STATUS OF THESIS

Title of thesis | Design and Characterization of Low-Power Low-Noise All-Digital Serial Link for Point-to-point Communication in SoC

I MOHAMMED H K ALSER

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

The contents of the thesis will remain confidential for _____ years.

Remarks on disclosure:

Mohammed M.

Signature of Author

Endorsed by

Signature of Supervisor

Permanent address:
Sheikh Redwan, ElJala' Street,
Gaza City, Palestine

Name of Supervisor
Dr. Maher Assaad

Date : 14 / March / 2013

Date : 14/3/2013

UNIVERSITI TEKNOLOGI PETRONAS

DESIGN AND CHARACTERIZATION OF LOW-POWER LOW-NOISE ALL-DIGITAL SERIAL LINK FOR POINT-TO-POINT COMMUNICATION IN SOC

by

MOHAMMED H K ALSER

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree stated.

Signature: _____

Main Supervisor: Dr. Maher Assaad

Signature: _____

Co-Supervisor: Dr. Fawnizu Azmadi Hussin

Signature: _____

Head of Department: Dr. Rosdiazli B Ibrahim

Date: 14-3-13

Dr Rosdiazli Bin Ibrahim
Head of Department
Electrical & Electronic Engineering Department
Universiti Teknologi PETRONAS

DESIGN AND CHARACTERIZATION OF LOW-POWER LOW-NOISE ALL-DIGITAL SERIAL LINK FOR POINT-TO-POINT COMMUNICATION IN SOC

by

MOHAMMED H K ALSER

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

MASTER OF SCIENCE

ELECTRICAL AND ELECTRONIC ENGINEERING

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR,

PERAK

MARCH 2013

DECLARATION OF THESIS

Title of thesis | Design and Characterization of Low-Power Low-Noise All-Digital Serial Link for Point-to-point Communication in SoC

I MOHAMMED H K ALSER

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

Signature of Author

Signature of Supervisor

Permanent address:
Sheikh Redwan, ElJala' Street,
Gaza City, Palestine

Name of Supervisor
Dr. Maher Assaad

Date : 14/ March / 2013

Date : 14/3/2013

ACKNOWLEDGEMENTS

First and foremost, I would like to thank Allah the almighty for granting me the patience, strength and guidance to accomplish this work and for having my first baby Hasan, which makes me the happiest man in the world.

I am so grateful for meeting many inspiring and knowledgeable professors, colleagues, and friends, during the last two years in Malaysia. I was able to experience an enormous amount of educational, professional, and personal growth due to the support and encouragement of several people. For that, I would like to seize this opportunity to acknowledge all of them.

I would like to thank my supervisor, Dr. Maher Assaad, for teaching me and giving me valuable feedback throughout this challenging period. I would not forget the days we worked closely and discussed about papers together. I would also like to thank my co-supervisor Dr. Fawnizu Azmadi Hussin for his technical guidance during my research proposal defence. His dedication to research has inspired me and has significantly influenced my research.

I am grateful to University Technology of PETRONAS (UTP) for funding this research under URIF grant and for giving me the opportunity to gain valuable experience as a graduate assistant.

To my wife Nour, my father Dr. Hasan, my mother Mrs Itaf, my brothers, and my sister, I would like to extend my appreciation and faithful thank to them. Their love, care, and never ending support always give me the internal strength to move on whenever I feel like giving up or whenever things become too hard. I thank them for always believing in me and being by my side.

I would also want to thank all my relatives, my friends and those who have directly or indirectly assisted me during this research. I wish you all success and fruitful achievements all the time.

ABSTRACT

The fully-digital implementation of serial links has recently emerged as a viable alternative to their classical analogue counterpart. Indeed, reducing the analogue content in favour of expanding the digital content becomes more attractive due to the ability to achieve less power consumption, less sensitivity to the noise and better scalability across multiple technologies and platforms with inconsiderable modifications. In addition, describing the circuit in hardware description languages gives it a high flexibility to program all design parameters in a very short time compared with the analogue designs which need to be re-designed at transistor level for any parameter change. This can radically reduce cost and time-to-market by saving a significant amount of development time. However, beside these considerable advantages, the fully-digital architecture poses several design challenges.

