

**PRELAYOUT DESIGN OF CONFIGURABLE SERDES FOR HIGH
SPEED SIGNALING IN MULTIDIE INTERCONNECT**

By

CHIEW CHONG GIAP

**A Dissertation submitted for partial fulfilment of the requirement for
the degree of Master of Science (Microelectronic Engineering)**

August 2016

Acknowledgement

First and foremost, I would like to convey my heartfelt gratitude to my supervisor, Dr Patrick Goh Kuan Lye, for giving me the opportunity to work under him. His sound advice and expert guidance throughout the dissertation is invaluable.

Special thanks to my company supervisors, CY Cheoh and KH Choe for being thoughtful throughout the whole year. I would also like to express gratitude to my colleagues YK Lum and HC Ng for sharing their expertise and experience. Furthermore, I would like to thank my family, as this journey would have not been possible without their remarkable encouragement and patience.

Last but not least, I reserve my sincere appreciation for the people that are not mentioned above, who had offered advice in this research and enabled the thesis to be done on time.

Table of Contents

Acknowledgement.....	ii
Table of Contents	iii
List of Figures	vi
List of Tables.....	viii
List of Abbreviations.....	ix
Abstrak	xi
Abstract	xii
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.1.1 On Chip Communication	2
1.1.2 Parallel Link	3
1.1.3 Serial Link	4
1.1.4 Serialism over Parallelism	5
1.1.5 Shortcomings of Parallel Link and Advantages of Serial Link.....	5
1.1.6 Skew and Jitter	6
1.2 Problem Statement	8
1.3 Objective	9
1.4 Project Scope	9
1.5 Thesis Outline.....	10
CHAPTER 2 LITERATURE REVIEW	11
2.1 Overview	11
2.2 Serial Transmission Clocking Schemes	11
2.2.1 Mesochronous Clocking	11
2.2.2 Plesiochronous Clocking.....	12
2.2.3 Asynchronous Clocking	13
2.3 SerDes Conventional Implementation.....	14
2.3.1 Overview of Conventional SerDes Design	14
2.3.2 The Transmitter	16
2.3.3 The Interconnect.....	16
2.3.4 Serializer	17

2.3.5	The Receiver	18
2.3.6	Deserializer	19
2.4	Challenges in Serial link.....	20
2.5	Transmission with Clock Data Recovery	21
2.6	The Multi-Level Transmission Scheme	22
2.7	Chapter Summary	24
CHAPTER 3 METHODOLOGY		26
3.1	Overview	26
3.2	Workflow for the Clockless SerDes Interconnect Design.....	27
3.3	Simulation and Test Bench Setup	28
3.4	The Test Bench.....	30
3.5	Block Diagram of the Proposed Clockless SerDes Design	31
3.6	Flow Chart of the Proposed Clockless SerDes Design	32
3.7	Configurable Clockless SerDes Interconnect Sub-Blocks Implementation	35
3.7.1	Increasing the Data Sampling Rate	36
3.7.2	Three-Level-Transmission Scheme	37
3.7.3	The Serializer	38
3.7.4	The Deserializer	42
3.7.5	Three-Level Signal Encoder	45
3.7.6	Three-Level Signal Decoder	49
3.8	System Level Implementation.....	51
3.8.1	System Level Implementation for SiP with 80 Bit Transmission.....	51
3.8.2	System Level Implementation for SiP with 40 Bit Transmission.....	52
3.9	Slew Rate Analysis.....	52
3.10	Duty Cycle Distortion	53
3.11	Timing Analysis	54
3.12	RMS Current	57
3.13	Chapter Summary	57
CHAPTER 4 RESULTS AND DISCUSSION		60
4.1	Overview	60
4.2	Simulation Strategy	60
4.3	Simulation Results.....	62

4.4	System Implementation of 80-to-20 Interconnect for 40 GTps Link.....	65
4.5	System Implementation of 40-to-20 Interconnect for 20 GTps Link.....	70
4.6	Slew Rate Analysis.....	74
4.7	Duty Cycle Distortion	79
4.8	Timing Analysis	80
4.9	RMS Current	82
4.10	Design Summary and Overall Discussion.....	84
CHAPTER 5 CONCLUSION AND RECOMMENDATIONS		87
5.1	Overview	87
5.2	Conclusions	87
5.3	Future Recommendations.....	88
REFERENCE.....		90
APPENDIX A		93
APPENDIX B		94
APPENDIX C		95
APPENDIX D		96
APPENDIX E		97
APPENDIX F.....		101
APPENDIX G		102

List of Figures

Figure 1.1: Block diagram of parallel multidie interconnect and serial multidie interconnect	3
Figure 1.2: Parallel link interconnect for multi-core signaling	4
Figure 1.3: Clock skew in sequentially adjacent registers	7
Figure 1.4: Clock jitter in a periodic clock pulse	8
Figure 2.1: Mesochronous clocking serial link	12
Figure 2.2: Plesiochronous clocking serial link	13
Figure 2.3: Asynchronous clocking serial link	14
Figure 2.4: Overall block diagram of conventional SerDes system in [11].....	15
Figure 2.5: Block diagram of the conventional SerDes transmitter [11]	16
Figure 2.6: Block diagram of serializer [18]	18
Figure 2.7: Block diagram of the conventional SerDes receiver [11].....	19
Figure 2.8: Block diagram of deserializer [18]	20
Figure 3.1: Block diagram of QDR clockless SerDes interconnect transceiver	27
Figure 3.2: HSPICE simulation flow	29
Figure 3.3: HSPICE simulation setup (Spicedeck)	30
Figure 3.4: The test bench in HSPICE for functional simulation	31
Figure 3.5: Block diagram of the proposed clockless SerDes design	32
Figure 3.6: Flow Chart of the proposed SerDes design	34
Figure 3.7: QDR sampling waveform.....	36
Figure 3.8: Comparison of SDR, DDR and QDR.....	37
Figure 3.9: Three-level-transmission signal encoding scheme	38
Figure 3.10: Schematic of the serializer.....	39
Figure 3.11: Data output waveform in 4-to-1 serialization.....	41
Figure 3.12: Data output waveform in 2-to-1 serialization.....	42
Figure 3.13: Schematic of the deserializer.....	43
Figure 3.14: Flowchart of embedding clock and data to three-level signal	46
Figure 3.15: Schematic of three-level signal encoder.....	47
Figure 3.16: Output drive arrangement at low clock pulse	47
Figure 3.17: NMOS transistor arrangement as passgate	48
Figure 3.18: Small signal model for NMOS	48
Figure 3.19: Schematic of three-level decoder	50
Figure 3.20: Three-level clock recovery and data extraction scheme.....	50
Figure 3.21: Transition and pulse parameter of a clock pattern.....	53
Figure 3.22: Duty cycle parameters of a pulse waveform	54
Figure 3.23: Clock domain crossing timing path	56
Figure 3.24: Clock domain crossing timing path data launch and capture	56
Figure 3.25: Block diagram of multidie interconnect system implementation with 80 bit signal density.....	58

