High-Speed Serial Link for Low-Power Memories

ANELY CATALINA MELENDEZ RASGADO
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

High-Speed Serial Link for Low-Power Memories

Anely Catalina Melendez Rasgado
an7038me-s@eit.lth.se

Department of Electrical and Information Technology
Lund University

Supervisor: Babak Mohammadi
babak.mohammadi@eit.lth.se

Examiner: Pietro Andreani
pietro.andreani@eit.lth.se

June 20, 2018

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

A bidirectional serial link on-chip implementation is going to be assessed so as to set the option of using it as a replacement of the actual parallel interconnection used to transfer data between different memory banks in an embedded low-power memory unit. Asynchronous communication is the protocol selected and current mode pulse signaling is the technique used to transfer data. A 32-bit data packet is transmitted with a throughput of 10.66 Gbps. The interconnect was designed using 28nm CMOS BULK technology from TSMC and was simulated with Cadence Analog Spectre; it occupies 902.21 μm^2 and consumes 4.93 pJ/bit. The research was done in collaboration with the company Xenergic.

Acknowledgements

I would first like to express my sincere gratitude to my supervisor Babak Mohammadi, CEO of Xenergic, who gave me this opportunity and offered me his continuous support during this research. I deeply appreciate the time he has dedicated to answer my questions and to give me guidance whenever I needed it. I thank him for his patience, enthusiasm and his endless encouragement that helped me to stay motivated during the completion of this project. I specially appreciate his constructive advice and feedback, which helped me to improve the quality of this project.

I would like to thank my professors from the Faculty of Engineering LTH at Lund University, Joachim Rodrigues and Pietro Andreani for always offering me their support and help when I needed it. I deeply appreciate your valuable comments and the knowledge that you have shared with me during the last two years.

I am grateful to Lund University Global Scholarship programme and CONACYT in Mexico for making it possible for me to study in Sweden. I want to express my deep gratitude to all my friends, in Mexico and in Sweden for their kindness, generosity and for being a major source of support through this journey. Heartfelt thanks go to Berta, who shared with me long days at work, technical discussions and also fun afternoons.

A very special word of thanks goes to my family. First to my parents: Ana and Luis, to whom I dedicate this milestone, for always prioritizing my education, giving me unconditional love, support and encouragement and for always believing in me. Their courage, resilience and strength are an inspiration to me and I can not express with words how much I appreciate all their sacrifices. Thank you. And finally to my sisters: Lorena and Ashly, for always being there for me and giving me continuous moral support. Their encouragement kept me going in the most difficult times and this accomplishment would not have been possible without them.

Popular Science Summary

Electronic devices play an important role in different aspects of our daily lives. We are surrounded by gadgets that possess a lot of features and in order to make this possible, the integration of different systems within a single chip is necessary. The development of new technologies and innovation in circuit design are required in order to be able to increase the functionalities of a chip, optimize area and minimize energy consumption. Among different types of circuits that exist, memories are an important element and one of the biggest building blocks within a chip, so when designing memories, area is always a concern and a parameter to optimize.

The information that is going to be used and processed by other circuits, is usually stored and read from a memory. This information (data) will be transferred across different functional units and in order to do so, wires routed in parallel are commonly used. This is a critical aspect that needs to be improved since it represents a high area cost. Moreover, different phenomena arise when having metals routed in parallel, which in turn represent a detriment in the overall system performance.

That is why this thesis aims to assess the cost of replacing the parallel interconnection with a serial link in order to optimize area. One of the concerns of doing so, is to be able to transfer the same amount of data per unit time as with a parallel link. The reason of it, is that in order to make this possible, the system needs to operate at higher clock frequencies.

On-chip communication can be synchronous, meaning that transmission and reception of data are both coordinated with one another by using the same clock signal, or it can be asynchronous, meaning that the transmitter and the receiver are timed independently from each other. In this thesis, the asynchronous communication protocol is implemented, since it is simpler and less expensive.

Data representation and proper modeling of the transmission channels are necessary in order to design the system's circuits. Proper architectures were selected in order to accomplish data transfer at the required speed.

The initial assessment results show that the proposed system is a viable solution for intermemory communication that can be implemented as a way of memory design optimization.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Thesis Motivation

Today's systems on chip integrate a large number of analog and digital circuits, being embedded memories the ones that occupy the largest area in a die, around 60% - 70% and 75% - 85% of the transistor count in some of today's ICs [1]. Different memories are available to choose from, according to the required purpose on the SoC and specific design criteria. SRAM memories represent a good compromise between speed, power consumption and area occupation among different embedded memory solutions and because of that, they are the preferred memory type for applications like cache memories in microprocessors and ASICs. As new technologies emerge, the scalability of such memories is achieved by employing different design techniques and architectures. For instance, in [2] the number of bits per bitline is reduced where the target memory macro is selected using a new decoding and multiplexing scheme achieving energy efficiency. However, a higher number of memory banks and more data buses are necessary to maintain the memory's capacity, making inter-memory communication more intricate.

It is a common practice to transmit data between memory banks using parallel buses, providing high-throughput at the cost of large area occupation and higher power dissipation, added to a limited performance due to coupling effects such as crosstalk and complex routings. The latter is also limited to the available clock rate and skew [3]. Moreover, delay uncertainty introduced by repeaters, layout parasitic effects and process variations [4] have also a big impact on the quality and speed of transmission as well as in signal degradation along the interconnect.

One alternative to bit-parallel buses is to have a bit-serial interconnection. Several advantages derive from this type of communication, such as less parasitics, smaller area occupation and reduced power consumption since less number of wires are needed to transmit data. However, the major challenge when implementing a serial communication link is to maintain the same throughput as with its parallel counterpart. An implementation of the aforementioned alternative is presented as a proof of concept in this thesis along with an initial area and energy assessment aiming to provide an option to minimize memory area.

Project Specifications

The objective of this project is to design a high-speed link for bit serial transmission in 28nm CMOS BULK process. The specifications are set by the existing memory to which the present link is going to be integrated .

Specification	**Value**
Throughput	10.66 Gbps
Packet's Size	32 bits
Max. Available Area	1500 µm ^{2}
Routing	M4 and M6

Table 1.1: High-speed serial link design specifications

In addition to the requirements specified in table 1.1, the serial link should be able to transmit and receive data at both ends (bidirectional), passive devices can not be used and layout parasitics should be minimized.

Thesis Organization

The outline of this thesis is listed below:

- **Chapter 2:** technical knowledge collected during the development of this project is presented so as to have an understanding of important concepts in which the design is based on.
- **Chapter 3:** the system model is presented along with design considerations.
- **Chapter 4:** it consists of the design process of all the circuits involved, including detailed explanations of the selected architectures in each case.
- **Chapter 5:** layout design is discussed.
- **Chapter 6:** the results that were obtained in pre and post-layout simulations are examined and compared.
- **Chapter 7:** conclusions are drawn and future work is mentioned.

Chapter 2

Background Research

Common Terminology

Link: the medium through which information travel from source (transmitter) to destination (receiver). For this project, the link is the wire, and its design involves not only the physical connection between receiver and transmitter but driver and load as well.

Packet: a set of information that is going to be transmitted along the interconnects. It can contain control and data bits from memory.

Symbols: a symbol refers to a certain state of the channel with a specific duration of time. Their purpose is to represent data, as it is encoded either in each unit or within state changes. The unit interval is the lapse between transitions and it can be measured on an eye diagram.

Latency: it is the time that it takes for a packet to travel along the channel. From the time it is sent until it is received at the end where it is going to be processed.

Throughput: this is the real amount of data that traverse the interconnect. The value is a metric of the actual system's performance and it can be affected by different external factors. It is expressed in bits-per-second.

ISI: Intersymbol interference refers to the effect that causes a distorted signal when one or more symbols interfere with each other while being transmitted causing noise on the arriving signal at the receiver. Different causes provoke this phenomenon such as multi-path propagation (in the case of wireless communication), limited bandwidth, reflections, signal delays, etc. As this is undesirable in communication systems, its occurrence must be minimized . One way of doing so, is by implementing equalization techniques and error correcting codes [5].

LEDR: Level-Encoded Dual-Rail signaling protocol is an NRZ encoding scheme based on binary current sensing [6]. That is, the signal traveling along one pair of wires changes its value every time a new bit is transmitted. This scheme is delay insensitive and it encodes one bit of data in a pair of wires (or rails). One of the wires will contain data information and the other one will contain the phase that corresponds to each data bit. The advantages of this scheme include that there is no need to send a reset bit in between data bits (NRZ) which represents a significant system-level throughput asset. Moreover, it also increases energy efficiency, since only one transition occurs on the wire per data transfer [7].

Crosstalk: the capacitance between two or more adjacent wires causes an undesired transfer of signals among different communication channels. The importance of this effect is more relevant for long wires and it increases with high switching frequencies of the signals traveling along these conductors. As it is shown in Figure 2.1, if the signal in wire0 is meant to maintain its actual state and a neighboring signal (in wire1) switches, the latter will introduce noise to the former. In this case, wire0 will be the *victim* and wire1 will be the *agressor* [8]. In order to minimize crosstalk, different approaches can be taken:

- Maximize spacing between wires running in parallel or avoid parallel routing altogether.
- Add shields to critical signals.
- Interleave buses with different switching times.
- Implement crosstalk cancellation techniques such as staggered repeaters, charge compensation, and twisted differential signaling [9].

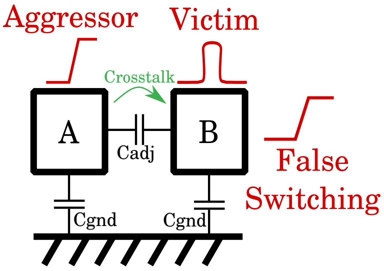


Figure 2.1: Crosstalk between two adjacent wires A and B

Random Binary Data

A serial data signal is a binary sequence of logical "0"s or "1"s that carry information. The speed of this data type is based on the bit rate and determines the channel capacity [10]. If the time interval corresponding to each bit is (T_b)

seconds, the bit rate is $\mathrm{R_b}$ and it is given by:

$$R_{\mathrm{b}} = \frac{1}{T_{\mathrm{b}}} \frac{bits}{second}$$

.

If the logical "0"s are represented by a zero voltage or current, the DC content (or average value) is non-zero whereas in a zero DC content signal both "1"s and "0"s have equal values but with opposite polarity. A binary sequence of random generated data can contain long chains of either "0"s or "1"s (*runs*) (see Figure 2.2). In other words, the signal exhibits a "low transition density", which in turn provokes failures and makes transceiver operations more challenging in general. According to optical communication standards, the maximum "run length" may be as long as 72 bits and to avoid exceeding it, data encoding is implemented at the transmitter [11]. Data bits are usually accompanied by extra bits that denote data validity, contain control information or are meant to perform error detection and correction. These type of bits are called *overhead* and along with data bits are part of a protocol frame or transmission packet.

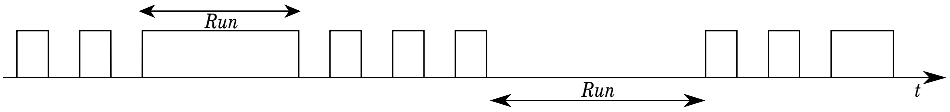


Figure 2.2: Long run in a random data sequence

Data Formats

There are different formats to represent data, among which are: NRZ, RZ, 8B/10B, Manchester coding, amplitude and FSK coding.

- **NRZ and RZ Data:**

  In NRZ each bit is represented either by the high or the low state of a pulse, which has a predetermined duration. This format is highly employed in high-speed applications.

  RZ data consists of pulses that last half bit period, going back to the low state afterwards. This means that in between two symbols containing information, there exists a "0" symbol. One of the drawbacks of this format is that bandwidth occupation is twice as much as with NRZ data [11].

  The aforementioned formats don't contain any clock information while in low state and have DC content.

- **8B/10B Coding:**

  In order to have a DC balanced signal and limit the maximum "run length", the overhead is increased by 25%, i.e. instead of having 8 bits, the sequence is increased to 10 bits with a maximum run length of 5 bits.