This work seeks to elucidate these design challenges involved in the design of fully-digital serial links. Further, it presents a low-power low-jitter fully-digital complete serial link synthesized and implemented on Altera Cyclone II FPGA. Unlike the conventional architectures, it is totally synthesized from a Verilog hardware description language and eliminates the need for any analogue or off-chip components. Key circuit blocks include a built-in PRBS generator for testing purposes, a high-resolution frequency synthesizer circuit, an eight-phase generator, and a low-power clock and data recovery (CDR) circuit using quarter-rate clocking to mitigate the speed requirement of the data recovery process. The serial link achieves a BER lower than 10^{-12} (using bathtub curve), consumes 0.97 mW power while operating at 167.36 Mb/s, and operates continuously over the range of 167.32 Mb/s to 193.6 Mb/s per channel. The measured RMS and peak-to-peak jitters of the recovered clock are 49.8 ps and 295 ps, respectively, for 167.36 Mb/s 2^{23}-1 PRBS input stream.

ABSTRAK

Pelaksanaan pautan digital sesiri sepenuhnya baru-baru ini, muncul sebagai satu alternatif yang berdaya maju berbanding dengan sistem analog klasik yang lain. Sememangnya, ini dapat mengurangkan kandungan analog dengan memperluaskan kandungan digital menjadikan ia lebih menarik disebabkan keupayaannya untuk mencapai penggunaan kuasa yang lebih rendah, kurang sensitiviti kepada hingar dan berkemampuan untuk pembangunan yang lebih baik di antara pelbagai platform dan teknologi dengan pengubahsuaian yang dikehendaki. Tambahan pula, untuk mengkaitkan litar dalam istilah perkakasan, ia memberikan fleksibiliti yang tinggi untuk merancang semua parameter reka bentuk dalam masa yang sangat pendek berbanding dengan reka bentuk analog yang perlu direka semula di peringkat transistor untuk perubahan sebarang parameter. Dengan ini pengurangankan kos pembangunan dan masa-untuk-memasarkan dapat dijimatkan seawal peringkat pembangunan lagi. Walau bagaimanapun, seni bina digital sepenuhnya menimbulkan beberapa cabaran dalam peringkat reka bentuk.

Tugasan ini bertujuan menjelaskan cabaran-cabaran yang terlibat dalam mereka bentuk hubungan sesiri digital sepenuhnya. Selain itu, ia mempamerkan kuasa-rendah rendah-gugup pautan sesiri digital sepenuhnya yang disintesis dan dilaksanakan pada Altera Cyclone II FPGA. Tidak seperti seni bina konvensional, ia benar-benar disintesis daripada bahasa Verilog dan menghapuskan keperluan untuk mana-mana komponen analog atau komponen luaran cip. Blok-blok litar utama termasuk penjana PRBS dalaman untuk tujuan pengujian, litar penjana pemasa beresolusi tinggi, penjana lapan-fasa, dan litar pemasa dan pemulihan data berkuasa rendah (CDR) menggunakan masa kadaran suku untuk menurunkan kelajuan proses pemulihan data.