Figure 3.26: Block diagram of multidie interconnect system implementation with 40 bit signal density.....	59
Figure 4.1: Simulation waveform of 4-to-1 serialization.....	62
Figure 4.2: Simulation waveform of 2-to-1 serialization.....	63
Figure 4.3: Eye diagram of signals in 4-to-1 serialization mode	64
Figure 4.4: Eye diagram of signals in 2-to-1 serialization mode	65
Figure 4.5: Top level schematic of 80-to-20 interconnect system implementation	67
Figure 4.6: Schematic of 80-to-20 interconnect system implementation (TX)	68
Figure 4.7: Schematic of 80-to-20 interconnect system implementation (RX)	69
Figure 4.8: Simulation results of 80-to-20 interconnect system implementation	70
Figure 4.9: Top level schematic of 40-to-20 interconnect system implementation	71
Figure 4.10: Schematic of 40-to-20 interconnect system implementation (TX)	72
Figure 4.11: Schematic of 40-to-20 interconnect system implementation (RX)	73
Figure 4.12: Simulation results of 40-to-20 interconnect system implementation	74
Figure 4.13: Deserialized data rise time in 4-to-1 serialization mode	75
Figure 4.14: Deserialized data fall time in 4-to-1 serialization mode.....	75
Figure 4.15: Deserialized data rise slew rate in 4-to-1 serialization mode	76
Figure 4.16: Deserialized data fall slew rate in 4-to-1 serialization mode.....	76
Figure 4.17: Deserialized data rise time in 2-to-1 serialization mode	77
Figure 4.18: Deserialized data fall time in 2-to-1 serialization mode.....	77
Figure 4.19: Deserialized data rise slew rate in 2-to-1 serialization mode	78
Figure 4.20: Deserialized data fall slew rate in 2-to-1 serialization mode.....	78
Figure 4.21: Deserialized data duty cycle distortion in 4-to-1 serialization mode	80
Figure 4.22: Deserialized data duty cycle distortion in 2-to-1 serialization mode	80
Figure 4.23: Simulation waveform of setup analysis.....	81
Figure 4.24: Simulation waveform of hold analysis	82
Figure 4.25: RMS current in 4-to-1 serialization mode across PVT.....	83
Figure 4.26: RMS current in 2-to-1 serialization mode across PVT.....	84

List of Tables

Table 1.1: Comparison between parallel link and serial link.....	5
Table 2.1: Design comparison.....	24
Table 3.1: Input vector configuration of the serializer.....	40
Table 3.2: Output vector configuration of the serializer	40
Table 3.3: Input vector configuration of the deserializer.....	44
Table 3.4: Output vector configuration of the deserializer	44
Table 4.1: Simulation conditions	61
Table 4.2: Specification on output slew rate.....	75
Table 4.3: Area distribution of sub-blocks in the design	84
Table 4.4: Power consumption of the design	85
Table 4.5: Design comparison of this work against previous proposed designs	86

List of Abbreviations

Abbreviation	Meaning
ASIC	Application Specific Integrated Circuit
ASSP	Application Specific Standard Product
CDR	Clock and Data Recovery
DDR	Double Data Rate
DLL	Delay Locked Loop
ECO	Engineering Change Order
FPGA	Field Programmable Gate Array
HSSI	High Speed Serial Interface
IC	Integrated Circuit
IO	Input/output
IP	Intellectual Property
LSB	Least Significant Bit
MLT	Multi-Level Transmission
MSB	Most Significant Bit
NoC	Network-on-Chip
NRZ	Non-Return-to-Zero
PLL	Phase Locked Loop
PnP	Plug-and-Play
PRBS	Pseudo-Random Bit Sequence
QDR	Quad Data Rate

RMS	Root-Mean-Square
RX	Receiver
SDR	Single Data Rate
SerDes	Serializer and Deserializer
SiP	System in Package
SoC	System-on-Chip
TX	Transmitter

REKA BENTUK PRA SUSUN ATUR SERDES MAMPU DIKONFIGURASI UNTUK ISYARAT BERKELAJUAN TINGGI ANTARA BERBILANG DAI

Abstrak

Dengan kemajuan teknologi pemprosesan, saiz transistor menjadi lebih kecil dan lebih litar elektrik intelektual (IP) disepadukan ke dalam litar bersepadu (IC). Dalam usaha untuk menampung fungsi kompleks serta meningkatkan prestasi litar bersepadu, pereka IC telah menggalakkan integrasi pelbagai dai dalam cip tunggal. Komunikasi antara dai memerlukan rangkaian komunikasi pada cip yang memerlukan ruang reka bentuk yang intensif. Dengan meningkatkan ketumpatan transistor dan mengecilkan cip, rangkaian antara dai tidak mencukupi akibat trend perekaan IC yang semakin meningkat dalam menuntut rangkaian lebih lebar dan logam trek untuk sambungan antara dai. Penukaran bit data selari ke dalam aliran data siri untuk komunikasi antara dai dapat mengurangkan bilangan wayar diperlukan untuk disambung. Penghantaran siri segerak memerlukan dimensi reka bentuk yang besar dan blok tambahan yang memerlukan lebih kuasa untuk penyegerakan isyarat data dan isyarat jam. Ini dapat dielakkan dengan pelaksanaan skim pengisyarat tetapan masa sendiri yang mengelakkan penghantaran isyarat jam dalam wayar yang berasingan. Kajian ini bertujuan untuk membina sistem pengisyratan SerDes tanpa jam yang boleh diguna semula, dapat diskalakan dan dikonfigurasi sebagai rangkaian komunikasi antara berbilang dai. Reka bentuk yang dibina dapat mencapai kadar data 2 Gbps, mengambil ruang reka bentuk kecil dengan menggunakan 308 transistor, berkeluasan $38.17 \mu\text{m}^2$ dalam IC dan penggunaan kuasa yang serendah 1.10 mW.