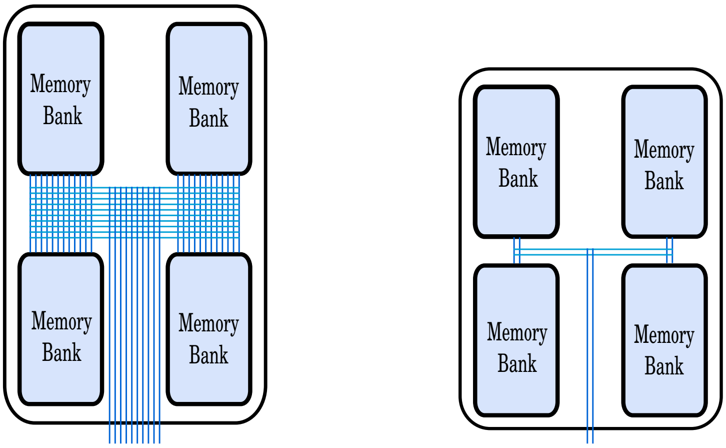
- **Amplitude and FSK coding:** Sine waves are used for encoding data signals instead of square-wave signals. Amplitude, frequency and phase modulation are possible. The signal spectrum of sine wave signals lacks harmonic waves, thus making it easier to comply with EMC specifications [12].

- **Manchester coding:** In this format, the phase angle of the signal contains the bit information. It has no mean values and it is rarely used in high speed systems [11].

Parallel vs Serial Communication

Parallel communication relies on N parallel wires that can carry N bits at the same time, whereas serial communication employs one or two wires to transfer one bit at each time. Among the differences in between these two types of communication are:

- Parallel links operates at the "RC" region while serial links operate at "RLC" region [3].
- There is low coupling noise in a serial link, since different signaling schemes can be employed in order to mitigate crosstalk.
- The skew in a serial link is much smaller than in a parallel link [3]. It can be neglected since communication will be delay insensitive (all the symbols will have the same delay).
- Area occupation of a parallel link increases with the size of a data packet, whereas for the case of a serial link it remains the same.



(a) Parallel link

(b) Serial link

Figure 2.3: Replacement of parallel bus with serial link

In Figure 2.3a the connection in between memory banks is shown as parallel buses. These connections are the ones that are replaced by a serial interconnect (Figure 2.3b).

Asynchronous Serial Data Communication

As the number of devices in a single chip increases with each new technology, clock distribution among all the blocks as well as synchronization to a reference clock become a serious task. One way to overcome this challenge is by having independently clocked subsystems that interact with each other. By doing this, idle modules can be powered down and clock skew related timing failures are compounded to locally synchronous subsystems [13]. In the memory context in which this system is intended to work, area occupation is the major concern, so the implementation of a clock generating and recovery circuits (PLL and CDR) is avoided by making the communication asynchronous.

Asynchronous communication relies on how the signals are going to be sent, since both ends of the transmission channel keep timing independent from each other. Some considerations need to be taken into account when attempting this type of communication between two different entities. For instance, the amount of data that is going to be transferred (number of bits), how this is going to be represented (signaling scheme) and the rate at which it is going to be transmitted and received. In asynchronous communication links, the LSB arrives first and one of the drawbacks is the large *overhead*, since several bits are needed for transmission control.

Pulse Signaling Scheme

The serial link employs *pulse dual-rail* encoding scheme to transmit data as it enables pulse signaling as well as differential signaling. The aforementioned techniques, used along with each other, eliminate the need of data decoding logic implementation. With this scheme, two wires are used to represent data. One wire represents bit "0" and the other one represents bit "1". When a valid bit is transmitted, there is a pulse and no-pulse pair on the wires.

For every clock cycle a bit is transmitted, however the pulse is available at the wire only during the high state of the clock (RZ scheme), returning to its original state at the end of each transmission (see Figure 2.4). Power consumption is reduced, since only a portion of the wire needs to be charged. It has been demonstrated that this type of signaling can save up to 50% of energy compared

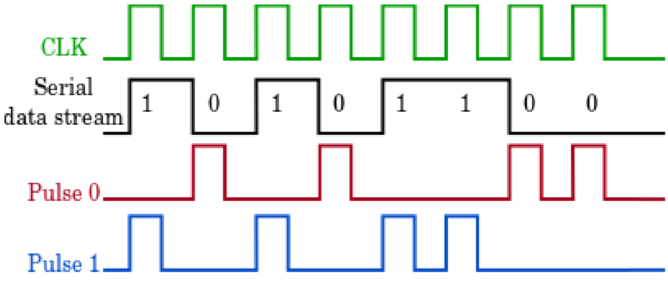


Figure 2.4: Pulse Dual-Rail Signaling

to level-based signaling (like LEDR) with repeater insertion [14]. Only two wires are necessary to achieve the integration of pulse dual-rail encoding with differential signaling. Another advantage of this protocol is that the RZ data format makes the effect of dispersion less severe by having sharp current pulses and receiver termination [15]. It is very important to mention the relevance of wire modeling as it has to be accounted in a lossy environment; making the wires wider and routing them in a high metal layer will contribute to maintain signal integrity and minimize attenuation [15].

Low-Swing Signaling: in order to improve performance when driving long wires, a small voltage swing V_{swing} is preferred rather than having to detect a full voltage swing. This is because power costs can be reduced by having the possibility to turn off the driver after the output has created enough voltage to be detected. Some drawbacks of this technique are that more complex driver and receiver circuits are needed and routing might become more expensive [8].

To take advantage of low-swing signaling, the signal should travel on differential pairs of wires and need to be equalized so as to prevent ISI (from previous transmitted data). Moreover, the differential signal should be amplified once it arrives at the receiver. One challenge that arises with the implementation of this type of signaling is the transmission of a self-timed clock from driver to receiver since having a full-swing signal takes more time [8]. Some of the advantages of differential signaling are the rejection of common-mode noise and the return of DC current solely [16].

Unipolar Current-Mode Signaling: in this type of signaling, the current present at the channel I is constantly sinked by the transmitter. As a termination load is necessary at the end of the receiver, the differential voltage at the RX's input is given by $\Delta V_{\text{in}} = R_{\text{T}} I$, where R_{T} is the impedance value of the termination load.

The advantages offered by this scheme are that the current present at the channel becomes immune to supply voltage (VDD) and GND bouncing since it is I_{tail} of the differential amplifier and it is fixed. Furthermore, the fully differential configuration of the amplifier at the RX offers a high common-mode rejection. Another factor that contributes to this characteristic is that current is being drawn by the transmitter from VDD and sinked into the GND rail all the time, minimizing switching noise associated with this block [17].

CMOS-CML circuits

Current-Mode circuits are commonly used in high-performance serial links since they can operate at low-voltages, minimizing the impact of low supply voltages (current swings are not limited by them). Moreover, these circuits can use AC or DC coupling [18] and offer a wider bandwidth than voltage mode circuits [17].

Based on the assets that current-mode circuits offer, different signaling schemes have been used. For instance, current-mode and current sensing signaling detect a signal with low-impedance at the RX termination. It is important to highlight that they are not the same. The former refers to detecting a voltage at the end of the channel so as to compare it with a reference voltage value and amplify it afterwards, whereas in the latter, the current is the value that is sensed and compared with a reference current so as to generate an output voltage [6]. In this project, current-mode signaling is used.

The main contributor to power dissipation in current-mode circuits is the static power. This is because there exists a direct path from VDD to GND through the termination load and current is drawn constantly. However, this can be minimized with different circuit techniques.

Transmission Lines

Transmission lines are the medium through which electromagnetic waves are going to travel from one source to a destination. They have different applications, being the transmission of signals the most common one. There exist different types of transmission lines, such as coaxial cables, optical fibers, microstrip lines and hollow waveguides [19]. As stated in the previous chapter, the main concern when replacing a parallel bus with a serial interconnection, is to maintain the same throughput as with the parallel interconnect, meaning that data transfer should occur at a much higher speed. In this project, to transfer a 32 bits data packet in a span of 3 ns, transmission at 10.66 GHz is required. As the frequency of operation increases, the electrical properties of on-chip interconnect dominate the overall performance of the circuit, that is why it is very important to model the line as a high-frequency transmission line. Bandwidth of the line will determine the signal attenuation and will set the limit for transmission distance.

Preserving *signal integrity* is one of the main concerns when designing a data transfer system. By comparing the shape of the waveform of original data being sent with the one retrieved at the receiver side, the quality of the signal can be determined. When traversing a wire, signals are affected by capacitive, resistive and inductive effects, which will determine the integrity of the transferred signal. Therefore, it is important to describe the interconnection line in terms of these parameters.

It is important to mention that in a lossy transmission line, distortion is present at the end of the line due to the presence of the R component. In this type of line, attenuation occurs at a much lower rate than the propagation speed [20]. The main electrical properties of a transmission line are *characteristic impedance* and *propagation velocity.*

Characteristic Impedance: it is the ratio of the voltage over current present at a given point of the transmission line. It is important to know the value of the characteristic impedance of the line in order to place a termination load with similar impedance so as to avoid signal reflections, which translate into signal losses.

Propagation Velocity: refers to how fast a signal traverse the interconnect.

Microstrip: a microstrip transmission line consists of a ground plane or substrate over which a metal wire (with width w and thickness t) is placed separated one from each other by a dielectric of height h and dielectric constant k [8] as it is shown in Figure 2.5.

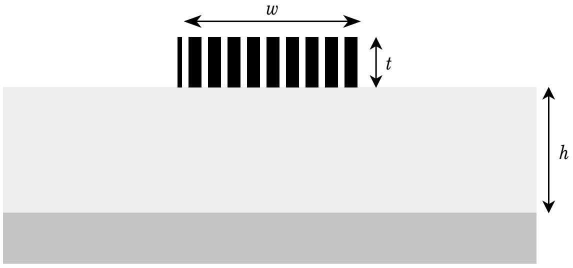


Figure 2.5: Microstrip

Random-Access Memories

This type of on-chip memories can be accessed in any moment to retrieve data. SRAM memories are a sub-type of this class and they are characterized by the possibility to perform both read and write operations, differing from ROM memories which can only be accessed to read data. Other feature that differentiate them, is that SRAM memories only hold data while power is supplied, that is, they are **volatile**, whereas ROM memories are **non-volatile** since they hold data indefinitely.

SRAM Memory

Memory unit: a group of memory banks forms a memory unit (Figure 2.6). Different stages in between them make the memory operations possible, such as timing controller, buffers, address decoders and pre-charge circuitry.

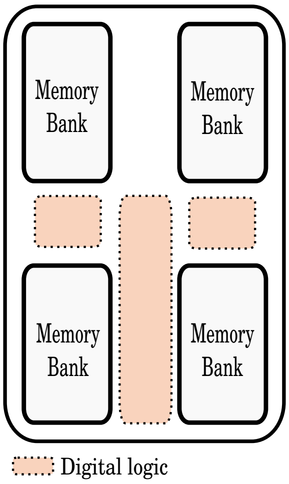


Figure 2.6: Memory unit

Memory bank: a cluster of memory macros that store data within a computer. The purpose of arranging memory macros into a bigger module (see Figure 2.7) is to increase the speed of access to all memory locations and improve the overall performance.

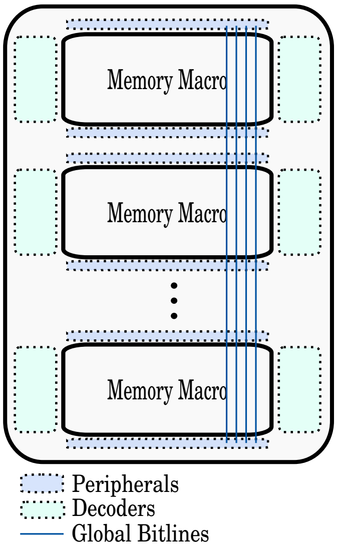


Figure 2.7: Memory bank

Memory macro: an array of memory bitcells in conjunction with peripheral circuits (Figure 2.8) that enables the performance of read and write operations. A memory macro consists of 2^n words of 2^m bits each. The memory unit cell storing a bit is called the *bitcell*. A word line WL is shared among all the cells in the same row and the bitline pairs (BL and $\overline{BL}$) are shared over each column. In order to keep the parasitics' value (capacitance and resistance) of these lines within an acceptable range as the memory size increases, the amount of rows and columns per macro is limited. Therefore, the memory array will contain 2^{n-k} words and 2^{m+k} columns. Data decoders allow the access to the bitcells by providing the corresponding row and column where each of these are located.

