6-GB/S SERIAL LINK TRANSCEIVER FOR NOCS

By

Safaa Ahmed Mohammed Abdelfattah

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Electronics and Communications Engineering

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2016
6-GB/S SERIAL LINK TRANSCEIVER FOR NOCS

By

Safaa Ahmed Mohammed Abdelfattah

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Electronics and Communications Engineering

Under the Supervision of

Prof. Serag E. D. Habib
Professor of Electronics
Electronics and Communications Engineering Department
Faculty of Engineering, Cairo University

Dr. Sameh A. Ibrahim
Assistant Professor
Electronics and Communications Engineering Department
Faculty of Engineering, Ain Shams University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2016
6-GB/S SERIAL LINK TRANSCEIVER FOR NOCS

By

Safaa Ahmed Mohammed Abdelfattah

A Thesis Submitted to the
Faculty of Engineering at Cairo University
in Partial Fulfillment of the
Requirements for the Degree of
MASTER OF SCIENCE
in
Electronics and Communications Engineering

Approved by the Examining Committee:

Prof. Serag E. D. Habib, Thesis Main Advisor

Prof. Mohamed Riad Elghoneimy, Internal Examiner

Prof. Mohamed A. Dessouky, External Examiner
Electronics and Communications Engineering Department,
Faculty of Engineering, Ain Shams University

FACULTY OF ENGINEERING, CAIRO UNIVERSITY
GIZA, EGYPT
2016
Engineer’s Name: Safaa Ahmed Mohammed Abdelfattah
Date of Birth: 22/11/1989
Nationality: Egyptian
E-mail: safaaelawamy@cu.edu.eg
Phone: +201285436641
Address: 13 Abd El-latif street, El salam City, Cairo
Registration Date: 1/10/2012
Awarding Date: -/-/2016
Degree: Master of Science
Department: Electronics and Communications Engineering

Supervisors:
Prof. Serag E. D. Habib
Dr. Sameh A. Ibrahim
(Electronics and Communications Engineering Department, Faculty of Engineering, Ain Shams University)

Examiners:
Prof. Mohamed A. Dessouky (External examiner)
(Electronics and Communications Engineering Department, Faculty of Engineering, Ain Shams University)
Prof. Mohamed Riad Elghoneimy (Internal examiner)
Prof. Serag E. D. Habib (Thesis main advisor)

Title of Thesis:
6-Gb/s Serial Link Transceiver For NoCs

Key Words:
SerDes; NoC; CUSPARC

Summary:
The design of a 6-Gb/s serial link for CUSPARC NoC is presented. The proposed SerDes consists of a serializer and a deserializer. The design targets TSMC digital 65-nm CMOS technology and 1.2-V supply. The use of serial links reduces the interconnect area of the network on chip by 93.96% relative to the design with parallel 32 bit data links. The traces between the cores achieved maximum tolerable clock skew between the Tx and the Rx up to ±36% of the clock period. The link consumes 6.9 mW power (1.15 pJ/bit).
Acknowledgements

All Praise to ALLAH for his many blessings on me and for completing this work.

I am really would like to thank my advisors

Prof. Dr. Serag E. D. Habib and Dr. Sameh A. Ibrahim

for their assistance, motivation and guidance through out my work.

I would express my gratitude to my family (especially my parents) for their love and encouragement to me since I was born.

I am very grateful for my dear fiancee’

Hussein Mohammed

for his support, encouragement, advice, patience, and help since we have met.

Thanks to Ahmed Reda for helping in the layout design.
Dedication

To my parents and my fiancee’ Hussein
# Table of Contents

Acknowledgements                          i  
Dedication                                 ii 
Table of Contents                          iii 
List of Tables                             v  
List of Figures                            vi  
List of Symbols and Abbreviations          viii  
Abstract                                   x  

## 1 INTRODUCTION
1.1 Motivation                            1  
1.2 Problem Statement                     2  
1.3 Thesis Outline                        2  

## 2 BACKGROUND
2.1 Parallel and Serial Links             3  
2.2 History of Serial Links               4  
   2.2.1 Visual Method                      4  
      2.2.1.1 Smoke Signal                  4  
   2.2.2 Wire Method                       4  
      2.2.2.1 Current Loop                  4  
      2.2.2.2 Morse code                    5  
2.3 Introduction to high speed serial links 5  
   2.3.1 Types of SERDES                   6  
      2.3.1.1 Parallel Clock SerDes         6  
      2.3.1.2 Embedded Clock SerDes          6  
      2.3.1.3 8b/10b SerDes                 7  
      2.3.1.4 Bit Interleaving SerDes        7  
      2.3.1.5 Shift Register SerDes          7  
2.4 Introduction to CUSPARC                8  
   2.4.1 CUSPARC Architecture               8  
      2.4.1.1 Processor Integer Unit (IU)    9  
      2.4.1.2 Instruction Cache (I-Cache)    9  
      2.4.1.3 Data Cache (D-Cache)           10  
      2.4.1.4 Cache Controller               10  
      2.4.1.5 Main processor bus             10  
      2.4.1.6 Memory Controller              10  
      2.4.1.7 Bridge                         11  
      2.4.1.8 Interrupt Controller           11  

# List of Tables

2.1 Instruction and Data Caches parameters ........................................... 10
2.2 SerDes Comparison ........................................................................... 13
2.3 WAFT and Basic SerDes Comparison .................................................. 14
2.4 Coplanar and Microstrip Waveguides Comparison ............................. 15
2.5 Comparison between Area of 2 Gb/s serial link and its corresponding parallel link Proposed In [14] ........................................................... 18
2.6 Comparison between Area of 8 Gb/s serial link and its corresponding parallel link Proposed In [14] ...................................................... 19
2.7 Area Optimization for the Routers proposed in [15] ......................... 21
2.8 Performance of 64-bit 8-mm Serial Links in [16] ............................... 22
2.9 Comparison Between Different SerDes Architectures ....................... 23

