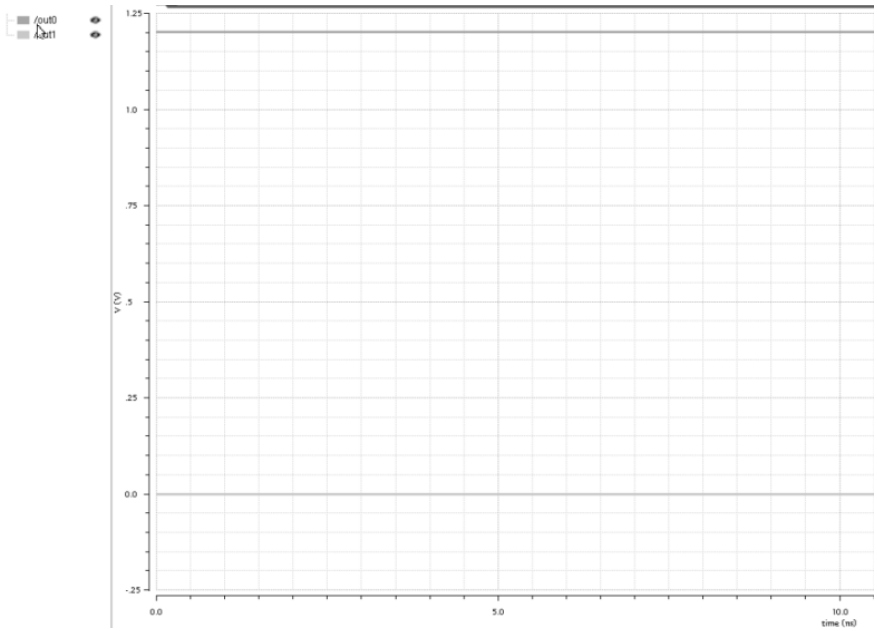


Fig. 52: Transient response of Tied circuit.

As expected, logic 1 occurs in one output and a logical 0 is presented in the other output.

3.1.11 Testbench of Decoder 2-3.

The test Bench is shown in Fig. 55. The transient response is shown in Fig. 56.

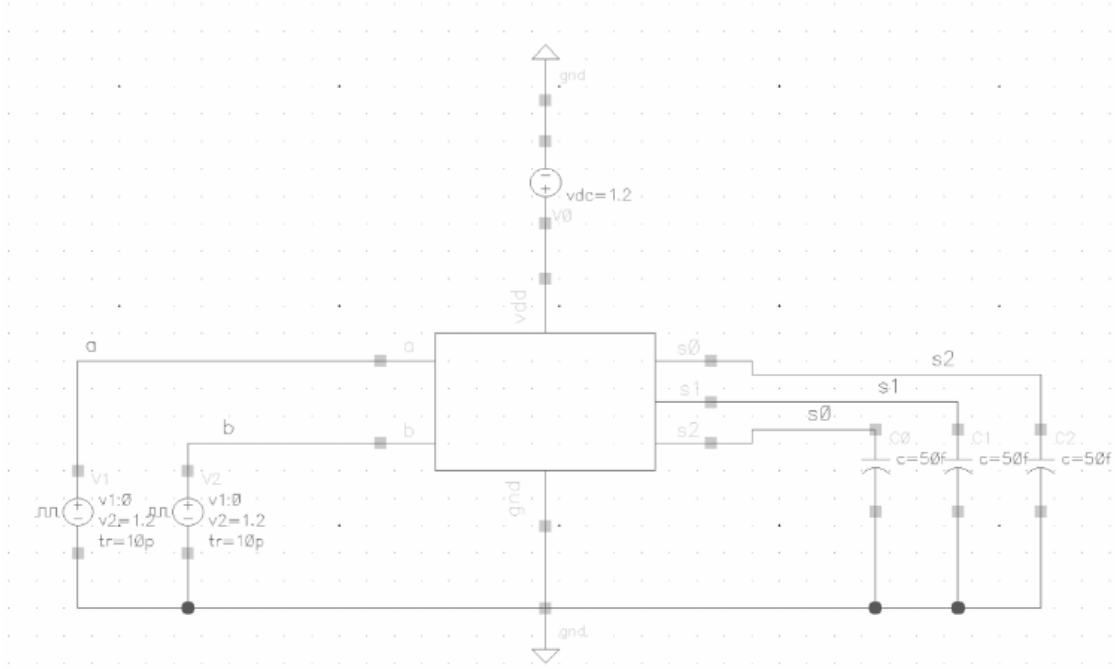


Fig. 53: Test bench of Decoder 2-3.

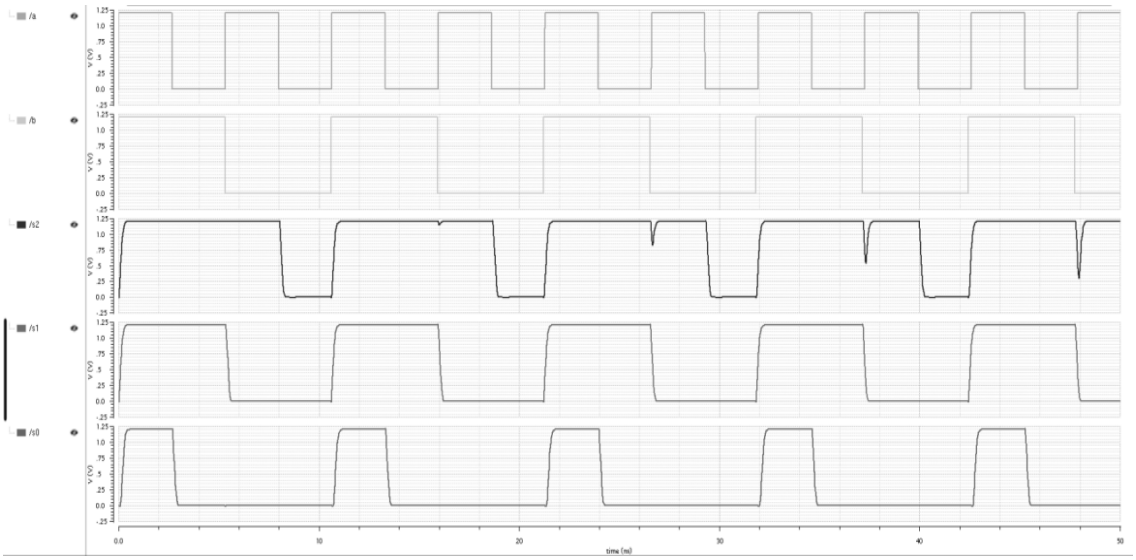


Fig. 54: Transient response of Decoder 2-3.

In this transient response we can observe the behavior of thermometer as described in [5].

3.1.12 Testbench of Decoder 4-12.

The test bench of Decoder 4-12 including all its internal blocks is presented in Fig. 30, Fig.31 and Fig.32. The transient response is shown in Fig. 57.

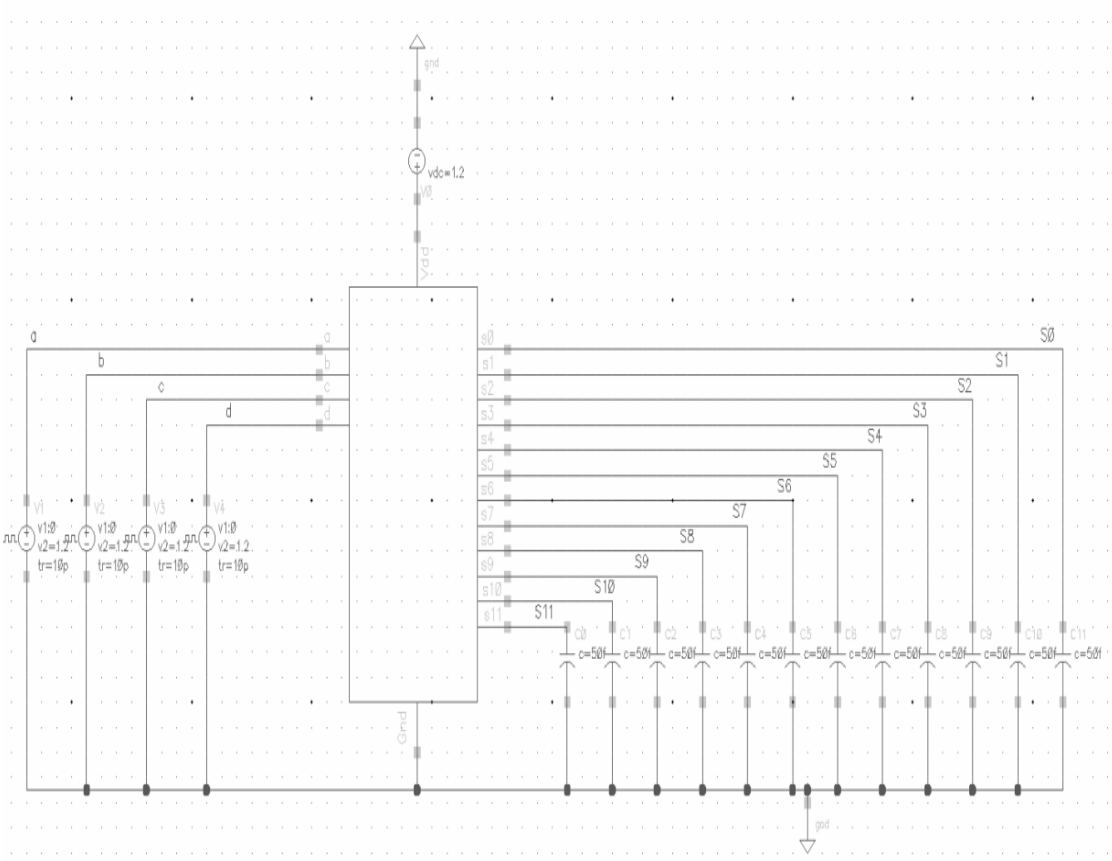


Fig. 55: Test bench of Decoder 4-12.

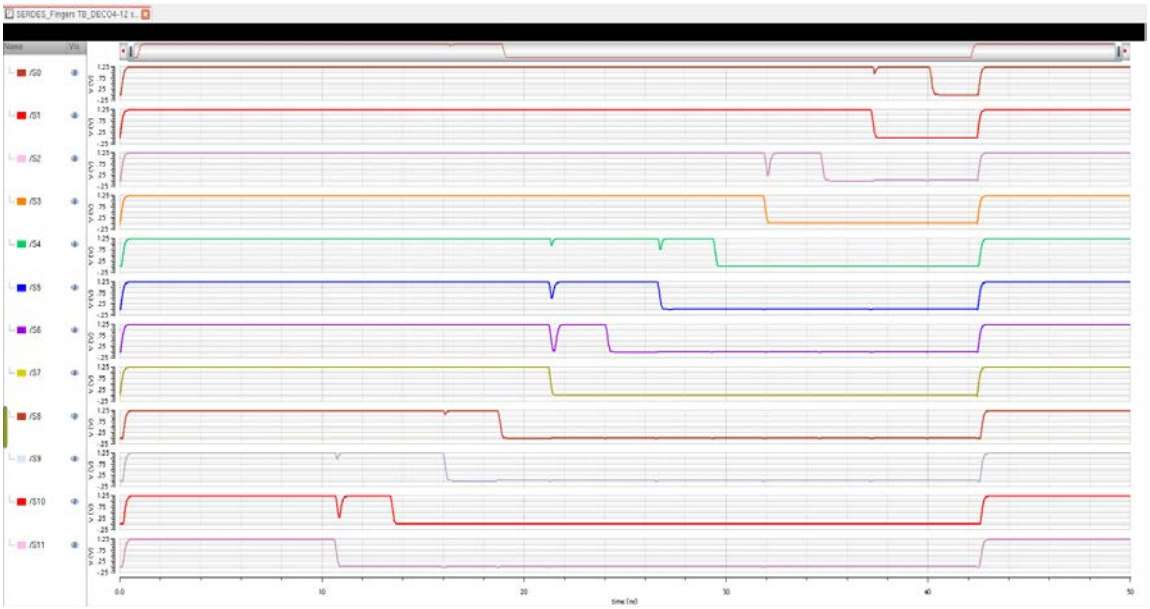


Fig. 56: Transient response of Decoder 4-12.

The transient response shows that the table presented in Fig. 11 is satisfied.

3.1.13 Testbench of ZAP UNIT.

The schematic of the ZAP UNIT was shown in Fig. 33. The testbench to verify the amplitude control of ZAP UNIT is shown in Fig. 59.

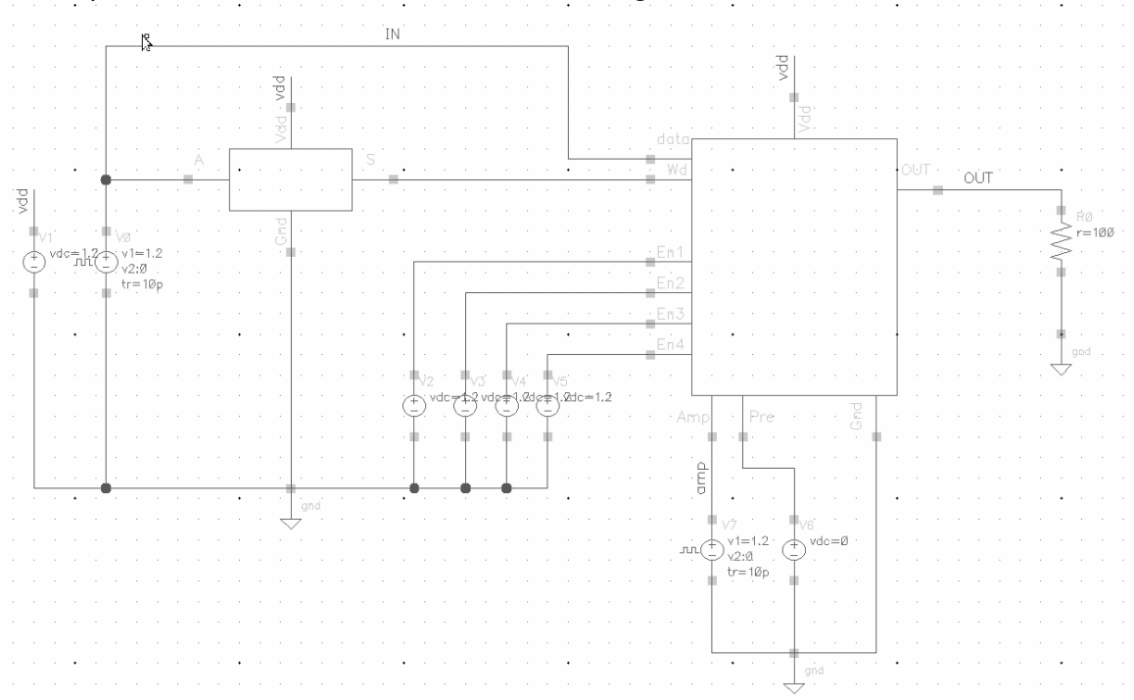


Fig. 57: Testbench of ZAP UNIT for amplitude response

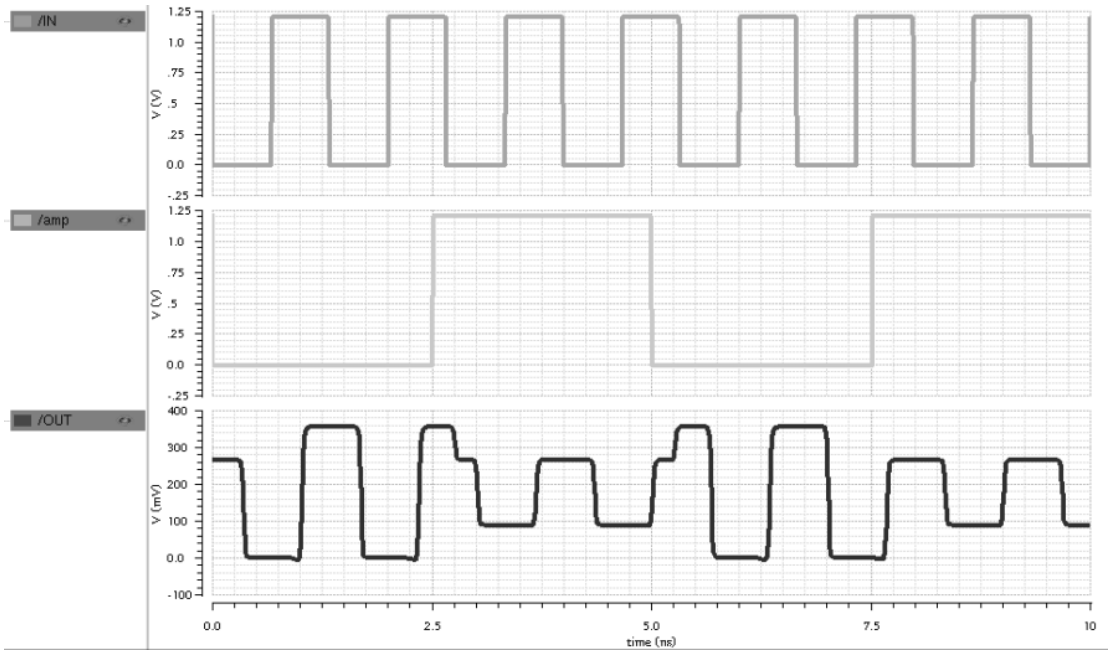


Fig. 58: Transient response (amplitude) of ZAP UNIT.

Fig. 60 shows the change in amplitude when one of the two controls is activated in the ZAP UNIT. 4 cases are considered, in this section only the first case is presented, other cases were validated in the response of the transmitter module.

Testbench to simulate impedance control of ZAP UNIT is shown in Fig. 61.