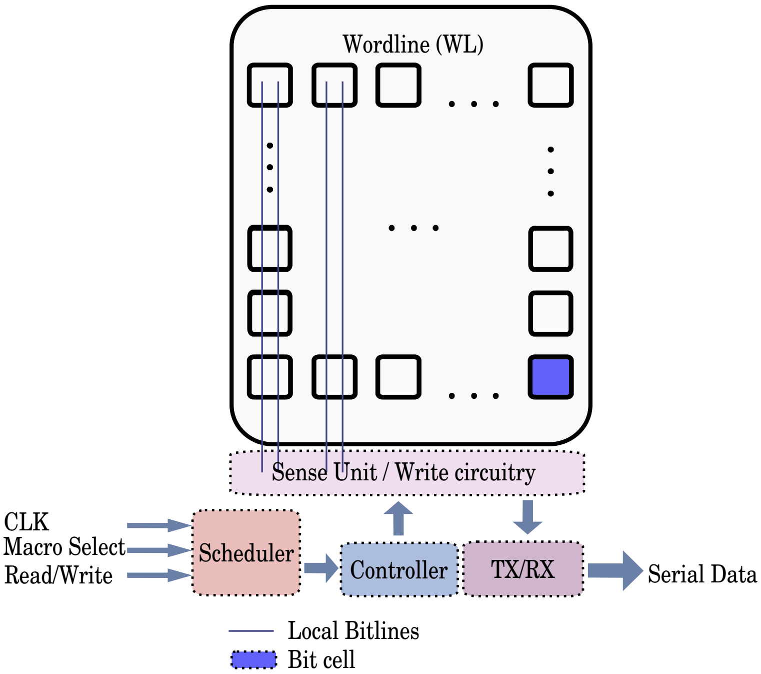


Figure 2.8: Memory macro

6T bitcell: The prevailing memory cell architecture used among ICs is the six-transistors (6T) bitcell. In this configuration, SRAM memories rely on a pair of cross-coupled (positive feedback) inverters to latch data, making high speed operations possible at the cost of bigger area per bit [1]. A standard 6T bitcell is shown in Figure 2.9. It consists of two cross-coupled inverters with their outputs connected to a pass transistor each, which are driven by the WL. Their source/drain terminals are connected to the respective bitlines, where data is going to be transferred either to be stored or to be acquired. One of these bitlines contains the actual data and the other one contains its complement. A generic SRAM architecture for each of both cases is presented in the following subsections.

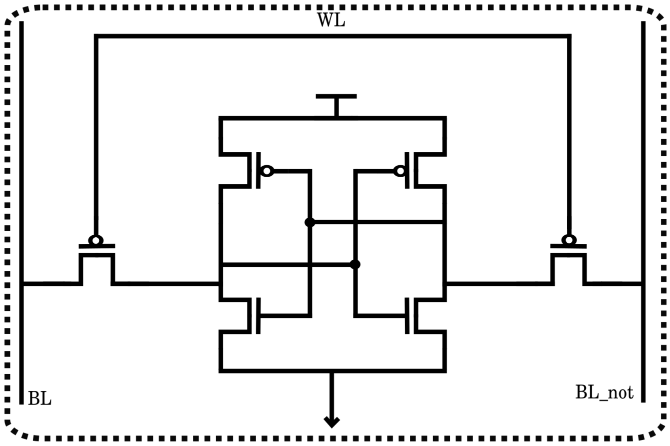


Figure 2.9: SRAM 7T bitcell

Read Architecture

The goal of the read operation is to maintain data integrity (stored data to not be destroyed) while this action is performed. In order to read data stored in a bitcell, the access transistors should be turned on by selecting the correspondent WL according to the row where the bitcell is located. Once the pass transistors are on, the bitlines (BL and $\overline{BL}$) might or might not be discharged. The initial voltage difference created between these lines is detected by a sense amplifier which provides an output depending on the stored value. Different control signals make possible to achieve timing requirements and proper operation such as macro selection (including row and column) circuitry, read enable, clock and bitlines selectors that will multiplex them.

In the serial link proposed in this thesis, the serializer circuit is implemented by re-using resources already present in the memory architecture. For instance, flip-flops are built by connecting latches to the sense amplifiers' outputs in order to reduce area cost.

Write Architecture

For a memory to be functional, it should be possible to overwrite data already present at the bitcell. The principle of write operation is to set a low level at one of the columns which is accomplished by connecting one of the bitlines (BL or $\overline{BL}$) to GND.

Chapter 3

System model

The interconnect system is shown in Figure 3.1. It consists of a serializer, transmitter (TX), receiver (RX), and a deserializer. In order to make bidirectional communication possible, switches have been added to the transmisison lines in order to be able to select the active modules, according to the direction in which communication should occur.

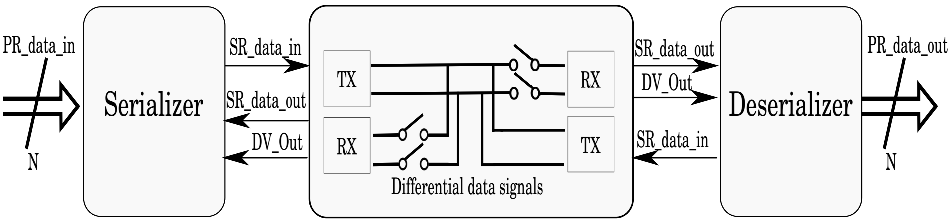


Figure 3.1: Bidirectional Interconnect System

Communication Protocol

During the read operation of the memory, bitlines might be discharged depending on the contents of the bitcell. After the sense unit detects this voltage difference, and read values are ready to be transferred, the parallel loading of data *PR_data_in* is performed. Once data is latched and ready to be serialized, an input clock drives 32 switches in order to generate a serial data stream *SR_data_in*. The transmitter TX is in charge of encoding the incoming serial data stream and conveying enough voltage to the transmission lines (*drive voltage*). In order to do so, a pulse dual-rail encoder generates pulses according to the bit meant to be represented and differential pulse current-mode signaling is used to transmit data along the lines.

At the receiver RX, data *SR_data_out* is retrieved directly from the arriving signals without the need of extra decoding circuitry. Similarly, a data validity indicator signal *DV_Out* is generated and used as a clock to shift data at the deserializer. Finally, the deserializer conveys data into a parallel bus *PR_data_out*.

Chapter 4

Circuit Design

Transmission Lines: distributed RLC model

The interconnect distance is set to 100 µm, which classifies as short and local range transmission [21]. This value was set as the reference for this project, since this is the distance that exists in between memory banks of the targeted memory.

For this project, it was not possible to get L parasitic values since data regarding technology material parameters was not available (substrate and oxide thickness). However, an initial distributed RLC approximation model is used in order to conduct the design of remaining system blocks. Distributed model (see Figure 4.1) was chosen over lumped model since the operating frequency is considered high and delay can be represented more accurately [21].

The highest L value on the range of 0.15-1.5 pH/ µm which are the most typical values for on-chip inductance [8] is taken for each section.

The RC values were approximated with parasitic values extracted from a pair of M6 wires with $\ell = 100\mu m$, $w = 4 \times w_{min}$, spaced to GND shields (with w_{min}) by *space*$_{min}$ *+ 40 %.* Width and spacing were maximized within the available area.

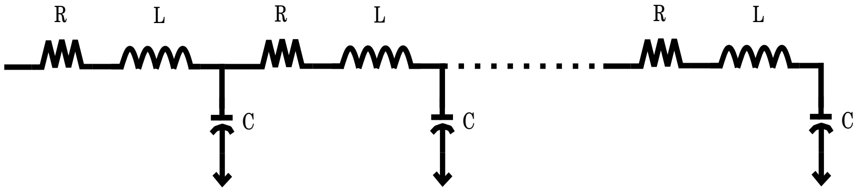


Figure 4.1: RLC model

Serializer

It is the first stage in the data path at the transmitter. Here, low-speed parallel data is combined into a high-speed serial data stream [11]. The possibility of reusing the sense amplifiers that are part of the Sense Units located in between memory macros was studied for the implementation of this block. A multi-phase multiplexer (see Figure 4.2) is employed since its size would increase linearly with the number of parallel bits and it has lower power consumption [11] .

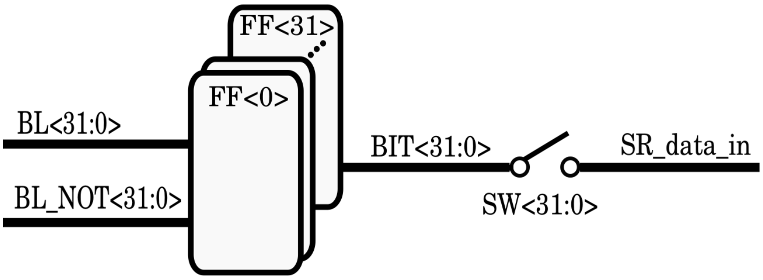


Figure 4.2: Serializer

Once the read operation has occurred, data bits will be loaded into the multiplexer which consists of 32 flip-flops connected in parallel. These flip-flops were built by connecting the sense amplifiers' outputs to a latch as shown in Figure 4.3. The next step to obtain a serial stream of data is to drive the switches, one on each CLK cycle so as to obtain the output signal *SR_data_in.*

This architecture requires that the switches are driven with very sharp non-overlapping clock pulses in order to avoid glitches at the output.

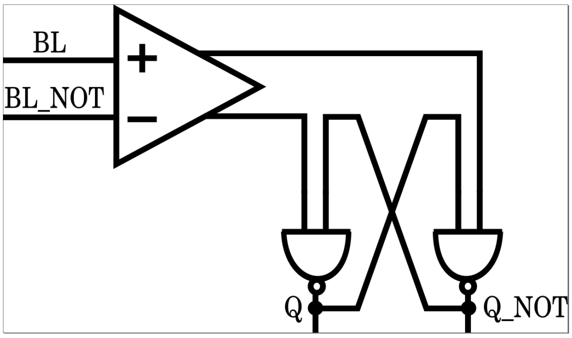


Figure 4.3: Flip-Flop with Sense Amplifier and Latch

Transmitter

The functions of this block are to encode the incoming serialized data stream and drive the transmission lines, i.e. to convey a sufficiently large current or voltage to the communication channels [17]. In order to do so, a dual-rail encoder and a differential driver were implemented. The top-level diagram of this block is shown in Figure 4.4.

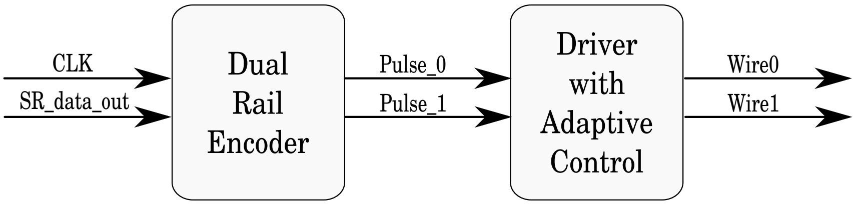


Figure 4.4: Transmitter

Dual Rail Encoder

This encoder consists of two NAND gates and two inverters (see Figure 4.5), that correspond to bit "0" and "1" respectively. It encodes data into a Pulse (P) and, No-Pulse (NP) pair according to the present data value, e.g., when the input *SR_data_in* is a "1", a pulse is present at the output of NAND1 and no pulse is present at the output of NAND0 [15]. Moreover, no pulse is present on neither of the wires when the clock is in low state (RZ format), serving as a spacer in between transmissions.