3.1 Simulation results of the SA sampler ................................................. 33

4.1 Data Rates ......................................................................................... 41
4.2 Input and encoded data of the encoder .............................................. 42
4.3 Parallel Encoded Data ...................................................................... 53
4.4 Clock Skew Test Cases ..................................................................... 57
4.5 Clock Skew results for Typical Typical Corner ................................. 57
4.6 Clock Skew results for Slow Slow Corner ......................................... 57
4.7 Clock Skew results for Fast Fast Corner .......................................... 57
4.8 Power Distribution .......................................................................... 58
4.9 SerDes Comparison ........................................................................ 60
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Parallel Transmission Example</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Serial Transmission Example</td>
<td>3</td>
</tr>
<tr>
<td>2.3</td>
<td>Serializer</td>
<td>5</td>
</tr>
<tr>
<td>2.4</td>
<td>Deserializer</td>
<td>5</td>
</tr>
<tr>
<td>2.5</td>
<td>Simple Parallel Clock SerDes Example</td>
<td>6</td>
</tr>
<tr>
<td>2.6</td>
<td>Simple Interleaving SerDes Example</td>
<td>7</td>
</tr>
<tr>
<td>2.7</td>
<td>Conventional SerDes [8]</td>
<td>8</td>
</tr>
<tr>
<td>2.8</td>
<td>CUSPARC Architecture [1]</td>
<td>9</td>
</tr>
<tr>
<td>2.9</td>
<td>WAFT SerDes [8]</td>
<td>14</td>
</tr>
<tr>
<td>2.10</td>
<td>Prototype SerDes for on Chip signaling [13]</td>
<td>15</td>
</tr>
<tr>
<td>2.11</td>
<td>The Transmitter Proposed In [13]</td>
<td>16</td>
</tr>
<tr>
<td>2.12</td>
<td>The Receiver Proposed In [13]</td>
<td>17</td>
</tr>
<tr>
<td>2.13</td>
<td>2-Gb/s SerDes Proposed In [14]</td>
<td>18</td>
</tr>
<tr>
<td>2.14</td>
<td>8-Gb/s Quasi Serial Link [14]</td>
<td>19</td>
</tr>
<tr>
<td>2.15</td>
<td>SerDes With Serialization Ratio n/p [15]</td>
<td>20</td>
</tr>
<tr>
<td>2.16</td>
<td>Proposed Serial Link in [16]</td>
<td>21</td>
</tr>
<tr>
<td>2.17</td>
<td>WP CML Serializer in [17]</td>
<td>22</td>
</tr>
<tr>
<td>2.18</td>
<td>WP CML Deserializer in [17]</td>
<td>23</td>
</tr>
<tr>
<td>3.1</td>
<td>The Whole 6-Gb/s Serial Link Platform</td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>Coding Method</td>
<td>25</td>
</tr>
<tr>
<td>3.3</td>
<td>10:1 Multiplexer Block Diagram</td>
<td>26</td>
</tr>
<tr>
<td>3.4</td>
<td>The Transmitter 5:1 Multiplexer Block Diagram</td>
<td>26</td>
</tr>
<tr>
<td>3.5</td>
<td>CMOS D Latch</td>
<td>27</td>
</tr>
<tr>
<td>3.6</td>
<td>Transmission Gate 2:1 MUX</td>
<td>27</td>
</tr>
<tr>
<td>3.7</td>
<td>Waveforms of the Selection Signals Used in the even 5:1 Multiplexer</td>
<td>28</td>
</tr>
<tr>
<td>3.8</td>
<td>Selections Circuit</td>
<td>29</td>
</tr>
<tr>
<td>3.9</td>
<td>JK FF Schematic</td>
<td>29</td>
</tr>
<tr>
<td>3.10</td>
<td>TSPC D Flip Flop</td>
<td>30</td>
</tr>
<tr>
<td>3.11</td>
<td>Driver Schematic</td>
<td>30</td>
</tr>
<tr>
<td>3.12</td>
<td>Driver with TL</td>
<td>31</td>
</tr>
<tr>
<td>3.13</td>
<td>Frequency Divider</td>
<td>31</td>
</tr>
<tr>
<td>3.14</td>
<td>StrongARM Sampler</td>
<td>32</td>
</tr>
<tr>
<td>3.15</td>
<td>Optimized RS Latch</td>
<td>33</td>
</tr>
<tr>
<td>3.16</td>
<td>The Receiver 1:10 Demultiplexer Block Diagram</td>
<td>34</td>
</tr>
<tr>
<td>3.17</td>
<td>D Flip Flop</td>
<td>34</td>
</tr>
<tr>
<td>3.18</td>
<td>The Receiver Selection Circuit</td>
<td>35</td>
</tr>
<tr>
<td>3.19</td>
<td>Transmitter Layout</td>
<td>36</td>
</tr>
<tr>
<td>3.20</td>
<td>Receiver Layout</td>
<td>37</td>
</tr>
<tr>
<td>3.21</td>
<td>Synchronization Process</td>
<td>38</td>
</tr>
<tr>
<td>3.22</td>
<td>Timing Diagram</td>
<td>39</td>
</tr>
<tr>
<td>3.23</td>
<td>Gate based 2:1 MUX</td>
<td>40</td>
</tr>
<tr>
<td>3.24</td>
<td>New 5:1 MUX</td>
<td>40</td>
</tr>
</tbody>
</table>
vii
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two-Dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three-Dimensional</td>
</tr>
<tr>
<td>AMS</td>
<td>Analog Mixed Signal</td>
</tr>
<tr>
<td>ASIC</td>
<td>Application-Specific Integrated Circuit</td>
</tr>
<tr>
<td>CDR</td>
<td>Clock and Data Recovery</td>
</tr>
<tr>
<td>CML</td>
<td>Current Mode Logic</td>
</tr>
<tr>
<td>CMOS</td>
<td>Complementary Metal Oxide Semiconductor</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CUSPARC</td>
<td>Cairo University SPARC processor</td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
</tr>
<tr>
<td>D-Cache</td>
<td>Data Cache</td>
</tr>
<tr>
<td>DE</td>
<td>Delay Element</td>
</tr>
<tr>
<td>DEMUX</td>
<td>Demultiplexer</td>
</tr>
<tr>
<td>DMA</td>
<td>Direct Memory Access</td>
</tr>
<tr>
<td>FF</td>
<td>Flip Flop</td>
</tr>
<tr>
<td>FF Corner</td>
<td>Fast Fast Corner</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>FSM</td>
<td>Finite State Machine</td>
</tr>
<tr>
<td>I^2C</td>
<td>Inter-Integrated Circuit</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines</td>
</tr>
<tr>
<td>I-Cache</td>
<td>Instruction Cache</td>
</tr>
<tr>
<td>IR</td>
<td>Instruction Register</td>
</tr>
<tr>
<td>ISI</td>
<td>Inter-Symbol Interference</td>
</tr>
<tr>
<td>ITRS</td>
<td>International Technology Roadmap for Semiconductors</td>
</tr>
<tr>
<td>IU</td>
<td>Integer Unit</td>
</tr>
<tr>
<td>LSB</td>
<td>Least Significant Bit</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>MSB</td>
<td>Most Significant Bit</td>
</tr>
<tr>
<td>MUX</td>
<td>Multiplexer</td>
</tr>
<tr>
<td>NoC</td>
<td>Network on Chip</td>
</tr>
<tr>
<td>PCI Express</td>
<td>Peripheral Component Interconnect Express</td>
</tr>
<tr>
<td>PLL</td>
<td>Phase Locked Loop</td>
</tr>
<tr>
<td>R</td>
<td>Router</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RD</td>
<td>Running Disparity</td>
</tr>
<tr>
<td>RS</td>
<td>Reset Set</td>
</tr>
<tr>
<td>Rx</td>
<td>Receiver</td>
</tr>
<tr>
<td>SA</td>
<td>Strong Arm</td>
</tr>
<tr>
<td>SAFF</td>
<td>Strong Arm Flip Flop</td>
</tr>
<tr>
<td>SATA</td>
<td>Serial ATA</td>
</tr>
<tr>
<td>SerDes</td>
<td>Serializer and Deserializer</td>
</tr>
<tr>
<td>SONET</td>
<td>Synchronous Optical Networking</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static Random Access Memory</td>
</tr>
<tr>
<td>SS Corner</td>
<td>Slow Slow Corner</td>
</tr>
<tr>
<td>TL</td>
<td>Transmission Line</td>
</tr>
<tr>
<td>TSMC</td>
<td>Taiwan Semiconductor Manufacturing Corporation</td>
</tr>
<tr>
<td>TSPC</td>
<td>True Single-Phase-Clocked</td>
</tr>
<tr>
<td>TSV</td>
<td>Through Silicon Via</td>
</tr>
<tr>
<td>TT Corner</td>
<td>Typical Typical Corner</td>
</tr>
<tr>
<td>Tx</td>
<td>Transmitter</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver/Transmitter</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>UTOPIA</td>
<td>Universal Test and Operations Physical Interface for Atm</td>
</tr>
<tr>
<td>VLSI</td>
<td>Very Large-Scale Integration</td>
</tr>
<tr>
<td>WAFT</td>
<td>Wave-Front Train</td>
</tr>
<tr>
<td>WP</td>
<td>Wave Pipelined</td>
</tr>
</tbody>
</table>
Abstract

Compared to parallel data transmission, serial transmission has the advantage of smaller number of ports/pins, high immunity to interference, lower power consumption and smaller area. Concurrent with the trend to replace parallel buses with fast high-speed serial links, there is an ongoing trend to replace the multi-core memory-shared processor with a loosely-coupled, many-core processor Network on Chip (NoC).

This work introduces a fast serial link for many-core NoCs. The proposed serial link consists of a Serializer and a Deserializer (SerDes). The serializer contains an 8b/10b encoder, a 10:1 multiplexer (MUX) and a driver. The deserializer contains a sampler, a 1:10 demultiplexer (DEMUX), regenerative latches and a 10b/8b decoder. The design is modeled using a digital 65-nm CMOS technology and a 1.2-V supply using an Analog Mixed Signal (AMS) simulation tool. This SerDes design is integrated with the previously designed 16-core NoC based on Cairo University SPARC processor (CUSPARC) arranged in a 2D mesh architecture.

The SerDes works at 6 Gb/s. The use of serial links reduces the interconnect area of the network on chip by 93.96% relative to the design with parallel 32 bit data links. The traces between the cores are modeled using metal layer number eight achieving high percentage of allowable clock skew between the transmitter and the receiver. The design can tolerate clock skew between the transmitter and receiver of up to ±36% of clock period at TT corner. The transceiver consumes 6.9 mW power (1.15 pJ/bit).
Chapter 1: Introduction

1.1 Motivation

The number of transistors on chip almost doubles every two years, a technology observation famously known as Moore’s law. So the development in the VLSI fabrication enabled the fabrication of transistors smaller in size and faster in performance with higher transistor density on the same area. A faster transistor means higher operational clock frequency and higher performance.

Throughout time, the operational frequency kept increasing with VLSI technology scaling until it was hit by power limitations. To overcome this problem a new era came which is the Many-Core era. In this era, instead of having a single core processor with high frequency, designers used many cores on the same chip and each core operates with low frequency. However they achieve higher overall performance than the single-core processor and the power consumption is almost the same.

According to the International Technology Roadmap for Semiconductors (ITRS), as the technology scales, the delay of the interconnect increases. The problem now is no longer in the performance of the cores but in the communication between the cores. Hence, the overall performance will be affected.

Now the focus should be directed to the communication between the cores. Most of the on-chip communications are carried out through parallel communication buses. Parallel communication sounds like the best solution for the first glance because each bit has its own dedicated bus which is available all the time. However, the interconnect scaling problem makes the parallel communication not very efficient in terms of speed, area, and power consumption.

When it comes to a design with large number of bits, the routing of the parallel buses becomes very complicated because of the increase in the number of lines. The increase of power consumption and timing errors due to jitter, skew and crosstalk are the main drawbacks of parallel communication.

With these drawbacks of parallel communication, designers now tend to use serial communication on-chip. Serial communication of course don’t have the drawbacks of area, power consumption and complexity in routing like parallel buses. In the case of parallel buses with N buses each working at F Gbps, the serial bus must operate at $N \times F$ Gbps. Thus, serial buses have to operate at higher speeds.

Our work aims at replacing parallel buses NoC with serial buses NoC to use higher data rates for transmission and make it smaller in area.
1.2 Problem Statement

This work aims to design on-chip SerDes for Many-core NoCs to transmit data through serial links with high speed. As parallel interconnects were predominant in transmitting data between frameworks. However, because of their sensitivity to delay mismatches and interference, high cost and area, nowadays serial buses are essential. By diminishing the number of pins and interconnects, the serial buses offer high speed-links that can cope with the incremental speed of the Central Processing Units (CPUs).

We aim to apply this design to CUSPARC-based Many-core NoC designed by Soliman et al [1] which employed parallel data transmission.

1.3 Thesis Outline

The thesis is organized as follows:

- Chapter 2 provides a background about NoCs, the problems of parallel buses in NoC and a literature review on serial links.
- Chapter 3 presents the proposed 6-Gb/s SerDes design.
- Chapter 4 presents the simulation results of the proposed SerDes.
- Chapter 5 concludes this work.
Chapter 2: Background

2.1 Parallel and Serial Links

Serial communication transmits bits sequentially through a channel. Parallel communication transmits bits simultaneously through multiple channels with a synchronous clock.

The main difference between serial and parallel transportation is the number of transmitting channels. Parallel communication uses higher number of channels than serial communication. For example, if we have eight bits needed to be transmitted, in parallel communication it uses eight channels, but in serial communication it uses only one channel as shown in Figure 2.1 and Figure 2.2.

![Figure 2.1: Parallel Transmission Example](image1)

![Figure 2.2: Serial Transmission Example](image2)

There are other differences between serial and parallel buses:

- Parallel buses are simpler than serial buses, so parallel ports are easier to be implemented. Serial buses require an interface like Universal Asynchronous Receiver/Transmitter (UART) to send the data.
• Parallel buses can’t use high frequency to send the data. All bits should arrive at
the same time at the receiver side, but this is difficult to achieve for high data rates
as the signal delay time is not equal for all the links. Hence the receiver must wait
until all the data arrive and this decreases the data rate and throughput.

• In parallel buses, the interference between the lines is significant. As a result, only
short buses are allowed.

• Integrated circuits are expensive if they have large number of pins, so to decrease
the number of pins, integrated circuits use serial links rather than parallel links
especially if the speed is not an issue.

Parallel buses were used in the integrated circuits and memories and in other peripherals, but in the modern computers serial links are dominant.

Serial transmission is usually used in long channels to overcome the problems of cost
and interference that exist in parallel transmission. Moreover serial communication is
used in short channels especially in computers buses because serial buses improve the
signal integrity.

Serial Peripheral Interface (SPI), Inter-Integrated Circuit I²C, Peripheral Component
Interconnect Express (PCI Express), Universal Serial Bus (USB), Serial ATA (SATA), and
Synchronous Optical Networking (SONET) are all examples of serial buses.