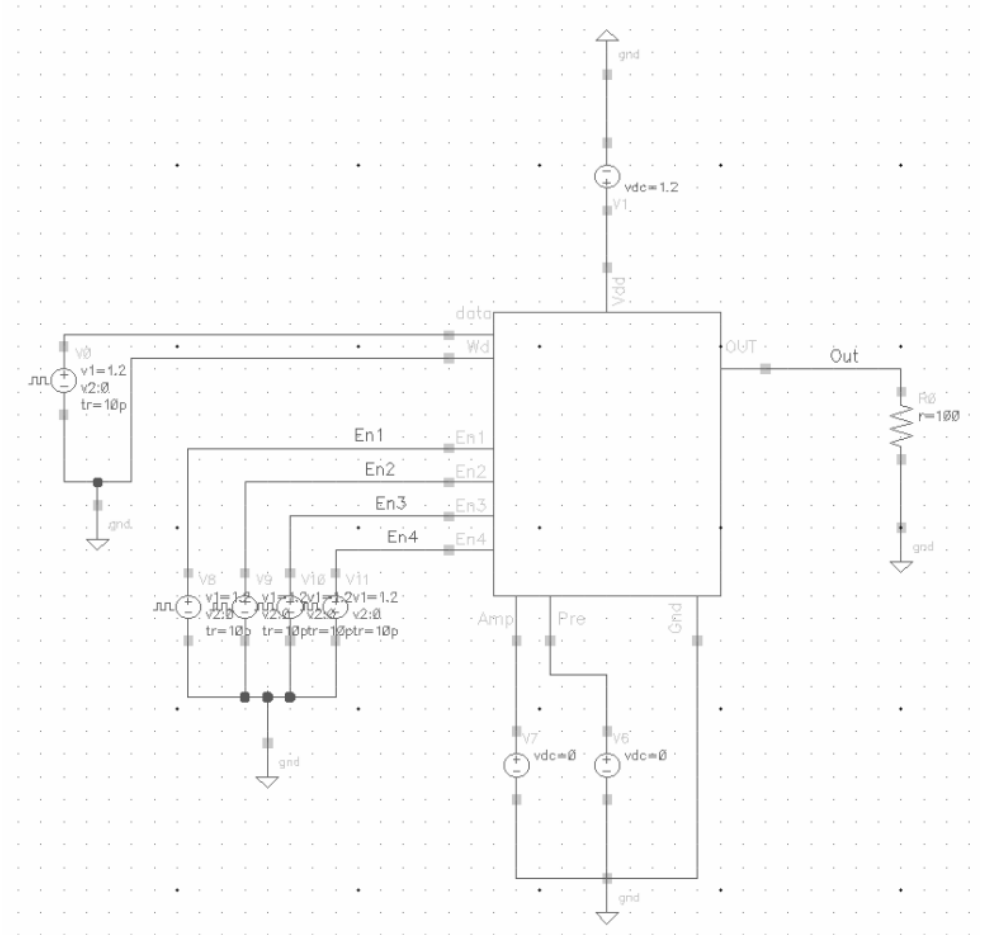


Fig. 59: Testbench to simulate the impedance response of ZAP UNIT.

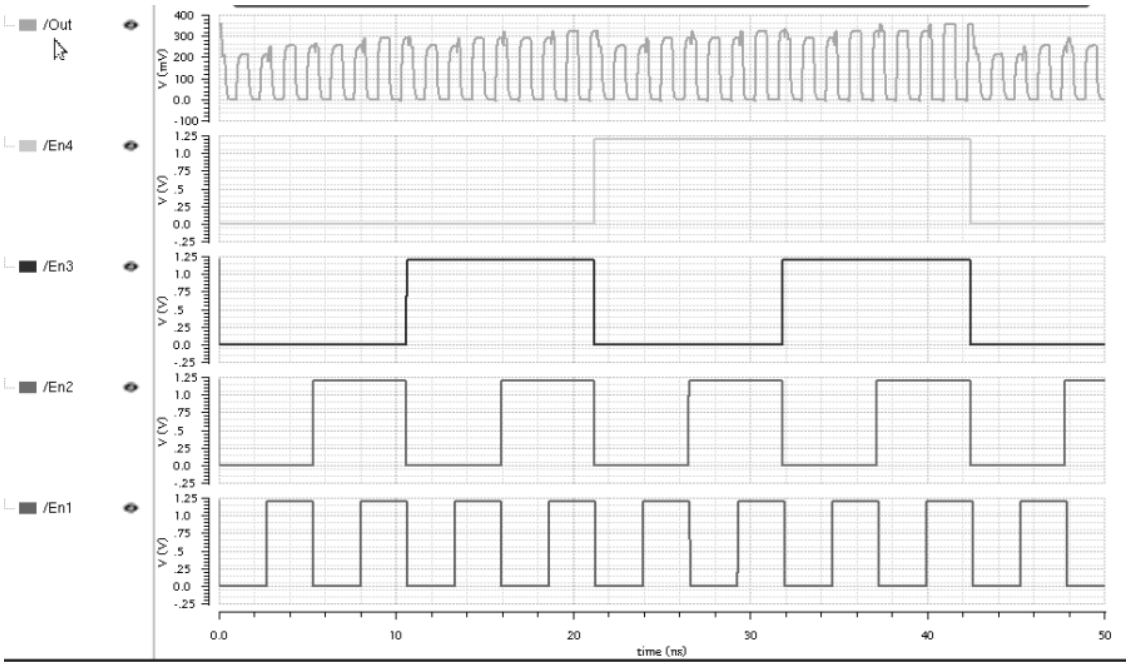


Fig. 60: Transient response (impedance) of ZAP UNIT.

Figure 62 shows the results of the transient response of the ZAP UNIT for the characteristic of impedance. As can be seen the output value changes as the control signals (EN1, EN2, EN3 and EN4) are activated.

Testbench to verify the pre emphasis control of ZAP UNIT is shown in Fig. 63.

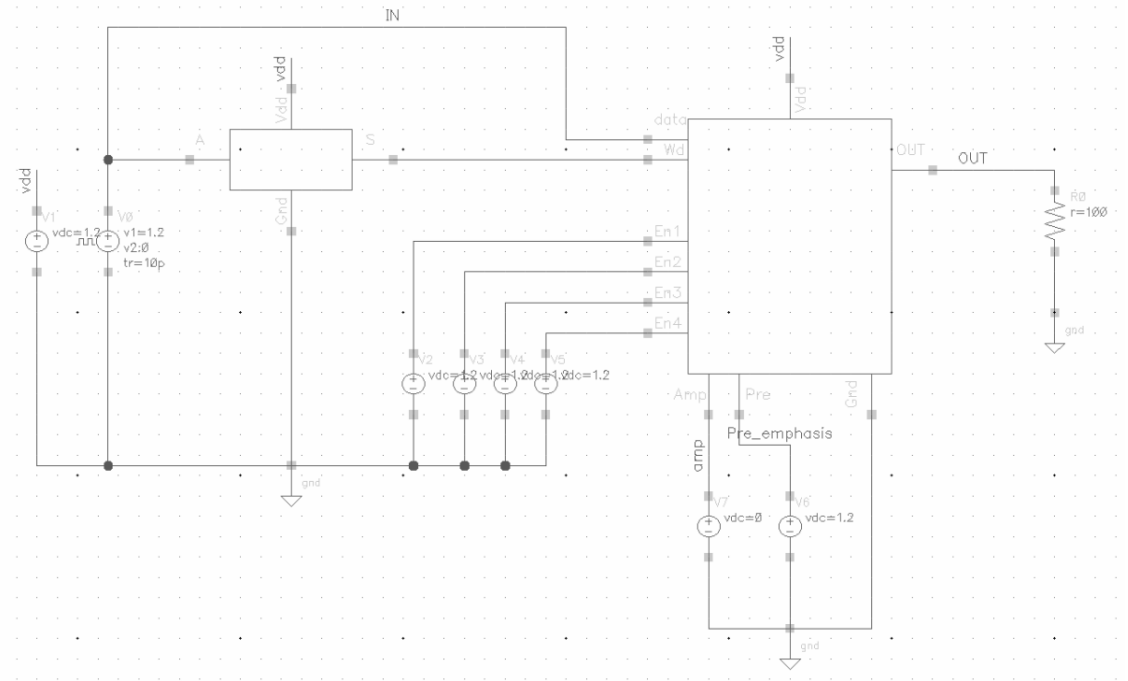


Fig. 61: Testbench for simulation the pre-emphasis response of ZAP UNIT.

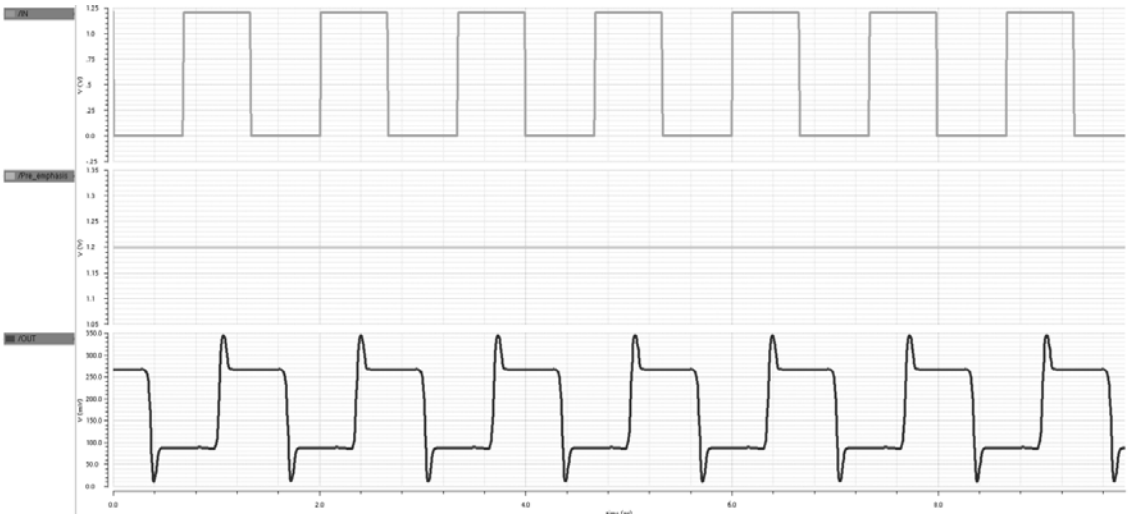


Fig. 62: Transient response (pre-emphasis) of ZAP UNIT.

Fig. 64 shows the change in pre-emphasis when one of the two controls is activated in the ZAP UNIT. Only one of 4 cases is considered, other cases were validated in the response of the full Transmitter module, because it is necessary to place three ZAP UNIT parallel, plus two Decoder 2-3 to generate the other levels of pre emphasis.

3.2 Simulation of Transmitter module (transistor level version).

Fig. 65 shows the test bench just to validate the generation of Normal and Complementary outputs. If we compare Tx and TxBar outputs shown on Fig. 63 we note the same amplitude and period but 180° phase shifted, meaning that Transmitter module designed at transistor level generates the expected Tx and TxBar outputs. Also pre-emphasis characteristic is validated. Fig. 66 presents the transient response with all the possible cases of pre-emphasis, results are shown in table 12

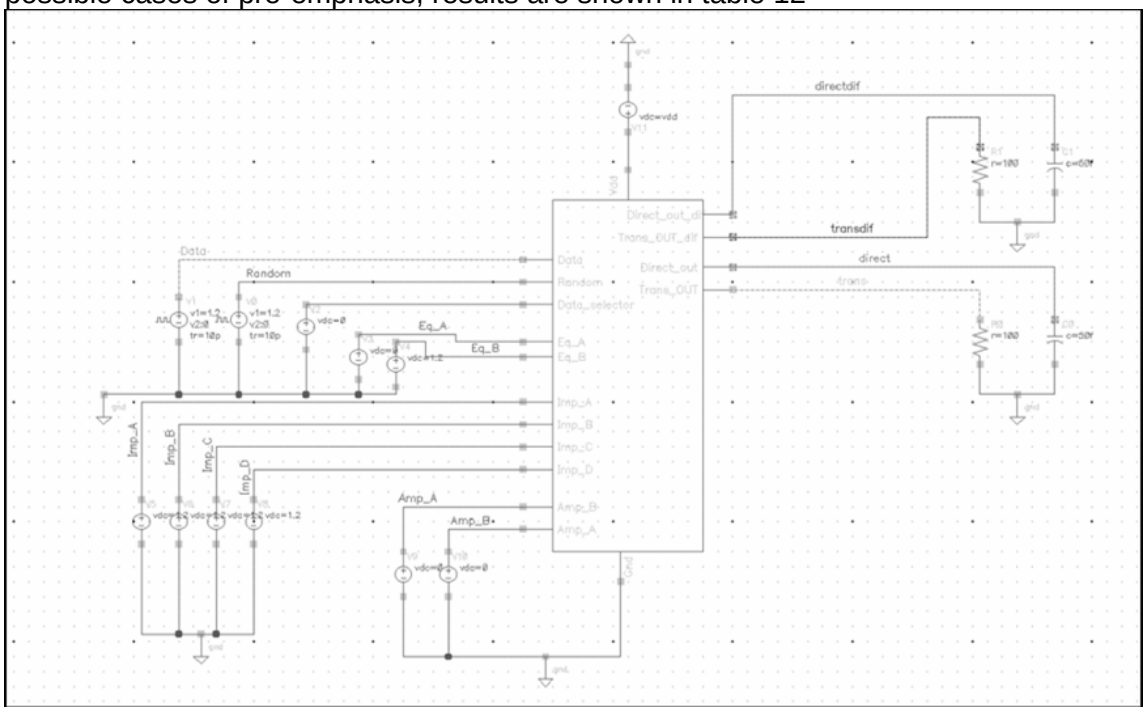


Fig. 63: Test bench to validate the response for Pre emphasis characteristic of complementary Transmitter module.

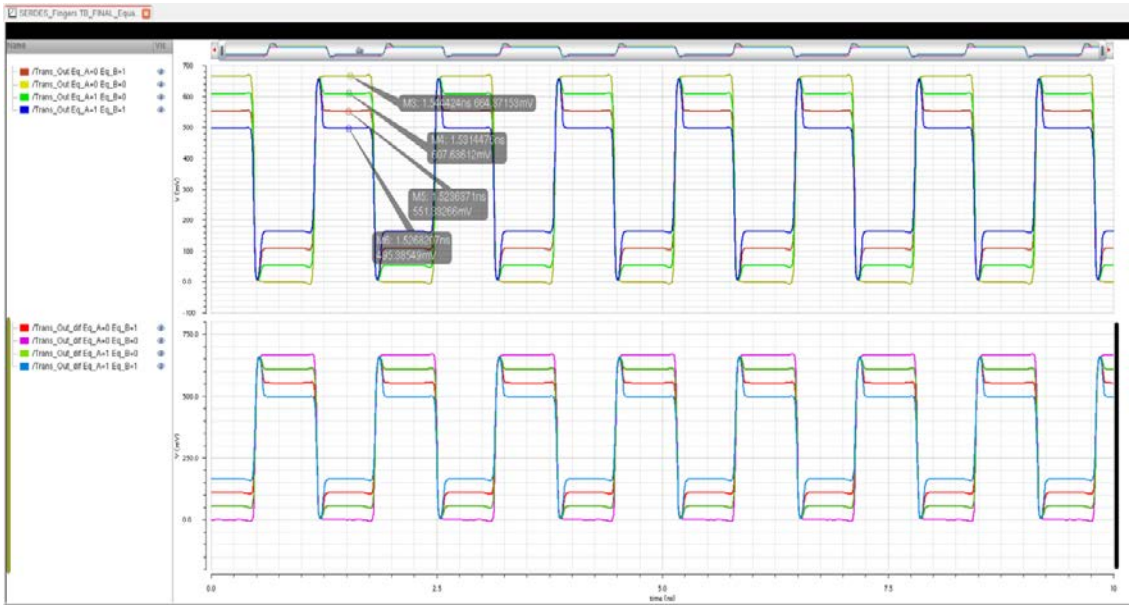


Fig. 64: Transient response for Pre emphasis characteristic of complementary Transmitter module with.

Configuration on pre-emphasis	Output	Pre emphasis
Eq_A=0 Eq_B=0	664.37153 mV	0
Eq_A=1 Eq_B=0	607.63612 mV	-.77 dB
Eq_A=0 Eq_B=1	551.33266 mV	-1.61 dB
Eq_A=1 Eq_B=1	495.38549 mV	-2.54 dB

Table 12: Values of Pre emphasis.

The amplitude characteristic of the Transmitter module is performed with the testbench shown in Fig. 67, Fig. 68 presents the transient response with all the possible cases of pre-emphasis, results are shown in table 13.

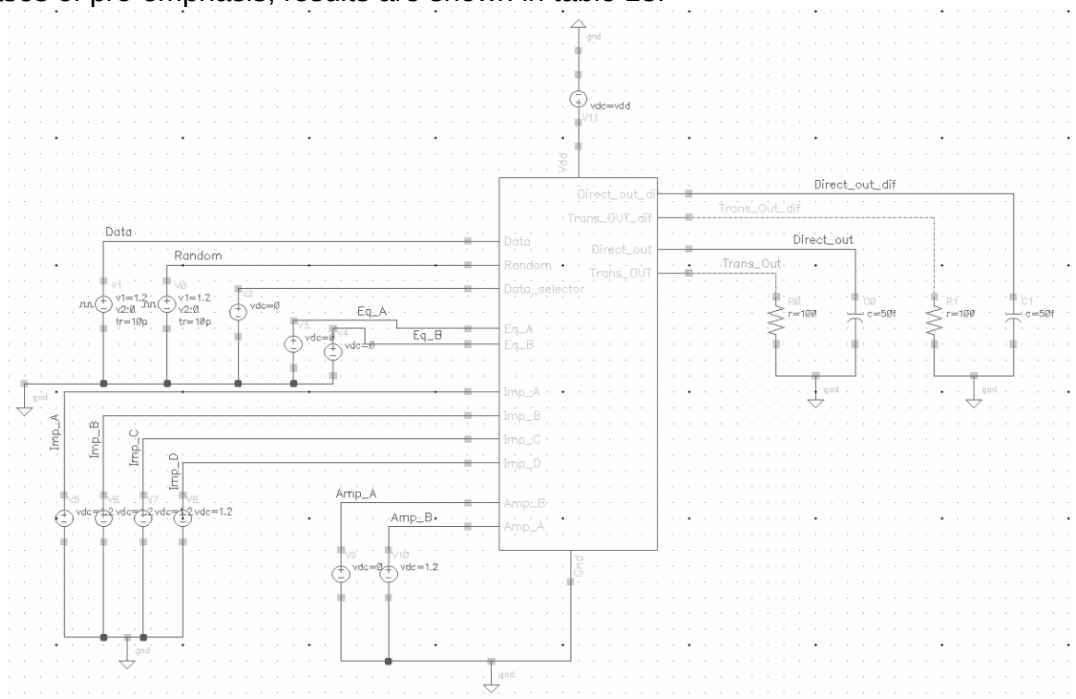


Fig. 65: Test bench to validate the Amplitude response of complementary Transmitter module.

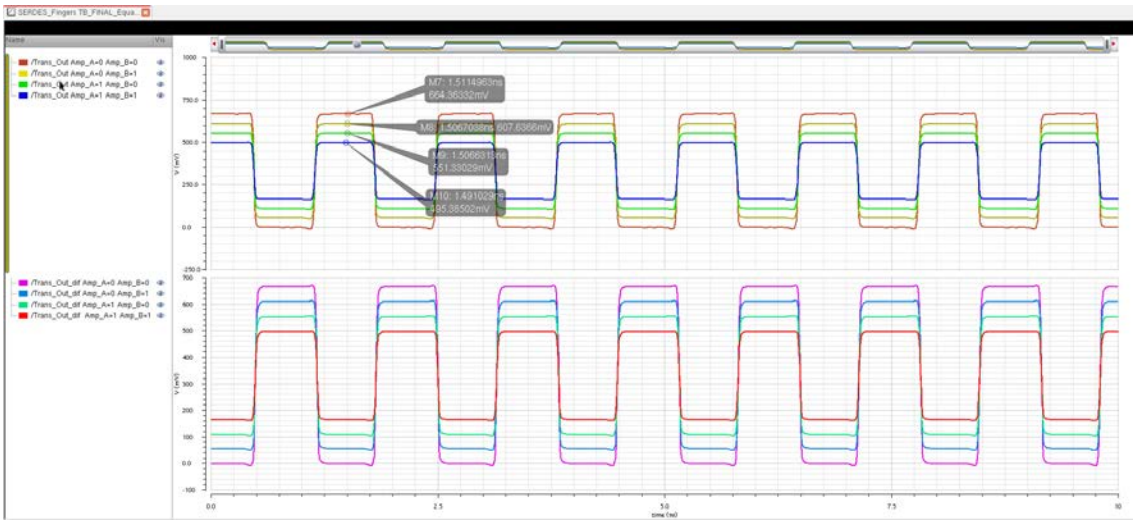


Fig. 66: Transient response for amplitude characteristic of complementary Transmitter module.