TABLE OF CONTENT

ABSTRACT vii
ABSTRAK viii
LIST OF FIGURES xiii
LIST OF TABLES xvi
CHAPTER 1 INTRODUCTION 17
1.1 Overview of Data Communication in System-on-Chip 17
1.2 Conventional Bus Limitations 18
1.3 Point-to-Point Links 19
1.4 Why Less Analogue Content Is Better? 22
1.4.1 Less Power Consumption 22
1.4.2 Less sensitivity to PVT variations and exhibits less phase noise 25
1.4.3 Cost-effectiveness and size-reduction 27
1.4.4 Better scalability and seamless programmability 27
1.5 Research Objectives 28
1.6 Scope of Research 28
1.7 Research Methodology 29
1.8 Thesis Organization 31
CHAPTER 2 LITERATURE REVIEW 32
2.1 Chapter Overview 32
2.2 A Typical Serial Link 32
2.3 The Key Elements of a Link 35
2.3.1 Transmitter Circuit 35
2.3.1.1 Pulse Amplitude Modulation (PAM) Transmitters 35
2.3.1.2 Multiplexing-based transmitters 36
2.3.2 Frequency Synthesizer Circuit 37
2.3.2.1 Direct-Digital Frequency Synthesizer (DDFS) 37
2.3.2.2 PLL and DLL-based Frequency Synthesizers 38
2.3.3 Receiver Circuit 41
2.3.4 Clock and Data Recovery (CDR) Circuit 41
2.3.4.1 PLL-Based CDR Circuit without Reference Clock 42

2.3.4.2 Combination of FLL/DLL Based CDR 43
2.3.4.3 Full-Rate and Fractional-Rate CDR Architectures 44
2.3.5 Serial Link Performance Metrics 45
2.3.5.1 Data Rate 45
2.3.5.2 Power Consumption 46
2.3.5.3 Bit Error Rate (BER) 46
2.3.5.4 Jitter Analysis 46
CHAPTER 3 FPGA HARDWARE PLATFORM 53
3.1 Chapter Overview 53
3.2 Choice of Technology (Target Hardware Platform) 53
3.3 Altera DE2-70 FPGA Development Board 54
3.4 Functioning of FPGAs 57
3.5 FPGA Design Flow and Development Tools 58
CHAPTER 4 PROPOSED HARDWARE ARCHITECTURE 60
4.1 Chapter Overview 60
4.2 Fully-Digital Serial Link Design Challenges 60
4.3 Serializer Design 61
4.3.1 8-to-1 Serializer 62
4.3.2 Frequency Synthesizer Circuit 63
4.3.2.1 Digitally Controlled Oscillator 65
4.3.2.2 Integer Divider 67
4.3.2.3 Frequency Detector (FD) 67
4.3.2.4 Up/Down Counter and Full Subtractor 69
4.3.2.5 Digitally Controlled Delay Line (DCDL) 70
4.3.2.6 DLL Controller 70
4.3.2.7 Phase Detector (PD) 71
4.3.3 Built-In PRBS Generator 72
4.4 Deserializer Design 72
4.4.1 Clock and Data Recovery (CDR) Circuit 73
4.4.1.1 Early-Late Phase Detector (ELPD) 73
4.4.1.2 Setup/Hold Time Violation in ELPD 75
4.4.1.3 Delay Line Controller (DLC) 76

4.4.1.4 Digitally Controlled Delay Line (DCDL) 77
4.4.1.5 Quarter-Rate Frequency Detector (QRFD) 77
4.4.1.6 Up/Down Counter and Full Subtractor 79
4.4.1.7 DCO and Eight-phase Signal Generator 80
4.4.2 4-to-8 Deserializer 81
CHAPTER 5 RESULTS AND DISCUSSION 83
5.1 Chapter Overview 83
5.2 FPGA Synthesis and Implementation 83
5.2.1 Resources Utilization Analysis 83
5.2.2 Power Consumption Analysis 85
5.3 Experimental Results 85
5.3.1 Serializer Performance 87
5.3.2 Deserializer Performance 90
5.4 Performance Comparison 96
5.4.1 Frequency Synthesizer Performance Comparison 96
5.4.2 CDR Performance Comparison 97
5.4.3 Serial Link Performance Comparison 97
CHAPTER 6 CONCLUSION AND RECOMMENDATIONS 104
6.1 Chapter Overview 104
6.2 Conclusion 104
6.3 Thesis Contributions 105
6.4 Future Work 107
REFERENCES 108