2.2 History of Serial Links

There are many earlier methods for the serial links.

2.2.1 Visual Method

2.2.1.1 Smoke Signal

Americans and Chinese developed primitive serial communication technique using smoke
signaling [2]. The smoke signal is a type of a visual transmission. When the fire is covered
with a blanket and removed quickly, there is a smoke. The size, timing and shape of
the puffs of the smoke can be controlled. Within the visual range of the smoke, anyone
can observe the puffs. Stations were created to maximize the visual range to cover large
distances.

2.2.2 Wire Method

2.2.2.1 Current Loop

Analog and digital loops are two types of the current loops. An analog current loop uses
4-20 mA of current where 4 mA corresponds to zero output or 0%, and 20 mA corresponds
to full scale output or 100%. In the digital current loop, the high signal is represented by the absence of current, and the low signal is represented by the presence of the current.

2.2.2.2 Morse code

In 1832, Samuel F.B. Morse developed the single wire telegraph concept [3]. The Morse code became in some earlier time the dominant language of the telegraph. Morse was able to transmit signals over long distance wires. Professor Leonard Gale, a professor of chemistry at New York University, helped Morse in sending a message along 16 km distance and this was a great development.

2.3 Introduction to high speed serial links

High speed serial links consists mainly of a Serializer and a Deserializer (SerDes). The serializer acts as a transmitter and the deserializer acts as a receiver.

The serializer main goal is to achieve high bit rate, low power consumption and low noise. It consists of a multiplexer to convert from parallel data to serial ones, a pre-driver and a driver that makes the output signals able to transmit properly [4]. Figure 2.3 shows a simple serializer.

![Serializer Diagram](image)

Figure 2.3: Serializer

The deserializer main goal is to achieve low Bit Error Rate (BER), high bit rate, and low power consumption. It consists of a pre-amplifier which compensates the loss of the channel, a sampler or a slicer which is responsible for extracting the data from the received signal, and a DEMUX which converts serial data to parallel data [5]. Figure 2.4 shows a simple deserializer.

![Deserializer Diagram](image)

Figure 2.4: Deserializer
Deserializer can contain a Clock and Data Recovery (CDR) block which selects an optimal sampling point for the data by observing the data transitions.

2.3.1 Types of SERDES

There are many types of SerDes design [6] [7]: parallel clock SerDes, embedded clock SerDes, 8b/10b SerDes, bit interleaving SerDes, and shift register SerDes.

2.3.1.1 Parallel Clock SerDes

This type is widely used in Universal Test And Operations Physical Interface For Atm (UTOPIA), PCI, and processor buses. It consists of several serial links in parallel. Thus, a bank of n:1 MUXs can be used instead of using one big MUX as shown in Figure 2.5.

![Figure 2.5: Simple Parallel Clock SerDes Example](image)

Each MUX is responsible for serializing its section of the bus separately. The serial streams travel to the Rx in parallel with a differential clock signal pair that can be used for the Rx to recover the data. The skew between the clock and the data must be considered and minimized for proper operation. Parallel clock SerDes is efficient in price and performance and it is an efficient way to transmit a wide parallel bus through long cables.

2.3.1.2 Embedded Clock SerDes

This type of SerDes serializes both data and clock onto one serial pair. The transmitted frame consists of: two bits of the clock (low and high bits) are embedded into the serial stream, a start signal, payload of the data, and a stop bit at the end of the frame. After powering up, the Rx searches for the periodic embedded clock rising edge. Since the data
bits change value over time while the clock bits do not, the Rx is capable of recognizing
the clock bits and synchronize to it. Consequently, the Rx can recover the data once locked.
The embedded clock SerDes is commonly used in applications that transmit data with
other control signals such as parity bits, synchronization bits, status bits, etc. There is
another feature of this type of SerDes which is that the Rx lock automatically to data.
This is useful especially for a remote Rx and in the case of having one Tx that sends
data to many receivers, so each Rx can lock to the data without interrupting the traffic by
transmitting training characters.

2.3.1.3 8b/10b SerDes

It maps eight parallel bits to ten bits parallel bits and then serializes these ten bits onto a
serial pair. The ten bits code is generated by IBM Corporation in early 1980’s. 8b/10b
encoding is used to generate balanced number of ones and zeros in the transmitted data and
to guarantee many transitions every cycle to make the Rx synchronize to the incoming data
stream. The Rx can locate the ten bits code word boundaries in the stream by observing
a special symbol called a comma character which was sent by the Tx. The bit sequence
in the comma character never appears in the normal data. Once the Rx observed the 10
bit code, it maps them back to eight bits and flagging an error if it detects an invalid ten
bits code. 8b/10b SerDes is used in many standards such as Ethernet, Fiber Channel, InfiniteBand.

2.3.1.4 Bit Interleaving SerDes

It multiplexes slower SONET or 8b/10b serial streams into one faster serial stream by
interleaving the bits as shown in Figure 2.6.

```
| A | B | C | D |
```

D C B A
D' C' B' A'

Figure 2.6: Simple Interleaving SerDes Example

The Rx demultiplexes the bits back to the slower streams. Bit interleaving SerDes is
used for switches and routeres to get more bandwidth.

2.3.1.5 Shift Register SerDes

This type is considered the fastest conventional SerDes. It consists of D flip-flops which
load and shift the data, and MUXs which select between the output data from the D
flip-flops and the parallel data as shown in Figure 2.7
2.4 Introduction to CUSPARC

CUSPARC is the first Egyptian processor which was developed at Cairo University. CUSPARC development started in 2004 in a graduation project. Since then, CUSPARC passed through many development phases to boost its performance. IBM CMOS 130nm process was used in implement two versions of CUSPARC (V1 and V2) on silicon. The CUSPARC design was also ported to TSMC 65 nm technology. Soliman etal [1] designed a 16-core NoC based on CUSPARC V1 cores.

2.4.1 CUSPARC Architecture

Figure 2.8 shows the architecture of the first generation of CUSPARC.
2.4.1 Processor Integer Unit (IU)

The IU is a very important part in the CUSPARC. The fetching, decoding, and execution are done in this unit. The processor has four stages of pipeline and about 80 instructions are implemented as per the SPARC ISA standard. IU additionally contains a windowed register file with four windows and 72 registers. Additionally, various control/status registers which control the entire processor operation are actualized in the IU. In 2010, an integer multiplier has been added to the IU to enhance its performance.

2.4.1.2 Instruction Cache (I-Cache)

A basic finite state machine (FSM) controls the instruction cache which is used in the fetch stage. The instruction of the address is read inline and stored in the Instruction Register of the decode stage (IR) by the I-Cache. If there is a cache miss, the processor pipeline is sitting tight for the I-Cache to have the required block from the primary memory. Consequently, there is a cache miss state at which the I-Cache sends a request to the controller of the cache to read the block from the memory.

Table 2.1 shows the I-Cache parameters.
2.4.1.3 Data Cache (D-Cache)

While the I-Cache is controlled by a simple FSM, a more complex FSM controls the D-Cache because it is responsible for the operations of the read and write. Moreover, the D-Cache is responsible for managing the Input / Output (I/O) devices. The controller unit of the D-Cache checks the address coming from the IU to recognize if it corresponds to an I/O devices or not. Then if the address is from an I/O device the D-Cache sends a request for read or write operation to the controller of the cache to write or read data to or from the I/O device. Data is not kept in the RAM of the cache to follow along the data in the devices. Furthermore, if the address is not for an I/O device the D-Cache carries out a comparison between the TAG memory and the coming address to see if there is a miss or a hit in the cache.

Table 2.1 illustrates the D-Cache parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>I-Cache</th>
<th>D-Cache</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miss latency</td>
<td>8 Clock Cycles</td>
<td>8 Clock Cycles</td>
</tr>
<tr>
<td>Replacement Policy</td>
<td>Direct Map</td>
<td>Direct Map</td>
</tr>
<tr>
<td>Write Policy</td>
<td>NA</td>
<td>Write Through</td>
</tr>
<tr>
<td>Size</td>
<td>4 KB</td>
<td>4 KB</td>
</tr>
</tbody>
</table>

2.4.1.4 Cache Controller

A simple FSM controls the cache controller. In case of a cache miss or I/O access the cache controller works like a server to answer for the requirements form the I-Cache and D-Cache. The cache controller is considered the only unit on the main bus of the processor, which made the implementation of the bus simpler.

2.4.1.5 Main processor bus

The main bus of the CUSPARC is the 64-bit Wishbone bus. This wishbone bus is an open source standard created by the OpenCores organization [9]. The Wishbone bus make different bus widths simple and flexible. Moreover, it offers portability to different IC technologies.

There are two types of the interfaces to the wishbone bus: MASTER and SLAVE. The cores that can generate bus cycles are called MASTER interfaces and the cores that receive the cycles of the bus are called SLAVE interfaces.

2.4.1.6 Memory Controller

The memory controller interfaces the flash memory and the main RAM memory to the processor main bus. The design of the memory controller incorporates flexible software control of the SRAM and flash memory timing signals. The memory controller consists of several blocks:
- Main Unit: which receives the requests that are coming from the controller of the cache through the main processor bus (64-bit wishbone bus) and recognizes if the address is a flash memory or RAM memory.

- Flash Controller: responsible for producing the control signals of the FLASH memory and waits to get the valid data from the bus of the memory.

- SRAM Controller: responsible for producing the control signals of the SRAM and waits to get the valid data from the bus of the memory.

- Control Registers: consists of different registers which control the units of the memory controller like registers with data addresses and instructions and memory delay counter.

- Boot Loader: is a Direct Memory Access (DMA) which is responsible for moving the data and instructions from the FLASH to SRAM memory at the booting stage. To determine the start of the boot loader, a software is used to get the locations of the destination and the source of the data and instructions that will be moved.

2.4.1.7 Bridge

The bridge connects the 64-bit wishbone bus to the 8-bit wishbone bus. A FIFO buffer is responsible for saving the data coming from the main bus (64-bit wishbone bus) to be written into the devices connected to the slow bus (8-bit wishbone bus) in the write operation. This FIFO buffer is not used in the read operation, instead the fast bus just waits for the data to come from the slow bus.

2.4.1.8 Interrupt Controller

The interrupt controller handles the interrupt requests coming to the CUSPARC processor from external devices.

2.4.1.9 Peripherals

CUSPARC can deal with slow external peripherals through the 8-bit wishbone bus interface. These peripherals should be compatible with the standard of the wishbone bus. CUSPARC can deal with 256 devices simultaneously. There are other peripherals such as UART and interrupt inputs IRQ1, IRQ2 and IRQ3. Every peripheral has a specific address space.