PRELAYOUT DESIGN OF CONFIGURABLE SERDES FOR HIGH SPEED SIGNALING IN MULTIDIE INTERCONNECT

Abstract

As the process technology advances, transistor size shrinks and more intellectual properties (IPs) are integrated onto chip. In order to accommodate the current complex functionalities as well as improving the performance of design, integrated circuit (IC) architecture has encouraged the integration of multiple die on a single chip. Communication between die requires full network-on-chip (NoC) which is area intensive. In deep sub-micron process nodes, high speed signaling between multiple die becomes one of the main challenges in multidie chip design. Methods to increase the routability have been proposed as the use of parallel interconnect appears to be the bottleneck of high speed multidie communication. Conversion of parallel data bits into serial data streams before transmission effectively reduced the number of wires required for the interconnect. Synchronous serial transmission requires large design dimension and power hungry auxiliary blocks for synchronization between the transmitted data and clock signals. This is avoided with the implementation of self-timed transmission scheme which eliminates the need to transmit the clock signal in a separate wire. This research is conducted to develop a reusable, scalable and configurable clockless version of SerDes system as the interconnect between multiple die. The proposed design achieves a data rate of 2 Gbps small area $38.71 \mu\text{m}^2$ with architectural simplicity with 308 transistor count and low power consumption of 1.10 mW.

CHAPTER 1

INTRODUCTION

1.1 Background

Integrated circuit (IC) semiconductor chips are designed to serve many functions such as microprocessor, transceiver or memory. The advantage of ICs is small in size as they are made in silicon wafer which contains up to millions of transistors. Programmable and application specific IC such as FPGA and application specific integrated circuit (ASIC) or application specific standard product (ASSP) respectively are great design challenges as the development of ICs is moving towards multi-die and higher density devices. As technology node advances, the process of fabrication shrinks transistor size in IC, the chip density becomes higher and higher. The chip design becomes more complex when more intellectual properties (IPs) are integrated on the same chip. In some high density IC, the multi-die approach is implemented where multiple cores are integrated on a single chip. The cores interact with each other and if the IPs are located on different die or core, the IPs require interconnection for communication. The network-on-chip (NoC) communication needs an efficient communication infrastructure to achieve lower input/output (IO) power, high speed and reliable passing of data from one core to another [1]. As chip design gets more complex, the IP's bandwidth increase and requires greater bus width to incorporate the data carrying capacity. Wider interconnection needs more interconnect resources. With the growing number of IP and bandwidth, to achieve more efficient use of interconnect resources, parallel interfacing has become the bottleneck to high speed communication.

1.1.1 On Chip Communication

The recent IC design implementations are heading towards high density and small sized transistor down to sub-micron and beyond. IC with large number of transistors has driven the realization of multicore chip such as SoC. This makes the design not only has block level communication problem, but also created cross die communication issues. The primary element to achieve communication at high frequency and bit rate between die will be an on chip interconnect with low power consumption [2].

On chip signaling can be implemented with two interconnect architectures as shown in Figure 1.1. The parallel link has interconnect wire count as much as the signal bus width. On the other hand, the serial link serializes the parallel data into a high frequency signal data stream before transmission. The transceiver and the transmission link must have a generic structure to support wide and range of blocks with different signal specifications.

As chip designs come in variants which offere different design scale, the interdie communication interface must offer scalability to support across design scales. Configurable transceiver is preferred so that it is adaptable and flexible to plug and play across different designs. A robust interconnect interface system can be used in the integration of different subsystems on different die. As IC technology moves towards sub-micron designs, the focus is to reduce the complexity of the transceiver circuit and minimize area and power consumption with high speed data transmission.

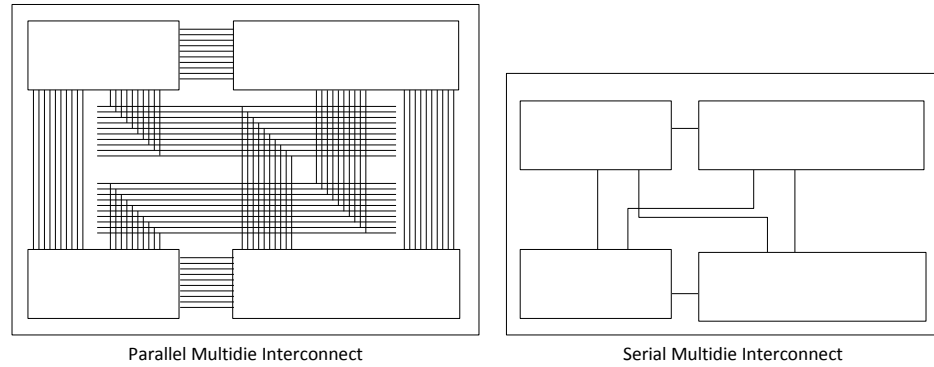


Figure 1.1: Block diagram of parallel multidie interconnect and serial multidie interconnect

1.1.2 Parallel Link

In simpler designs, the most straight forward way of data transmission was bus-based interconnect where data are sent with set of wires in parallel. Therefore, this implies more conductor needed to build the interconnect. Figure 1.2 shows parallel link interconnect among multiples cores. For the transmission of data to be synchronized between the sender core and recipient core, the clock signal is also sent in a separate wire in the parallel link. Timing for parallel signaling is referenced to the transmitted clock and is therefore synchronous. This signaling scheme has less complex transceiver architecture, as no serialization or encoding is required. As modern IC is running at high frequency, timing of the parallel transmission becomes crucial as it is hard to guarantee all the signals to arrive at the receiver simultaneously. A major drawback of parallel link is the crosstalk in the transmission lines. Higher frequency will give more pronounced crosstalk. Also, the trade off in parallel signaling scheme is the routing resources and chip area to build multiple transmission links which contributes greatly to production cost. Commonly used in block level and smaller scale designs, parallel communication are not preferred in multi-core designs for the drawbacks discussed in section 1.1.4.

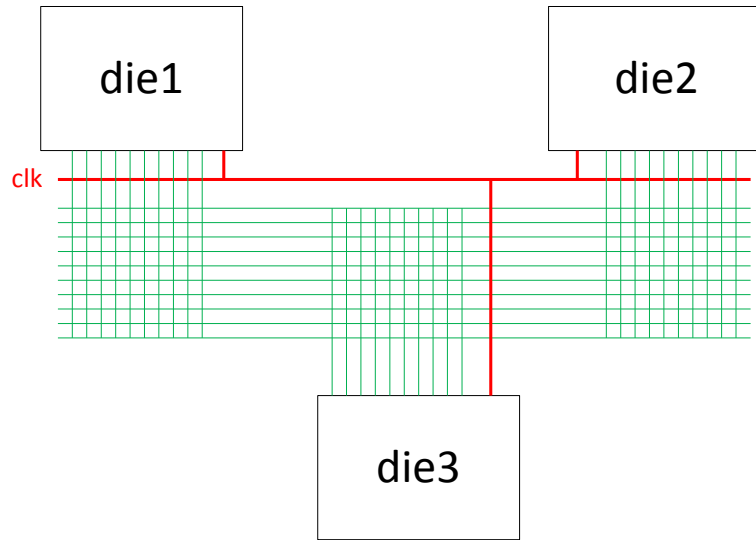


Figure 1.2: Parallel link interconnect for multi-core signaling

1.1.3 Serial Link

A high speed serial link implements a serializer to convert low frequency parallel data and serialize it into a stream of data with a higher frequency synthesized clock. The serialized data is transmitted through the transmission link to the receiver and converted back to parallel data at the original frequency with a deserializer. This implementation of serial signaling is known as a SerDes [3].