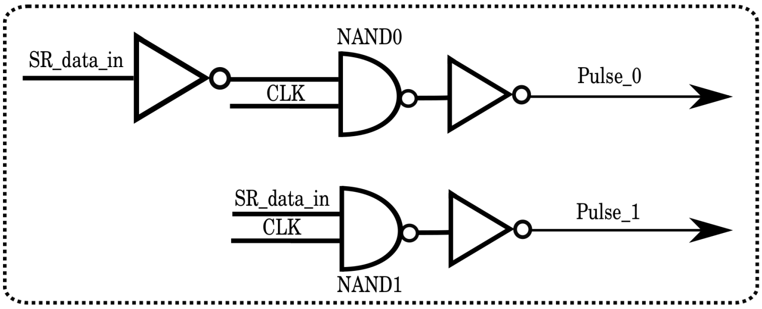
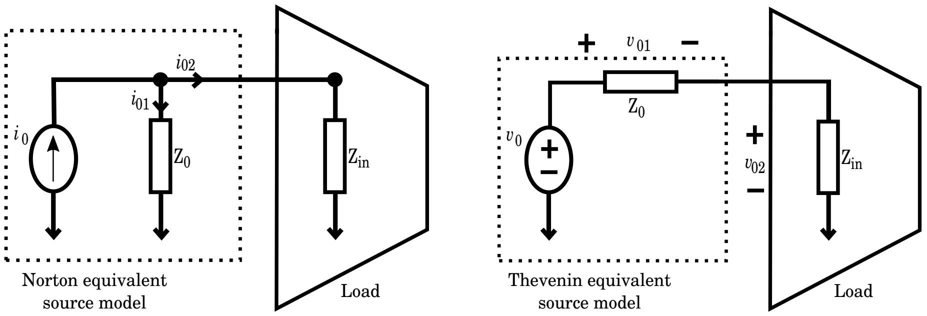


Figure 4.5: Dual Rail Encoder

Differential Driver

There are two different modes of producing an output drive voltage: either by a current-mode driver or by a voltage-mode driver. The former uses a Norton-equivalent parallel termination (high output impedance) and the latter uses a

Thevenin-equivalent series termination (low output impedance). A model diagram of both types of drivers is shown in Figure 4.6.



(a) Current-mode driver

(b) Voltage-mode driver

Figure 4.6: Loading effect on current and voltage mode circuits

Bandwidth of voltage-mode circuits is limited due to higher noise levels and the large voltage swing that is necessary for operation. Moreover, the rising and falling times for each node's voltage are large, giving as a result poor performance at higher data rates (Gb/s) [17]. Current-mode drivers allow high-speed operation (fast transient response due to low nodal impedance and low voltage swing of the critical nodes of the circuits proper of current-mode circuits [17]), therefore the architecture selected is an open-drain driver as shown in Figure 4.7). In this configuration, transistors M0 and M1 are source-coupled and act as complementary switches which steer the total tail current I_{tail}.

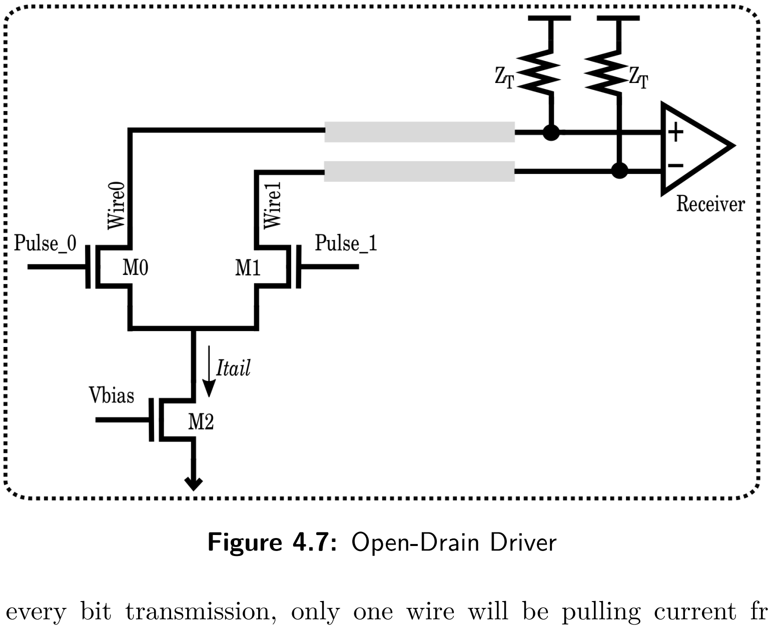


Figure 4.7: Open-Drain Driver

For every bit transmission, only one wire will be pulling current from the receiver to create the symbols that are going to be sent along the channel. When a "1" is transmitted, M1 is conducting, thus drawing current from Wire1. As no pulse is generated in Wire0, M0 is off, and no current is steered from that wire. The signaling method implemented in this project is a RZ scheme, therefore no current will be drawn from any of the wires while the CLK is in low state or in between data packet's transmission. It is important to mention that the open-drain driver provides an unipolar signaling i.e. current flows in only one direction.

As this configuration is resembling a Norton-equivalent , it requires a parallel termination load at the far end of the channel (Z_T).

The advantages inherent to this type of driver are:

- **Reduced Noise Levels:** these are minimized due to the constant current being drawn from the power supply which in turn reduces the AC component of power supply noise[15].
- **Swing of the signal:** the current swings are not limited when there is a low power supply voltage [17].
- **Well defined output current:** it will be independent of power variations and ground bouncing [17] since it depends on the constant current source (I_{tail}), the termination load and the impedance of the line.
- **Less power consumption:** it consumes less power compared with inverter drivers at multi-Gb/s data rate [22].

In table 4.1, the bias operating point of transistor M2 is presented.

I_{tail}	150 µA
V_{gs}-V_{th}	70.32 mV
V_{bias}	418.9 mV

Table 4.1: Transistor M2 DC-operating point

Termination Load

As mentioned in section 4.3.2, a current-mode circuit has a low input impedance and high output impedance, consequently, a termination load in a parallel scheme is required to minimize the loading effect caused by the finite output impedance of these circuits [17]. In this case, a diode-connected transistor (active termination) is connected at the end of each line to have low-impedance ($Z_{in,RX} \ll Z_{o,driver}$). By having a low-impedance termination load, the signaling operation will be possible at a lower noise margin since the current that is present at the wires is stable and well determined. Furthermore, it will shift the dominant pole of the system and will reduce the time constant, thereby decreasing the delay and increasing the bandwidth of the channel [6]. A double termination (Figure 4.8) was implemented in this system. This is due to the fact that it is preferred over single one for high performance serial links since it stands for best signal quality (reduces the error for mis-termination [23]). The differential peak-peak RX voltage swing is

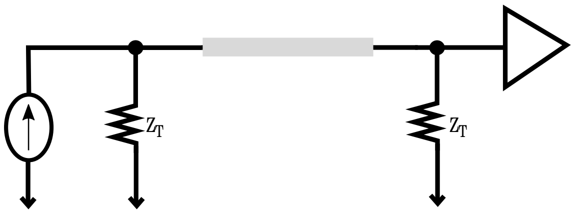


Figure 4.8: Double Termination

± IR/2 with double termination [18]. One drawback of a diode-connected load is that they consume voltage headroom, thus creating a trade-off between the output voltage swings, the voltage gain and the input common-mode range [6]. In the configuration used in this project (see Figure 4.9) transistor M_t is added to regulate the transconductance of M_{load} [15]. The idea is to lower it by reducing the current and not the aspect ratio of M_{load} [24]. The signaling scheme implemented in this project does not require decoding circuitry at the receiver, allowing DC coupling. The RX common-mode level is determined by the transmitter signal level: RXcommon-mode = IR/2 [18].

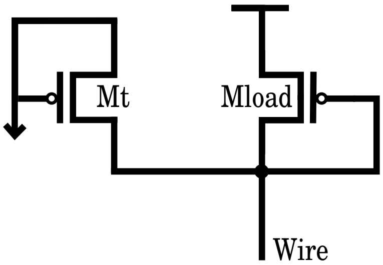


Figure 4.9: Diode-Connected Load with transconductance control

Adaptive Driver Control

The driver's output might be distorted before it is conveyed to the communication channels. This signal degradation is caused by high frequency impedance changes along the wire. When the signal switches slowly (low-frequency) the characteristic impedance of the channel increases whereas when fast switching takes up, the channel impedance decreases due to *skin effect* making a stronger drive necessary.

To alleviate this frequency-dependent problem and convey an equalized signal to the channel, a circuit that generates an initial delay and provides control to a variable load is connected to the driver's outputs. The configuration in Figure 4.10 consists of a chain of inverters, a NAND gate and an NMOS pull-down transistor for each wire. It can drive fast data transients and when there is a long run length present on the channel or after a long idle period, it mitigates the distortion that would be present at the signal in the next transition by reducing the impedance to the driver and adding extra load. Once this is accomplished, the NAND gate

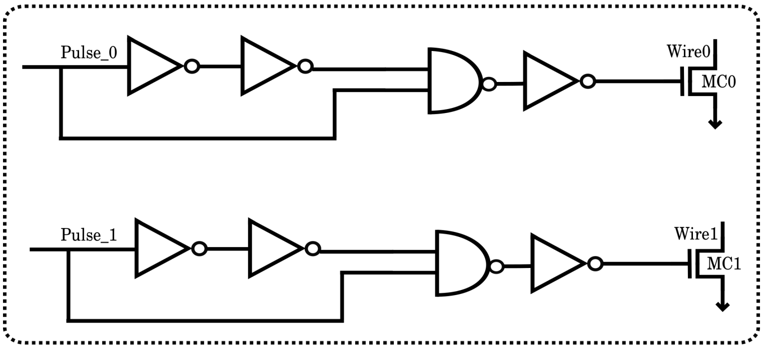


Figure 4.10: Adaptive Control Circuit

output will be "1" (thus "0" after the inverter), turning off the NMOS and resuming normal operation [4]. Transistors MC0 and MC1 are matched since they will drive differential signals. Sizing is the result of iterations to achieve maximum output voltage swing at the RX with given current value and line length. Figure 4.11 shows the difference in between signals at the wire with and without this block.

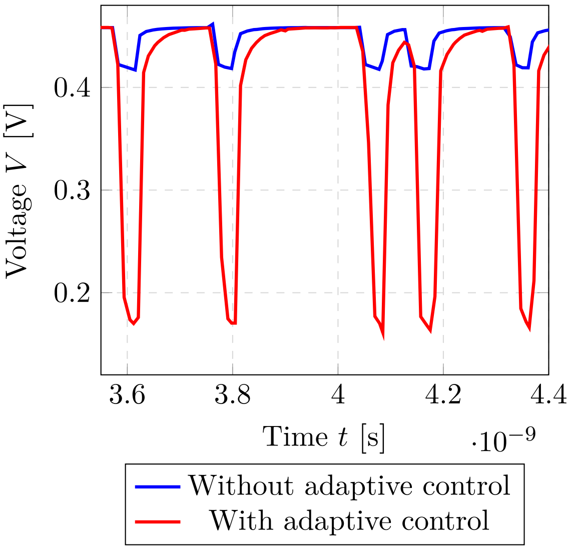


Figure 4.11: High-frequency compensation

Bidirectional Switches

One of the requirements of this project is to be able to send and receive data on both directions. In order to achieve this, both TX and RX are present on each end. Furthermore, a pair of switches are connected to the wires, one at each of the RX's inputs in order to select the active module, which in turn defines the direction of transmission as shown in table 4.2. These switches are implemented with a *transmission gate*, which is a voltage-controlled bilateral switch controlled by an enable signal. The gates are connected to the enable signal in an opposite way, meaning that only one of the switches is on at a given time (Figure 4.12).

Enable	Direction
1	Left to Right
0	Right to Left

Table 4.2: Direction of transmission set by enable

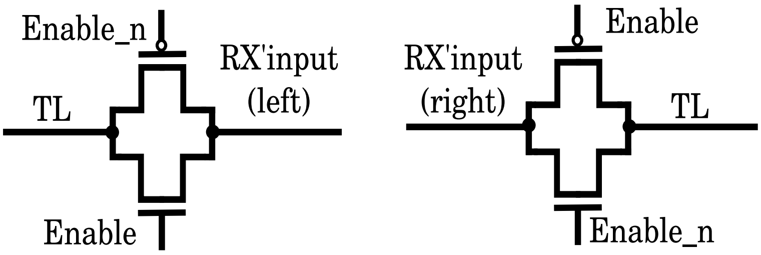


Figure 4.12: Transmission gates at the RXs' inputs

As it can bee seen in Figure 4.13, it exists signal distortion and the voltage swing is reduced by approximately 20 mV after passing through the switches. The reason of this is the switch on-resistance R_{on} and the high-frequency components.