Configuration on amplitude	Amplitude
Amp_A=0 Amp_B=0	664.363 mV
Amp_A=1 Amp_B=0	607.636 mV
Amp_A=0 Amp_B=1	551.33 mV
Amp_A=1 Amp_B=1	495.385 mV

Table 13: Values of Amplitude.

The impedance characteristic of the Transmitter module is performed with the testbench shown in Fig. 69 three levels of impedance are considerate. Fig. 70 presents the transient response of impedance characteristic.

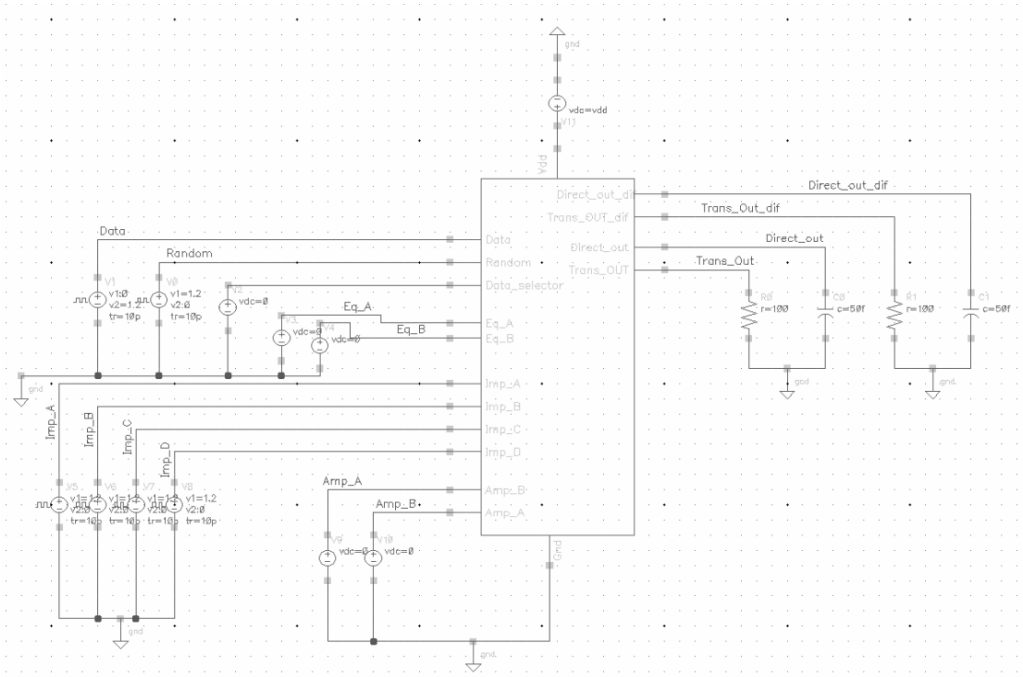


Fig. 67: Test bench to validate the response for impedance characteristic of complementary Transmitter module.

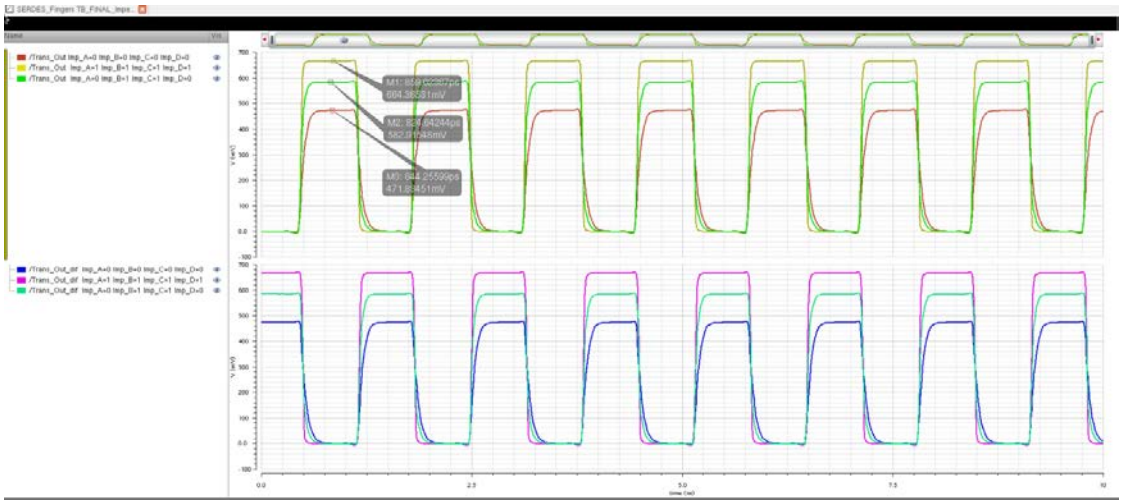


Fig. 68: Transient response for impedance characteristic of complementary Transmitter module.

The impedance characteristic is calculated with Eq. 1.

Configuration on Impedance	Impedance
Imp_A=0 Imp_B=0 Imp_C=0 Imp_D=0	146 Ω
Imp_A=0 Imp_B=1 Imp_C=1 Imp_D=0	100.4355 Ω
Imp_A=1 Imp_B=1 Imp_C=1 Imp_D=1	76.62 Ω

Table 14: Values of Impedance.

3.3 Validation of Transmitter module (behavioral model version)

In order to test the designed behavioral model of the Transmitter we use the same schematic of the Transmitter module in 130nm but with internal blocks replaced by their Verilog-A models. Fig. 71 shows the testbench used for impedance modulation test.

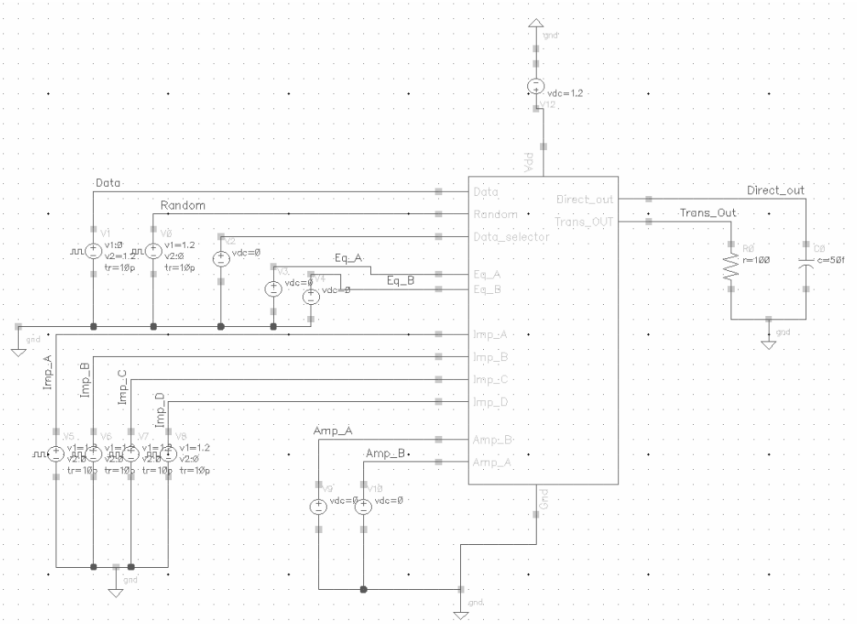


Fig. 69: Testbench of transient response for Impedance modulation test of Transmitter module on its transistor level version.

Fig. 70: Transient response for Impedance of Transmitter module based on its transistor level version.

Figure 73 shows the transient response of the Transmitter module replacing the module Decoder 4-12 at transistor level by its equivalent module in Verilog-A. It is important to mention that the results are very similar compared with transient response of the Transmitter module at transistor level (Fig. 72) .These results validate the correct implementation of Decoder 4-12 macromodel.

Fig. 71: Transient response for Impedance of Transmitter module with Decoder 4-12 Verilog-A module.

Figure 74 shows the transient response of the Transmitter module replacing the module Decoder 2-3 at transistor level by its equivalent module in Verilog-A. It validates the correct implementation of Decoder 2-3 macromodel.

Fig. 72: Transient response for Impedance of Transmitter module with DECO2-3 Verilog-A Module.

Figure 75 shows the transient response of the Transmitter module replacing the module ZAP UNIT at transistor level by its equivalent module in Verilog-A. Once again, this validates the correct implementation of ZAP UNIT macromodel.

Fig. 73: Transient response for Impedance of Transmitter module with ZAP UNIT Verilog-A Module.

Fig. 76 shows the transient response of impedance modulation of Verilog-A Transmitter module with all its internal blocks replaced by its macromodels. Here again, the response of Transmitter macromodel is similar to the obtained for transistor level version of Transmitter module. As expected, in Fig. 76, we do not observe delay time, neither rise time or fall time degradation, this is because module is implemented with ideal components.

Fig. 74: Transient response for Impedance of Verilog-A Transmitter module.

3.4 PVT corners simulation of Transmitter Module designed in 130 nm.

Table 15 shows the configuration of PVT corners used in ADE XL to simulate impedance, pre-emphasis and amplitude variations under different operation conditions (PVT corners)., We performed simulation with a set of three critical PVT corners i. e., temperatures, and three process parameters, slow – slow, fast – fast and typical.

Temperature	Transistors		
Centigrades C	ss	ff	tt
-40	yes	yes	yes
27	yes	yes	yes
125	yes	yes	yes

Table 15: Temperatures and corners set up.

Table 16 summarizes the labels that appear in figures 79, 80, 81 and 82. The first column (from left to right) means the label, the second column means the corner used and the third column is the temperature.

Out_trans_tt_0	tt	-40 Centigrade
Out_trans_tt_1	tt	27 Centigrade
Out_trans_tt_2	tt	125 Centigrade
Out_trans_ss_0	ss	-40 Centigrade
Out_trans_ss_1	ss	27 Centigrade
Out_trans_ss_2	ss	125 Centigrade
Out_trans_ff_0	ff	-40 Centigrade
Out_trans_ff_1	ff	27 Centigrade
Out_trans_ff_2	ff	125 Centigrade

Table 16: Temperatures and transistors configuration.

An important element in the Transmitter architecture is the ZAP UNIT. A key component in the ZAP UNIT is the basic cell (Fig. 77), which controls the impedance / resistance value depending on the value of the switch state, that is, if switch is open, an

impedance of 1640 Ω is obtained, by contrary if the switch is closed an impedance of 820 Ω is obtained [5].

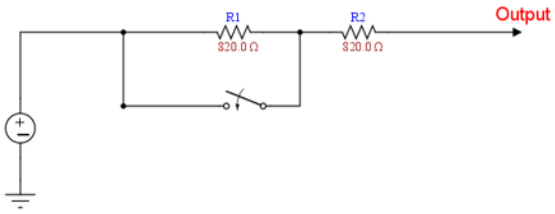


Fig. 75: Ideal Basic Cell Diagram.

In the Transmitter module, we have 12 basic cells, so we have 12 different values of impedance; these values are controlled with enable inputs Imp_A, Imp_B, Imp_C and Imp_D. The maximum value of impedance that we can have is 1640//12 = 136.67 Ω, and the minimum value of impedance that we have is 820//12 = 68.33 Ω. Figure 78 shows the testbench to evaluate the impact on impedance modulation of Transmitter module under PVT conditions described in Table 16.

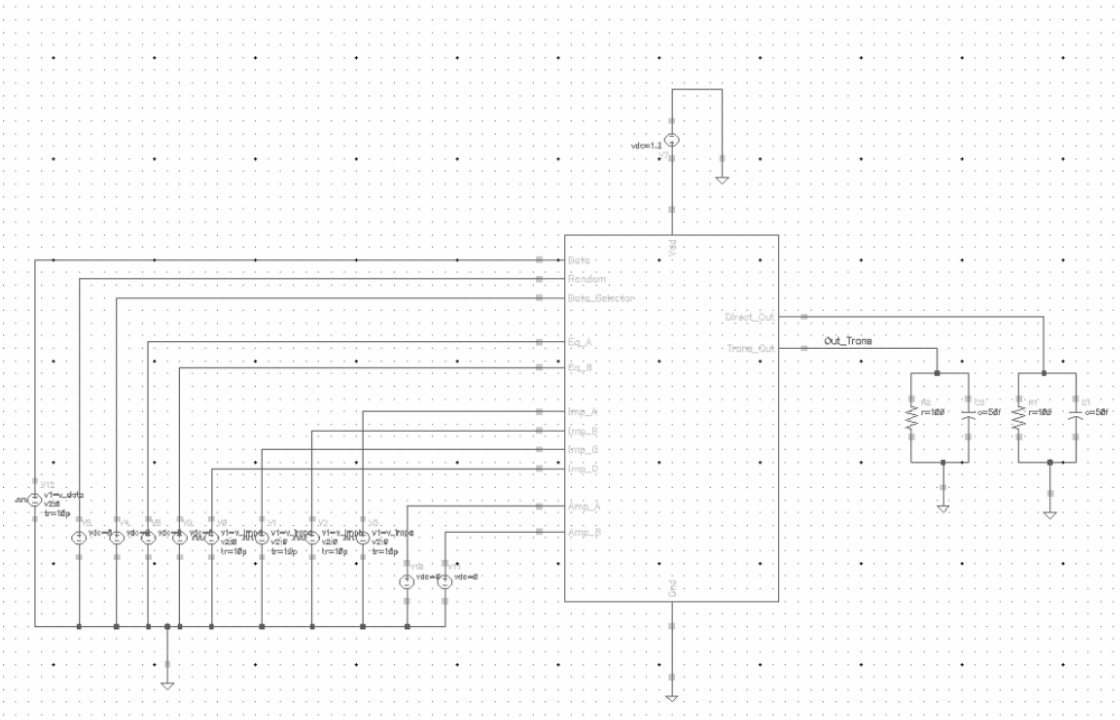


Fig. 76: Test bench to validate the Impedance modulation of Transmitter module in 130nm

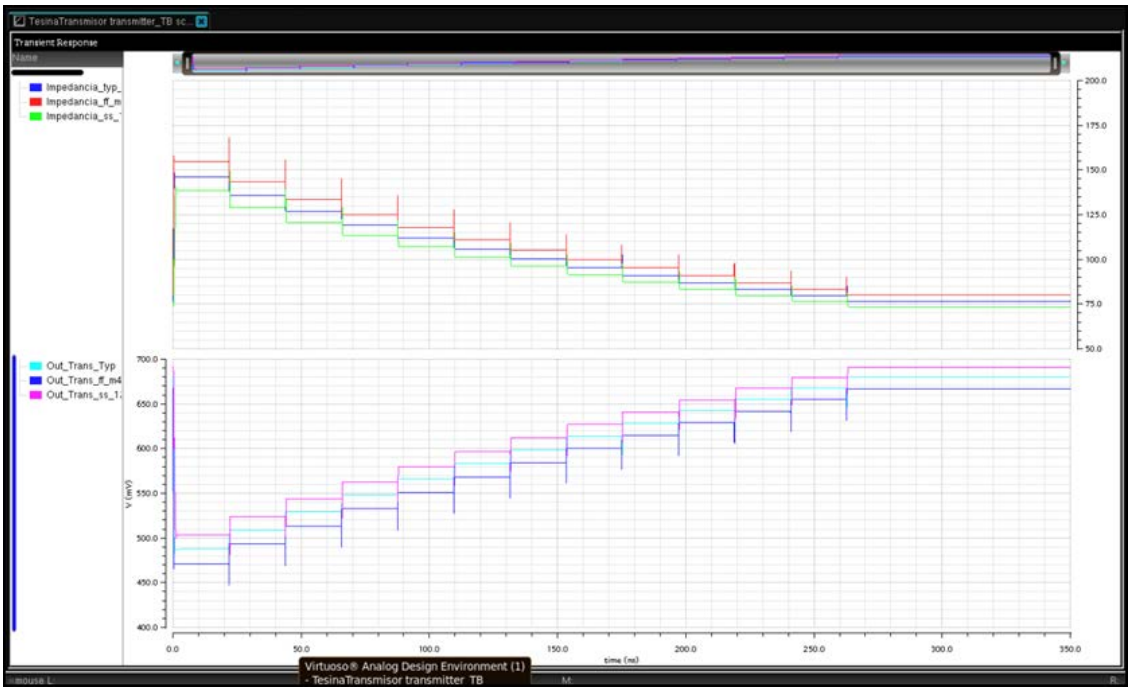


Fig. 77: Impedance variation using transistors ff(red), ss(green) and tt(blue).

Fig. 79 shows on the top the transient response of output impedance by applying Eq. 1, and using calculator functions of Spectre Virtuoso. On the bottom shows the transient response obtained in simulation for critical PVT corners (typical process and room temperature; fast-fast transistors and -40 °C; slow-slow transistors and 125 °C). Table 17 summarizes the values of output impedance observed in Fig. 79 which are compared with the ideal impedances values obtained by Eq. 1 [5].

Typical 27 °C (Ω)	% Variation vs theoretical Values	Fast – Fast -40C (Ω)	% Variation vs Theoretical Values	Slow – Slow 125C (Ω)	% Variation vs Theoretical Values	Ideal values (Ω).
146	6.83%	154.5168	13.06%	138.3191	1.21%	136.67
135.7	7.57%	143.3101	13.60%	128.8845	2.17%	126.15
126.78	8.23%	133.6472	14.09%	120.6594	3.00%	117.14
118.96	8.81%	125.2185	14.53%	113.4243	3.74%	109.33
112.07	9.34%	117.8	14.93%	107.0109	4.40%	102.5
105.932	9.81%	111.2199	15.29%	101.2856	4.99%	96.47
100.4355	10.24%	105.3421	15.62%	96.44356	5.85%	91.11
95.4841	10.63%	100.0594	15.93%	91.5	6.01%	86.31
91	10.98%	95.28531	16.20%	87.28369	6.44%	82
86.9211	11.30%	90.9492	16.46%	83.44023	6.84%	78.095
83.1932	11.61%	86.99326	16.71%	79.92143	7.22%	74.54
79.7732	11.88%	83.3692	16.93%	76.68759	7.56%	71.3
76.6242	12.14%	80.03675	17.13%	73.7	7.86%	68.33

Table 17: Impedance variation using transistors ff, ss and tt.