To achieve the same bit rate, an N bit serializer will transmit data with a clock N times faster than the N bit parallel link. Serial link requires N times fewer interconnect between cores and therefore reducing the requirement in routing resources and space of the interconnect. There are two types of serial link signaling techniques and will be discussed on the following section.

1.1.4 Serialism over Parallelism

Serial link has gradually dominated over parallel link in multi-core high speed communications. As discussed in 1.1.1, while transistors scale down, high density chip design is moving towards the trend of multi-core implementation. This has caused the delay of interconnect wires to increase and the need of a robust network on chip (NoC) has emerged [4]. In addition, the cost of speed and high frequency in inter-core communication is increasing. In sub-micron designs, the global wire delay becomes a significant matter to timing. The complexity of the parallel interconnect wires will have problems with relative area, power overhead, signal degradation due to crosstalk, synchronization, bandwidth limitation and scalability [5, 6]. Due to possible skew among parallel interconnect wires, the maximum signaling speed is also limited. Therefore, serial NoC interconnect has been the solution for high speed inter-core communication [7] through the implementation of SerDes. The comparison between serial and parallel communication for NoC is summarized in Table 1.1.

Table 1.1: Comparison between parallel link and serial link

Specifications	Parallel Link	Serial Link
Area	Large	Small
Power	High	Low
Crosstalk	Yes	No
Frequency	Low	High

1.1.5 Shortcomings of Parallel Link and Advantages of Serial Link

An early analysis [8] shows that having a parallel link mesh interconnect of network on chip (NoC) takes a large interconnect area due to the necessity of large amount of routings and shielding. The work did a comparative analysis and relieved a

large increase in power efficiency with the use of serial link. Interconnect are extensively required in multichip designs, which are preferable to have small scale in power and area consumption. Parallel links are also harder to route and have low immunity to noise [2].

Serial link provides more efficient and cost saving method to enable the communication between multiple die, as aligned to the objective of having system-on-chip (SoC) designs, which targets to integrate complex systems in a single package. This is something parallel interconnect can never offer.

A serial connection needs fewer wires in the transmission link. For an N bit SerDes, the required physical link for data transmission is reduced by a scale of N , and creates design space for better isolation of the wires. Due to lesser wires in the transmission link, the transceiver is less prone to crosstalk issue. Furthermore, in asynchronous clocking serial link, skew and jitter is not an issue. Therefore serial link is more preferred in modern designs with the need of NoC.

1.1.6 Skew and Jitter

In IC designs, circuits are working at a very high frequency. Due to many factors such as wire-interconnect length, temperature variations, material imperfections, capacitance and inductance, the same clock signal might arrive at different time at different devices or components. Clock skew is the differences in the arrival time of the same clock signal at the device clock pin. Figure 1.3 shows this misalignment when the same clock propagates through routing and arrive at different registers. At high

frequency, pulse width becomes short and clock skew becomes crucial in high speed signaling [9].

Clock jitter is the time deviation of a clock edge from the reference edge. It is observed that a presumed periodic clock signal is not ideal and the clock edge might arrive earlier or later than the controlled position. This introduces timing uncertainty of the clock pulse as the discrepancy will distort the clock pulse duty cycle. Controlling clock jitter is critical in signal detection at the receiving end as it may jeopardize the synchronicity between the transceiver pair [10]. Figure 1.4 shows the possible occurrences of uncertainty in the clock edge timing of a symmetrical clock pulse [9].

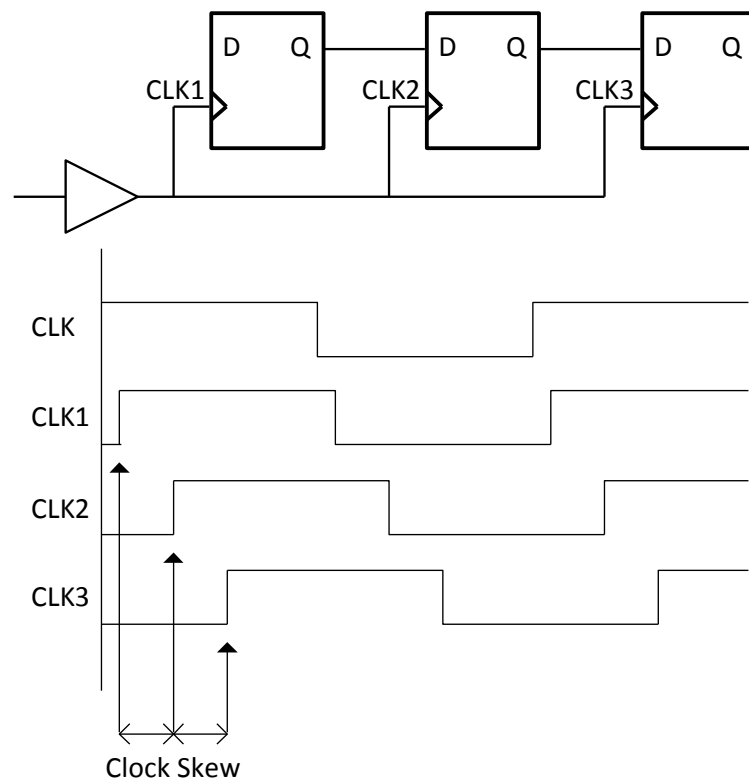


Figure 1.3: Clock skew in sequentially adjacent registers

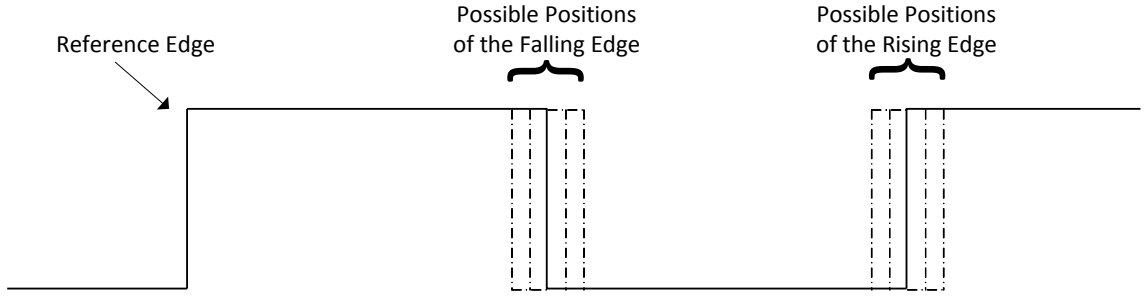


Figure 1.4: Clock jitter in a periodic clock pulse

1.2 Problem Statement

As the process technology advances, transistor size shrinks and more IPs are integrated onto chip. The increasing trend in system-on-chip (SoC) and system in package (SiP) designs demand the integration of more bus width and metal tracks on the interconnection between IPs. With the increasing of transistor density and downsizing of chips, the current micro-bumps interconnect between die will not be sufficient. This causes the communication between cores to become the performance bottleneck.