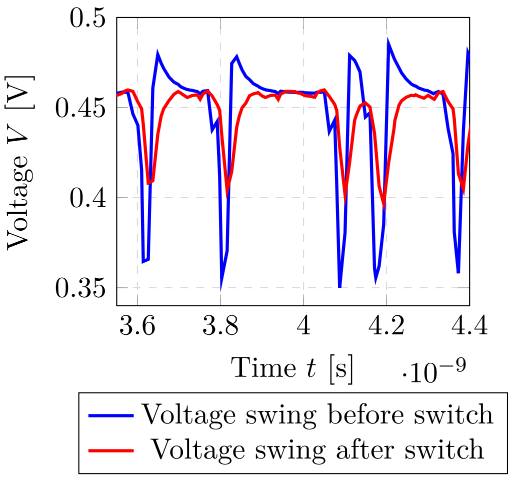


Figure 4.13: Voltage swing reduction after transmission gates

Receiver

The main function of the serial link receiver is to detect the bit that was sent and generate a data validity signal *DV_Out* which will be then used as a CLK to the shift register at the deserializer. The transmitter employs current-mode signaling and for this reason, the receiver needs to sense the voltage difference between the two wires so as to amplify it and provide a full-swing output signal that can drive digital logic.

Data Recovery

To perform the data recovery operation, a self-biased amplifier is used along with a chain of inverters (see Figure 4.14).

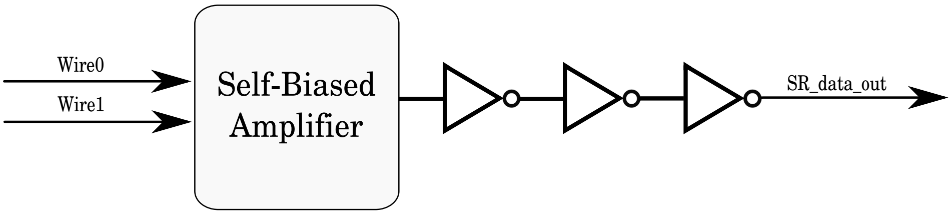


Figure 4.14: Data Recovery

As shown in Figure 4.15, this is an architecture that consists of two differential amplifiers, which are serving as the load to one another, making it fully complementary.

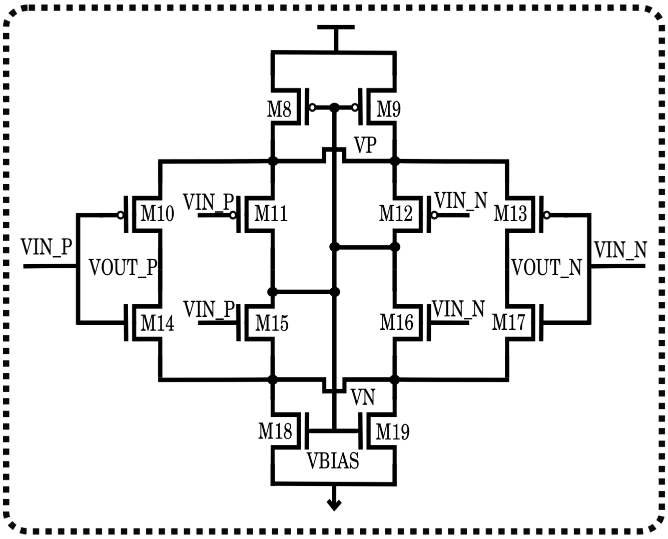


Figure 4.15: Self-biased amplifier

Self-biasing through negative feedback ensures that the bias voltages are going to be very stable since any shift in V_{bias} from its nominal value will be corrected. It also presents a high common-mode noise rejection and can operate at high-speed. The reason of this is that its switching currents are higher than its bias current. Added to these properties, the aforementioned configuration has a higher differential gain than other amplifiers and large common-mode input range because its bias operating point tunes itself according to the input signal levels.

Data Validity Decoder

The circuit used to generate a data validity indicator signal is presented in Figure 4.16. A single-stage of the circuit uses a differential amplifier with an active current mirror (Figure 4.17) as a load to sense the voltage at the end of line and compare it with a fixed reference in order to deliver a single-ended output (output voltage referenced to GND).

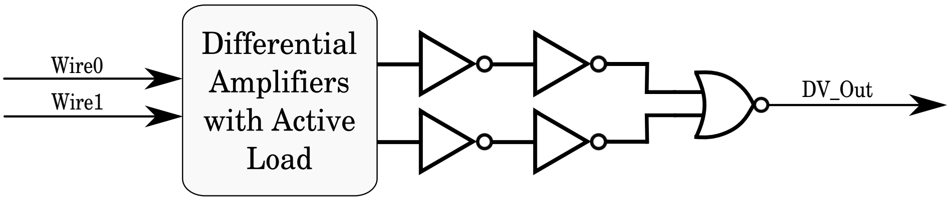


Figure 4.16: Data Validity Decoder

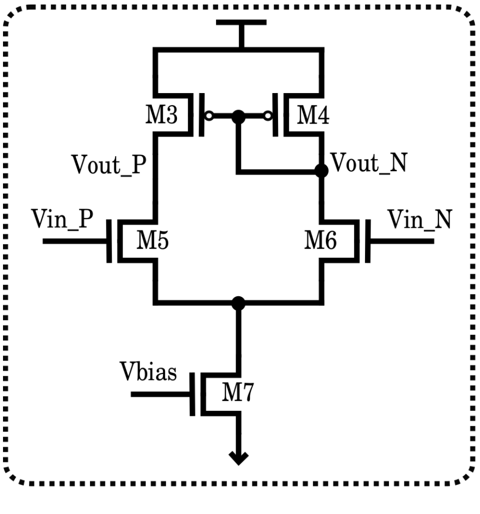


Figure 4.17: Differential Amplifier with active load

In this case, the voltage reference is set to 433 mV for the first stage which is close to the DC value of transmitted signals. This voltage reference is set with an ideal voltage source since the generation of such reference is not within the scope of this project. As there will be current in only one of the wires during each bit transfer and no current on neither of the wires in between consecutive bit transmissions, a signal that denotes the validity of the received bit is generated.

Three stages of this amplifier (as shown in Figure 4.18) for each input signal were necessary to remove the offset of the signal and generate a large swing output that is fully regenerated after two inverters. By feeding these two signals to a NOR, a signal composed by a train of pulses with same period as the CLK at which the transmitter sent the information is generated.

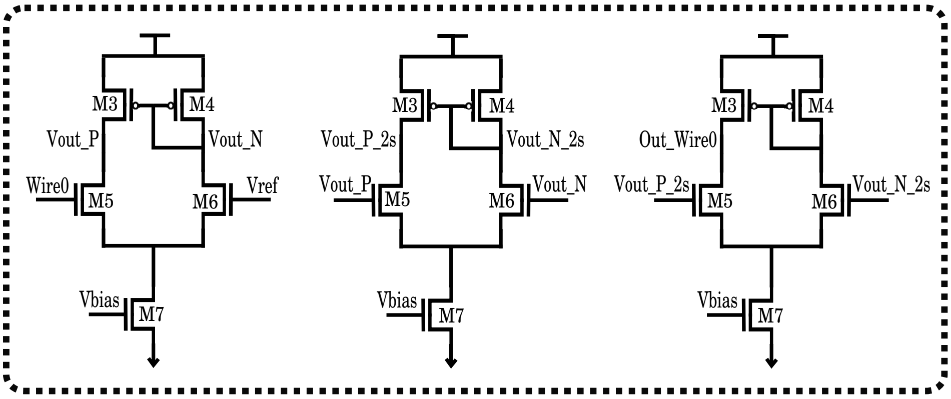


Figure 4.18: Amplifier stages to recover Clock

Bias operating points of each stage are reported in following tables:

Stage	$\mathbf{I_{tail}}$ **(µA)**	$\mathbf{V_{gs}}$**-**$\mathbf{V_{th}}$**(mV)**	$\mathbf{V_{bias}}$
1st	381.8	38.69	418.9 mV
2nd	463	39.92	418.9 mV
3rd	514.1	40.83	418.9 mV

Table 4.3: M7 DC-operating point of three stages

Transistor	$\mathbf{I_d}$ **(µA)**	$\mathbf{V_{gs}}$**-**$\mathbf{V_{th}}$**(mV)**
M5	207.2	42.5
M6	174.6	23.34
M3	-207.2	3.66
M4	-174.6	11.1

Table 4.4: 1st stage DC-operating point

Transistor	I_d (μA)	V_{gs}-V_{th}(mV)
M5	300.1	11.19
M6	162.9	79.14
M3	-163	-8.55
M4	-300.1	-32.15

Table 4.5: 2nd-stage DC-operating point

Transistor	I_d (μA)	V_{gs}-V_{th}(mV)
M5	332.6	280.3
M6	181.6	27.99
M3	-332.5	-22.19
M4	-181.6	8.26

Table 4.6: 3rd-stage DC-operating point

Deserializer

Once data has been recovered by the receiver, it should be parallelized in order to make it available to the accepting module, which in this case is another memory bank. In order to do so, 32 switches are multiplexed with non-overlapping pulses (*SW<31:0>*). Each bit is going to be latched afterwards by a D-latch using tri-state buffers (see Figure 4.19). The signal *SW_D* and its complement *SW_D_NOT* are going to enable/disable the tri-state buffers so as to operate one of them in active mode and the other one in high-impedance mode. When the clock signal *SW_D* is in high state, the corresponding bit will be transferred to the output *BIT<n>*. If *SW_D* is in low state, data bit will be latched by the two back-to-back inverters [25].

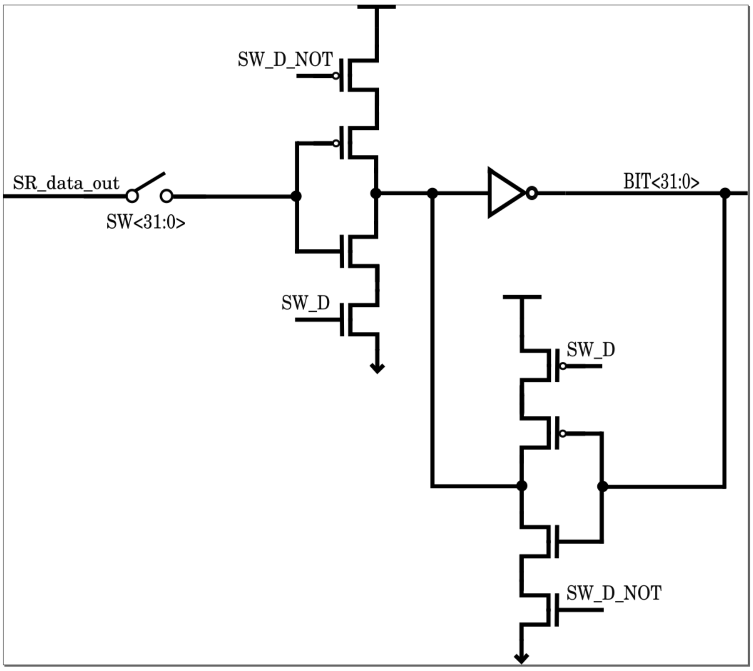


Figure 4.19: Deserializer: D-latch with tri-state inverters

Chapter 5

Layout

After the completion of transistor level design, it is necessary to draw the layout and extract the parasitics in order to analyze the overall performance of the circuit.

Three parameters were key for the realization of the serial link layout: area, matching and symmetry. In Figure 5.1 the layout view of one end of the interconnect is shown. A view of the complete system is in Figure 5.2.

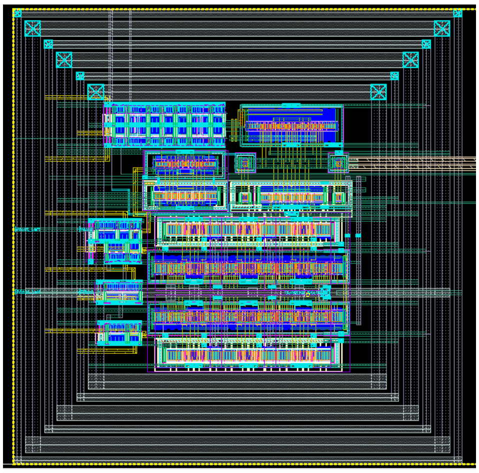


Figure 5.1: TX-RX, one end of the interconnect.