2.5 Introduction to NoCs

As per Pollack’s rule [10], the performance of the processors increases proportional to the square root of the increase in complexity. This leads to a slower increase in the
performance than the power. A Multi-Core processor offers a solution to this problem.

Multi-Core-processors result in higher performance through increasing the number of cores rather than increasing the frequency. However, by increasing the number of cores, the interconnects become an obstacle for enhancing the performance. This is because of the errors and the delays that will result from very long wires. NoCs provide a solution to this problem. In NoCs, shorter interconnects are used between the cores as each core communicates with the neighboring cores only even if there are thousands of cores [11].

NoCs consist of topology, routing, flow control, router design, and link design.

- **Topology:** determine the layout and the links between the NoC nodes.
- **Routing:** for a certain topology, routing method determine the links between nodes that the message will travel through.
- **Flow control:** is responsible for allocating channels and buffers for messages.
- **Router:** consists of input buffers, router logic and state, allocators, and a switch. Pipelining technique is used to improve throughput.

### 2.5.1 Problems of parallel Buses in NoC

Although NoC offers a solution for many multi-core challenge as discussed in Section 2.5, it introduces other challenges on its own especially in a large NoC with parallel data transmission between cores.

Parallel buses must work with low data rates to decrease the power dissipation in the cores of the NoC. Moreover, interference exists between multiple adjacent wires. Parallel buses consume area, and increase the complexity in routing. The increase in the routing complexity can be solved with different metal layers, but to a certain limit [11].

Serial buses result in simpler layout and result in lower area, cost, and interference between wires. However, serial buses introduce Inter-Symbol Interference (ISI) at high data rates. Proper SerDes design is needed to overcome this issue [11].

### 2.6 Literature Review

A lot of research is available in the field of serial links and made modifications over the basic SerDes designs discussed in section 2.3.1. Table 2.2 shows comparison between some designs related to on-chip NoC designs with serial communication links over the period from year 2005 to 2014.

A new serdes technique, wave-front train (WAFT), is presented in [8] to get over the problems of the basic serializer/deserializer. The shift register SerDes is considered the fastest conventional SerDes. It contains D flip-flops which load and shift the data, and MUXs which select between the output data from the D flip-flops and the parallel data. But this architecture has several problems:
Table 2.2: SerDes Comparison

<table>
<thead>
<tr>
<th>paper number/ specs</th>
<th>[8]</th>
<th>[12]</th>
<th>[13]</th>
<th>[14]</th>
<th>[15]</th>
<th>[16]</th>
<th>[17]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Summary</td>
<td>A new serialization technique WAFT SerDes used in NoC</td>
<td>Prototype serial link employing pulsed current-mode signaling</td>
<td>Serial link transceiver for on-chip</td>
<td>Synchronous parallel 3D links TSV</td>
<td>Asynchronous 3D-NoC router</td>
<td>Self-timed serial link</td>
<td>WP-CML SerDes structure</td>
</tr>
<tr>
<td>Fabrication</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Technology</td>
<td>0.18 µm</td>
<td>0.18 µm</td>
<td>0.13 µm</td>
<td>90 nm</td>
<td>TSMC 65 nm</td>
<td>TSMC 65 nm</td>
<td>TSMC 65 nm</td>
</tr>
<tr>
<td>Data Rate (Gb/s)</td>
<td>3</td>
<td>8</td>
<td>9</td>
<td>2</td>
<td>16</td>
<td>9.09 (one link)</td>
<td>12.67</td>
</tr>
<tr>
<td>Energy per bit (pJ/bit)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.95</td>
<td>0.98 (one link)</td>
<td>1.12</td>
</tr>
<tr>
<td>Area of SerDes (µm²)</td>
<td>-</td>
<td>-</td>
<td>177500</td>
<td>460.681</td>
<td>19000</td>
<td>33124</td>
<td>-</td>
</tr>
</tbody>
</table>

- Limitation of the maximum clock frequency: as it is limited by the delay time of the D FF.

- The overhead of the high clock of the system.

- Synchronization problem: If the channel is long, there will be a skew between the receiver and the transmitter which should be taken into account.

The WAve-Front Train (WAFT) technique is a new method of serialization and deserialization to overcome these problems. Table 2.3 shows the difference between WAFT SerDes and the basic SerDes.

As shown in Figure 2.9 which is 4:1 WAFT circuit, when the EN signal equals to zero (Low) the data (D0 : D3) is loaded into the delay elements (QS0 : QS3). VDD is loaded
in QP and the ground signal is loaded into OUT and this means that the serializer is off. The second case if the EN equals to 1 (High), the loaded signal (D0 : D3) are shifted through the MUXs to OUT. The transmitted data shifts through the deserializer until the VDD signal reaches at the end of the receiver. The unit delay of both the receiver and the transmitter is the same, which equals to the delay of the delay element (DE) and the MUX delay. Consequently, if the VDD signal reaches to the stop point, the data (D0 : D3) will reach at (Q0 : Q3) at the same time. When the stop signal is equals to 1 (High) the output is latched.

The WAFT main concept is the necessity of having equal delays for both the transmitter and receiver, so any variation between the delays of the DE will increase the jitter at the deserializer and lower the performance. An NoC is fabricated in [8] using 0.18 µm technology. The interconnection between switches were the serial links with WAFT SerDes. The eight to one WAFT had a data rate 3 Gb/s at 1.8-V supply.

In [12] a prototype serial link with pulsed current-mode signaling is fabricated with a 1 GHz clock using 0.18 – µm technology with data rate of 8 Gbps. The wires on chip have impedance with inductive part. Proper design of the interconnection wires can utilize the inductive part of the wire impedance to lower the interconnection delays. The implemented links are co-planar wave-guide links which is better in performance than the micro-strip wave-guide. Table 2.4 summarize the difference between co-planar and micro-strip wave-guide.
Table 2.4: Coplanar and Microstrip Waveguides Comparison

<table>
<thead>
<tr>
<th>Coplanar</th>
<th>Microstrip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistent with layout image</td>
<td>Not consistent with layout image</td>
</tr>
<tr>
<td>Flexible to provide limited increasing of Zo</td>
<td>Not flexible to provide limited increasing of Z</td>
</tr>
</tbody>
</table>

The driver consists of eight current mode drivers and the receiver is a StrongARM sense amplifier latches. A prototype link was fabricated depending on the concept of on chip pulsed current mode signaling to achieve the speed of light latency across a transmission line using TSMC 0.18–μm technology with interconnect length of 3 mm. The link achieves 8–Gb/s data rate with a clock frequency of 1 GHz. The Driver at the transmitter and the receiver have the same clock, even though there is a clock skew between them which can be adjusted at the start up of operation.

On chip global signaling with serial link transceiver is presented in [13] and fabricated. One of the factors that leads to Inter-Symbol Interfernce (ISI) is the series resistance. This problem can be solved by making a limit for the frequency, but this makes the implementation of the link more complex. Another way of solving this problem is using very thick lines for the interconnections. The data can be transmitted through only one channel or two channels at most (differential signals) through the serial transmission. If the data consists of 8-bits, then the speed of the serial link is higher than the parallel bus by eight times. The high clock frequency is a challenge in the serial link as the receiver may require higher clock frequency to sample the data. The transceiver used is shown in Figure 2.10. It consists of PLL, Tx, Rx and self error checking block.

![Figure 2.10: Prototype SerDes for on Chip signaling [13]](image)

The transmitter shown in Figure 2.11 consists of serializer, drivers and frequency
divider. The data coming from the self test block is serialized and transmitted using the drivers.

![Diagram of transmitter](image)

Figure 2.11: The Transmitter Proposed In [13]

The receiver as shown in Figure 2.12 composed of phase interpolator and a filter to tune the phase, comparator to sample the data and First In First Out Buffer (FIFO) to deserialize the data.
The on chip channel length is 5.8 mm. The used technology is 0.13 μm. The data is transmitted with 9–Gbps data rate and the area of the chip is 3.57 mm².

In [14] a novel method to achieve the high bandwidth offered by through silicon vias (TSVs) is proposed. To get over the synchronization problem of the long interconnects, the data is transmitted using parallel buses. In 3D NoC, data can be transferred synchronously between two tiers through “3D” vias. Those “3D” vias are known as Through Silicon Vias (TSVs), but using parallel buses will lead to large area and cost. By using serial links, data can be sent through limited number of TSVs and hence solving the problem of the area and the synchronization by the large bandwidth offered by the TSVs. Serial links are used instead of parallel 3D links to gain area and increase yield by using 8:1 MUX and 1:8
DEMUX. The design used 90-nm technology and the output data rate is 2 Gbps with area gain of 74% relative to the case of parallel buses. The first case studied in this paper is a serial link that has a data rate of 2 Gb/s and consists of 8:1 MUX on a tier, 1:8 DEMUX on another tier and TSVs as shown in Figure 2.13.

![Figure 2.13: 2-Gb/s SerDes Proposed In [14]](image)

The 8:1 MUX at which the data is propagated through eight shift registers using 2 GHz clock. This circuit is implemented using 90 nm technology. The 1:8 DEMUX consists of some 1:2 DEMUXs. Each group of the 1:2 DEMUXs operates with different clock frequency. This DEMUX is implemented using 90 nm kit with a data rate of 8 Gb/s. The used TSVs have spacing and diameters of 20 µm. Table 2.5 shows the comparison between the area of serial link and parallel link.