Routing congestion will occur when the bus width increases. The routing of data bus becomes difficult and detour will cause increase in wire length and adding delay, which ultimately causes difficulty to close timing. The engineering change order (ECO) to pass timing and improve chip performance will result in increase of power consumption and affect signal integrity. The consequence of conventional bus interconnect scheme will increase the risk of manufacturing defect as the chances of having open or short routings and vias are higher. Due to this high routing density, the average cost per device is also higher.

Methods to increase the routability have been proposed with different serial clocking scheme such as the mesochronous serial transmission [11] and plesiochronous

transmission [12, 13]. Conversion of parallel data bits into serial data streams before transmission greatly reduced the number of wires required for the interconnect [14]. However, this serialization and deserialization scheme involves extensive use of analogue auxiliary circuits to minimize the skew and jitter of the data and clock signals [8]. Clock and data recovery circuits are power and design area intensive. The complexity of conventional high speed interconnect schemes imposes challenge on the system's reusability, portability and scalability.

1.3 Objective

1. To propose a configurable clockless SerDes design with reduced power consumption.
2. To implement the SerDes in a modular and configurable manner that supports Plug-and-Play in any digital IP with minimum modification.

1.4 Project Scope

The main objective of this research is to implement a full system achieving a high speed on-chip serial communication between die. The communication is through a serial link which includes the transmitter, the receiver, the encoder, the decoder and the interconnect between the two cores. Designing a robust transceiver without transmission of clock signal in a separate line and capable to send high speed data up to 2 Gbps with low power consumption is the scope of this project.

This research will cover the design of the clockless SerDes serial link, evaluation of the proposed algorithm through pre-layout simulation, analysis and comparison with the other previously proposed method.

1.5 Thesis Outline

This research aims to implement an on-chip serial link for high speed communication between multiple die. The scope is to design the transmitter and receiver pair, the encoder and decoder pair to transmit binary signal over a lossy interconnect. The final objective is to enable the serial communication between two cores at high speed and low power.

The background of serial link is discussed in Chapter 1. It includes the shortcomings of parallel links and the basics of interconnect signaling schemes. Previous proposed work and related publications are reviewed in Chapter 2. The pros and cons of past works are also described and compared.

The proposed clockless SerDes design is discussed in Chapter 3. The signaling scheme to achieve advantage of high speed serial communication with minimum slew time and jitter is described and summarized.

Measurement results from simulation are analysed and discussed in Chapter 4. The system is placed on a testbench for verification on functionality and performance. Finally, the conclusion highlights the key findings and contributions of the work. Chapter 5 concludes the research and recommends potential improvements for relevant future works.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

In the previous chapter, the basic signaling background and the need of serial link is discussed. In this chapter, the types of serial transmission clocking schemes are reviewed. The design of conventional SerDes are also discussed and the deficiency of conventional SerDes will be explained. Existing works of clockless SerDes designs are reviewed and compared in different degree of details respectively. This chapter provides an introduction to serial link systems and also gives an outline of the thesis, which is devoted to the development of inter-die high-speed, low power serial link.

2.2 Serial Transmission Clocking Schemes

The inter die transmission of signal includes sending the clock signal to the recipient die. Synchronicity of the data and clock signal is the key to transmission of correct data. In serial transmission, there are multiple clocking schemes implementing different methods to achieve synchronicity.

2.2.1 Mesochronous Clocking

Serial link with mesochronous clocking requires two transmission line in the interconnect. Figure 2.1 depicts the mesochronous clocking where the clock requires a dedicated wire in the transmission, which consumes routing resource and design area on the chip. Both the data signal and the clock signal are transmitted separately to the

receiver. As the interconnect will have introduced an uncertain amount of delay, the transmitter (TX) and receiver (RX) clocks will have an unknown skew although they are the same clock of the same source. In this serial signaling scheme, additional circuits such as the phase detector is usually included in the RX to detect phase shift of the received clock signal and adjust for synchronicity [10]. Such circuit is power hungry and design area intensive, as well as introducing complexity and reducing reusability of the SerDes.

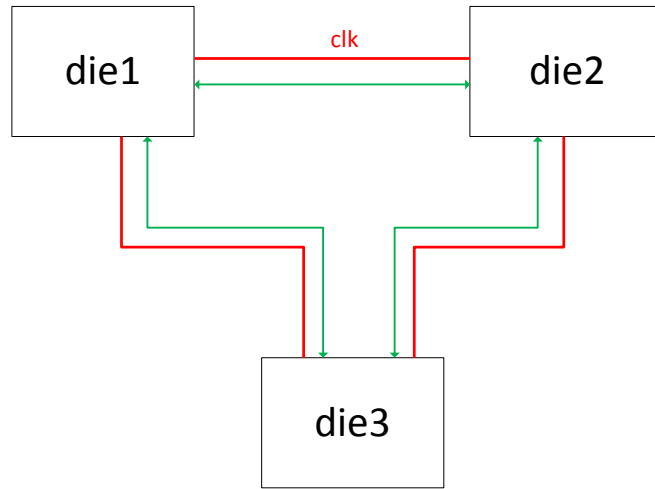


Figure 2.1: Mesochronous clocking serial link

2.2.2 Plesiochronous Clocking

In order to further reduce the transmission line in serial link, plesiochronous clock serial link is introduced. Such signaling scheme further reduces the wire count in inter-core signaling where the need of transmitting the clock signal over a dedicated wire is removed. In IC design, wire tracks carrying clock signal usually takes up more design space as additional shielding and buffer is required. Figure 2.2 shows the plesiochronous signaling scheme where the clock is not sent in inter-core communication, but fed distinctly to each core. However, there will be a phase mismatch on the clock signal at

each core, and frequency mismatch if they are running at different frequency. Additional circuits are needed at the RX to synchronize the received signals, which are also power hungry and area intensive [11].