On Table 17 one can observe a difference between theoretical values and the typical values. An explanation of this difference is related to fact that the transistor is not

a perfect switch, so there is a drop in voltage when the enable input is off in the basic cell.

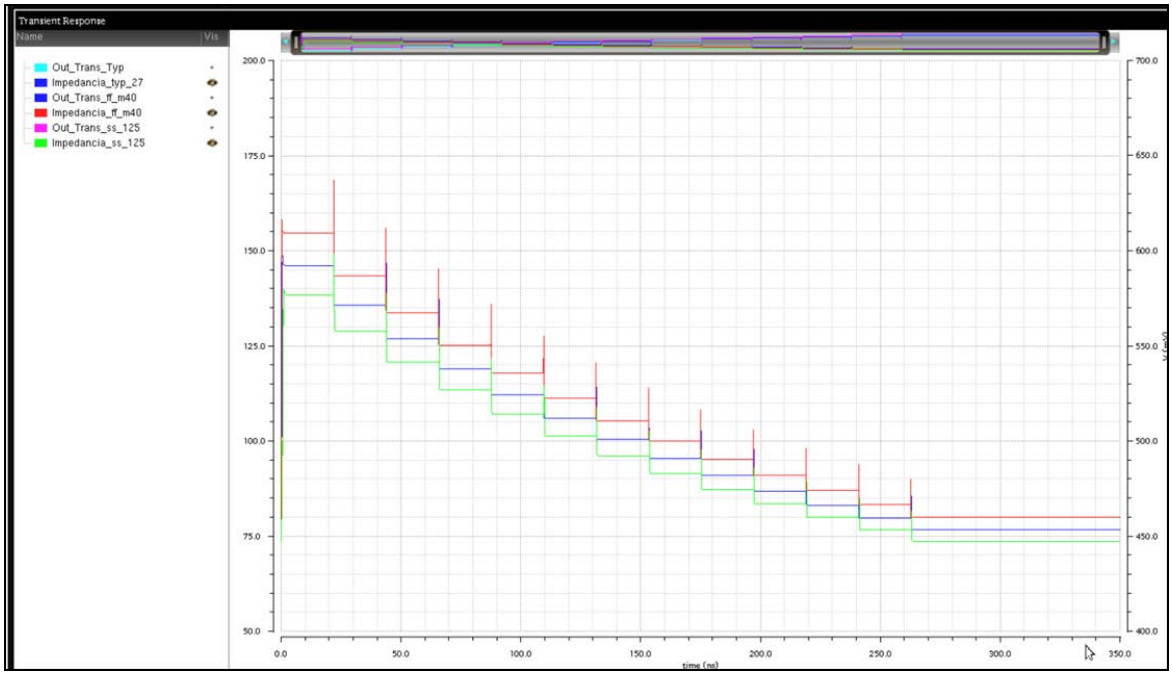


Fig. 78: Change in voltage measured at Tx output.

Figure 80 shows the change in voltage measured at Tx output, these changes in voltage are in accordance with the table 17 and can be verified using equation (Eq. 3). Notice that for ff transistor and -40 °C impedance are higher than the value for ss transistors and high temperature (125 °C) conditions.

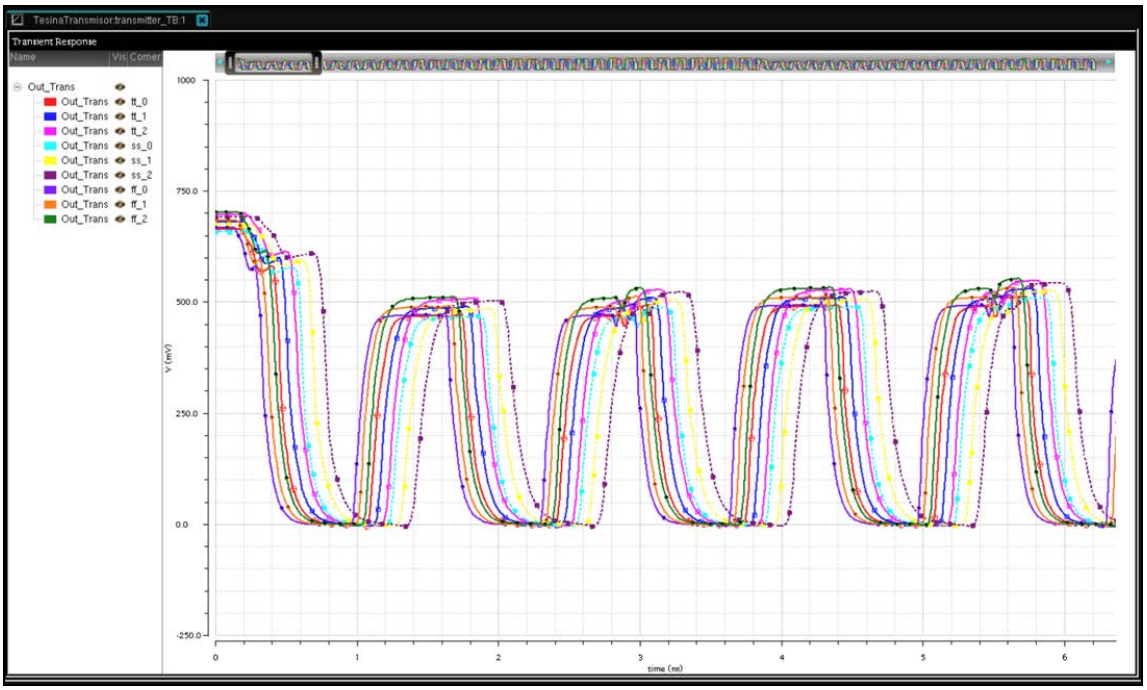


Fig. 79: Transient response at corners.

Fig. 81 shows the transient response of the Transmitter module for impedance characteristic for all the corners on table 16.

In Fig. 82 we show the transient response for impedance characteristic of the Transmitter module for the two extremes of the corners, fast – fast @ 125C and slow – slow @ 125C; in which we can appreciate a delay time of 352.9 pS between these two corners.

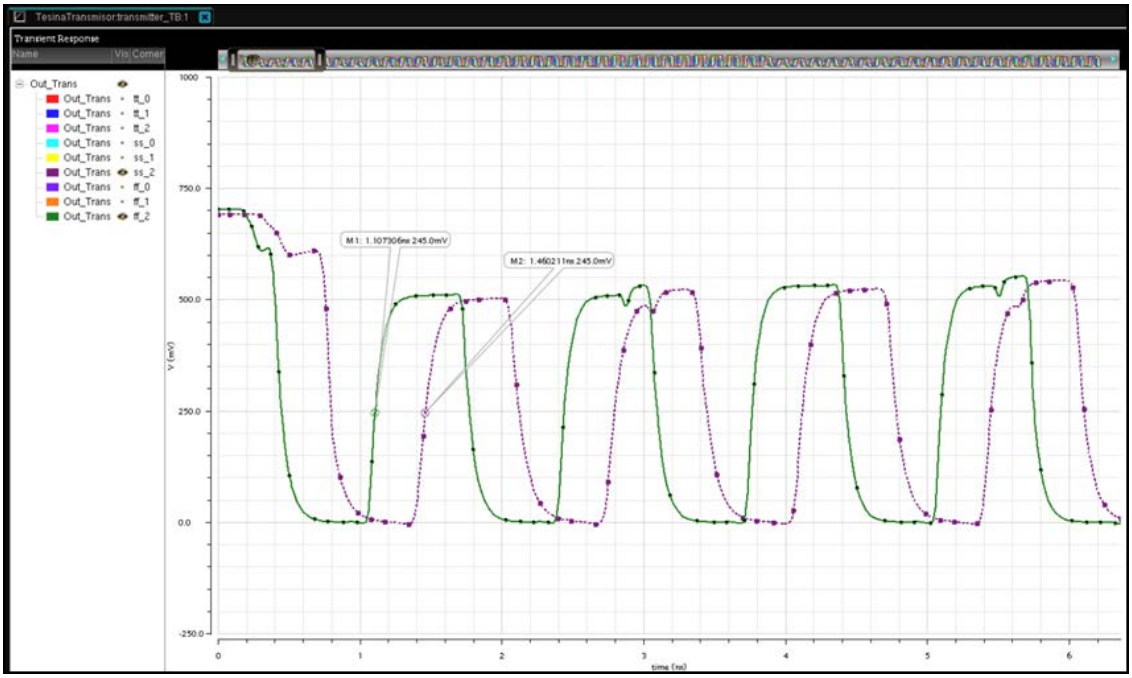


Fig. 80: Delay of 352.9 pS in the response using ss configuration compared with a ff configuration @ temperature of 125C.

In the performed PVT corners simulation of Transmitter module, we observe significant variations in impedance features of Transmitter module working under PVT corners. Further analysis using Monte – Carlo should be done in order to determine if the changes in amplitude (impedances) affect the Transmitter behavior and the impact on it.

CHAPTER 4: PHYSICAL DESIGN OF TRANSMITTER MODULE

4.1 Layout design of internal blocks of the Transmitter module.

The first design of Layout was done for every block of Transmitter module. With the purpose of integration in a same way of using standard cells, complex blocks such as DECO 4-12 and DECO 2-3, Buffer, and others (Figs. 86 to 91), low-level digital cells (NAND,NOR and Inverter) were designed with the same height (Figs. 83 to 85). We verify the layout of each cell with CALIBRE (DRC and LVS).

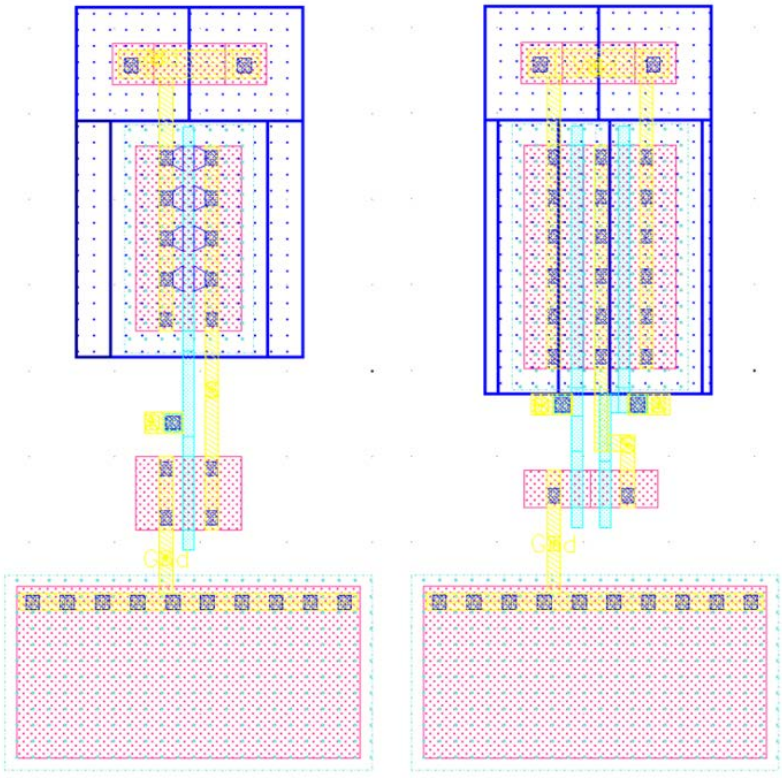


Fig. 81: Layout view of Inverter (left) and NAND2 (right).

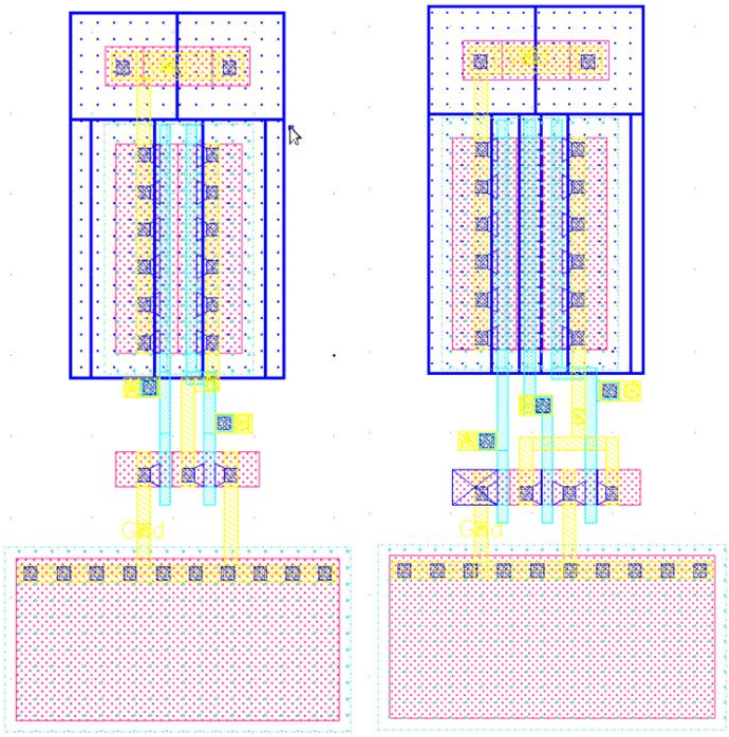


Fig. 82: Layout view of NOR2 (left) and NOR3 (right).

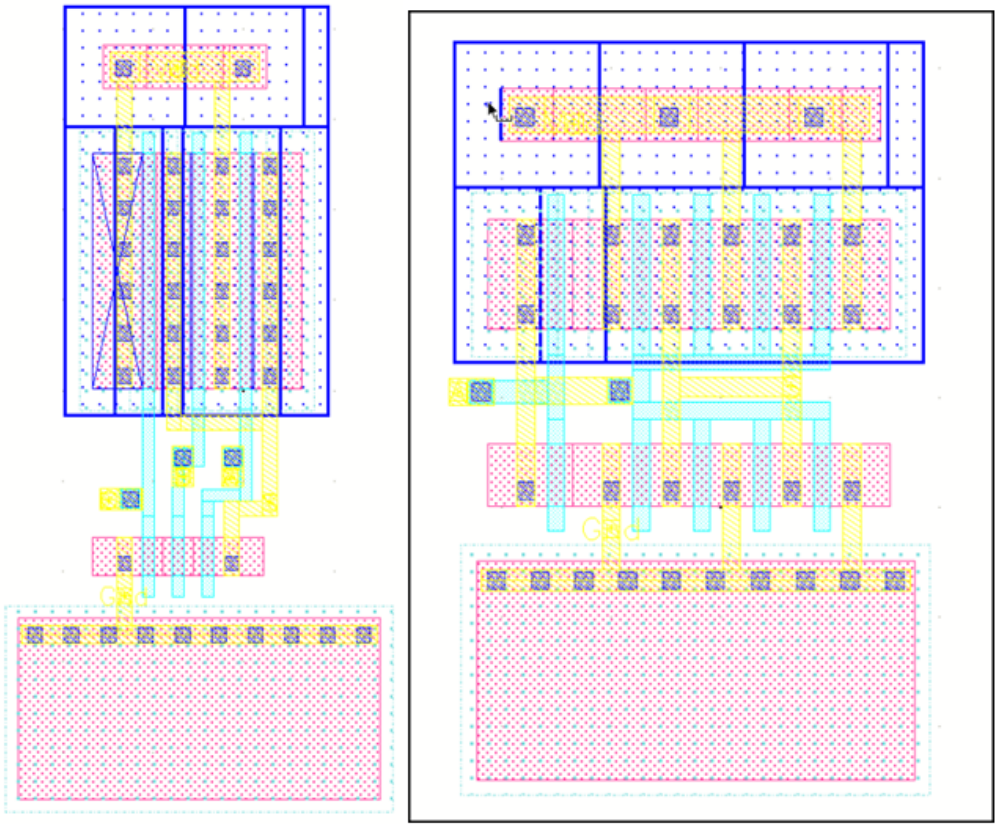


Fig. 83: Layout view of NAND3 (left) and Buffer (right).

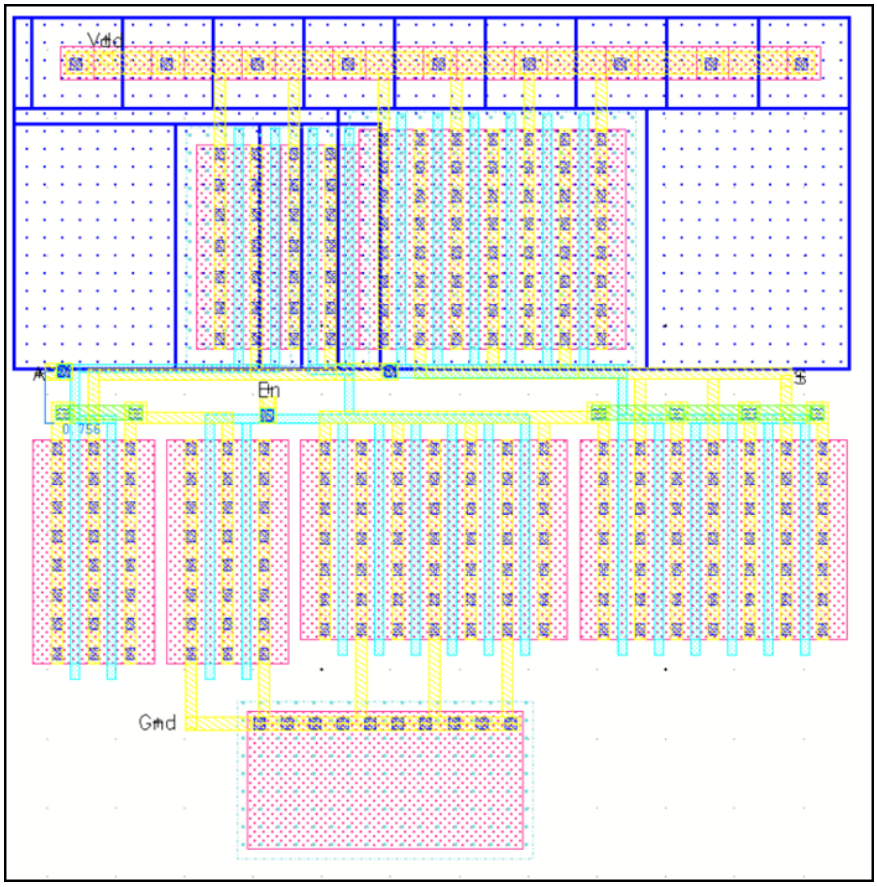


Fig. 84: Layout view of Buffer 3 state.