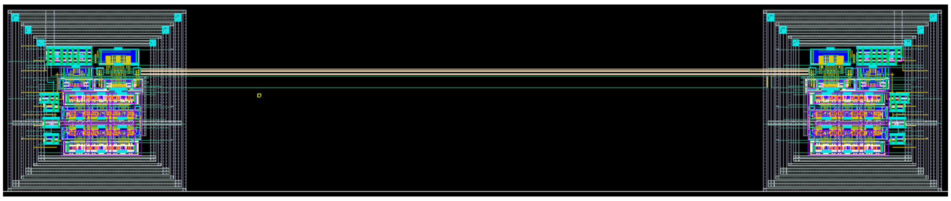


Figure 5.2: Layout of bidirectional serial link

Floorplan

On each of the interconnect ends there is a TX and a RX. As it is shown in Figure 5.3, the TX is placed at the top and the RX at the bottom. It is important to mention that the layout of serializer and the deserializer was not done, since these blocks are only going to be simulated. The total area occupied including the power lines routes is 902.21 µm^2 (active area is 336.7 µm^2).

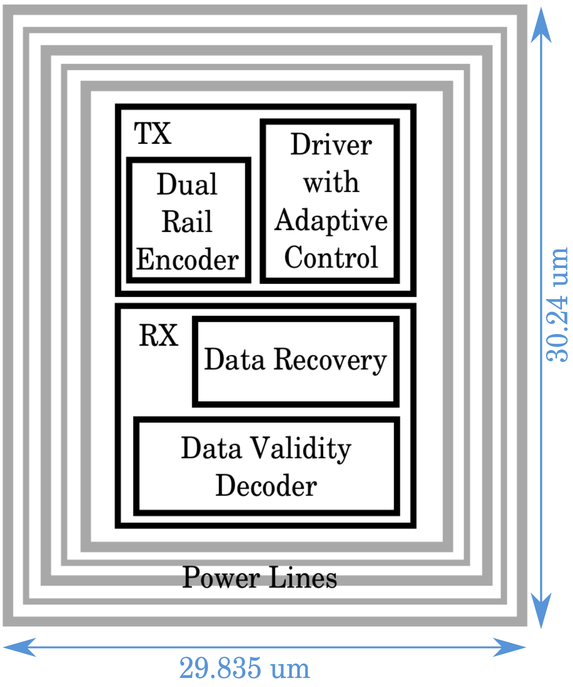


Figure 5.3: Floorplan of one end of the interconnect

Symmetry and Matching

Differential circuits are affected by asymmetries, thus mismatches should be minimized. By doing this, the circuit will present a higher common-mode noise rejection and increase the minimum input signal level [24]. It is important, to place the transistors in a symmetric pattern and maintain an uniform environment around them so that all of them are affected in the same way.

Regarding the layout of all the amplifiers contained in this system, the following guidelines were followed to maximize matching in between critical transistors:

- Transistors were arranged in a common-centroid configuration (Figure 5.4), i.e. they all are on the same OD, have the same orientation, are aligned to each other and mirrored in respect with Y axis.

- Dummy transistors were added at the edges of the transistors' arrays along 1.5 µm.
- Long gates were avoided by folding transistors into fingers, which will also improve the proportion in between PMOS and NMOS arrays, thus facilitating routing. Transistors' finger width was left as three times the minimum allowed in the technology used (except for the amplifier of the data validity decoder where finger width is 6 times the minimum).
- Metal routings and transistors' surroundings were kept the same on both sides of the Y axis.
- Substrate connections surround arrays of matched transistors in order to avoid latch-up.

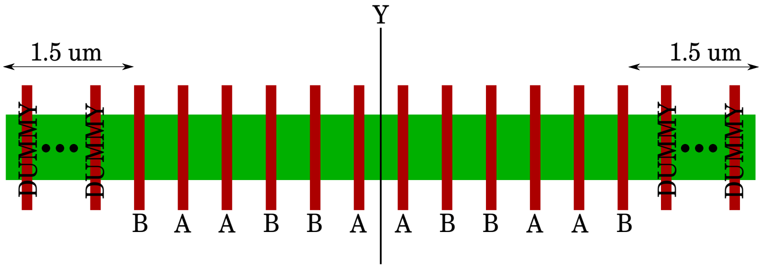


Figure 5.4: Transistors A and B in a common-centroid configuration

Channels' wiring

M6 wires that are 4×w_{min} wide had been used for the transmission lines. Minimum width GND shields were interleaved with these wires (see Figure 5.5) in order to reduce external interference and noise injection. If a parallel bus would have been used, in the case of a single-port memory with same throughput, the total wiring area would have been approximately 6 times larger (accounting for w_{min}+10 % and $spacing_{min}$+10 %).

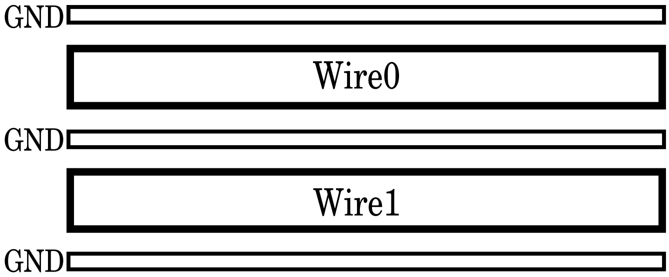


Figure 5.5: Channels' wiring in the serial link

Chapter 6

System Integration and Simulation Results

Test-Bench

The initial stage of simulation consisted on analyzing the critical path functionality, that is, driver, receiver and data validity decoder so as to do a proper architecture selection for each circuit and the subsequent sizing of transistors. In order to be able to do this, a random serial data stream was generated with a Verilog-A block which was later replaced with the Serializer stage. The supply voltage VDD was set to 0.9 V since this is the nominal operating voltage for the transistors used in this project (low-voltage transistors) and a clock signal CLK with t = 93.75 ps is used.

Generation of Input Serial Data Stream

A block diagram of the system used to generate the serial stream is shown in Figure 6.1. The parallel data bus *BL<31:0>* is fed to this block along with a control parallel bus signal *SW<31:0>*. The former was simulated with 32 supply voltages with either a 0 or a 0.9 V value and the latter was generated with a ring counter.

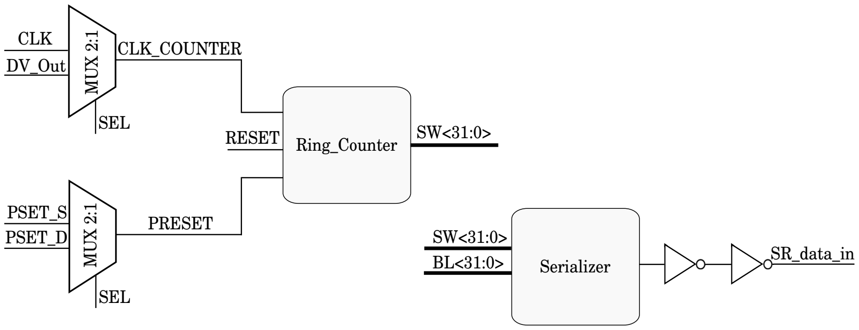


Figure 6.1: Generation of serial input data stream

Ring Counter: this block consists of 32 flip-flops connected in series, with the last one's output fed back to the input of the first one. CLK and *PRESET* signals are going to be defined depending on the block that they are going to control (see table 6.1) by a multiplexer:

	SERIALIZER	DESERIALIZER
CLK_COUNTER	CLK	DV_Out
PRESET	PSET_S	PSET_D

Table 6.1: CLK and PRESET signals for Ring-Counter

The ring counter shifts a logical ONE (SET of first flip-flop in Figure 6.2) each clock cycle in order to generate signal *SW<31:0>*, which consists of 32 non-overlapping pulses.

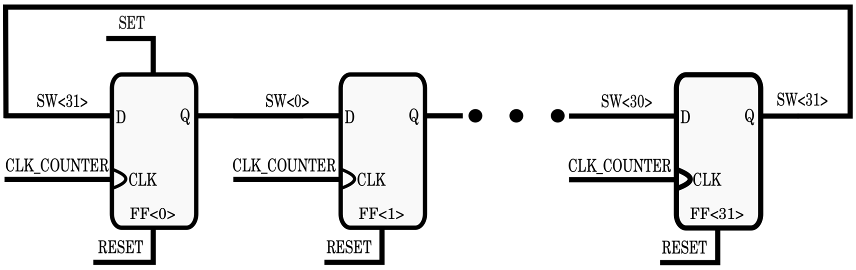


Figure 6.2: Ring Counter

Generation of Output Parallel Data

Following the same method as with the serializer, the deserializer (Figure 6.3) has a control bus signal *SW_D<31:0>* and the output of the Data Recovery block *SR_data_out* as inputs. As each switch inside the deserializer turns on, data bits are going to pass through, thereby becoming available at the output of this block.

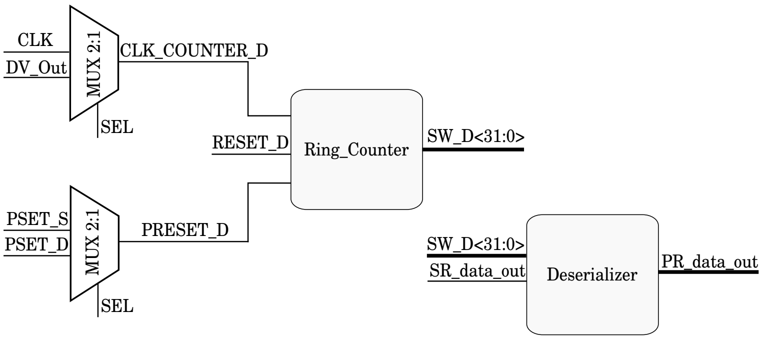


Figure 6.3: Generation of parallel output data

Once serial data was available, design of the TX and RX was implemented. In Figure 6.4 the serial link test-bench configuration is shown:

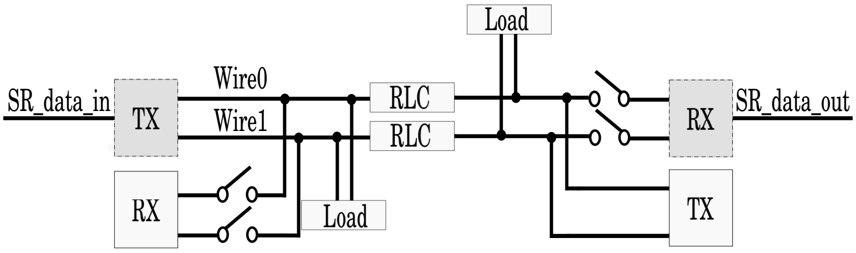


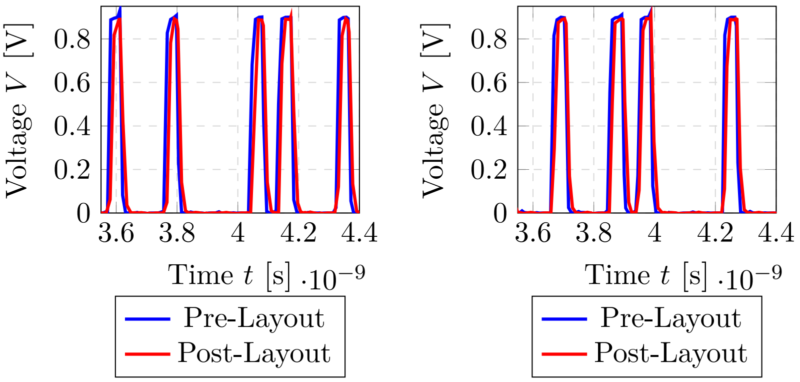
Figure 6.4: Serial link

Pre-Layout vs Post-Layout Simulation Results

All the following Figures and results were obtained with TT corner otherwise mentioned. Moreover, the post-layout extraction was typical RC-coupled.