<table>
<thead>
<tr>
<th>Component</th>
<th>Area of serial link(µm²)</th>
<th>Area of 8-bit parallel link(µm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUX</td>
<td>440</td>
<td>0</td>
</tr>
<tr>
<td>DEMUX</td>
<td>443</td>
<td>0</td>
</tr>
<tr>
<td>Frequency Divider</td>
<td>21.2</td>
<td>0</td>
</tr>
<tr>
<td>TSV number</td>
<td>4xTSV</td>
<td>18xTSV</td>
</tr>
<tr>
<td>TSV array</td>
<td>6400</td>
<td>28800</td>
</tr>
<tr>
<td>Total Link</td>
<td>7304</td>
<td>28800</td>
</tr>
<tr>
<td>Area %</td>
<td>25.36%</td>
<td>100%</td>
</tr>
</tbody>
</table>

The second case studied is 8Gb/s quasi serial link for NoCs, which consists of 32:4 MUX, 4:32 DEMUX, PLL and some frequency dividers as shown in Figure 2.14. Table 2.6 shows the comparison between the area of serial link and parallel link.
Figure 2.14: 8-Gb/s Quasi Serial Link [14]

Table 2.6: Comparison between Area of 8 Gb/s serial link and its corresponding parallel link Proposed In [14]

<table>
<thead>
<tr>
<th>Component</th>
<th>Area of serial link(µm²)</th>
<th>Area of 8-bit parallel link(µm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUX</td>
<td>1282.6</td>
<td>0</td>
</tr>
<tr>
<td>DEMUX</td>
<td>11854.4</td>
<td>0</td>
</tr>
<tr>
<td>Frequency Divider</td>
<td>21.2</td>
<td>0</td>
</tr>
<tr>
<td>TSV number</td>
<td>12×TSV</td>
<td>66×TSV</td>
</tr>
<tr>
<td>TSV array</td>
<td>19200</td>
<td>105600</td>
</tr>
<tr>
<td>Total Link</td>
<td>21689.2</td>
<td>105600</td>
</tr>
<tr>
<td>Area %</td>
<td>20.54%</td>
<td>100%</td>
</tr>
</tbody>
</table>

An asynchronous 3D-NoC router with a novel serialization scheme in the vertical directions is fabricated in TSMC 65-nm technology in [15]. The 3D-NoC router presented in this work supports two vertical links. Vertical TSVs are more costly than the horizontal links, which has its impact on reducing the number of vertical connections. A vertical serial link is used in the vertical connection and two different channels are used in the horizontal intra tier connections. The serial links reduces the number of interconnects between dies and hence increasing the bandwidth of the vertical connections. Consequently, the layout of the TSVs will be simpler. The m:1 serializer is shown in Figure 2.15. It consists of groups of MUXs and the 1:m deserializer consists of tree of DEMUXs. The used MUXs and DEMUXs are self controlled to get the minimum overhead of the controller part in the serializer and deserializer.
The design of the NoC router and serial link is implemented by place and route tools. Table 2.7 shows the area optimization between different routers.

The output data rate is 16 Gb/s and the power consumed by the serial links (up and down) is 31.2 mwatt with area gain of 57 % than parallel buses architecture.

In [16] a serial link with pulse dual-rail encoding techniques is designed using TSMC 65-nm technology. The semi-serial link composed of eight serial links. The main topic
Table 2.7: Area Optimization for the Routers proposed in [15]

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Parallel links Router</th>
<th>Serial Link with External TSV</th>
<th>Serial Link with Internal TSV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Router Logic</td>
<td>0.170 mm²</td>
<td>0.170 mm²</td>
<td>0.185 mm²</td>
</tr>
<tr>
<td>Serial Link Logic</td>
<td>0 mm²</td>
<td>0.02 mm²</td>
<td>0.02 mm²</td>
</tr>
<tr>
<td>TSV</td>
<td>0.36 mm²</td>
<td>0.085 mm²</td>
<td>0.02 mm²</td>
</tr>
<tr>
<td>Total Area</td>
<td>0.53 mm²</td>
<td>0.275 mm²</td>
<td>0.225 mm²</td>
</tr>
<tr>
<td>Area %</td>
<td>100 %</td>
<td>52 %</td>
<td>43 %</td>
</tr>
</tbody>
</table>

of this paper is the design of circuits and methods used in the serializer with the encoder which is dual rail type. The deserializer at the receiver composed of shift registers. There is an interface circuit between the serial link and the beginning of the router. The proposed serial link is shown in Figure 2.16.

The serializer consists of a clock generator, a shift register, a counter and interfacing circuits. The shift register shifts the data, using the clock generated, which consists of True Single-Phase-Clocked (TSPC) flip-flops (FFs) having small delay. The TSPC is developed to have the ability of loading parallel data. The channel between the transmitter and receiver is a distributed RLC model with a length of 8 mm. The deserializer composed of a shift register (TSPC FFs) and interfacing circuit between the receiving circuit and the deserializer. The receiving circuit is responsible for receiving and amplifying the difference of the voltage between the differential wires. The link simulated is composed of eight bit-serial links each of them is eight bit to one bit serial link. The simulation results are compared of 64-bit to one-bit serial link. The semi-serial link had a lower energy dissipation than the bit-serial link. This is due to a couple of reasons:

- The increase of the channel bandwidth is greater than the increase of the power dissipation.
- SerDes circuits in the semi serial link has lower power dissipation than the bit serial link because there is no need for the 63-bit counter and using seven-bit counter instead of it. Moreover, there is no need to replicate the clock generator and the control circuits. Instead, buffers are used to propagate the clock and control signals instead.
The semi serial link has a better performance and is more energy efficient than a bi-serial link especially for a long channel distance. As shown in Table 2.8, the throughput is doubled with the number of the parallel links, but the power and energy dissipation did not double.

<table>
<thead>
<tr>
<th>No. of parallel serial links</th>
<th>Throughput (Gbps)</th>
<th>Power Consumption(mW)</th>
<th>Energy per bit(pJ/bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.09</td>
<td>8.931</td>
<td>0.982</td>
</tr>
<tr>
<td>2</td>
<td>18.182</td>
<td>9.020</td>
<td>0.496</td>
</tr>
<tr>
<td>4</td>
<td>36.364</td>
<td>13.890</td>
<td>0.382</td>
</tr>
<tr>
<td>8</td>
<td>72.728</td>
<td>20.852</td>
<td>0.286</td>
</tr>
</tbody>
</table>

Table 2.8: Performance of 64-bit 8-mm Serial Links in [16]

The design is simulated in Cadence Analog Spectre and results in 72.72-Gbps data rate and 20.852 mwatt power consumption for eight parallel eight bit to one bit serial links with 8-mm channel.

Wave pipelined CML (WP-CML) SerDes structure is designed in [17] with each component designed in CML mode using TSMC 65-nm technology. The simulations result in 12.67-Gbps data rate and 14.3-mwatt power. The independence from the clock in asynchronous WP SerDes is a great benefit as the control signals guarantee the validity of data transmission and they are used instead of a PLL to adjust the clocks in the Tx and Rx. Even though it is a must to wait for the handshake signal and this will slow down the operation. The insertion of CML makes faster transition between the levels and require small delay between stages. The design implemented is based on pipelined wave SerDes. The design does not depend on the clock and this is considered an advantage for this design. The serializer consists of MUXs, Delay Elements (DE) and CMOS to CML blocks. The deserializer consists of MUXs, FF, DE, and CML to CMOS block. Figure 2.17 and Figure 2.18 show the block diagrams of the serializer and deserializer.

![Figure 2.17: WP CML Serializer in [17]](image-url)
The MUXs in the serializer are responsible for loading new data if the load signal is high. When the load signal is low the DE shifts the data throughout the stages. The signal (Ser Out) is sent to the deserializer that should have similar structure like the serializer to guarantee correct sampling. The deserializer uses MUXs for latching the signal. The latch-configured MUX stores the output value if the transmission is stopped. To start new data transmission, the (FF en) signal becomes high to load the received data in eight output FFs. This makes the data valid until a new data is transmitted and registered. The technique used in this paper is similar to the WAFT technique of [8]. Table 2.9 shows the comparison between different SerDes Architectures.

Table 2.9: Comparison Between Different SerDes Architectures

<table>
<thead>
<tr>
<th>Design</th>
<th>Technology(nm)</th>
<th>Speed(Gb/s)</th>
<th>Power(mW)</th>
<th>Power per bit(pJ/bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP-CMOS [18]</td>
<td>180</td>
<td>3.9</td>
<td>2.44</td>
<td>0.62</td>
</tr>
<tr>
<td>WAFT [8]</td>
<td>180</td>
<td>4.3</td>
<td>NA</td>
<td>–</td>
</tr>
<tr>
<td>CMOS-CML [19]</td>
<td>65</td>
<td>10</td>
<td>106</td>
<td>10.6</td>
</tr>
<tr>
<td>Self Timed [20]</td>
<td>65</td>
<td>12</td>
<td>15.5</td>
<td>1.29</td>
</tr>
<tr>
<td>WP-SR [21]</td>
<td>65</td>
<td>67</td>
<td>150</td>
<td>2.23</td>
</tr>
</tbody>
</table>
Chapter 3: Proposed SERDES Design

This chapter presents our proposed 6-Gb/s serial link transceiver in details. The proposed serial link transceiver is composed of a Tx and a Rx. The Tx consists of encoder, MUX, and drivers. The Rx consists of sampler, DEMUX, retimers, and decoders.

Two designs of our proposed SerDes are presented in this thesis: Design A and Design B. Design A presents SerDes that works only in the TT corner with high power consumption. Design A is presented in section 3.1. Design B presents a second version of design A after power reduction and it works for all corners. Design B is presented in section 3.5.

3.1 System Description (Design A)

The platform for the 6-Gb/s serial link is shown in Figure 3.1. It consists of two main blocks; a transmitter and a receiver.

3.2 Transmitter Design

The transmitter consists of an 8b/10b encoder to adjust the number of consecutive ones and zeros, a 10:1 multiplexer (MUX) responsible for providing the 6-Gb/s data and a driver to generate a signal suitable for transmission.

3.2.1 8b/10b Encoder

8b/10b encoder encodes the eight bits input data to ten bits parallel data using 600MHz clock, so the data rate output from this block is 600Msymbols/s. After the encoding is done, the serial data has the same number of zeros and ones for a given length which is called a Direct Current (DC) balanced data. The maximum number of continuous zeros or ones in the serial data without transitions is five (the duration of run equals to five) which helps the receiver to extract the data correctly [22]. If clock and data recovery is to be implemented. This also reduces the ISI.
The encoded data has a Running Disparity (RD) which is the difference between the number of ones and zeros in the code. The RD can take three values +2, -2 and zero [22]. If the coded vector has non-zero RD, then the next encoded data must have a different RD per the coding rules [22]. Hence each coded ten bits have two disparities: current RD which is the disparity of the encoded vector and next RD which is next disparity calculated from the current RD. RD can be divided into two categories: RD+ (RD = 1) when RD equals to -2 or zero and RD- (RD = 0) when RD equals to +2 or zero [22].

The code consists of 256 data characters (Dx.y) and 12 control characters (Kx.y). The coding method divide the eight bits data inputs into two categories of bits: three most significant bits (MSB) (x) and five least significant bits (LSB) (y) denoted by H, G, F and E, D, C, B, A (from MSB to LSB). The three bits block is encoded into four bits with names j, h, g, f. The five bits block is encoded into six bits with names i, e, d, c, b, a. The encoded four bits and six bits are the ten bits parallel outputs [22] as shown in Figure 3.2.