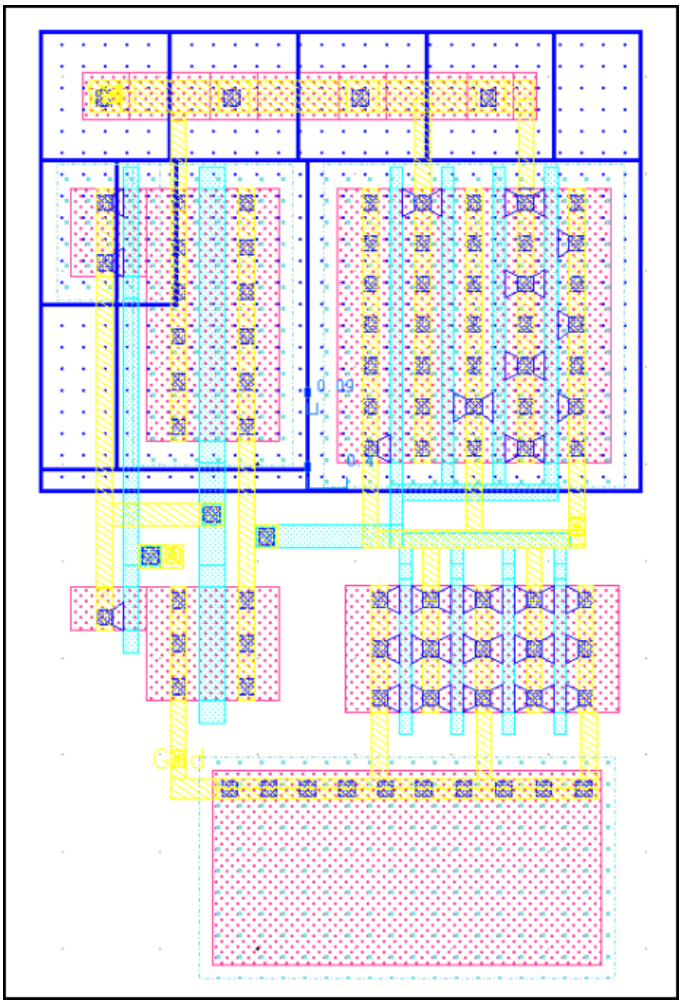


Fig. 85: Layout view of Delay Inverter.

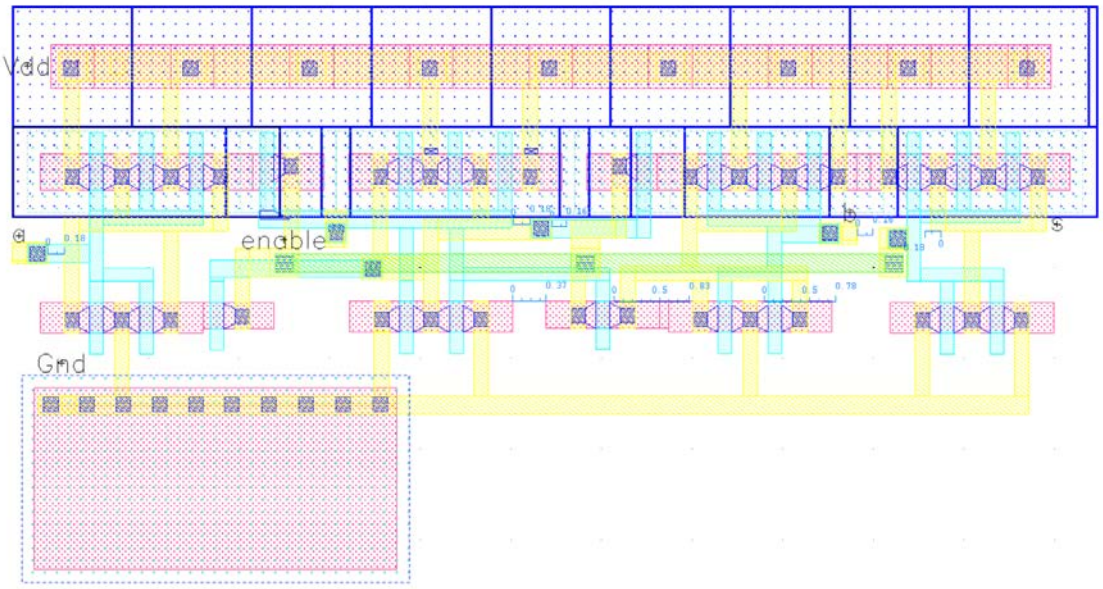


Fig. 86: Layout view of Multiplexer 2-1.

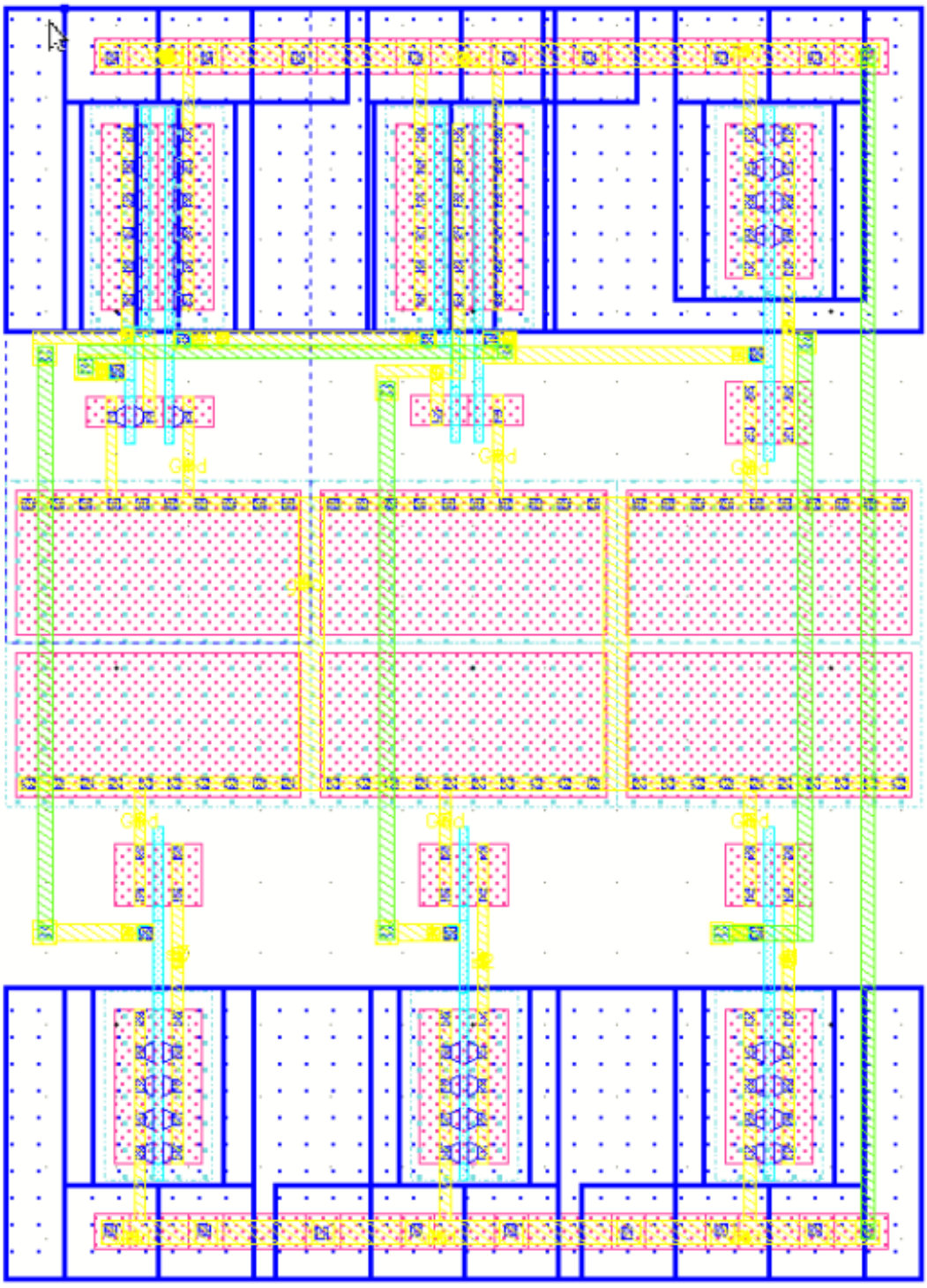


Fig. 87: Layout view of DECO 2-3.

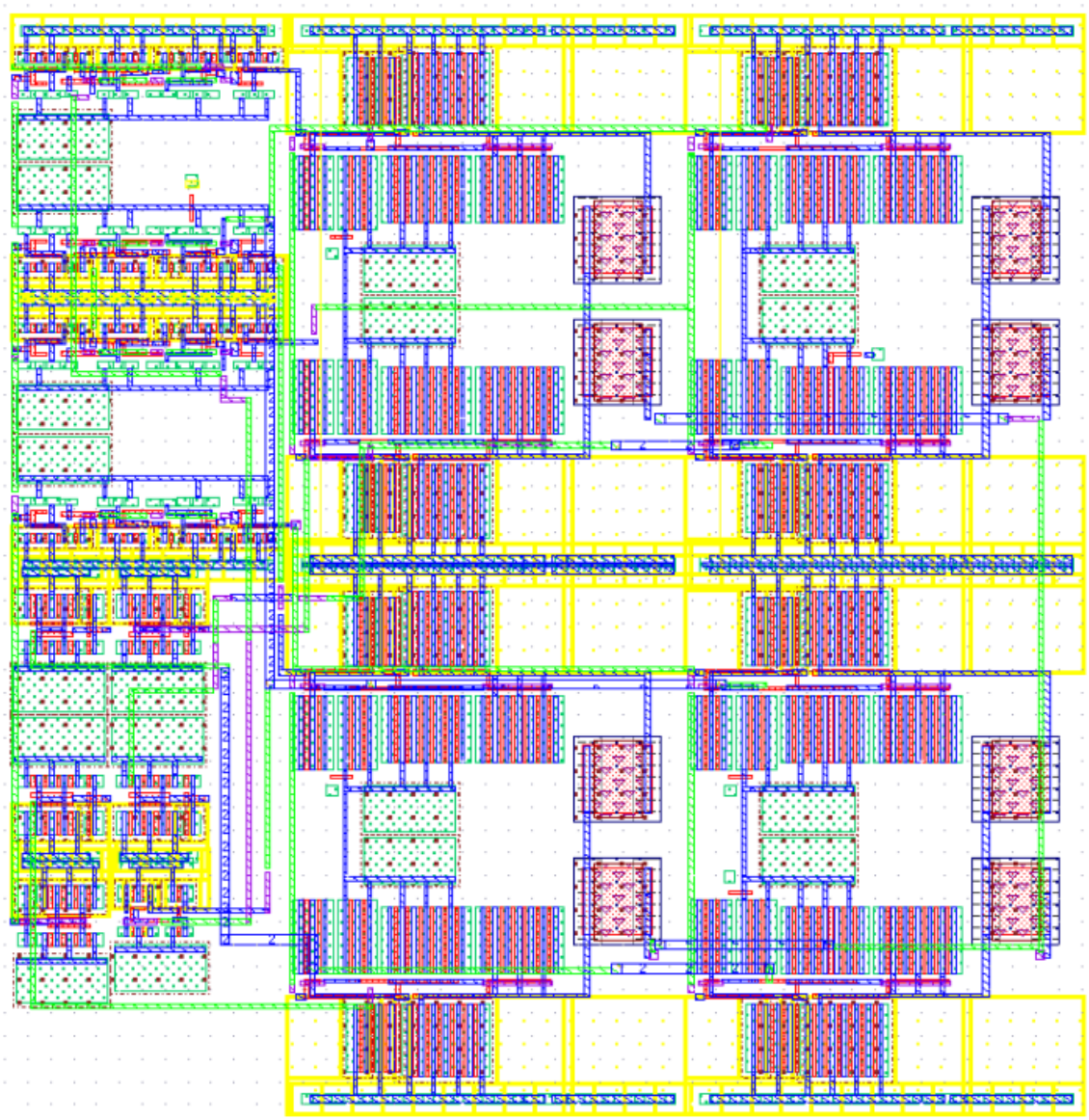


Fig. 88: Layout view of ZAP UNIT.

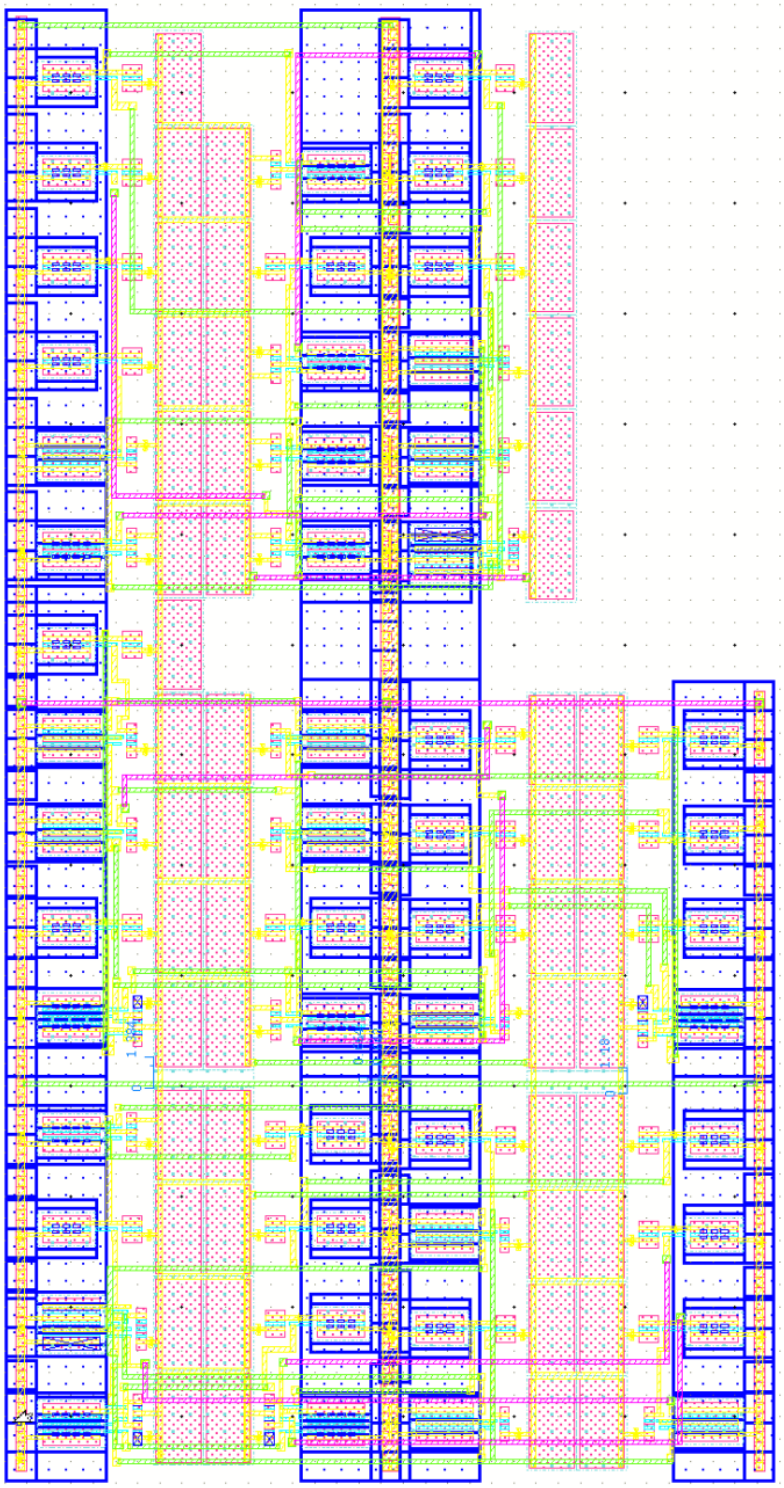


Fig. 89: Layout view of DECO 4-12.

4.2 Layout of Transmitter module.

The layout of internal blocks of Transmitter module, whose designs were presented from Fig. 83 to Fig. 91 were integrated into a single cell (Fig. 92). Notice this layout corresponds to the design version of Transmitter module with Tx and TxBar outputs. The area occupied by the Transmitter module is 130 µm x 140 µm.

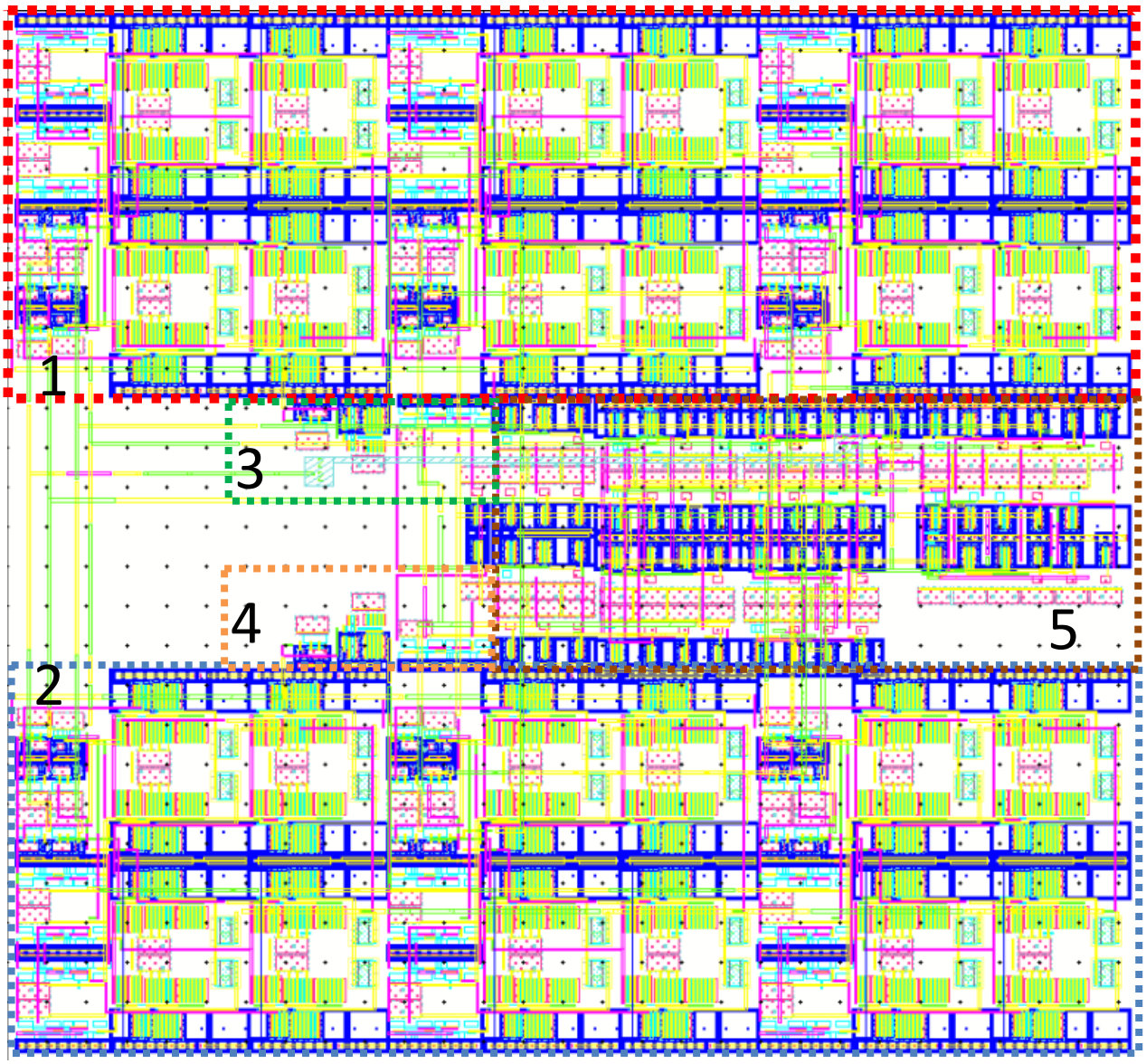


Fig. 90: Final Layout of Transmitter Module with Complementary Outputs.