Transmitter

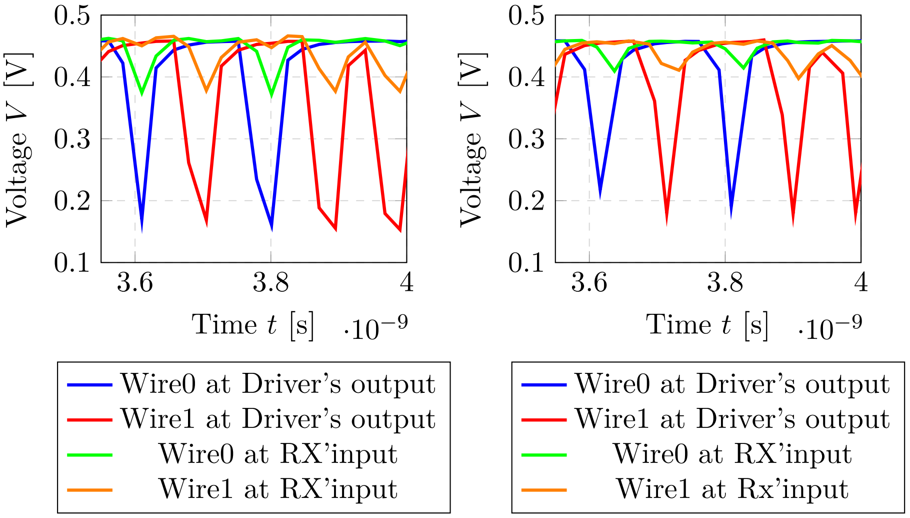
Dual-Rail Encoder: the generation of pulses is achieved and layout parasitics have no significant effect on the outputs. In Figure 6.5 a fraction of the 32-bit data stream is shown : "010110010".



(a) Pulse Bit 0 **(b)** Pulse Bit 1

Figure 6.5: Encoded data Pulse signals

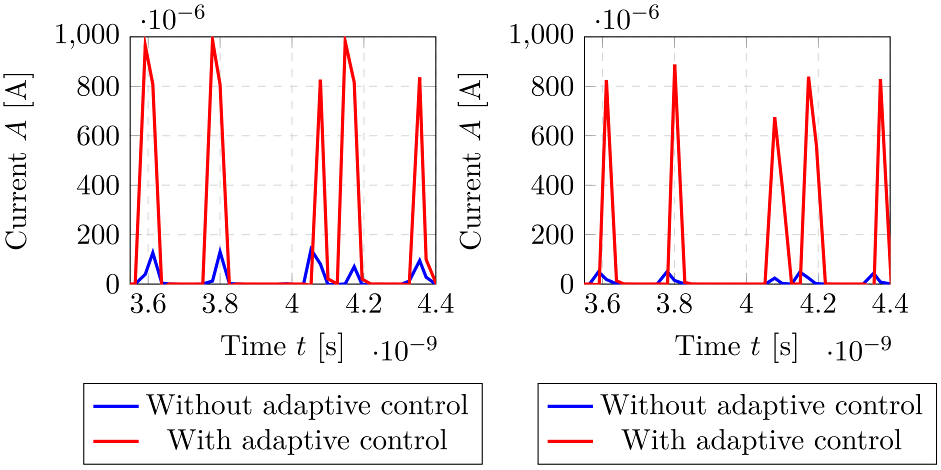
Differential Driver: as it can be seen in Figure 6.6, the voltage swing at the driver's outputs is bigger than at the RX inputs. There is a reduction of approximately 200 mV due to losses along the transmission channel. Moreover, the voltage swing at the RX inputs is 50 mV after layout whereas in pre-layout simulations it was 60 mV.



(a) Pre-layout **(b)** Post-layout

Figure 6.6: Voltage signals at the channels

In Figure 6.7, it can be observed that current sinked by the driver during transmission goes up to 820 µA in post-layout, proving that large current pulses are generated by the adaptive control circuit.

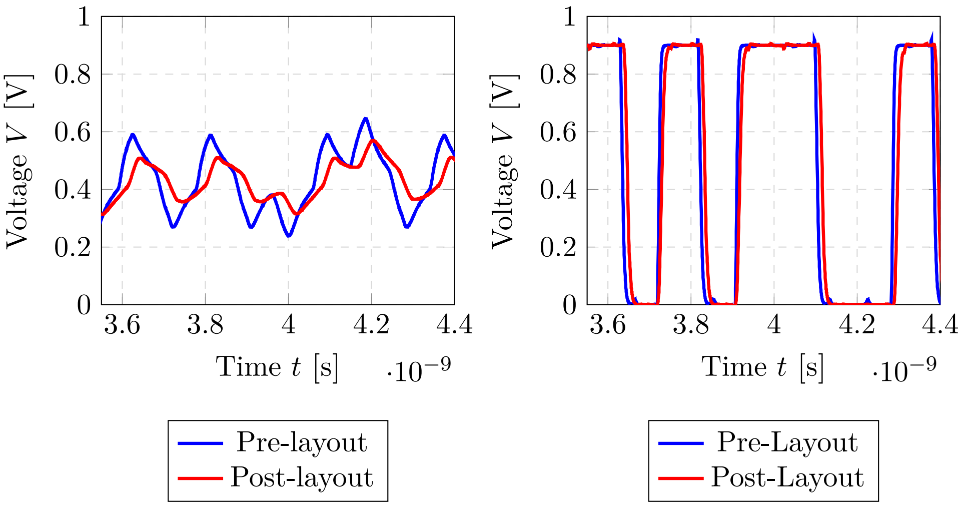


(a) Pre-layout **(b)** Post-layout

Figure 6.7: Current signals at driver

Receiver

Self-Biased Amplifier: the output of this amplifier is shown in Figure 6.8. It can be seen that the common-mode voltage is 450 mV, which corresponds to VDD/2, this allows full regeneration of the signal after connecting digital logic (inverters) at the output.



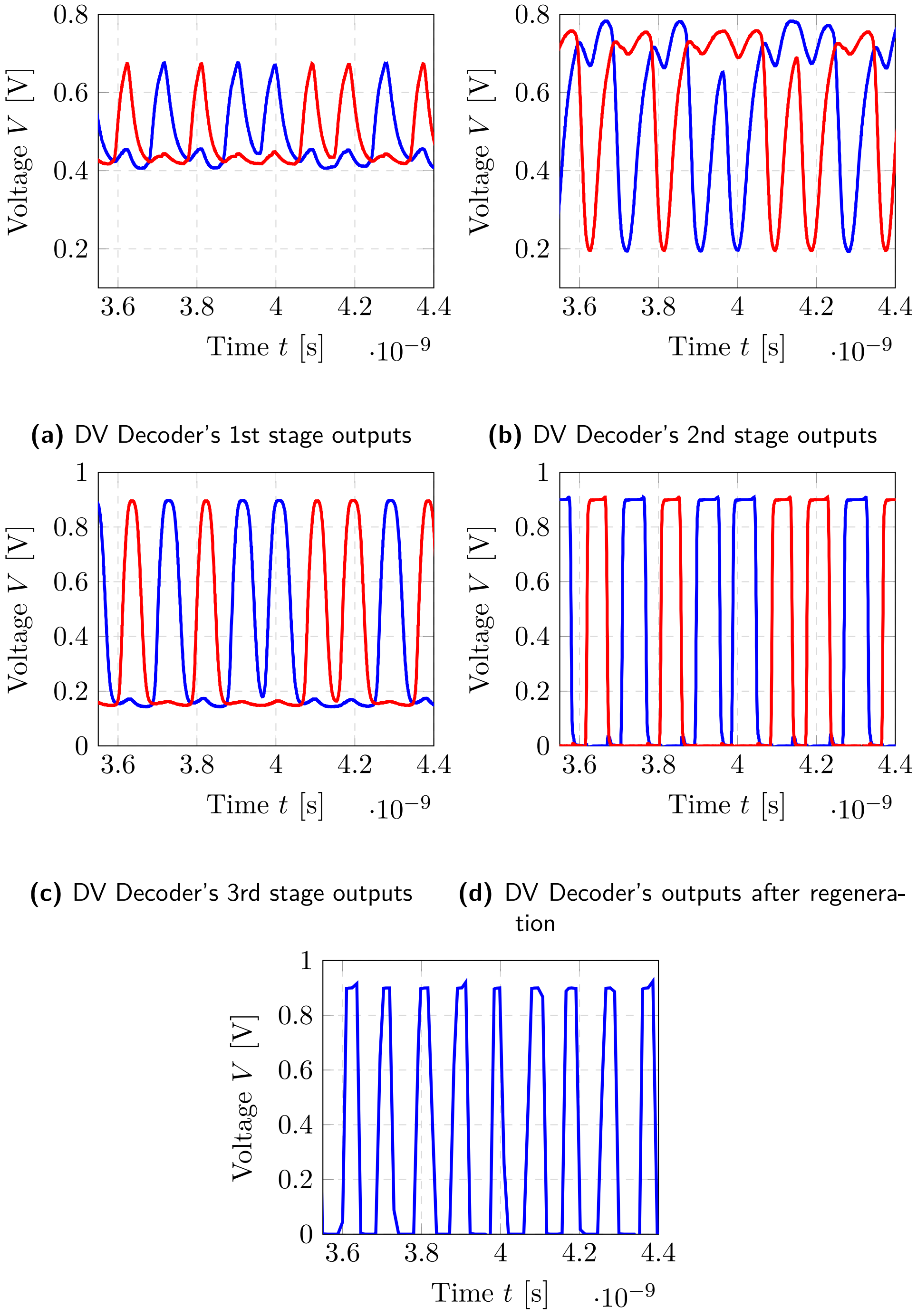
(a) Self-biased amplifier's output

(b) Self-biased amplifier's output after regeneration

Figure 6.8: Retrieved Data

It is important to mention that the output is affected by layout parasitics and degradation of the input signal, that is why the output has a higher delay and is attenuated, however it still allows data retrieval.

Data Validity Decoder: three stages of a differential amplifier with active load were necessary per wire. The first stage reduces the offset level, the second stage increases the voltage swing of the output and the third one inverts it and provides a signal that can be regenerated into a square wave so as to drive digital logic. In Figures 6.9 and 6.10 outputs of each stage and final reconstructed signal are shown.