LIST OF FIGURES

Figure 1.1: Conventional bus-based communication in SoC..... 19
Figure 1.2: Point-to-point parallel link based communication in SoC. 20
Figure 1.3: Area and power for serial and parallel links versus technology node [5]. 21
Figure 1.4: (a) Power efficiency and (b) jitter performance of recently published serial links for chip-to-chip communication.....21
Figure 1.5: Basic CMOS gate. 22
Figure 1.6: Basic MCML gate. 24
Figure 1.7: Power versus frequency in CMOS and MCML inverters for a CMOS 0.18-μm process, V_{dd} = 1.8 V [19]. 25
Figure 1.8: Hierarchy and scope of research. 29
Figure 2.1: Serial link based communication in SoC: (a) forwarded clock architecture and (b) embedded clock architecture. 34
Figure 2.2: Block diagram of an embedded clock serial link. 35
Figure 2.3: The 4-PAM transmitted signal timing diagram.....36
Figure 2.4: Simplified top-level block diagram of a SERDES system.....37
Figure 2.5: Simplified block diagram of a direct-digital frequency synthesizer. 38
Figure 2.6: Basic block diagram of the all-digital PLL-based frequency synthesizer. 39
Figure 2.7: Operation for the all-digital DLL-based frequency synthesizer.....40
Figure 2.8: Block diagram of a PLL-based CDR without a reference clock.....42
Figure 2.9: Block diagram of a combination of FLL/DLL based CDR architecture...43
Figure 2.10: Waveforms for (a) full-rate, (b) half-rate, and (c) quarter-rate CDR.....44
Figure 2.11: Effect of jitter on (a) clock signal and (b) data stream.....47
Figure 2.12: Classification of jitter. 48
Figure 2.13: Characterization of random jitter in an eye diagram by Gaussian statistics.....49
Figure 2.14: Example of (a) a reference signal data eye and the influence of (b) periodic jitter, (c) duty-cycle distortion jitter, and (d) data-dependent jitter on it.....50
Figure 2.15: (a) Characterization of total jitter in an eye diagram. (b) Fitting the histogram with Gaussian curves. 51

Figure 2.16: Bathtub curve and the total jitter from a bit-error-rate perspective..... 52
Figure 3.1: Altera DE2-70 FPGA development board [74]..... 55
Figure 3.2: Simplified Cyclone II EP2C20 FPGA block diagram..... 57
Figure 3.3: Overall FPGA design flow..... 59
Figure 4.1: (a) Block diagram and (b) timing diagrams of the 8-to-1 serializer..... 63
Figure 4.2: Block diagram of the proposed frequency synthesizer circuit..... 64
Figure 4.3: Functional block diagram of the digitally controlled oscillator..... 65
Figure 4.4: Measured ring oscillator output frequency F_{OSC} versus chain length. The number of bits of N defines the number of frequency steps between the two extreme limits (440 and 280 MHz)..... 66
Figure 4.5: Block diagram of the fractional divider and behavioural Verilog code.... 67
Figure 4.6: Block diagram of integer frequency divider..... 67
Figure 4.7: (a) Block diagram and (b) Timing diagrams of the frequency detector.... 68
Figure 4.8: Block diagram of the DCDL..... 70
Figure 4.9: Block diagram of the phase detector..... 71
Figure 4.10: Gate-level simulation waveform of DLL..... 71
Figure 4.11: Block diagram of the built-in programmable PRBS generator..... 72
Figure 4.12: Block diagram of the proposed CDR circuit..... 73
Figure 4.13: Block diagram of the ELPD..... 74
Figure 4.14: Detecting conditions of the ELPD (a) Locked state (b) Late state (Shift_Right is high) (c) Early state (Shift_Left is high)..... 74
Figure 4.15: Timing diagrams showing the condition to avoid setup/hold time violation in the ELPD..... 75
Figure 4.16: Block diagram of the DCDL..... 77
Figure 4.17: (a) Block diagram and (b) Timing diagrams of the QRFD architecture. 78
Figure 4.18: (a) Block diagram of the proposed 8-phase generator..... 80
Figure 4.19: (a) Block diagram and (b) timing diagrams of the 4-to-8 deserializer architecture..... 82
Figure 5.1: FPGA resources usage summary report..... 84
Figure 5.2: Layout of the Cyclone II FPGA chip..... 85
Figure 5.3: Measurement setup used to validate the serial link..... 87