![Diagram of Coding Method](image)

**Figure 3.2: Coding Method**

### 3.2.2 10:1 MUX

It consists of two 5:1 MUXes and one 2:1 MUX [23],[24] as shown in Figure 3.3.
The first 5:1 MUX is used for multiplexing the even bits (D0, D2, D4, D6, D8) and the second 5:1 MUX is used for multiplexing the odd bits (D1, D3, D5, D7, D9). The output 2:1 MUX multiplexes between the two streams to generate one line consisting of the ten bits (D0-D9) using a 3-GHz clock. Figure 3.4 illustrates that each 5:1 MUX consists of five CMOS D latches. Figure 3.5 depicts the D latch design. Four CMOS 2:1 MUX (Figure 3.6) are used for selecting the data by using selection lines [23],[25].
Figure 3.5: CMOS D Latch

Figure 3.6: Transmission Gate 2:1 MUX
The clocking scheme for the 10:1 MUX is shown in Figure 3.7. The selection pulse width is equal to \((\frac{1}{3})\) nsec, D8 is transferred to the output of the D latch using 3-GHz clock. The MUX chooses D6 when SEL1 is high and chooses D8 when it is low, then the data output from the first MUX is transferred to the output of the second D latch. The second MUX chooses D4 when SEL2 is high and chooses D6 and D8 when it is low. After that the data transferred to the output of the third D latch, and SEL3 chooses D2 when it is high and chooses D4, D6 and D8 when it is low. SEL4 which is the last selection signal chooses D0 when it is high and D2, D4, D6 and D8 when it is low.

![Figure 3.7: Waveforms of the Selection Signals Used in the even 5:1 Multiplexer](image)

### 3.2.3 Design of Selection Circuit

The selection circuit consists of a 3-bit synchronous counter with reset input which counts till four. The counter consists of 3 J-K flip flops and some logic gates as shown in Figure 3.8. Figure 3.9 shows the schematic of JK FF.
Figure 3.8: Selections Circuit

Figure 3.9: JK FF Schematic
The selections signals are logic functions from the outputs of the counter as shown in Figure 3.8.

Figure 3.10 illustrate the schematics of TSPC D FF.

![Figure 3.10: TSPC D Flip Flop](image)

### 3.2.4 Driver

The driver used is a chain of inverters, each inverter is scaled by $f$ from the previous one as shown in Figure 3.11.

![Figure 3.11: Driver Schematic](image)

The number of inverters and the scaling factor $f$ are chosen to get minimum delay through the driver when loaded by the Transmission Line (TL) and the input of the receiver. The TL is modeled as a wire on metal 8 layer (M8 layer) of the TSMC digital 65-nm CMOS technology with a length of 600 $\mu$m and a width of 0.4 $\mu$m. The TL is modeled by its S parameters as illustrated in Figure 3.12.
3.2.5 Frequency Divider

The 3-GHz clock is generated from a frequency divider which is a T flip flop as shown in Figure 3.13. T FF is a JK FF with its J port is connected to its K port.

3.3 Receiver Design

The receiver contains a sampler to get the digital signal back, a 1:10 demultiplexer (DEMUX) to get the parallel data back, a retimer for proper synchronization, and a 10b/8b decoder to remove the encoding performed at the transmitter.
3.3.1 Sampler

The sampler is used to sample the received signals to extract the correct data. The operation of the sampler is divided into two main phases:

- Track phase: where the output from this phase is amplified.
- Hold Reset phase: the output here is sometimes amplified through positive feedback and sometimes the held value of the output is reset [5].

The two types of latches under consideration in this thesis are: Strong Arm (SA) and Current Mode Logic (CML). To form a flip flop from the SA latch, cascade Reset Set (RS) latch after SA latch. To form a flip flop from CML, cascade another CML latch after the first CML latch.

3.3.1.1 SA and CML flip flops

The sampler consists of a Strong Arm Flip Flop (SAFF) which consists of StrongARM latch (Figure 3.14) [26] followed by an optimized Reset-Set (R-S) latch (Figure 3.15) [27].

![StrongARM Sampler Diagram](image-url) 

Figure 3.14: StrongARM Sampler
The SAFF has advantages of:

- No static power dissipation.
- Full CMOS output levels.

The optimized RS latch has symmetric pull up and pull down paths which allows equal delays for Q and Q’. Another advantage of the optimized RS latch is that during the evaluation phase, only one pmos transistor is activated to change the output data which speeds up the operation.

The SAFF simulation results are shown in Table 3.1.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average power at clock 6 GHz</td>
<td>71.46 µW</td>
</tr>
<tr>
<td>Max. speed</td>
<td>24 Gb/s</td>
</tr>
<tr>
<td>Tcq at clock 6 GHz</td>
<td>31.2 psec</td>
</tr>
<tr>
<td>Setup time at clock 6 GHz</td>
<td>10 psec</td>
</tr>
<tr>
<td>Hold time at clock 6 GHz</td>
<td>18 psec</td>
</tr>
<tr>
<td>Input referred offset $\sigma_{vt}$</td>
<td>14.9 mV</td>
</tr>
</tbody>
</table>

### 3.3.2 1:10 DEMUX and Retimers

The 1:10 DEMUX consists of 12 D flip flops (CMOS), the first two are used to extract even and odd samples (D0, D2, D4, D6, D8) and (D1, D3, D5, D7, D9) using 3-GHz clock. The other ten flip flops select the ten samples (D0 : D9) using delayed versions of SEL4 as shown in Figure 3.16. Figure 3.17 shows the schematic of D FF.
The output of the DEMUX consists of ten bits that are passed through regenerative latches and retimers to retime the data. The ten regenerative latches are optimized RS latches [27] and the retimers are CMOS D flip flops that retime the data with 600-MHz clock.
3.3.3 Design of Selection Circuit

The selection circuit in the receiver side is the same circuit of selection number four in the transmitter side (SEL4) with a delayed versions. Figure 3.18 shows the even selection signals. Delaying the even selection signals by buffers, the odd selections can be produced.

![Diagram of Receiver Selection Circuit]

Figure 3.18: The Receiver Selection Circuit

3.3.4 10b/8b Decoder

It maps ten bits parallel data inputs to eight bits parallel data outputs [22] using 600-MHz clock. The decoding process is the opposite of the encoding process done at the transmitter.
3.4 Platform Layout of Design A

3.4.1 Transmitter Layout

The Tx layout is shown in Figure 3.19. The Area based on TSMC 65-nm technology is $36 \times 29 = 1044 \, \mu m^2$, The metal stack is M1 to M3.

![Transmitter Layout](image)

Figure 3.19: Transmitter Layout

3.4.2 Receiver Layout

The Rx layout is shown in Figure 3.20. The Area based on TSMC 65-nm technology is $47 \times 24 = 1128 \, \mu m^2$, The metal stack is M1 to M3.
3.5 Design B

Design A work only for the TT corner because the data input to the MUX are not synchronized to each other. Thus the system was enhanced to design B that works for all corners by making synchronization in the data with respect to each other. Design B is also an enhancement for design A corresponding to the reduction of the power consumption.

3.5.1 Synchronization Process

To make the data synchronized with each other, delaying elements are added as follows and as shown in Figure 3.21.

- A D latch has been added before (D6, D7 and sel1) to make it synchronized with the output of the first latch inside the 5:1 MUX.
- Two latches and a 2:1 MUX have been added before (D4, D5 and sel2)
- Three latches and two 2:1 MUXs have been added before (D2, D3 and sel3)
- Four latches and three 2:1 MUXs have been added before (D0, D1 and sel4)
The corresponding timing diagram is shown in Figure 3.22
Figure 3.22: Timing Diagram
3.5.2 Power Reduction

All CMOS-latches inside the MUX are replaced by TSPC flip flops which are faster than the CMOS flip flops and needs single ended clock only. Any CMOS flip flop is replaced by TSPC flip flop and the transmission gate 2:1 MUX is replaced by 2:1 MUX using CMOS nand gates, which needs only single ended selection signals resulting in power reduction. The new 2:1 MUX shown in Figure 3.23.

![Figure 3.23: Gate based 2:1 MUX](image)

The new 5:1 MUX is shown in Figure 3.24. The delays elements discussed in section 3.5.1 are used but not shown in Figure 3.24 for simplicity.

![Figure 3.24: New 5:1 MUX](image)

In the receiver side, all the CMOS flip flops inside the DEMUX are also replaced by TSPC flip flops using single ended clock and selection signals.
Chapter 4: Simulation Results

4.1 Circuit Level Simulation Results of Design B

An analog and digital co-simulation of the full system is carried out using the Analog Mixed Signal (AMS) simulation tool. The encoder and decoder are designed using Verilog coding while the rest of the blocks are designed at transistor level.

Each block works at a specific data rate as shown in Figure 4.1 and illustrated in Table 4.1.

![Figure 4.1: Block Diagram with Data Rates and Eye Diagrams](image)

Table 4.1: Data Rates

<table>
<thead>
<tr>
<th>Block Name</th>
<th>Data Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder</td>
<td>600 MS/s</td>
</tr>
<tr>
<td>Mux. o/p</td>
<td>6 Gb/s</td>
</tr>
<tr>
<td>Sampler</td>
<td>6 Gb/s</td>
</tr>
<tr>
<td>De-Mux o/p</td>
<td>600 MS/s</td>
</tr>
<tr>
<td>Retimers</td>
<td>600 MS/s</td>
</tr>
<tr>
<td>Decoder</td>
<td>600 MS/s</td>
</tr>
</tbody>
</table>

4.1.1 Simulation Results of the Transmitter

The data is read from file into the encoder. The output data from the encoder is passed through the MUX and the driver to be transmitted. Table 4.2 shows a sequence of input data that was read into the platform.
Table 4.2: Input and encoded data of the encoder

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Encoded Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000 (0x00)</td>
<td>0010111001 (0xB9)</td>
</tr>
<tr>
<td>00000001 (0x01)</td>
<td>0010101110 (0xAE)</td>
</tr>
<tr>
<td>00000010 (0x02)</td>
<td>0010101101 (0xAD)</td>
</tr>
<tr>
<td>00000011 (0x03)</td>
<td>1101100011 (0x363)</td>
</tr>
<tr>
<td>00000100 (0x04)</td>
<td>1101010100 (0x354)</td>
</tr>
<tr>
<td>00000101 (0x05)</td>
<td>0010100101 (0xA5)</td>
</tr>
</tbody>
</table>

Table 4.2 illustrate the main frame data we sent. This frame is sent three times respectively. Figure 4.2(a) and Figure 4.2(b) show the simulated encoder inputs and outputs.