The numbers in the layout on Fig. 92 indicates the location of internal blocks:

1- Transmitter block for differential output (TxBar).
2- Transmitter block for Normal output (Tx).
3- Input block for complementary output (TxBar).
4- Input block for normal output (Tx).

5- Control Block of Transmitter module characteristics (Amplitude, impedance and pre emphasis).

Layout verification with Calibre DRC and Calibre LVS tools of all the internal blocks of Transmitter module, pass without DRC or LVS errors. As an illustrative example, Fig. 93 and Fig. 94 presents the DRC and LVS display results from Calibre.

```
RULECHECK GR131_MA_T3 ..... TOTAL Result Count = 0                    (0)
RULECHECK GRMA908 ......... TOTAL Result Count = 0                    (0)
RULECHECK GRMA953 ......... TOTAL Result Count = 0                    (0)
RULECHECK GRT3W131f ....... TOTAL Result Count = 0                    (0)
RULECHECK GRT3W131f_Mx .... TOTAL Result Count = 0                    (0)
RULECHECK GRT3WQCAP24 ..... TOTAL Result Count = 0                    (0)
RULECHECK GRT3WQCAP24a .... TOTAL Result Count = 0                    (0)
RULECHECK GRT3WQCAP24b .... TOTAL Result Count = 0                    (0)
RULECHECK GR594_M1 ........ TOTAL Result Count = 0                    (0)
RULECHECK GRT3W594a_M1 .... TOTAL Result Count = 0                    (0)
RULECHECK GRT3W594b_M1 .... TOTAL Result Count = 0                    (0)
RULECHECK GR595_M1 ........ TOTAL Result Count = 0                    (0)
RULECHECK GRT3W595_M1 ..... TOTAL Result Count = 0                    (0)
RULECHECK GR594_M2 ........ TOTAL Result Count = 0                    (0)
RULECHECK GRT3W594a_M2 .... TOTAL Result Count = 0                    (0)
RULECHECK GRT3W594b_M2 .... TOTAL Result Count = 0                    (0)
RULECHECK GR595_M2 ........ TOTAL Result Count = 0                    (0)
RULECHECK GRT3W595_M2 ..... TOTAL Result Count = 0                    (0)
RULECHECK GR594_M3 ........ TOTAL Result Count = 0                    (0)
RULECHECK GRT3W594a_M3 .... TOTAL Result Count = 0                    (0)
RULECHECK GRT3W594b_M3 .... TOTAL Result Count = 0                    (0)
RULECHECK GR595_M3 ........ TOTAL Result Count = 0                    (0)
RULECHECK GRT3W595_M3 ..... TOTAL Result Count = 0                    (0)
--------------------------------------------------------------------------------
--- RULECHECK RESULTS STATISTICS (BY CELL)
---
--------------------------------------------------------------------------------
--- SUMMARY
---
TOTAL CPU Time:                  5
TOTAL REAL Time:                 9
TOTAL Original Layer Geometries: 3210 (59059)
TOTAL DRC RuleChecks Executed:   1655
TOTAL DRC Results Generated:     0 (0)
```

Fig. 91: Calibre DRC verification for Transmitter module without errors.

```
                ################################################
                ##                                            ##
                ##       C A L I B R E   S Y S T E M          ##
                ##                                            ##
                ##            L V S   R E P O R T             ##
                ##                                            ##
                ################################################


REPORT FILE NAME:         Final_Diferencial_inverter.lvs.report
LAYOUT NAME:              /home/jnunez/8sf_IBM/Final_Diferencial_inverter.sp ('Final_Diferencial_inverter')
SOURCE NAME:              /home/jnunez/8sf_IBM/Final_Diferencial_inverter_out.cdl ('Final_Diferencial_inverter')
RULE FILE:                /home/jnunez/8sf_IBM/_cmrf8sf.lvs.cal_
CREATION TIME:            Wed Jul 13 18:34:06 2016
CURRENT DIRECTORY:        /home/jnunez/8sf_IBM
USER NAME:                jnunez
CALIBRE VERSION:          v2015.2_27.20    Tue Jun 2 10:53:48 PDT 2015


                          OVERALL COMPARISON RESULTS


                      #       ###################
                    #         #                 #       +   +
                  #   #       #     CORRECT     #         |
                   # #        #                 #       \___/
                    #         ###################


********************************************************************************************************************
                                    CELL  SUMMARY
********************************************************************************************************************

  Result         Layout                       Source
  -----------    -----------                  --------------
  CORRECT        Final_Diferencial_inverter   Final_Diferencial_inverter
```

Fig. 92: Calibre LVS verification for Transmitter module without errors.

4.3 Post layout simulation

We have evaluated the post-layout performance of all internal blocks of Transmitter module. As illustrative examples, below we present the case of Decoder 2-3, Multiplexer 2-1 and ZAP UNIT post-layout responses.

For cell Decoder 2-3 we extract parasitics from layout using Calibre-PEX, then we perform postlayout simulation with typical process values on extracted Calibre view using the testbech of Fig. 95. In the obtained post-layout transient response (Fig. 96) we note a delay time of 12.805 pS with respect to pre-layout response.

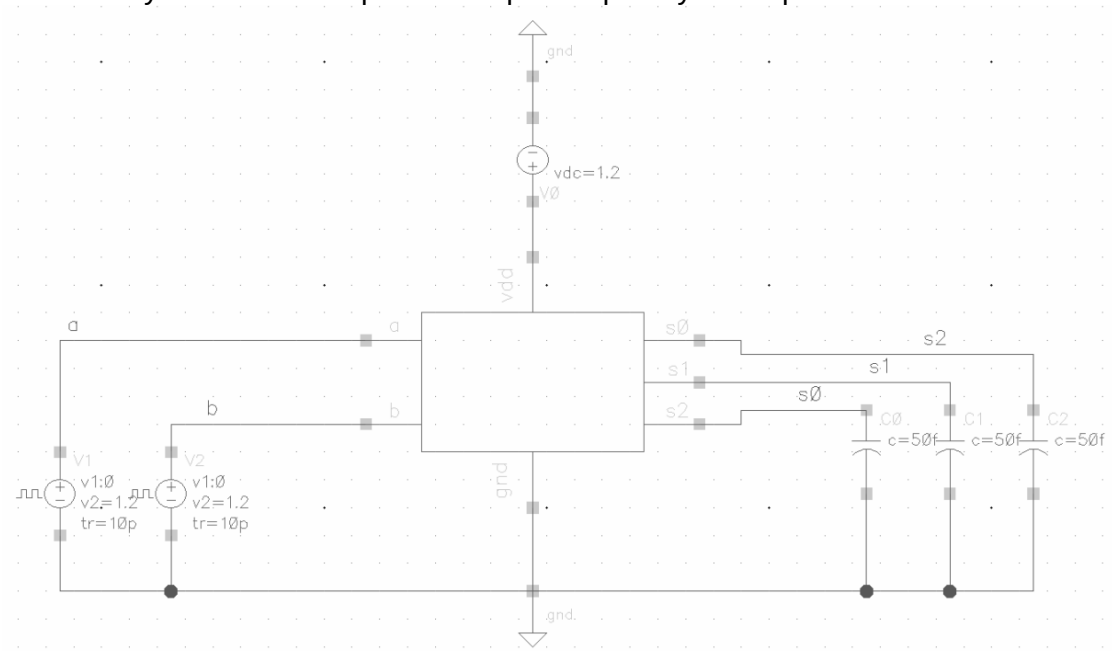


Fig. 93: Test bench for post layout simulation of Decoder 2-3.

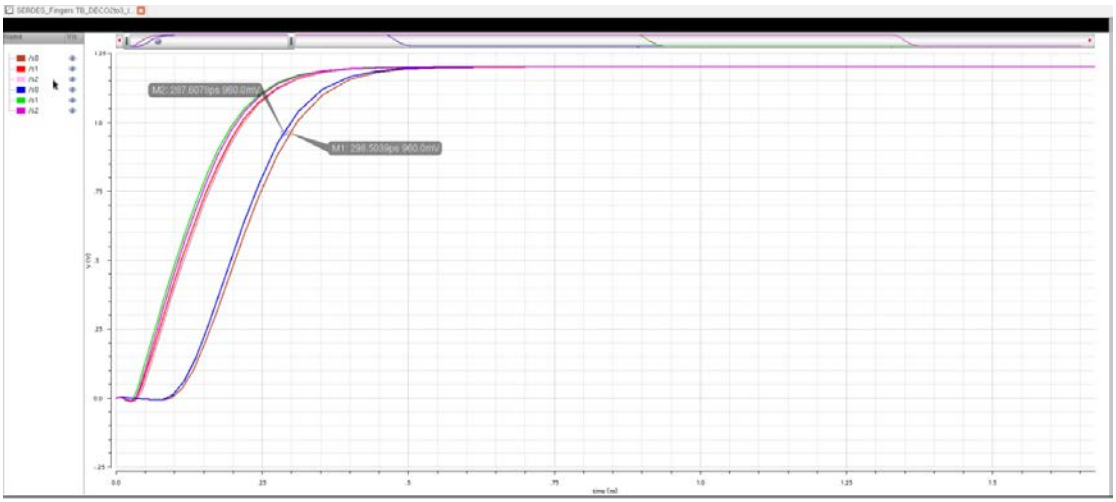


Fig. 94: Post layout transient response of Decoder 2-3.

Similarly, for cell Multiplexer 2-1 we extract parasitics from layout, then we perform postlayout simulation using the testbech of Fig. 97. In the obtained transient response (Fig. 98) we observe a delay time of 69.19 pS, also a reduction of 1.63% in amplitude of signal in the post-layout response. These changes are attributed to the parasitic capacitances. Notice however, these changes are not significant and it is expected not serious affectation in the performance of Transmitter module.

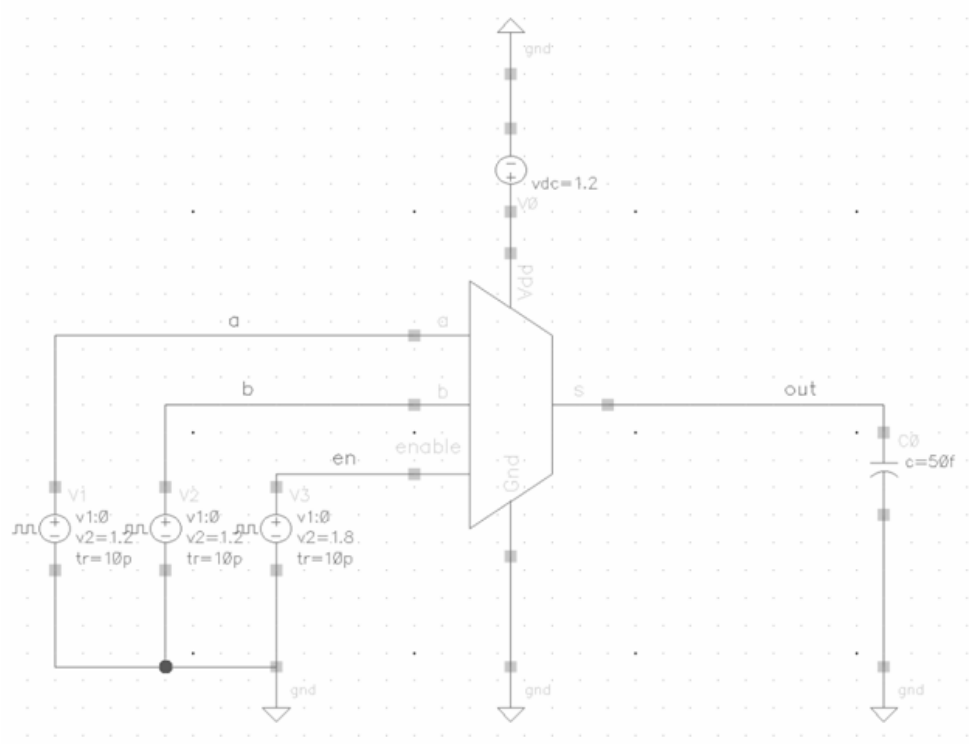


Fig. 95: Testbench for post layout simulation of Multiplexer 2-1.

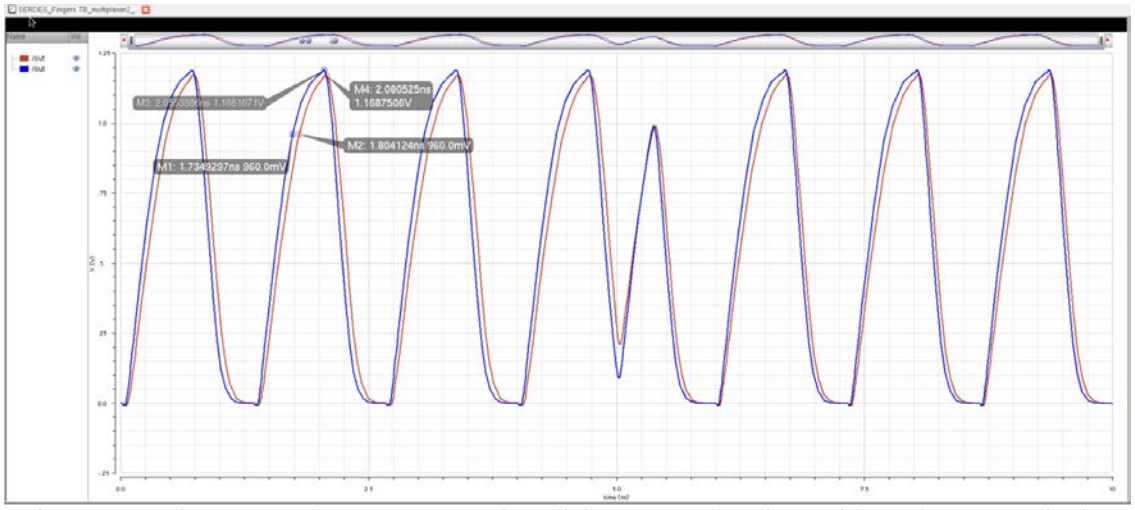


Fig. 96: Post layout transient response of Multiplexer 2-1, showing a delay of 69.19 pS in the rise time and a little reduction in amplitude.

For cell ZAP UNIT we extract parasitics from layout, and then we perform postlayout simulation using the testbech of Fig. 99. We observe a delay time of 94.43 pS, also a reduction of 2.18% in amplitude of signal in the post-layout response.

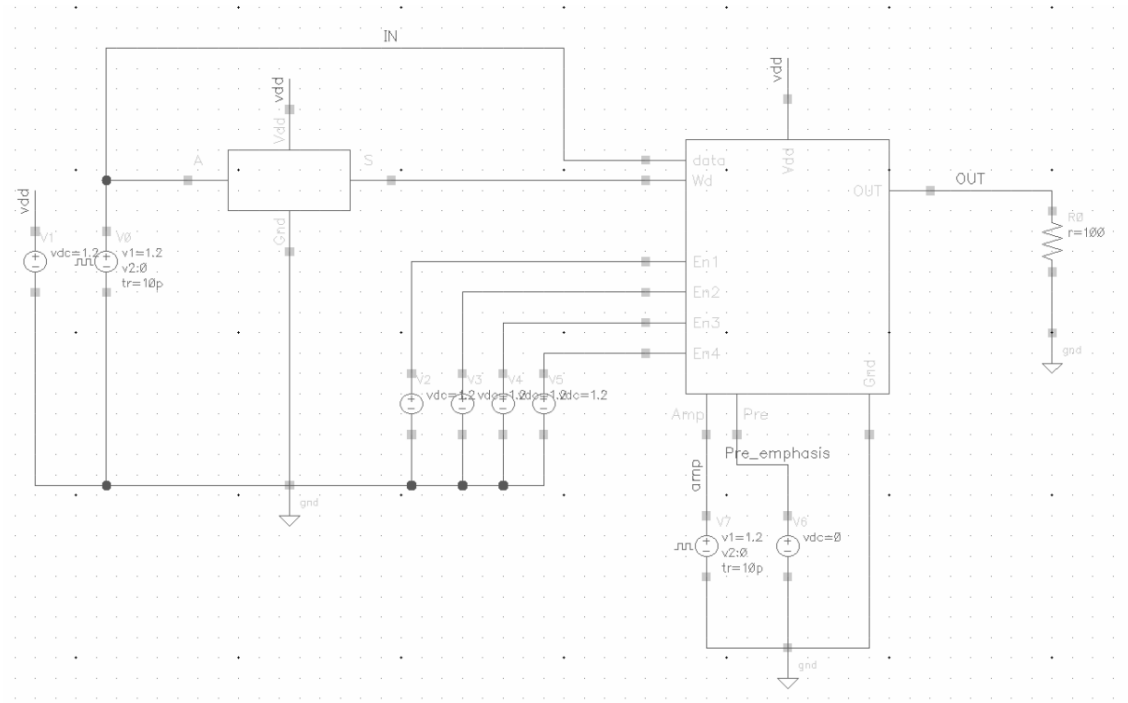


Fig. 97: Testbench for post layout simulation of ZAP UNIT.