Figure 5.4: Measured RMS and peak-to-peak jitters of the frequency synthesizer output. 88
Figure 5.5: Waveforms of (a) the PRBS generator, (b) the 8-to-1 serializer at the Locked state. 89
Figure 5.6: Measured serializer output: (a) waveform and (b) eye diagram. 90
Figure 5.7: Measured eye diagram of the received data stream. 91
Figure 5.8: Measured (a) spectrum, (b) jitter histogram, and (c) phase noise of the recovered clock for 2^{23}-1 PRBS input data stream. 91
Figure 5.9: (a) Waveforms of the CDR internal signals and (b) A screen-shot of an oscilloscope for the locked state behaviour. 92
Figure 5.10: Measured eye diagrams of the (a) 1-to-4 and (b) 1-to-8 recovered data, for 2^{23}-1 PRBS input data stream..... 93
Figure 5.11: (a) Peak-to-peak jitter and (b) RMS jitter of the recovered clock versus bit length of PRBS (2^n-1) at 41.85 Mb/s. 94
Figure 5.12: (a) Peak-to-peak jitter and (b) RMS jitter of the recovered data versus bit length of PRBS (2n-1) at 20.9 Mb/s. 94
Figure 5.13: Measured BER bathtub curve of the 1-to-8 recovered data. 95
Figure 6.1: (a) Power efficiency and (b) Jitter performance of the proposed system and recently published serial links. 105

LIST OF TABLES

Table 4.1: Fractional divider input and output signals 69
Table 4.2: Delay line controller operations 76
Table 4.3: QRFD internal signals (A and B) states 79
Table 4.4: QRFD truth table 79
Table 5.1: Performance comparison for the proposed frequency synthesizer with existing architectures 98
Table 5.2: Performance comparison for the proposed CDR with the prior art 100
Table 5.3: Performance comparison for the complete serial link with the prior art .. 102

CHAPTER 1

INTRODUCTION

1.1 Overview of Data Communication in System-on-Chip

The secret behind the success of modern integrated circuit (IC) technologies is due, in large part, to the realization of two essential and inseparable functions: high-speed data processing and reliable data communication. Over the years, advances in IC fabrication technology have led to an exponential growth of IC's speed and integration levels. These advances allow the number of integrated transistors to grow much more rapidly than the number of input and output (I/O) pins of the chip. While Intel's latest-generation processor called "Ivy Bridge" containing over 1.4 billion transistors incorporated in one chip and continues to double every two years in accordance with Moore's law, the speed performance of the I/O pins has historically lagged behind, doubling only about every six years.

Furthermore, due to the dramatic increase in the number of modules that can be incorporated into a single chip, the emerging demand for higher intra/inter chip communication bandwidth has significantly intensified. Such chip is commonly referred to as system-on-chip (SoC), and typically consists of various heterogeneous modules such as processors, dedicated hardware components, memories, I/O interfaces, and external interface IP (intellectual property) blocks that need to communicate with each other. However, the on/off-chip communication bandwidth does not scale up well with the continuous advancement on the on-chip computation capacity and thereby becomes the dominant performance bottleneck. This has prompted extensive research efforts during the last two decades in the direction of improving the on/off-chip communication networks to cope with the increasing

performance requirements. Recently, serial point-to-point data links have gradually dominated for the significant advantages over the multi-bit conventional buses and the parallel point-to-point links. Hence, they have the potential to fulfil the strongly emerging need for a higher communication bandwidth, more efficient data transmission, lower cost, and longer distance reach.