(a) DV Decoder's 1st stage outputs

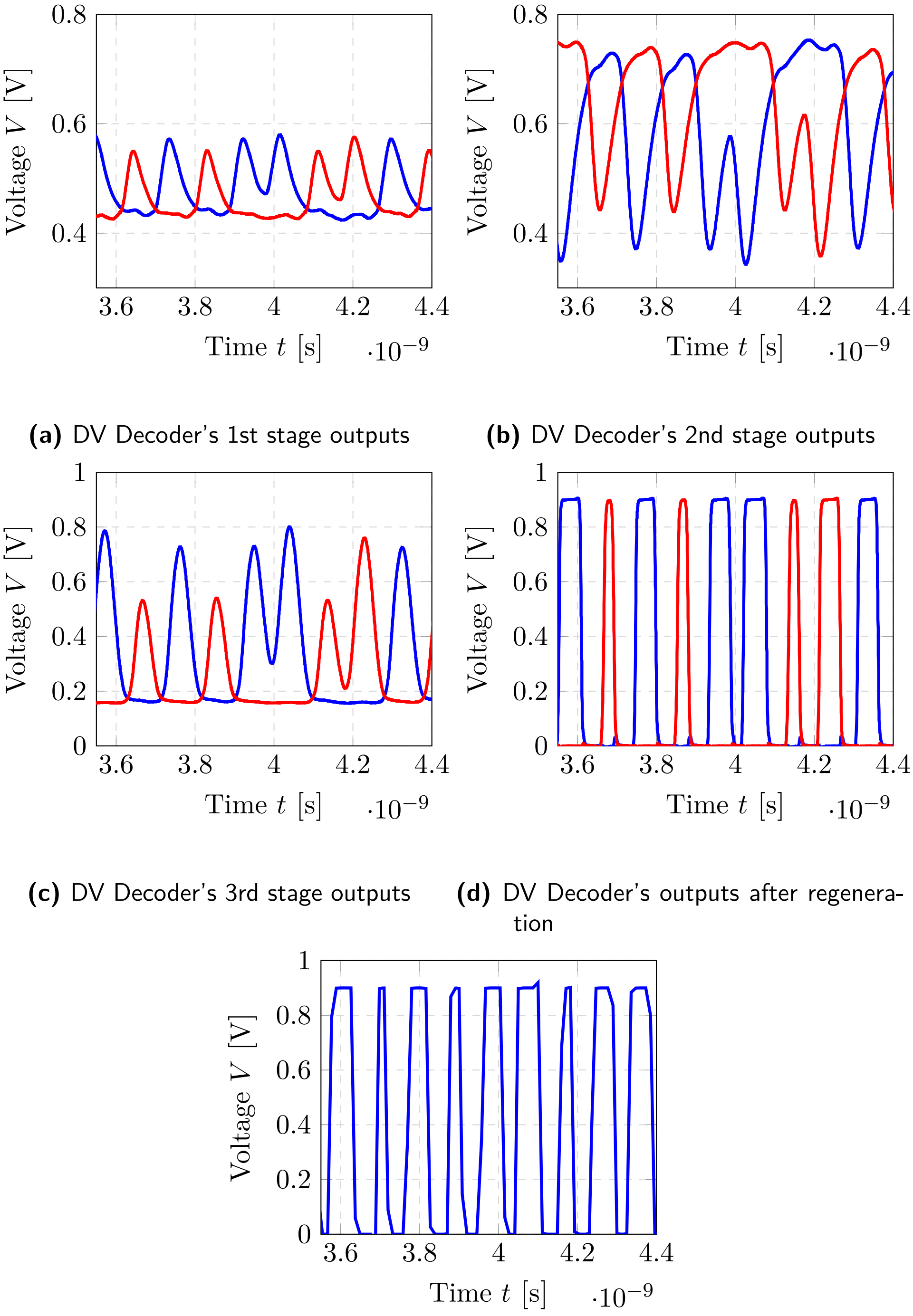
(b) DV Decoder's 2nd stage outputs

(c) DV Decoder's 3rd stage outputs

(d) DV Decoder's outputs after regeneration

(e) DVOut after NOR

Figure 6.9: Data Validity Indicator signal decoding Pre-Layout



(a) DV Decoder's 1st stage outputs

(b) DV Decoder's 2nd stage outputs

(c) DV Decoder's 3rd stage outputs

(d) DV Decoder's outputs after regeneration

(e) DVOut after NOR

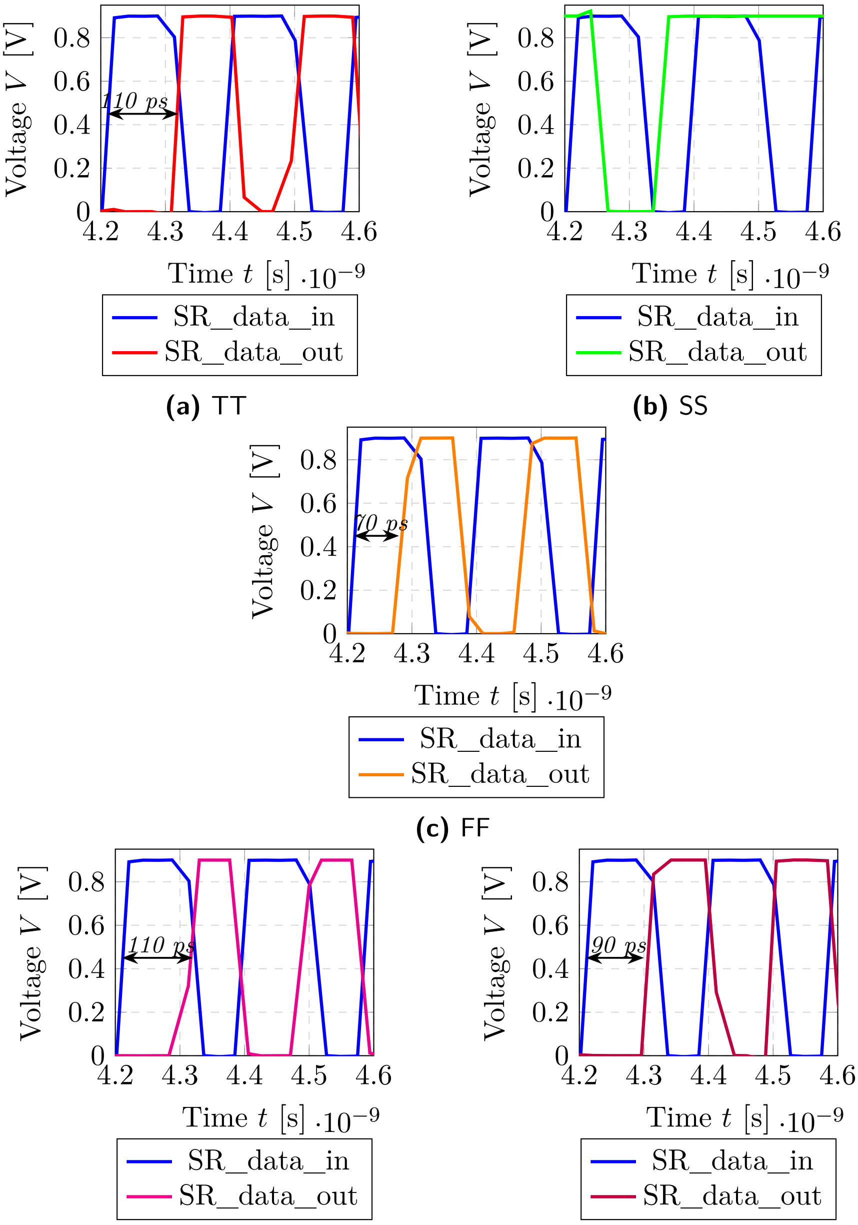
Figure 6.10: Data Validity Indicator signal decoding Post-layout

Latency:

The delay that exists in between sent and retrieved data is within the range of 70 ps - 110 ps depending on the corner. It is important to mention that due to the signaling mode used this is a delay insensitive communication protocol and this delay does not have a big influence in the overall system performance. It can be seen in Figure 6.11 that there exists data loss in SS corner. More work need to be put into in order to optimize the design so as to make it functional in all different corners.

Energy

The energy per bit is accounted as 4.93 pJ with a CLK frequency = 10.66 GHz during active transmission and 3pJ/bit during no transmission.



(a) TT

(b) SS

(c) FF

(d) SF

(e) FS

Figure 6.11: Latency of data signal in different corners

Chapter 7

Conclusion and Future Work

Conclusion

The design and implementation of a serial link was explored and tested in this project with the purpose of using it within an embedded memory. After completion of simulations, the resulting system has achieved the targeted throughput and area. It employs current-mode differential pulse signaling and the communication protocol is asynchronous. Analysis on different corners showed acceptable results in all of them except for SS, where these were not satisfactory. The energy per bit is 4.93 pJ. This interconnect system demonstrated to be a good candidate for high-performance on-chip communication and wiring area reduction. It has the advantage of scalability, being that increasing the number of bits per packet won't increase the routing area.

Future Work

It is important to explore the limitations of a serial interconnect and the feasibility of its implementation within different memory architectures. The data validity signal generation needs to be optimized in order to have a working system in all corners. Another future task is to run Monte Carlo simulations, since they would help to analyze how process and mismatch variations between devices will affect the overall behavior of the system.

The power-down of inactive modules will reduce energy consumption and it is something that needs to be addressed as well as the generation of current sources and voltage reference. The construction of the serializer and deserializer was made with only system simulation purposes. No layout was drawn for these blocks as they were not the central part of the system design assessment, therefore it is left as future work the optimization of its design as well as the control interface with memories.

References

[1] J. Singh and B. Raj, “Sram cells for embedded systems,” 03 2012.

[2] B. Mohammadi, O. Andersson, J. Nguyen, L. Ciampolini, A. Cathelin, and J. N. Rodrigues, “A 128 kb 7t sram using a single-cycle boosting mechanism in 28-nm fd-soi,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 4, pp. 1257–1268, April 2018.

[3] R. R. Dobkin, A. Morgenshtein, A. Kolodny, and R. Ginosar, “Parallel vs. serial on-chip communication,” in *Proceedings of the 2008 International Workshop on System Level Interconnect Prediction*, ser. SLIP ’08. New York, NY, USA: ACM, 2008, pp. 43–50. [Online]. Available: http://doi.acm.org/10.1145/1353610.1353620

[4] R. Dobkin, M. Moyal, A. Kolodny, and R. Ginosar, “Asynchronous current mode serial communication,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 7, pp. 1107–1117, July 2010.

[5] “What is intersymbol interference?” http://www.tech-faq.com/intersymbol-interference.html, [Online; accessed 26-March-2018].

[6] E. E. Nigussie, *Variation Tolerant On-Chip Interconnect.* USA: Springer Science+Business Media LLC, 2012.

[7] P. B. McGee, M. Y. Agyekum, M. A. Mohamed, and S. M. Nowick, “A level-encoded transition signaling protocol for high-throughput asynchronous global communication,” in *2008 14th IEEE International Symposium on Asynchronous Circuits and Systems*, April 2008, pp. 116–127.

[8] D. M. H. Neil H.E. Weste, *Integrated Circuit Design.* USA: Pearson Education, 2010.

[9] R. Ho, K. Mai, and M. Horowitz, “Managing wire scaling: a circuit perspective,” in *Proceedings of the IEEE 2003 International Interconnect Technology Conference (Cat. No.03TH8695)*, June 2003, pp. 177–179.

[10] L. Frenzel, “What’s the difference between bit rate and baud rate?” http://www.electronicdesign.com/communications/what-s-difference-between-bit-rate-and-baud-rate, 2012, [Online; accessed 1-May-2018].

[11] B. Razavi, *Design of Integrated Circuits for Optical Communications.* New Jersey: John Wiley Sons,Inc, 2012.

[12] Samson, "Serial data transmission," https://www.samson.de/document/l153en.pdf, [Online; accessed 14-March-2018].

[13] E. Nigussie, J. Plosila, and J. Isoaho, "Reliable asynchronous links for soc," in *2005 International Symposium on System-on-Chip*, Nov 2005, pp. 124–127.

[14] P. Wang, G. Pei, and E. C. C. Kan, "Pulsed wave interconnect," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 5, pp. 453–463, May 2004.

[15] E. Nigussie, S. Tuuna, J. Plosila, J. Isoaho, and H. Tenhunen, "Semi-serial on-chip link implementation for energy efficiency and high throughput," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 12, pp. 2265–2277, Dec 2012.

[16] S. Palermo, "Ecen620: Network theory,broadband circuit design lecture 19: High speed transmitters," https://pdfs.semanticscholar.org/presentation/93fc/a525deb2cfbf6d214886856031b2ecebe725.pdf, 2014, [Online; accessed 15-February-2018].

[17] F. Yuan, *CMOS Current-Mode Circuits for Data Communications.* USA: Springer Science+Business Media LLC, 2007.

[18] S. Palermo, "Ecen720: High-speed links circuits and systems spring 2017 lecture 5: Termination,tx driver, and multiplexer circuits," http://www.ece.tamu.edu/~spalermo/ecen689/lecture5_ee720_termination_txdriver.pdf, 2017, [Online; accessed 15-February-2018].

[19] W. C. Fernando L. Teixeira, Kaladhar Radhakrishnan, *Encyclopedia of RF and Microwave Engineering-High Frequency Transmision Lines.* USA: John Wiley and Sons,Inc, 2005.

[20] D. Kucar and A. Vannelli, "Interconnection modelling using distributed rlc models," in *The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications, 2003. Proceedings.*, June 2003, pp. 32–35.

[21] A. Deutsch, G. V. Kopcsay, P. Restle, G. Katopis, W. D. Becker, H. Smith, P. W. Coteus, C. W. Surovic, B. J. Rubin, R. P. Dunne, T. Gallo, K. A. Jenkins, L. M. Terman, R. H. Dennard, G. A. Sai-Halasz, and D. R. Knebel, "When are transmission-line effects important for on-chip interconnections," in *1997 Proceedings 47th Electronic Components and Technology Conference*, May 1997, pp. 704–712.

[22] I. F. T. O. K. K. M. O.-h. H. S. T. E. A.Tanabe, M. Umetani and F. Masuoka., "0.18/im cmos 10-gb/s multiplexer/de-multiplexer ics using current model logic with tolerance to threshold voltage fluctuation," pp. IEEE J. Solid–State Circuits, 36(6):988–996, 06 2001.

[23] C.-K. Yang, "Design of high-speed serial links in cmos," 06 2018.

[24] B. Razavi, *Design of Analog CMOS Integrated Circuits.* India: Mc Graw Hill Education, 2002.

[25] Radhika, N. Pandey, K. Gupta, and M. Gupta, "Low power d-latch design using mcml tri-state buffers," in *2014 International Conference on Signal Processing and Integrated Networks (SPIN)*, Feb 2014, pp. 531–534.

Printed by Tryckeriet i E-huset, Lund 2018

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2018-658
http://www.eit.lth.se