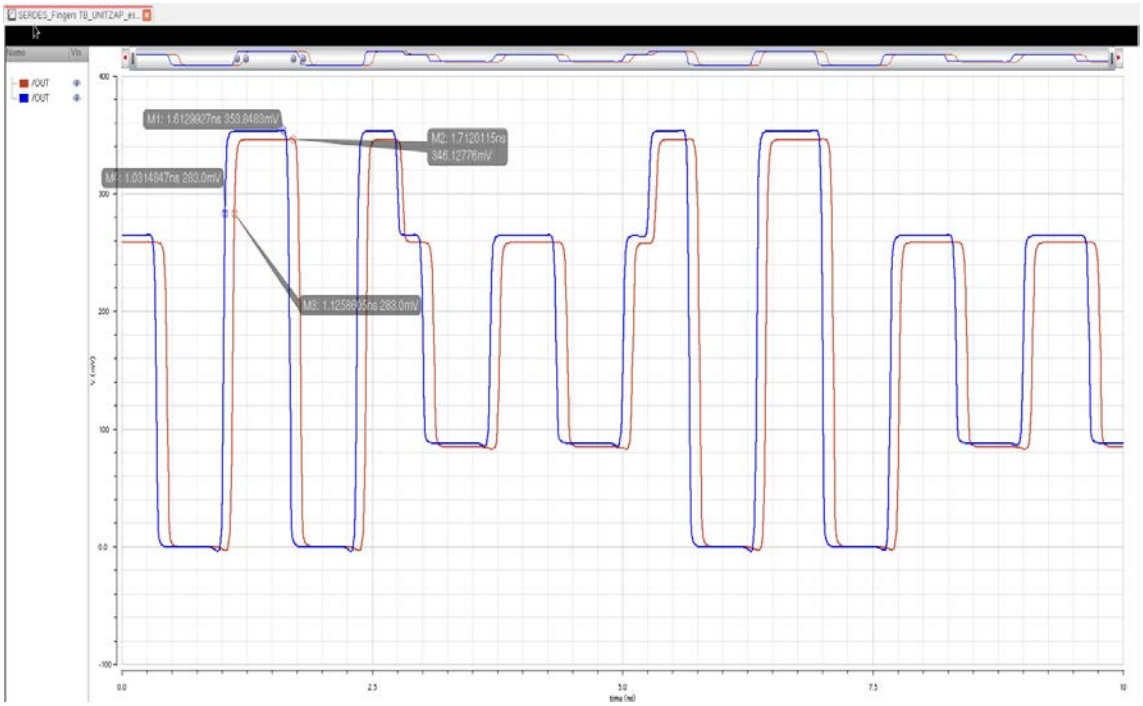


Fig. 98: Post layout transient response of ZAP UNIT, showing a delay of 94.439 pS in the rise time and a little reduction in amplitude.

To validate the pre emphasis characteristic of the Transmitter module we perform postlayout simulation using the testbech of Fig. 101 and Calibre view. There is a delay time of 228 pS, also a reduction of 3.07% in amplitude of signal in the post-layout response.

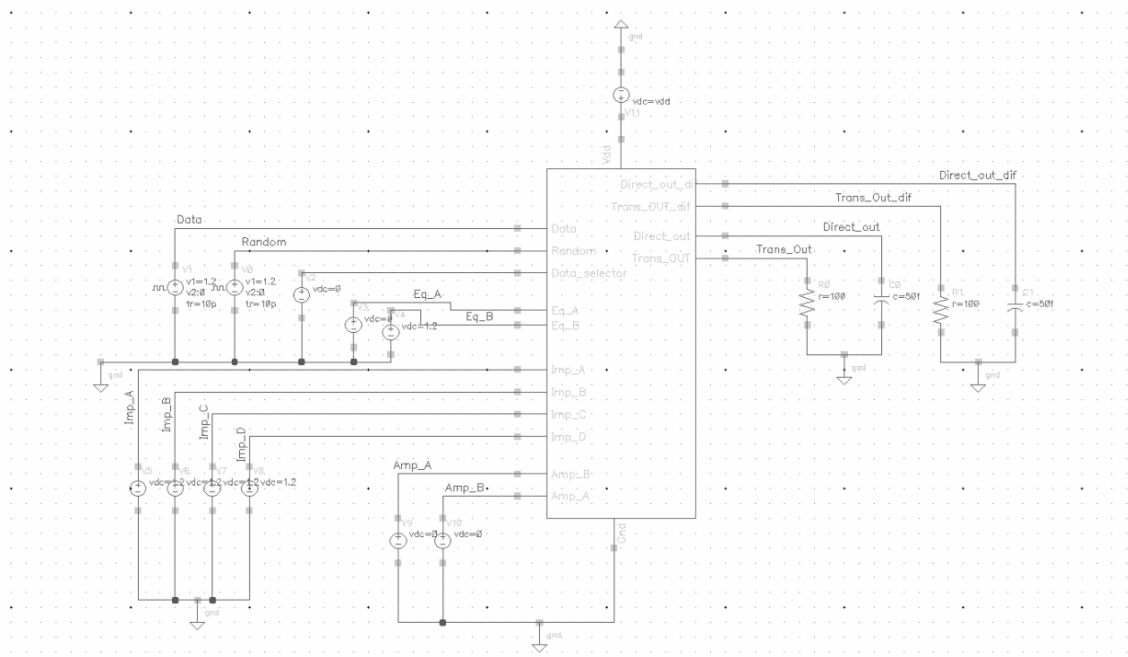


Fig. 99: Testbench for post layout simulation of pre emphasis characteristic of the Transmitter module.

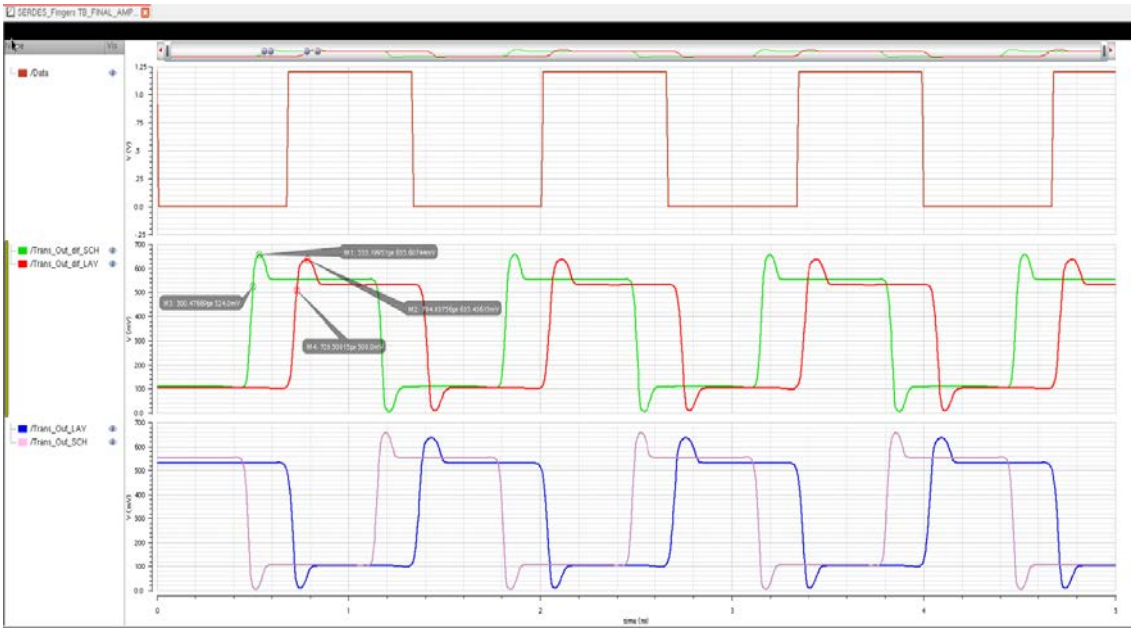


Fig. 100: Post layout transient response of pre emphasis characteristic of the Transmitter module, showing a delay of 228 pS in the rise time and a little reduction in amplitude.

In the same way to validate the amplitude characteristic of the Transmitter module we perform postlayout simulation using the testbech of Fig. 103 and Calibre view. In the obtained transient response (Fig. 104) we observe a delay time of 237 pS, also a reduction of 4.5% in amplitude of signal in the post-layout response.

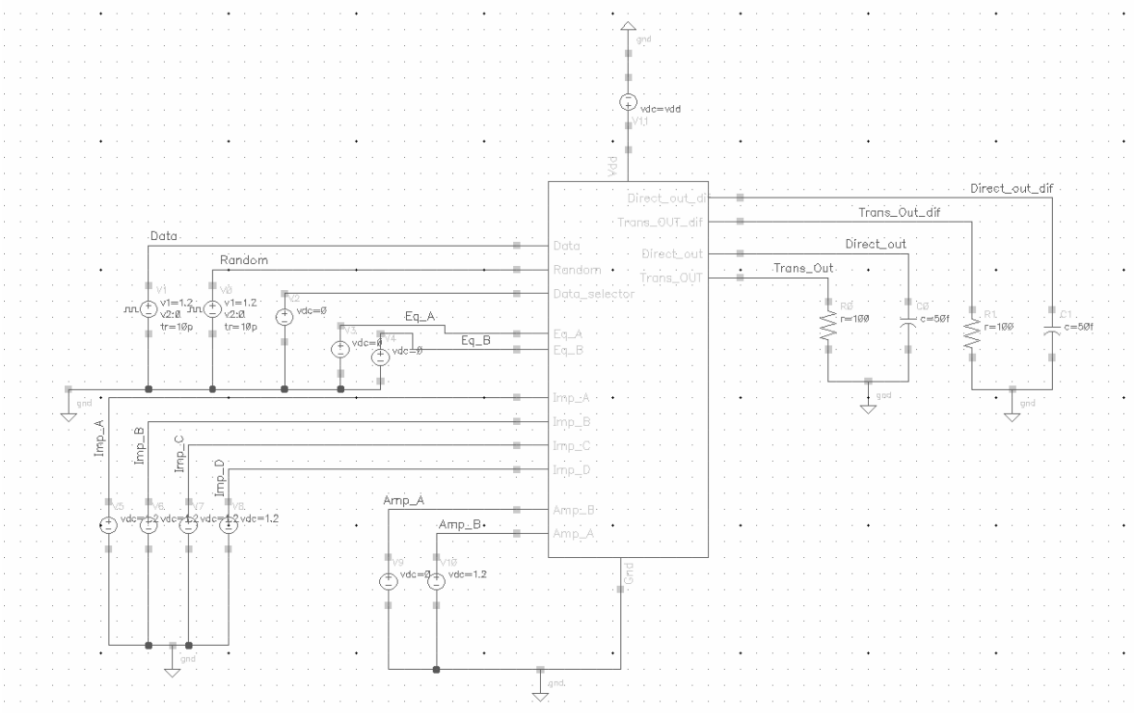


Fig. 101: Testbench for post layout simulation of pre emphasis characteristic of the Transmitter module.

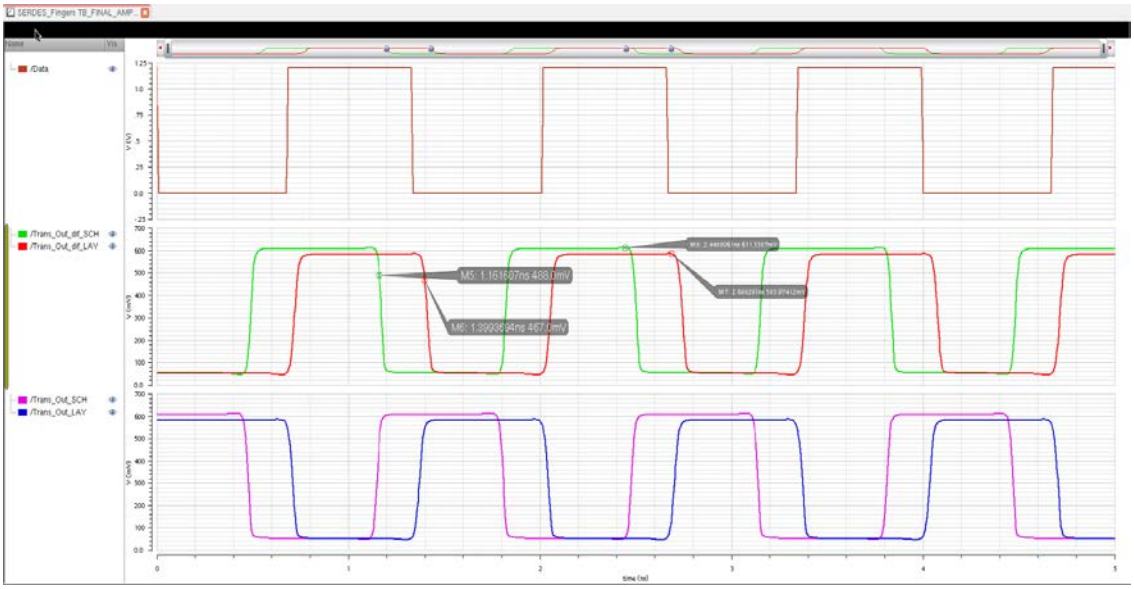


Fig. 102: Post layout transient response of pre emphasis characteristic of the Transmitter module, showing a delay of 237 pS in the rise time and a little reduction in amplitude.

In similar way we perform postlayout simulation using the testbech of Fig. 105 and Calibre view. We observe a delay time of 239.8 pS, also a reduction of 3.07% in amplitude of signal in the post-layout response (Fig 106). These changes are attributed to the parasitic capacitances and size of the transistors. In principle, these changes are not significant and it is expected not serious affectation in the performance of Transmitter module. However, to validate this assumptions further analysis such as Montecarlo must be performed

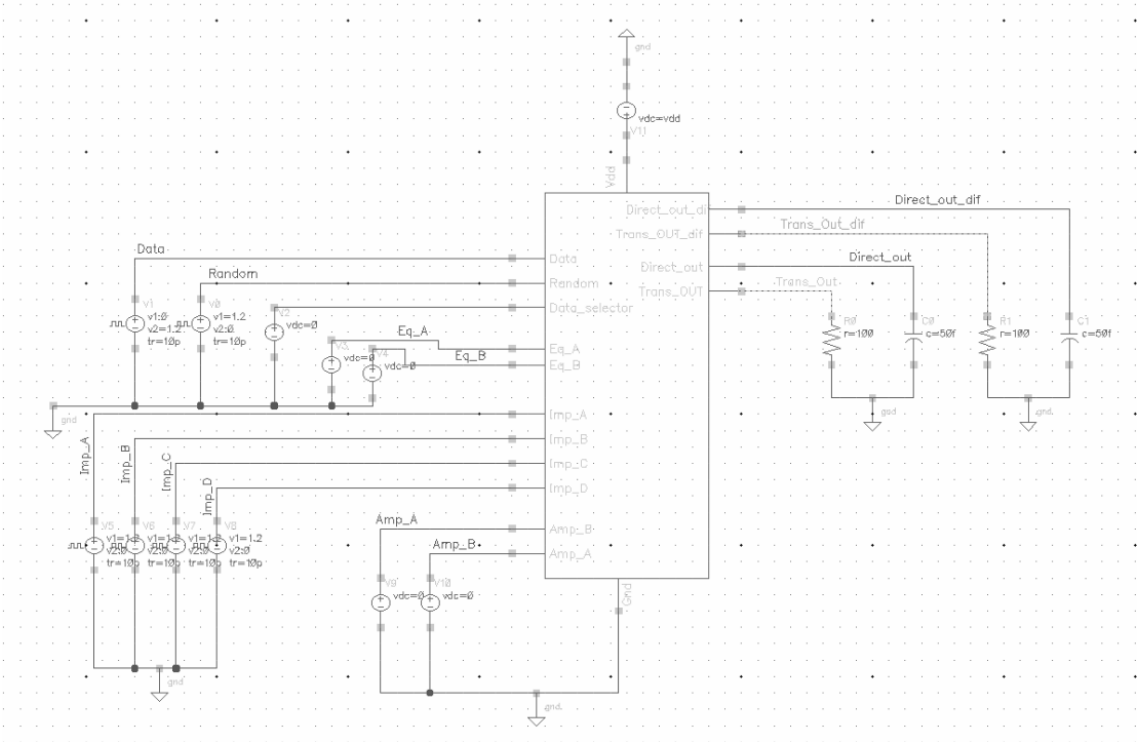


Fig. 103: Testbench for post layout simulation of impedance characteristic of the Transmitter module.

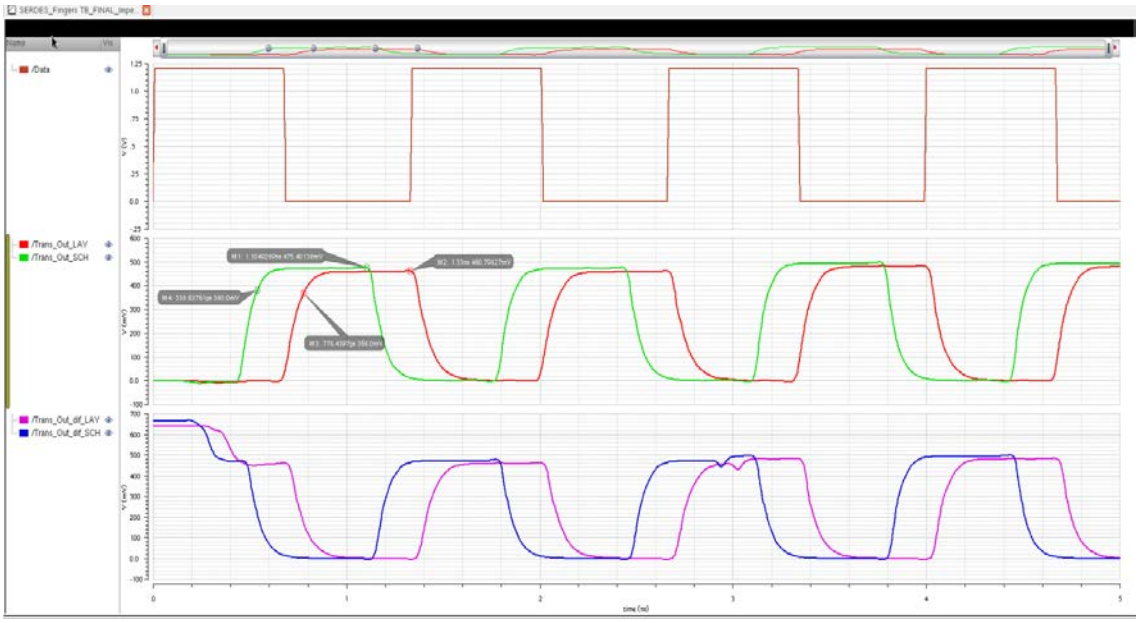


Fig. 104: Post layout transient response of impedance characteristic of the Transmitter module, showing a delay of 239.8 pS in the rise time and a little reduction in amplitude.

CONCLUSIONS

The design of the Transmitter module with complementary outputs was presented. The proper response of whole Transmitter module was validated by simulation using Spectre-Virtuoso.

The behavioral model of the Transmitter module with complementary outputs was completed and simulated. The behavioral model-based response shows that macromodel works properly.

In the performed PVT corners simulation of Transmitter module, we observe significant variations in impedance, pre-emphasis and amplitude features of Transmitter module working under ff and ss PVT corners. We have attributed the origin of such variations to the three state buffer block because it is not a perfect switch, then the voltage drop can affect the impedance in a similar way that in the transmission line. However further analysis using Monte – Carlo should be done in order to determine if the changes in amplitude (impedances) affect the Transmitter behavior and the impact on it.

The layout of Transmitter module with complementary outputs was finished; custom placement and routing of layout of individual blocks Transmitter was performed. The layout was verified using Calibre. Both verifications show not any DRC or LVS errors in the layout of Transmitter module.

Post Layout Simulations was performed. Some small differences between pre-layout and post-layout responses were found, they includes rise time, fall time and amplitude of the signal. Main differences are in timing but it is expected that these differences will not affect the Transmitter module performance.