1.2 Conventional Bus Limitations

Over the course of many years, a bus-based paradigm (common clock or synchronous architecture) [1] was the most common in data communication systems. Figure 1.1 shows the multi-bit conventional bus architecture, where various modules are connected through a set of shared parallel channels. Additionally, a separate channel is distributed to all modules carrying a global clock signal used for synchronous transmission and reception of data. Although this communication paradigm has been adequate in the past, increasing speeds accentuate several problems associated with the using of the multi-bit conventional bus system [2]. Since the parallel data streams carried by the bus must be synchronized with the global clock signal, three primary limitations have been arisen that are skew, crosstalk, and large area [3, 4]. First, skew or timing offset among parallel channels results from the unequal arrival time of bits transmitted at the same time. This reduces the speed of data transmission to the slowest of all of the parallel transmission channels. Second, crosstalk caused by the undesired effect created by the transmission of a signal on one channel in another channel. This creates interference between the parallel signal channels and hence causes data signal delay and noise and places an upper limit on the length of the data transmission. Third, the cost of using a bus is also a serious issue since they occupy a large area of silicon. Further, the distribution of a global clock with the required timing accuracy over all modules easily results in high system design efforts and leads to high system cost. Therefore, the use of multi-bit bus for on/off-chip communication with a global clock will make the communication unreliable and limit further improvement of future SoC.

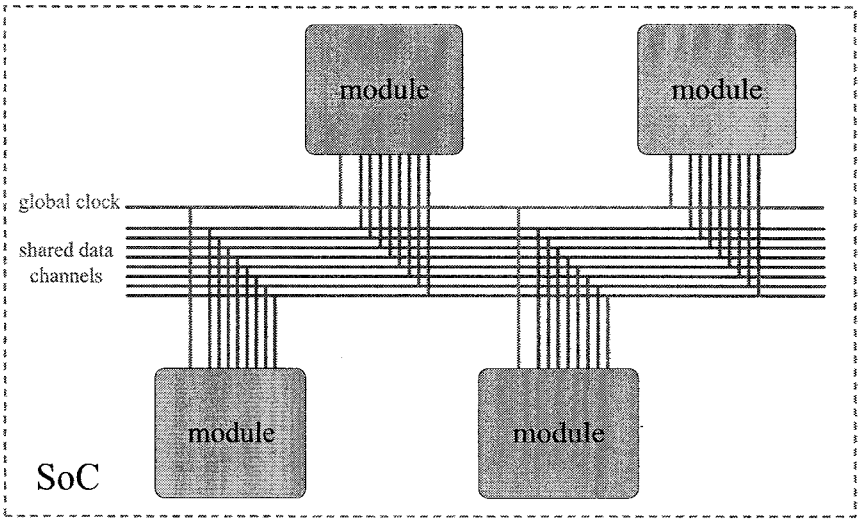


Figure 1.1: Conventional bus-based communication in SoC.

1.3 Point-to-Point Links

The aforesaid limitations of the multi-bit conventional buses make them viable for only small scale systems, which incorporate few modules, such as memory or peripheral buses. For larger scale systems, an alternative and attractive solution is to completely forsake the bus paradigm and use point-to-point links as a medium of communication. Point-to-point link architecture can be divided into two categories, namely parallel links and serial links. Parallel links are considered as the simplest method of communicating synchronous data between modules. Figure 1.2 shows a conventional source synchronous point-to-point parallel link. In this approach, each module is directly connected to each of the other modules by several parallel channels for transmitting the data, and an additional channel for transmitting the clock along with the data.

The architecture of the point-to-point parallel links inherently addresses many of the primary limitations of conventional buses. Nevertheless, improving the bandwidth of parallel links is uneconomical and impractical, since it is achieved by increasing the bit rate per channel and integrating a large number of channels into the system at the cost of large chip area, routing difficulty, noise and power consumption [5, 6].