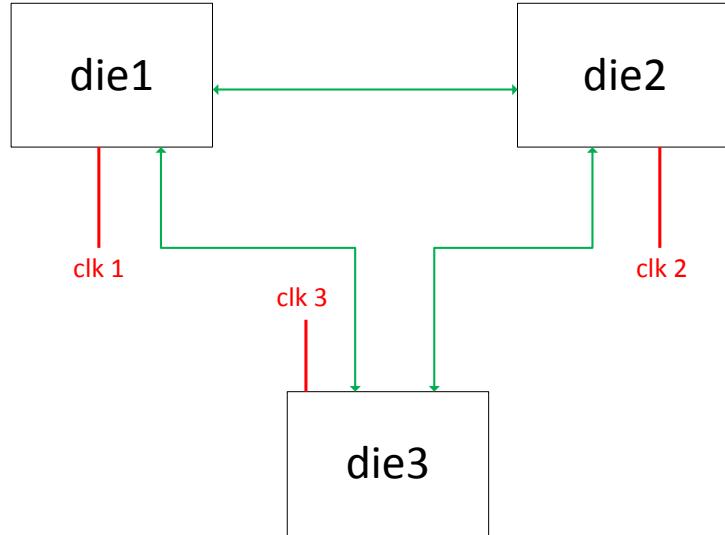


Figure 2.2: Plesiochronous clocking serial link

2.2.3 Asynchronous Clocking

Figure 2.3 illustrates the block diagram of asynchronous clocking serial link. An encoding circuit is added to the TX to send the serialized data signal with the clock embedded in it. On the other hand, a decoding circuit will be added to recover the data and clock signal from the received signal before deserializing it. The circuitry to recover the embedded clock is usually a clock and data recovery (CDR), or customized light weight, high speed decoding circuits [11, 12, 15, 16]. Such serial signaling design is also known as self-timed SerDes. Since the data and clock are extracted from the same signal, there will be no clock skew between the two and the signal is jitter insensitive. In this thesis, the SerDes design will be based on this signaling scheme. Asynchronous transmission requires no synchronization circuit at both ends of the system and possible

to achieve with simple designs which can be setup very fast. This also reduces the design and production cost for the reduction in hardware required to build the system.

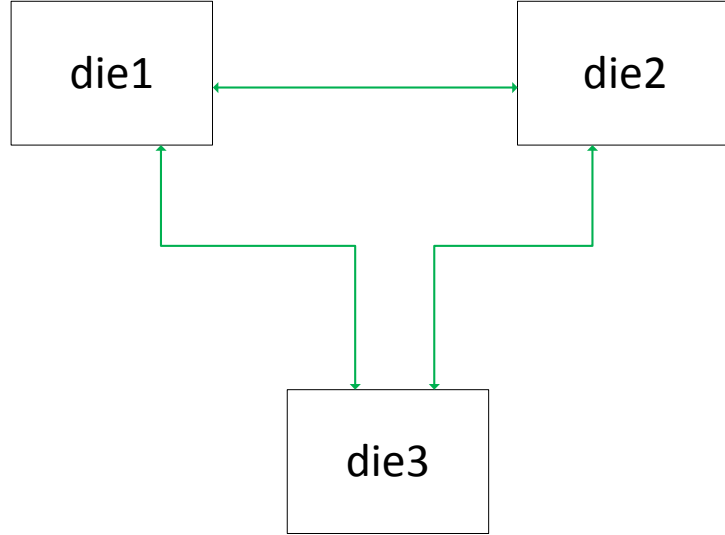


Figure 2.3: Asynchronous clocking serial link

2.3 SerDes Conventional Implementation

There are drawbacks in conventional SerDes designs. One of the shortcomings of conventional SerDes is the high power consumption, as complex circuits and power hungry circuits are required to ensure the transmitted data and clock signals are synchronized between the transmitter and receiver. The additional circuits to provide clock and data recovery (CDR), de-skew and reduce jitter of the clock and data signal will also introduce complexity to the design and thus takes up more design space.

2.3.1 Overview of Conventional SerDes Design

An earlier work presented an on-chip serial link over a lossy transmission line [11]. The transceiver was implemented in 0.13 μm CMOS process and transmits serial

signal at a data rate of 9 Gbps. Figure 2.4 shows the overall block diagram of the conventional SerDes system. This system sends the serialized data and clock signal separately over the transmission line. This mesochronous SerDes will have the same clock at the transmitter and receiver with clock skew. The receiver will need a phase tuning circuit to ensure the synchronicity of the data and clock signals.

The major drawback of this implementation is that phase adjusting circuit is power and area intensive [10]. The LC-oscillator-based phase locked loop (PLL) used to generate the 4.5 GHz clock for the transceiver and consumes 105 mW which is fairly large compared to other implementation of interconnect schemes. The use of analog module makes the design to have reduced reusability. In the following sections, the components in a conventional SerDes are described as a review of basic SerDes design.

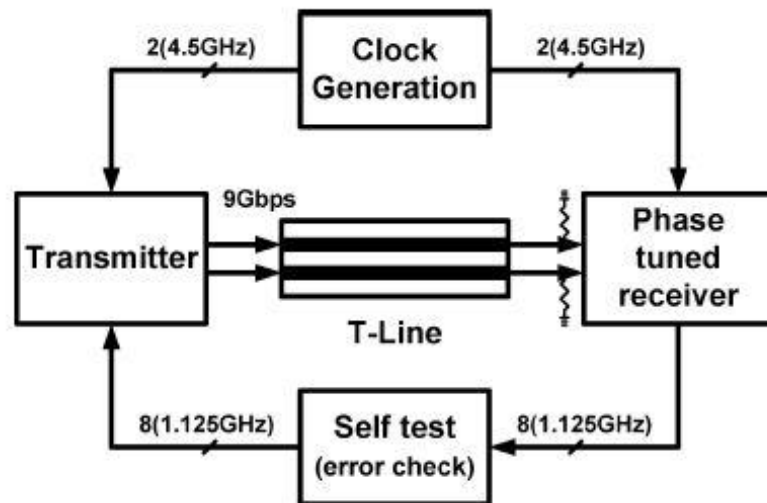


Figure 2.4: Overall block diagram of conventional SerDes system in [11]

2.3.2 The Transmitter

The transmitter sends signal to the recipient core through the inter-core transmission line. Figure 2.5 shows a conventional SerDes transmitter implemented in the work [11] which implements mesochronous transmission. The 9 Gbps SerDes transmits at double data rate with a clock at 4.5 GHz. The clock divider provides the frequency for the serializer for parallel bits conversion. This transmitter deploys two serializers, each running at 1.125 GHz to convert the parallel 1.125 GHz into 4.5 GHz data signal. As interleaved drivers are implemented, and both the serializers are clocked at the opposite clock edge, a 9 Gbps data signal is transmitted into a single data link with a driver. The clock signal is not embedded in the transmitted signal, but sent together with the data signal in another transmission line.