Figure 4.3 shows the eye diagram of D0 output from the encoder.

![D0 Encoder Output Eye Diagram](image)

Figure 4.3: D0 Encoder Output Eye Diagram

The even output from the 5:1 MUX inside the 10:1 MUX should be as following (10100 0100011000100110111111000) according to the Table 4.2. Figure 4.4 shows the even output from the 5:1 MUX inside the 10:1 MUX.
Figure 4.2: Encoder Data (a) Encoder Inputs (b) Encoder Outputs

Figure 4.4: Even Output from the 5:1 MUX

The odd output from the second 5:1 MUX inside the 10:1 MUX should be as following (0 111011110011101010000100110) according to the Table 4.2. Figure 4.5 shows the odd output from the second 5:1 MUX inside the 10:1 MUX.
The final output from the 10:1 MUX should be as following (10011101000111010100101101 100110011100110101010111010010100) according to the Table 4.2. Figure 4.6 shows the final output from the 10:1 MUX.

Figure 4.6: Final Output from the 10:1 MUX

Figure 4.7 shows the eye diagram of the MUX output.
The output data from the MUX is passed through the driver to make it suitable for transmission. Figure 4.8 shows the output after the driver for the three corners (TT, FF and SS) respectively.

Figure 4.8: Output after the Driver

Figure 4.9 shows the eye diagram of the Driver output.
4.1.2 Simulation Results of the Receiver

The data passes the transmission line and at the receiver side, the data is sampled by the SA sampler, demultiplexed, retimed and decoded. Figure 4.10 shows the output after the channel.
Figure 4.11 shows the eye diagram of the channel output.

Figure 4.11: Channel output Eye diagram

Figure 4.12 shows the output after the sampler.

Figure 4.12: Output after the Sampler

Figure 4.13 shows the eye diagram of the sampler output.
The output from the sampler is passed through the DEMUX to get back parallel bits. Bit number zero should be (101101) and this frame is repeated three times. Bit number one should be (010100) and so on. Figure 4.14 shows the output after the DEMUX.

Figure 4.14: Output after the DEMUX

Figure 4.15 shows the eye diagram for D0 output from the DEMUX.
Figure 4.15: Eye Diagram of D0 Output from DEMUX

Figure 4.16 shows the output after the regenerative latches.

Figure 4.16: Output after the Regenerative Latches

Figure 4.17 shows the eye diagram for D0 output from the regenerative latches.
Figure 4.17: Eye Diagram of D0 Output from REG. Latches

Figure 4.18 shows the output after the final flip-flop which is the input to the decoder.

Figure 4.18: Output after the Final FF

Figure 4.19 shows the eye diagram for D0 output from the FF.
Figure 4.19: Eye Diagram of D0 Output from FF

Figure 4.20 shows the outputs of the decoder for the three corners (TT, FF and SS) respectively showing correct operation.

By doing XOR between the transmitted data (input data to the encoder) and the received data (output data from the decoder) it results to zero output for the three corners as
shown in Figure 4.21.

![Graphs showing XOR results for TT Corner (a), FF Corner (b), and SS Corner (c).]

Figure 4.21: XOR Result (a) TT Corner (b) FF Corner (c) SS Corner

### 4.2 Post Layout Results of Design A

The parallel encoded data input to the TX is shown in Table 4.3. Thus, the transmitted data should be (0011010110100110001) corresponding to Table 4.3. The input data and the transmitted data are shown in Figure 4.22.
Table 4.3: Parallel Encoded Data

<table>
<thead>
<tr>
<th>Encoded Input Parallel Data Names</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIN0</td>
<td>00</td>
</tr>
<tr>
<td>VIN1</td>
<td>10</td>
</tr>
<tr>
<td>VIN2</td>
<td>00</td>
</tr>
<tr>
<td>VIN3</td>
<td>01</td>
</tr>
<tr>
<td>VIN4</td>
<td>11</td>
</tr>
<tr>
<td>VIN5</td>
<td>10</td>
</tr>
<tr>
<td>VIN6</td>
<td>01</td>
</tr>
<tr>
<td>VIN7</td>
<td>00</td>
</tr>
<tr>
<td>VIN8</td>
<td>01</td>
</tr>
<tr>
<td>VIN9</td>
<td>11</td>
</tr>
</tbody>
</table>

Figure 4.22: Encoded Input Parallel Data (VIN0:VIN9) and Serial Transmitted Data

The eye diagram of the transmitted data is shown in Figure 4.23
The output data from the FF which is the final stage in the receiver is shown in Figure 4.24.

4.3 CUSPARC NoC Clock Skew results of Design B

CUSPARC [28], [29] is the first Egyptian embedded intellectual-property processor. To boost its performance, a NoC based on this processor core was designed [1]. Figure 4.25 shows the NoC for 16 cores of CUSPARC. The data path is between any two routers (R) is shown as a solid line. The paths of the clock tree from the Phase Locked Loop (PLL) are shown as dotted lines.
As shown in Figure 4.26, the distance between any two neighboring cores is constant. For CUSPARC NoC design [1] targeting the TSMC digital 65-nm CMOS technology, this distance is 600 \( \mu \text{m} \). Moreover, having the PLL in the center of the NoC as shown in Figure 4.26 results in different path length for the clock distribution. This necessitates proper synchronization.
When the cores are on opposite sides of the PLL, for example CUSPARC cores (2) and (3), the paths are of equal length. Ideally, this results in zero skew between the transmitter and receiver. For 6-Gb/s data rate, these paths are 600 $\mu$m long resulting in a delay of 1.43 psec with no skew.

When the cores are on the same side of the PLL, for example CUSPARCs (1) and (2), or (3) and (4), the paths are of different length. This results in a clock skew that should be accounted for. In our design, the paths from PLL to core 3 and 4 are 600 $\mu$m and 1065 $\mu$m long and the clock delays are 1.43 psec and 5.12 psec, respectively. When designing the serial link, the receiver should sample the data correctly whether a skew is present or not.

The first case tested is when CUSPARC (2) sends data to CUSPARC (3) or vice versa. When CUSPARC (3) sends data to CUSPARC (4) or CUSPARC (2) to CUSPARC (1) this is the second case. The third case is when CUSPARC (4) sends to CUSPARC (3) or CUSPARC (1) sends to CUSPARC (2). All these cases with the clock delays are summarized in Table 4.4.

For the NoC of CUSPARC processor, the architecture proposed is tested for different clock phases for the transmitter and the receiver. By shifting the clock at the receiver side by a variable delay ($v$) from -166.66 psec to 166.66 psec, the maximum clock skew tolerable is up to $\pm 36\%$ in TT corner corresponding to the equation 4.1. Where V1 and V2 are the values of the variable V for positive and negative sweeps.
Table 4.4: Clock Skew Test Cases

<table>
<thead>
<tr>
<th>Cases</th>
<th>Transmitter clock delay</th>
<th>Receiver clock delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSPARC(2) sends to (3)</td>
<td>1.43 psec</td>
<td>1.43 psec</td>
</tr>
<tr>
<td>CUSPARC(3) sends to (4)</td>
<td>1.43 psec</td>
<td>5.12 psec</td>
</tr>
<tr>
<td>CUSPARC(4) sends to (3)</td>
<td>5.12 psec</td>
<td>1.43 psec</td>
</tr>
</tbody>
</table>

Tolerable clock skew percentage $= \frac{V_1 + |V_2|}{2 \times 166.66667} \%$ (4.1)

This maximum tolerable clock skew is much larger than the clock skew due to different wire delays from PLL to different routers.

The clock skew results for typical typical corner shown in Table 4.5.

Table 4.5: Clock Skew results for Typical Typical Corner

<table>
<thead>
<tr>
<th>Transmitter clock delay</th>
<th>Receiver clock delay</th>
<th>clock skew percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.43 psec</td>
<td>1.43 psec</td>
<td>± 35.7 %</td>
</tr>
<tr>
<td>1.43 psec</td>
<td>5.12 psec</td>
<td>± 36.0 %</td>
</tr>
<tr>
<td>5.12 psec</td>
<td>1.43 psec</td>
<td>± 34.8 %</td>
</tr>
</tbody>
</table>

The clock skew results for the slow slow corner shown in Table 4.6.

Table 4.6: Clock Skew results for Slow Slow Corner

<table>
<thead>
<tr>
<th>Transmitter clock delay</th>
<th>Receiver clock delay</th>
<th>clock skew percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.43 psec</td>
<td>1.43 psec</td>
<td>± 28.5 %</td>
</tr>
<tr>
<td>1.43 psec</td>
<td>5.12 psec</td>
<td>± 28.2 %</td>
</tr>
<tr>
<td>5.12 psec</td>
<td>1.43 psec</td>
<td>± 29.1 %</td>
</tr>
</tbody>
</table>

The clock skew results for the fast fast corner shown in Table 4.7.

Table 4.7: Clock Skew results for Fast Fast Corner

<table>
<thead>
<tr>
<th>Transmitter clock delay</th>
<th>Receiver clock delay</th>
<th>clock skew percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.43 psec</td>
<td>1.43 psec</td>
<td>± 45.0 %</td>
</tr>
<tr>
<td>1.43 psec</td>
<td>5.12 psec</td>
<td>± 44.7 %</td>
</tr>
<tr>
<td>5.12 psec</td>
<td>1.43 psec</td>
<td>± 43.5 %</td>
</tr>
</tbody>
</table>

4.4 Power Distribution of Design B

The total average power consumed in the transceiver equals to 6.9 mW (1.15 pJ/bit) which points to the high interconnect power efficiency of our on-chip SerDes design. The power distribution of each block illustrated in the Table 4.8.
Table 4.8: Power Distribution

<table>
<thead>
<tr>
<th>Blocks</th>
<th>Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUX</td>
<td>0.19</td>
</tr>
<tr>
<td>Driver 1</td>
<td>0.07</td>
</tr>
<tr>
<td>Driver 2</td>
<td>0.07</td>
</tr>
<tr>
<td>Sampler</td>
<td>0.07</td>
</tr>
<tr>
<td>DEMUX</td>
<td>0.07</td>
</tr>
<tr>
<td>Total Inverters</td>
<td>0.02</td>
</tr>
<tr>
<td>Total Latches and FFs</td>
<td>0.04</td>
</tr>
<tr>
<td>Selection circuits and its delays blocks at Tx</td>
<td>1.56</td>
</tr>
<tr>
<td>Selection circuits and its delays blocks at Rx</td>
<td>1.32</td>
</tr>
<tr>
<td>Frequency divider at Tx</td>
<td>0.55</td>
</tr>
<tr>
<td>Frequency divider at Rx</td>
<td>0.32</td>
</tr>
<tr>
<td>Delays blocks for data</td>
<td>0.27</td>
</tr>
<tr>
<td>Delays blocks for Rx clock</td>
<td>2.37</td>
</tr>
<tr>
<td><strong>Total power</strong></td>
<td><strong>6.9</strong></td>
</tr>
</tbody>
</table>