LIST OF REFERENCES

[1] Planet Analog, “The basics of SerDes (serializers/deserializers) for interfacing” [online] available at http://www.planetanalog.com/document.asp?doc_id=528099 Sep. 16, 2010 [Jan, 1, 2016].

[2] National Instruments, “PCI Express – An Overview of the PCI Express Standard” [online] avalaible at: http://www.ni.com/white-paper/3767/en/ nov 05, 2014 [Jan. 1,2016].

[3] Proyecto de intervención Freescale para Especialidad en Diseño de Sistemas en Chip/ITESO, July 2015.

[4] Federico Lopez Lobato “Behavioral Analog Design, Macromodelng with SPICE & Verilog A”, ITESO, Guadalajara, Jalisco, Notes of Course, Oct. 20, 2015.

[5] Romeo Covarrubias Larios, Enrique Maximiliano Calderón Ramos, Tesina “Diseño de un transmisor serial de datos con impedancia, amplitud y pre énfasis programable”, Departmento de Electronica, Sistemas e Informatica, ITESO, Octubre 2007.

[6] Silvaco, SamrtSpice training program, part 5: SmartSpice Verilog-A, Analog/Mixed signal simulation, available on line: http://www.silvaco.com/content/training/smartspice_training_part5.pdf, [May, 2016].

[7] Joel Antonio Nuñez Corona and Aarón Desiga Orendain, Internal Report, ESC-ITESO_R01-2015_JANC, 2016, Department of electronics, Systems and Informatics, ITESO, Dec., 2015

[8] Carlos Cesar López Félix, Tesina “Diseño del módulo Transmisor Serial de Datos de Sistema SerDes para Protocolo PCI express 1”, Departmento de Electronica, Sistemas e Informatica, ITESO, Guadalajara, Jalisco, Tesina, Diciembre 2015.

APENDIX

Apendix a

Multiplexer 2 to 1 Verilog – A.

```
`include "constants.vams"
`include "disciplines.vams"

module Mux_2to1(ina, inb, ctrl, out);

/*Defining inputs and outputs*/
input ina, inb, ctrl;
output out;
/*Defining electrical signals*/

electrical ina, inb, ctrl, out;
parameter real vdd=1.2;
parameter real rf_time=0.5n;
integer datoa, datob, control, sal;

// Module contents
analog
begin
@(timer(0)) begin
if (V(ina)>(vdd/2))
datoa=1;
else
datob=0;
if (V(inb)>(vdd/2))
datob=1;
else
datob=0;
if (V(ctrl)>(vdd/2))
control=1;
else
control=0;
end

@(cross(V(ina)-vdd/2,+1))
 datoa=1;
 @(cross(V(ina)-vdd/2,-1))
 datoa=0;
 @(cross(V(inb)-vdd/2,+1))
 datob=1;
 @(cross(V(inb)-vdd/2,-1))
 datob=0;
 @(cross(V(ctrl)-vdd/2,+1))
 control=1;
 @(cross(V(ctrl)-vdd/2,-1))
 control=0;
 if (control==0)
 sal=datoa;
 else
```

```
 sal=datob;

 V(out) <+ transition(sal*vdd,0, rf_time, rf_time);

end

endmodule // Mux_2to1
```

Apendix b

Basic cell code.

```
`include "constants.vams"
`include "disciplines.vams"

module basic_cell(in, en1, en2, out);

// Module contents

input in, en1, en2;
output out;

electrical in, en1, en2, out;
parameter real vdd=1.2;
parameter real R1=820;
parameter real R2=820;
parameter real trise=10p;
parameter real tfall=10p;
parameter tdelay=0;
integer en_1, en_2;

analog begin

@(timer(0)) begin
        if (V(en1)>(vdd/2))
   en_1=1;
   else
   en_1=0;
   if (V(en2)>(vdd/2))
   en_2=1;
   else
   en_2=0;
end

if(en_1==1&&en_2==0)
begin
        V(in,out)<+ R1*I(in,out);
        V(in,out)<+ R2*I(in,out);
end

else if (en_1==1&&en_2==1)
begin
        V(in,out)<+R2*I(in,out);
end

end
```

```
endmodule // basic_cell
```

Apendix c
Decoder 4 to 12 Verilog – A

```
`include "constants.vams"
`include "disciplines.vams"

module deco4_12(EN1,EN2,EN3,EN4,S0,S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11);

//Defining inputs and outputs
input EN1,EN2,EN3,EN4;
output S0,S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11;

//Defining variables and electrical quatities
electrical EN1,EN2,EN3,EN4,S0,S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11;
parameter real vdd=1.2;
parameter real rf_time =0.5n;
integer
dEN1,dEN2,dEN3,dEN4,dS0,dS1,dS2,dS3,dS4,dS5,dS6,dS7,dS8,dS9,dS10,dS11;

// Module contents

analog
begin
@(timer(0)) begin
 if (V(EN1)>(vdd/2))
 dEN1=1;
 else
 dEN1=0;
 if (V(EN2)>(vdd/2))
 dEN2=1;
 else
 dEN2=0;
 if (V(EN3)>(vdd/2))
 dEN3=1;
 else
 dEN3=0;
 if (V(EN4)>(vdd/2))
 dEN4=1;
 else
 dEN4=0;
 end

 @(cross(V(EN1)-vdd/2,+1))
 dEN1=1;
 @(cross(V(EN1)-vdd/2,-1))
 dEN1=0;
 @(cross(V(EN2)-vdd/2,+1))
 dEN2=1;
 @(cross(V(EN2)-vdd/2,-1))
 dEN2=0;
```

```
 @(cross(V(EN3)-vdd/2,+1))
 dEN3=1;
 @(cross(V(EN3)-vdd/2,-1))
 dEN3=0;

 @(cross(V(EN4)-vdd/2,+1))
 dEN4=1;
 @(cross(V(EN4)-vdd/2,-1))
 dEN4=0;


 if (dEN4==0&&dEN3==0&&dEN2==0&&dEN1==0)
 begin
 dS0=0;
dS1=0;dS2=0;dS3=0;dS4=0;dS5=0;dS6=0;dS7=0;dS8=0;dS9=0;dS10=0;dS11=0;
 end

 else if (dEN4==0&&dEN3==0&&dEN2==0&&dEN1==1)
 begin
 dS11=0;
dS10=0;dS9=0;dS8=0;dS7=0;dS6=0;dS5=0;dS4=0;dS3=0;dS2=0;dS1=0;dS0=1;
 end

 else if (dEN4==0&&dEN3==0&&dEN2==1&&dEN1==0)
 begin
 dS11=0;
dS10=0;dS9=0;dS8=0;dS7=0;dS6=0;dS5=0;dS4=0;dS3=0;dS2=0;dS1=1;dS0=1;
 end
 else if (dEN4==0&&dEN3==0&&dEN2==1&&dEN1==1)
 begin
 dS11=0;
dS10=0;dS9=0;dS8=0;dS7=0;dS6=0;dS5=0;dS4=0;dS3=0;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==0&&dEN3==1&&dEN2==0&&dEN1==0)
 begin
 dS11=0;
dS10=0;dS9=0;dS8=0;dS7=0;dS6=0;dS5=0;dS4=0;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==0&&dEN3==1&&dEN2==0&&dEN1==1)
 begin
 dS11=0;
dS10=0;dS9=0;dS8=0;dS7=0;dS6=0;dS5=0;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==0&&dEN3==1&&dEN2==1&&dEN1==0)
 begin
 dS11=0;
dS10=0;dS9=0;dS8=0;dS7=0;dS6=0;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==0&&dEN3==1&&dEN2==1&&dEN1==1)
 begin
 dS11=0;
dS10=0;dS9=0;dS8=0;dS7=0;dS6=1;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==1&&dEN3==0&&dEN2==0&&dEN1==0)
 begin
```

```
 dS11=0;
dS10=0;dS9=0;dS8=0;dS7=1;dS6=1;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==1&&dEN3==0&&dEN2==0&&dEN1==1)
 begin
 dS11=0;
dS10=0;dS9=0;dS8=1;dS7=1;dS6=1;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==1&&dEN3==0&&dEN2==1&&dEN1==0)
 begin
 dS11=0;
dS10=0;dS9=1;dS8=1;dS7=1;dS6=1;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==1&&dEN3==0&&dEN2==1&&dEN1==1)
 begin
 dS11=0;
dS10=1;dS9=1;dS8=1;dS7=1;dS6=1;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==1&&dEN3==1&&dEN2==0&&dEN1==0)
 begin
 dS11=1;
dS10=1;dS9=1;dS8=1;dS7=1;dS6=1;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==1&&dEN3==1&&dEN2==0&&dEN1==1)
 begin
 dS11=1;
dS10=1;dS9=1;dS8=1;dS7=1;dS6=1;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==1&&dEN3==1&&dEN2==1&&dEN1==0)
 begin
 dS11=1;
dS10=1;dS9=1;dS8=1;dS7=1;dS6=1;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end
 else if (dEN4==1&&dEN3==1&&dEN2==1&&dEN1==1)
 begin
 dS11=1;
dS10=1;dS9=1;dS8=1;dS7=1;dS6=1;dS5=1;dS4=1;dS3=1;dS2=1;dS1=1;dS0=1;
 end

 V(S0) <+ transition(dS0*vdd,0, rf_time, rf_time);
 V(S1) <+ transition(dS1*vdd,0, rf_time, rf_time);
 V(S2) <+ transition(dS2*vdd,0, rf_time, rf_time);
 V(S3) <+ transition(dS3*vdd,0, rf_time, rf_time);
 V(S4) <+ transition(dS4*vdd,0, rf_time, rf_time);
 V(S5) <+ transition(dS5*vdd,0, rf_time, rf_time);
 V(S6) <+ transition(dS6*vdd,0, rf_time, rf_time);
 V(S7) <+ transition(dS7*vdd,0, rf_time, rf_time);
 V(S8) <+ transition(dS8*vdd,0, rf_time, rf_time);
 V(S9) <+ transition(dS9*vdd,0, rf_time, rf_time);
 V(S10) <+ transition(dS10*vdd,0, rf_time, rf_time);
 V(S11) <+ transition(dS11*vdd,0, rf_time, rf_time);

end

endmodule // deco4_12
```

Apendix d
Inverter delay Verilog – A code.

```
`include "constants.vams"
`include "disciplines.vams"

module inv_delay(in, out);
input in;
output out;
electrical in, out;
parameter real vout_high = 1.2;
parameter real vout_low = 0;
parameter real vth = 0.6;
parameter real tdelay = 0.53n; //propagation delay
parameter real trise = 10p; //rise time
parameter real tfall = 10p; //fall time
real val;
analog begin
@( initial_step )
begin // initial condition
if ( V(in) > vth ) val = vout_low;
else val = vout_high;
end
@( cross( V(in) - vth, +1) ) val = vout_low; // Vin>vth => vout_low
@( cross( V(in) - vth, -1) ) val = vout_high; // Vin<vth => vout_high
V(out) <+ transition(val, tdelay, trise, tfall);
end
endmodule // inv_delay
```

Apendix e
Decoder 2 to 3 Verilog – A code

```
`include "constants.vams"
`include "disciplines.vams"

module decod2to3(I1, I0, S2, S1, S0);
//Defining inputs and outputs
input I1,I0;
output S2,S1,S0;

electrical I1, I0, S2, S1, S0;
parameter real vdd=1.2;
parameter real rf_time =0.5n;
integer dI1, dI0, dS2, dS1, dS0;

analog
begin
@(timer(0)) begin
 if (V(I0)>(vdd/2))
 dI0=1;
 else
 dI0=0;
 if (V(I1)>(vdd/2))
 dI1=1;
```

```
 else
 dI1=0;
end

@(cross(V(I0)-vdd/2,+1))
I0=1;
@(cross(V(I0)-vdd/2,-1))
dI0=0;
@(cross(V(I1)-vdd/2,+1))
dI1=1;
@(cross(V(I1)-vdd/2,-1))
dI1=0;


if (dI1==0&&dI0==0)
begin
        dS2=0; dS1=0;dS0=0;
end

else if (dI1==0&&dI0==1)
begin
        dS2=0; dS1=0;dS0=1;
end

else if (dI1==1&&dI0==0)
begin
        dS2=0; dS1=1;dS0=1;
end

else if (dI1==1&&dI0==1)
begin
        dS2=1; dS1=1;dS0=1;
end


 V(S0) <+ transition(dS0*vdd,0, rf_time, rf_time);
 V(S1) <+ transition(dS1*vdd,0, rf_time, rf_time);
 V(S2) <+ transition(dS2*vdd,0, rf_time, rf_time);

 end

endmodule // end of module decod2to3
```

Apendix f
Inverter Verilog – A code.

```
//this module is added to Transmitter Verilog-A code in order to generate the
complementary output

`include "constants.vams"
`include "disciplines.vams"

module inverter(in, out, vdd, gnd);
input in,vdd,gnd;
```

```
output out;
electrical in, out, vdd, gnd;
parameter real tdelay = 0.1p; //propagation delay
parameter real trise = 10p; //rise time
parameter real tfall = 10p; //fall time
real val;
analog begin
@( initial_step )
begin // initial condition
if ( V(in) > V(vdd)/2 val = vout_low;
else val = vout_high;
end
@( cross( V(in) - V(vdd)/2, +1) ) val = vout_low; // Vin> V(vdd)/2 => vout_low
@( cross( V(in) - V(vdd)/2, -1) ) val = vout_high; // Vin< V(vdd)/2 => vout_high
V(out) <+ transition(val, tdelay, trise, tfall);
end
endmodule // inverter
```

Appendix g
List of archives of Transmitter module design in Virtuoso cadence

Name of archive	Library	views
Final_Diferencial_Inverter	SERDES_Fingers	Schematic, symbol, layout
FINAL	SERDES_Fingers	Schematic, symbol, layout
TB_FINAL_AMPLITUD_Inverter 2	SERDES_Fingers	Schematic, config
TB_FINAL_AMPLITUD	SERDES_Fingers	Schematic,
TB_FINAL_Equalization_inverted	SERDES_Fingers	Schematic, config
TB_FINAL_Impedance_inverted	SERDES_Fingers	Schematic, config
TB_FINALDIF_Impedance	SERDES_Fingers	Schematic,
TX_VerA	SERDES_VerA	Schematic, symbol,
TX_VerA_Equalization	SERDES_VerA	Schematic
TX_VerA_Impedance	SERDES_VerA	Schematic
TX_VerA_Impedance_MUXVerA	SERDES_VerA	Schematic
TX_VerA_Impedance_ZAP UNIT_VerA	SERDES_VerA	Schematic
TX_VerA_Impedance _deco2to3verA	SERDES_VerA	Schematic
TX_VerA_Impedance _deco12	SERDES_VerA	Schematic
TX_VerA_Equalization_Inverter	SERDES_VerA	Schematic
TX_Ver_siniv	SERDES_VerA	Schematic
TB_FINAL_AMPLITUD	SERDES_VerA	Schematic,
Final_DECO2to3VerA	SERDES_VerA	Schematic,
Final_DECO4to12VerA	SERDES_VerA	Schematic,
Final_MUXVerA	SERDES_VerA	Schematic,
Final_ZAPverA	SERDES_VerA	Schematic,
BasicCell_VerA2	SERDES_VerA	Verilog-A, symbol
ZAP UNIT_VerA2	SERDES_VerA	Schematic, symbol
DECO2to3VerA2	SERDES_VerA	Verilog-A, symbol
DECO4to12VerA2	SERDES_VerA	Verilog-A, symbol
Inverter	SERDES_VerA	Verilog-A, symbol
InverterDel2	SERDES_VerA	Verilog-A, symbol
Mux2to1VerA2	SERDES_VerA	Verilog-A, symbol
TIED_verA2	SERDES_VerA	Verilog-A, symbol
TX_Final	SERDES_Fingers	Schematic, symbol, layout. calibre

Appendix h

Settings for voltage sources for testbench of transient response for Impedance of Transmitter module.

Label	Type	V1 (V)	V2 (V)	Period (S)	Pulse (S)	Fall time (S)	Rise time (S)
V0	vpulse	0	1.2	2.66 n	1.33 n	10 p	10 p
V1	vpulse	0	1.2	1.33 n	667 p	10 p	10 p
V2	vdc	0					
V3	vdc	0					
V4	vdc	0					
V5	vpulse	1.2	0	5.32n	2.66n	10p	10p
V6	vpulse	1.2	0	10.6n	5.32n	10p	10p
V7	vpulse	1.2	0	21.2n	10.6n	10p	10p
V8	vpulse	1.2	0	42.4n	21.2n	10p	10p
V9	vdc	0					
V10	vdc	0					

Appendix i

Settings for voltage sources for testbench to validate the Pre emphasis response of complementary Transmitter module.

Label	Type	V1 (V)	V2 (V)	Period (S)	Pulse (S)	Fall time (S)	Rise time (S)
V0	vpulse	0	1.2	2.66 n	1.33 n	10 p	10 p
V1	vpulse	0	1.2	1.33 n	667 p	10 p	10 p
V2	vdc	0					
V3	vdc	0					
V4	vdc	1.2					
V5	vdc	1.2					
V6	vdc	1.2					
V7	vdc	1.2					
V8	vdc	1.2					
V9	vdc	0					
V10	vdc	0					

Appendix j

Settings for voltage sources for testbench for Pre emphasis response validation of Transmitter module based on its macromodel.

Label	Type	V1 (V)	V2 (V)	Period (S)	Pulse (S)	Fall time (S)	Rise time (S)
V0	vpulse	0	1.2	2.66 n	1.33 n	10 p	10 p
V1	vpulse	0	1.2	1.33 n	667 p	10 p	10 p
V2	vdc	0					
V3	vdc	0					
V4	vdc	1.2					
V5	vdc	1.2					
V6	vdc	1.2					
V7	vdc	1.2					
V8	vdc	1.2					
V9	vdc	0					

V10	vdc	0					

Appendix k

Settings for voltage sources for testbench to validate the Impedance modulation of Transmitter module in 130nm

Label	Type	V1 (V)	V2 (V)	Period (S)	Pulse (S)	Fall time (S)	Rise time (S)
V0	vpulse	1.2	0	42.4n	21.2n	10p	10p
V1	vpulse	1.2	0	21.2n	10.6n	10p	10p
V2	vpulse	1.2	0	10.6n	5.32n	10p	10p
V3	vpulse	1.2	0	5.32n	2.66n	10p	10p
V4	vdc	0					
V5	vdc	0					
V6	vdc	0					
V7	vdc	1.2					
V8	vdc	0					
V9	vdc	0					
V10	vdc	0					
V11	vdc	0					
V12	vdc	1.2					

Appendix l

Settings for voltage sources for testbench bench for post layout simulation of DECO2-3.

Label	Type	V1 (V)	V2 (V)	Period (S)	Pulse (S)	Fall time (S)	Rise time (S)
V2	vpulse	0	1.2	10.6 n	5.32 n	10 p	10 p
V1	vpulse	0	1.2	5.32 n	2.66 n	10 p	10 p
V0	vdc	1.2					