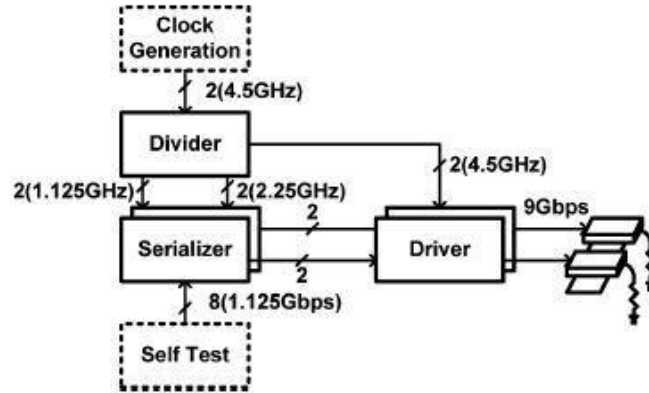


Figure 2.5: Block diagram of the conventional SerDes transmitter [11]

2.3.3 The Interconnect

The interconnect is the medium used for communication between multiple cores. In 2.5-dimensional (2.5D) design, it is usually a silicon bridge or silicon interposer connected with microbumps at the die. In [11], the interconnect is a differential pair,

intermediate metal layer with width of 6 μm with a spacing of 3 μm , shielded with 21 μm ground metal tracks. The transmission lines are resistively terminated to reduce reflection. The physical properties of the transmission line will not be covered in this research, but will be modelled to simulate the performance of the proposed SerDes.

2.3.4 Serializer

The basic serializer comprises of multiplexers that switch in sequence to propagate the parallel data. The multiplexers are connected in the topology as shown in Figure 2.6 at half of the required frequency each stage. Two techniques can be used to generate the different frequency clock, either with a clock divider to produce the lower frequency clock; or a frequency multiplier to produce the higher frequency clock [17]. The selector of the multiplexers are controlled by the clocks at twice the frequency at each stage, propagating each bit from the parallel data at both rising and falling clock edge and produces a single stream of high frequency data.

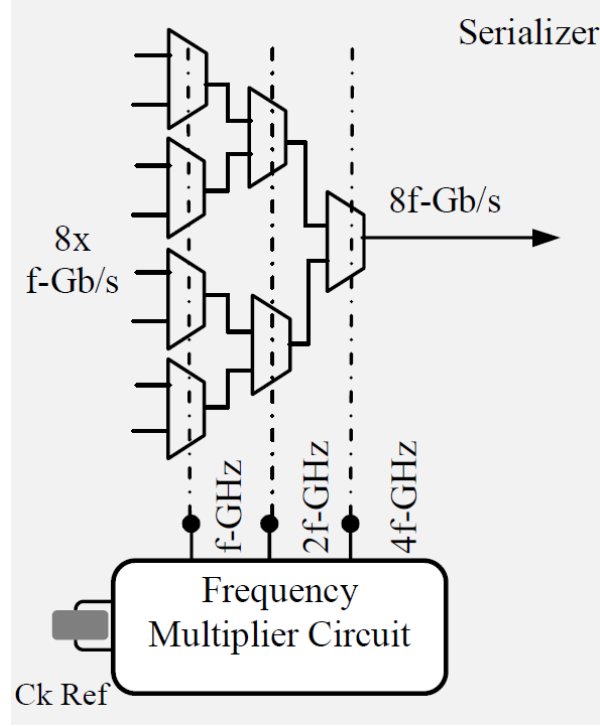


Figure 2.6: Block diagram of serializer [18]

2.3.5 The Receiver

The SerDes receiver resides in the signal recipient core and collects both the serialized data signal and the clock signal at the other end of the transmission link. In this work, the conventional CDR is used [18]. The conventional SerDes receiver implements a circuitry to detect the phase of the data signal and the clock signal. This also requires a clock generation circuit to produce a clock signal at the receiver end as a reference clock for the synchronization. The phase interpolator synchronizes the phase of the data signal at the receiver and the clock signal from the PLL. Finally, the clock signal clocks the internal components of the receiver to convert the serialized data to parallel data. The general block diagram of the receiver is illustrated in Figure 2.7.

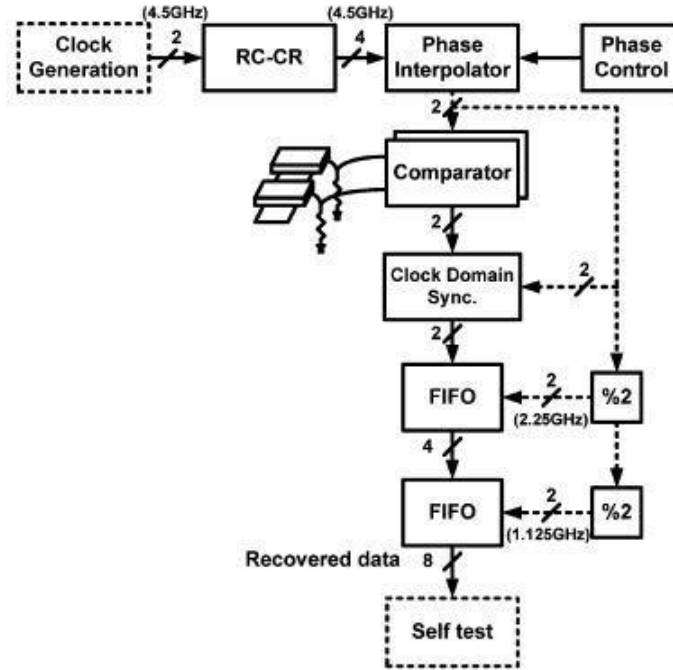


Figure 2.7: Block diagram of the conventional SerDes receiver [11]

2.3.6 Deserializer

The deserializer receives the serialized signal transmitted over the interconnect and converts it back to parallel data. Figure 2.8 shows a conventional deserializer which implements a de-multiplexer for deserialization. The high frequency clock signal is required at the deserializer to restore the parallel data. To achieve synchronicity, the clock signal is usually transmitted either through a dedicated interconnect wire and skew removed with additional circuits. The deserializer in Figure 2.8 implements a serialized signal with embedded clock which requires a CDR circuit to extract the transmitted clock for deserialization.