4.5 Comparison to Parallel Buses NoC Architecture

CUSPARC single core area with parallel buses transmission equals to 0.21 mm$^2$ [1]. L1 length in Figure 4.27 equals to 0.46 mm and L2 length equals to 0.032 mm. By adding the area of SerDes after the layout which is equal to 2172 µm$^2$, the area of the single core becomes 0.21217 mm$^2$. After converting 64 parallel lines between the routers (32 data inputs and 32 data outputs of the router) to four parallel lines (two input and output differential lines of the SerDes), L2 reduced by 4/64. Consequently, the routing area decreases from 0.489 mm$^2$ to 0.0295 mm$^2$ as shown in Figure 4.27. Thus the routing area decreases by 93.96%.
Figure 4.27: Area reduction [1]

4.6 Comparison With Related Works

Table 4.9 shows a comparison between this work (Design B) and previous works.
Table 4.9: SerDes Comparison

<table>
<thead>
<tr>
<th>paper number/specs</th>
<th>[8]</th>
<th>[12]</th>
<th>[13]</th>
<th>[14]</th>
<th>[15]</th>
<th>[16]</th>
<th>[17]</th>
<th>This Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Summary</td>
<td>A new serial-ization technique WAFT serdes used in NoC, fabricated</td>
<td>Prototype serial link employing pulsed current-mode signaling fabricated</td>
<td>Serial link transceiver for on-chip fabricated</td>
<td>Synchronous parallel 3D links TSV</td>
<td>Asynchronous 3D-NoC router fabricated</td>
<td>Self-timed semi serial link</td>
<td>WP-CML SerDes structure</td>
<td>A serial link for CUS-PARC NoC processor</td>
</tr>
<tr>
<td>Technology</td>
<td>0.18 (\mu m)</td>
<td>0.18 (\mu m)</td>
<td>0.13 (\mu m)</td>
<td>90 nm</td>
<td>TSMC 65 nm</td>
<td>TSMC 65 nm</td>
<td>TSMC 65 nm</td>
<td>TSMC 65 nm</td>
</tr>
<tr>
<td>Data Rate (Gb/s)</td>
<td>3</td>
<td>8</td>
<td>9</td>
<td>2</td>
<td>16</td>
<td>9.09 (one link)</td>
<td>12.67</td>
<td>6</td>
</tr>
<tr>
<td>Energy per bit (pJ/bit)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.95</td>
<td>0.98 (one link)</td>
<td>1.12</td>
<td>1.15</td>
</tr>
<tr>
<td>Area of SerDes ((\mu m^2))</td>
<td>-</td>
<td>-</td>
<td>177500</td>
<td>460.681</td>
<td>19000</td>
<td>33124</td>
<td>-</td>
<td>2172</td>
</tr>
</tbody>
</table>

The energy per bit for our work equals to 1.15 pJ/bit, which is smaller than [15] and nearly equals to [17]. Our design features a very attractive combination of low power per bit plus a small area (small cost).
Chapter 5: Conclusion and Future Work

5.1 Conclusion

The design of 6-Gb/s serial link for Many-Core CUSPARC processors, arranged in a 2D mesh architecture, is presented. The design is composed of a Tx and a Rx. The Tx consists of encoder, MUX and drivers. The Rx consists of sampler, DEMUX, retimers and decoders. The design targets TSMC digital 65-nm CMOS technology with 1.2-V supply. The design is simulated using AMS simulation tool. The simulation results in lower NoC area, high percentage of allowable clock skew between the transmitter and the receiver, and low power consumption. Our design features a very attractive combination of low power per bit plus a small area (small cost).

5.2 Future Work

The future work is to make the layout for design B to work for all corners and fabricate it. To enhance the maximum tolerable clock skew for design B, a clock and data recovery block may be introduced.
References


الملخص
مقارنة بنقل البيانات على التوازي، يتميز النقل المتتالي للبيانات بعدم منافذ صغيرة، مناعة عالية ضد التداخل، استهلاك منخفض للطاقة ومساحة أكثر. بالتزامن مع الاتجاه السائد حاليا لاستبدال الناقلات المتوازية بوصلات متتالية ذات سرعة عالية، يوجد اتجاه سائد آخر لاستبدال المعالج المتعدد النووي ذو الذاكرة المشتركة بشبكة متتالية

تُقدم هذه الرسالة وصلة متتالية سريعة للشبكات ذات الأبوية الكثيرة، الواصلة المتتالية المفترضة تتكون من مُسلسل و مُوازي (SerDes). المسلسل يحتوي على مُكود 8/10بت، مُعدل إرسال1:1 وسائق

والموؤژى يحتوي على أخذ العينات. مُعدل استقبال10بت، دوائر لف الشارة مع تجديدها ودائره فك الكود 8/10بت. تم وضع تصميم باستخدام تكنولوجيا CMOS 65 نانومتر الرقمية من شركة TSMC وتأتي تصميم الواصلة متتالية TES Des

الإعداد على رقائق مكونة من 16 نواة، كل منها عبارة عن معالج جامعة القاهرة سبارك (CUSPARC).

تعمل الوصلة متتالية المقدمة بسرعة 6 جيجا بت في الثانية. أمكن باستخدام هذه الوصلات المتتالية في بناء الشبكة المعمارية - ثنائية الإعداد ساقية الذكر - تقليد مساحة روابط الشبكة على الراكة بنسبة 93.96% من المساحة اللازمة لروابط 32 بت متوازية. التوصيات بين الأبوية تعتمد باستخدام الطبقة المعدنية رقم 8 في تكنولوجيا CMOS 65 نانومتر الرقمية من شركة TSMC. أثبتت المحاكاة عمل هذه الشبكة قدرتها على تحمل نسبة انحراف للساعة تصل إلى ±36% بين المرسل والمستقبل. تستهلك هذه الواصلة المتتالية طاقة 6.9 مبلي واط أي 1.15 حوالى بيكلو جو للنقل بت واحدة.
مهندة:
صفاء أحمد محمد عبد الفتاح
تاريخ الميلاد:
1989/11/22
 الجنسية:
مصرية
تاريخ التسجيل:
2012/10/1
تاريخ المنح:
2016/1
القسم:
هندسة الالكترونيات و الاتصالات الكهربية
درجة:
ماجستير العلوم
المشرفون:
أ.د. سراج الدين السيد حبيب
د. سامح عاصم إبراهيم
المتحدون:
أ.د. محمد أمين دسوقى (المتحدن الخارجي)
أ.د. محمد رياض النغيمي (المتحدن الداخلي)
أ.د. سراج الدين السيد حبيب (المشرف الرئيسي)
عنوان الرسالة:
مرسل ومستقبل لوصة متتالية تعمل عند 6 جيجابت في الثانية للشبكات على الرقائق
الكلمات الدالة:
مسلسل/موزى، شبكات على الرقائق، معالج جامعة القاهرة سبارك (CUSPARC)
ملخص الرسالة:
تم تصميم رابط 6 جيجابت / ثانية لوصة متتالية مناسبة للشبكات على الرقائق. تم استعمال هذه الوصلات في شبكة معمارية ثنائية الابعاد تحتوي أنوية كثيرة، كل منها عمارة عن معالج جامعة القاهرة سبارك. تم التصميم باستخدام تكنولوجيا CMOS 65 نانومتر مع مصدر جهد 1.2 فولت. استخدم الوصلات المتتالية قل صناعة روابط الشبكة على الرقائق بنسبة 93.96% مقارنة بالمساحة اللازمة لبناء وصلات بيانات 32 بت متوازية. أظهرت نتائج محاكاة عمل هذه الوصلات المتتالية قدرتها على تحمل انحراف للساعة بين المرسل والمستقبل يصل إلى ±36% من زمن دورة الساعة clock period. تستهلك الوصلة المتتالية 6.9 ملي وات أي حوالي 1.15 بيقو جول لكل بت.
مرسل ومستقبل لوصلة متاتالية تعمل عند 6 جيجابيت في الثانية للشبكات على الرقائق

إعداد
صفاء أحمد محمد عبد الفتاح

رسالة مقدمة إلى كلية الهندسة - جامعة القاهرة
كجزء من متطلبات الحصول على درجة ماجستير العلوم
في هندسة الإلكترونيات والاتصالات الكهربية

يعتمد من لجنة الممتحنين:

الاستاذ الدكتور: سراج الدين السيد حبيب
المشرف الرئيسي

الاستاذ الدكتور: محمد رياض الغنيمى
الممتحن الداخلي

الاستاذ الدكتور: محمد أمين دسوقى
الممتحن الخارجي

قسم هندسة الإلكترونيات والاتصالات الكهربية, كلية الهندسة,
جامعة عين شمس)

كلية الهندسة - جامعة القاهرة
الجيزة - جمهورية مصر العربية

2016
مرسل ومستقبل لوصلة متتالية تعمل عند 6 جيجابيت في الثانية للشبكات على الرقائق

إعداد
صفاء أحمد محمد عبد الفتاح

رسالة مقدمة إلى كلية الهندسة - جامعة القاهرة
كجزء من متطلبات الحصول على درجة ماجستير العلوم في هندسة الإلكترونيات والاتصالات الكهربية

تحت إشراف

أ.د. سراج الدين السيد حبيب
أستاذ دكتر
قسم هندسة الإلكترونيات والاتصالات الكهربية
كلية الهندسة - جامعة القاهرة

أ.د. سامح عاصم إبراهيم
مدرس
قسم هندسة الإلكترونيات والاتصالات الكهربية
كلية الهندسة - جامعة عين شمس

كلية الهندسة - جامعة القاهرة
الجيزة - جمهورية مصر العربية
2016
مرسل ومستقبل لوصة متتالية تعمل عند 6 جيجابيت في الثانية للشبكات على الرقائق

اعداد
صفاء أحمد محمد عبد الفتاح

رسالة مقدمة إلى كلية الهندسة - جامعة القاهرة
كجزء من متطلبات الحصول على درجة ماجستير العلوم
في
هندسة الالكترونيات والاتصالات الكهربائيه

كلية الهندسة - جامعة القاهرة
الجيزة - جمهورية مصر العربية
2016