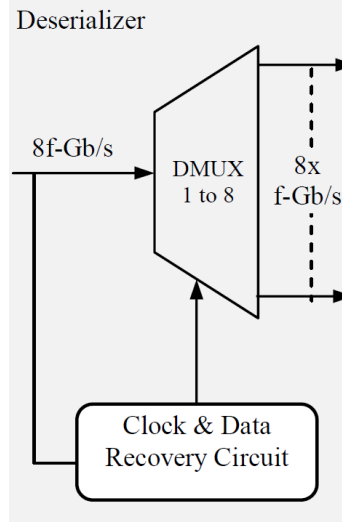


Figure 2.8: Block diagram of deserializer [18]

2.4 Challenges in Serial link

Serial data transmission between cores achieves a routing resource efficient design. However, the serialized data will have a bit rate at N times higher than the parallel data bit rate where N is the number of bits to be serialized. This will result in higher switching activity and increase power dissipation in the serial link. The work in [19] targets to reduce the switching activity by 40% with a new bit ordering technique. The involvement of the analogue circuits required to synchronize the data edge to the clock edge increases the complexity of the SerDes design and makes it not compatible as a reusable module. Clock and data recovery circuits are complex and usually includes analogue blocks which has the same problem. In conjunction with the growing density of chip design, it is desirable to construct a modular and scalable interconnect system that supports Plug-and-Play (PnP) is instantiable and is capable of reduce the number of interconnect (microbumps) by a desired factor. Due to the difficulties above, the SerDes is often customized as a transceiver and supports only certain critical features such as the

high speed signal interface (HSSI). Therefore, design resources are often being intensely focused on the SerDes transceiver and takes up high power consumption, high demand on routing resources and large physical area.

2.5 Transmission with Clock Data Recovery

Ideally, the data transmission is synchronous where the controlled clock runs the entire chip network. Having identical rates, no phase difference, no jitter and no skew, the clock at the transmitter and receiver end are perfectly aligned. In reality, this is impossible as there will always be mismatch. Transmitting serial signal and the clock signal in separate interconnect is called mesochronous transmission as described in section 2.2.1. Additional circuitry is required at the receiver end to restore the synchronicity between the data and clock.

A plesiochronous transmission scheme as described in section 2.2.2 is used to eliminate the need to transmit the clock in another wire. This can be done if the attribute of the clock such as frequency is known. A reference clock with the same attribute is generated at the receiver end and fed into CDR circuits to be phase aligned to the received data. However, there are challenges in implementation of CDR circuits. These analogue circuits are usually customized or redesigned each time it is ported to support a different IP. The complexity of analogue circuit increases as the operating frequency grows to meet tight timing requirement. When circuit performance needs to be guaranteed, large devices are used which contributes to increase in design area. This leads to the increase in parasitic of the circuit components which leads to the degradation in switching rate of the digital logic gates in the design which affects the performance of the blocks [13].

A CDR technique is presented in [13] which implements an enhanced CDR to reduce the aforementioned problems which often occur in conventional second-order CDR. It has relatively smaller design area and the system has low channel loss even with the analogue components which have large parasitic. The work achieves wider bandwidth and lower jitter in the clock and data recovery process and is capable of recovering low jitter clock from partially equalized eye. Despite having good timing margin, blocks used in the design such as the equalizer, the PLL, the delay locked loop (DLL) and phase detectors are complex analogue circuits. The system is still comparatively large compared to fully digital systems, and the analogue circuits attribute reduces the reusability.

An earlier proposed system in [5] made use of digital blocks which increase the scalability and reusability of the SerDes. The work achieves a better interconnect performance than parallel interconnect with the implementation of ring oscillators. However, the extensive use of sequential device causes timing to be a criteria to consider in the design. The system did not take advantage of both clock edges and only implements a single data rate (SDR) serialization. This plesiochronous design which involves an external clock to interact with the internal clock intensively requires auxiliary circuitries that involve analogue designs to support the clock synchronization. A third IP aside from the sender and receiver die is also present to support the inter-die communication.

2.6 The Multi-Level Transmission Scheme

An earlier work [12] proposed a SerDes Transceiver with multi-level transmission (MLT) which aims to eliminate the necessity to transmit the clock signal together with

the serialized data in separate lines. The clock is combined with the data signal to produce a signal with a third voltage level using a three level encoder circuit. The transmitted signal is decoded back to data and clock at the receiver side. This eliminates the need for equalizer circuit or edge detection circuit which is often complex. This technique solves the problem with clock jitter and skew as the clock and data are now within the same signal. However, the proposed decoder in this work uses a phase detector which occupies fairly a large area and is power hungry. A summary in [12] shows that the phase detector consumes 5 times the power of the Deserializer and 62.5% of the line driver power. The design will be less power efficient if used repeatedly to reduce the interconnect of a large signal scale design. Moreover, the serializer implementation is through double edge triggered flip flops which are timing crucial sequential devices. This is to compensate the problem in the three level encoding technique which represents each data logic in two clock cycles which loses half the data rate.

A solution has been proposed to hinder the loss of data rate as published in [16] which implements the same three-level signaling scheme as [12], but is capable of encoding two bits of data within a clock cycle. The technique utilizes four phases of the clock with frequency equal to the data rate to generate the three-level signal from the serialized data. The three-level signal produced by the encoder will transmit at double data rate even though four phases of clock are used. In order to comply with this encoding technique, additional circuit is introduced to the encoder which involves the use of customized multiplexer, customized three-level inverter and sequential devices. Although the blocks are reusable, these sub-blocks have significantly increased the design area consumption which makes it not feasible for instantiation in large scale

designs. Flip-flops used in the encoder design are sequential devices and have high transistor count, and needs to meet timing so that the circuit will not have issue with metastability. Compared to [12], the work in [16] achieves twice the data rate but consumes approximately 9 times the power. Table 2.1 demonstrates the performance specifications of previous proposed SerDes systems.

Table 2.1: Design comparison

Reference	[11]	[12]	[16]	[5]	[13]
Technique	Meso-chronous	Asyn-chronous, MLT	Asyn-chronous, MLT	Plesio-chronous	Plesio-chronous
Process	0.13um CMOS	65nm CMOS	65nm CMOS	0.18um UMC	28nm CMOS
Supply Voltage	N/A	N/A	1.2V	1.8V	N/A
Transistor Count	N/A	N/A	N/A	384	N/A
Transmission Bandwidth	9Gb/s	12Gb/s	24Gb/s	2Gb/s	40Gb/s
Internal clock	1.125GHz	24GHz	12GHz	200MHz	20GHz
External clock	4.5GHz	N/A	N/A	2.54GHz	N/A
Area	4.28mm ²	N/A	N/A	7.2mm ²	0.81mm ²
Power	765mW	15.5mW	109.6mW	4.19mW	927mW
Link	5.8mm, 30.6Ohm	3mm	L=5mm, W=5um	6mm	N/A

2.7 Chapter Summary

In this chapter, the serial transmission clocking schemes are discussed. Mesochornous and plesiochronous clocking require the clock phase to be known at the recipient die and additional circuits to achieve synchronicity. On the contrary, asynchronous transmission is self clocking and does not require synchronicity