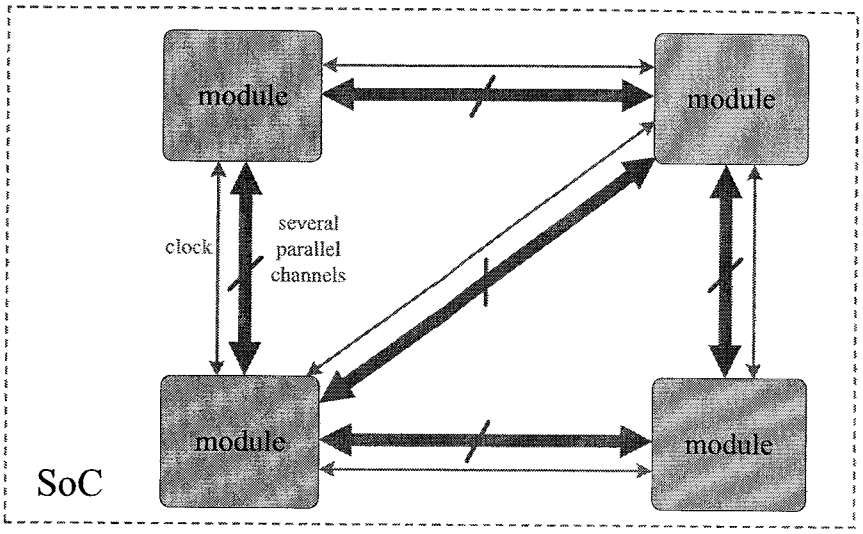


Figure 1.2: Point-to-point parallel link based communication in SoC.

However, parallel links have shown potential in short distance applications, such as networking switches, multiprocessor interconnections, and consumer products with extensive multimedia applications [7]. Alternatively, serial links have the ability to prolong the length of the data transmission channel by mitigating the issues of area, crosstalk, and skew. Therefore serial links have become the solution to higher and more efficient data transmission in order to meet the demands and trends of improving the on-chip and off-chip communication, including short and long distance communication and networking markets [5, 6].

A relatively analytical study [5], which extends a previous study [8], has been conducted in which comparing in terms of area and power serial to parallel links that have been implemented in various feature size of CMOS technologies. As illustrated in Figure 1.3, for any particular feature size of the CMOS technology, there is a limiting value of the link length above which it is better to implement the link as serial rather than parallel because it is more advantageous in term of power and area. This limiting value that defines the frontiers between the two types of the link implementations is scaling down as the relative scaling down of the CMOS technology feature size. In addition, the study shows that the serial links will become more attractive for shorter links in future technologies. Since future large SoC designs should employ serial links to mitigate the cost of communication in terms of area, bit-rate, power and latency.

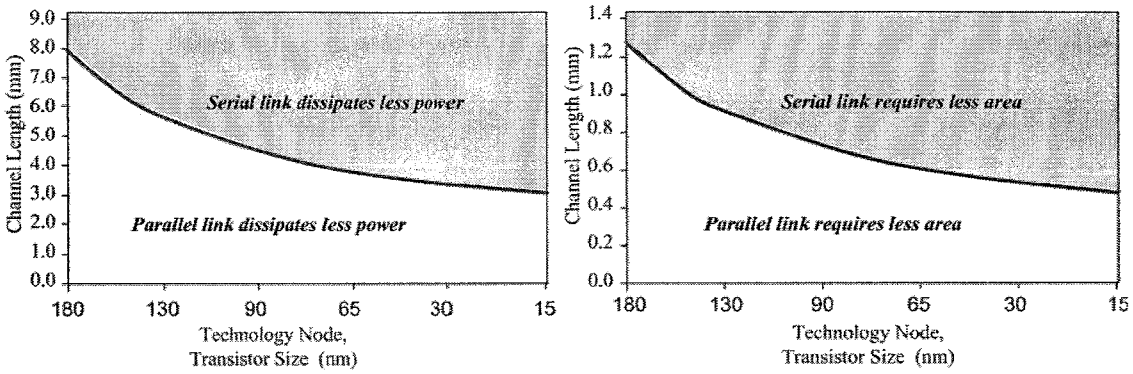
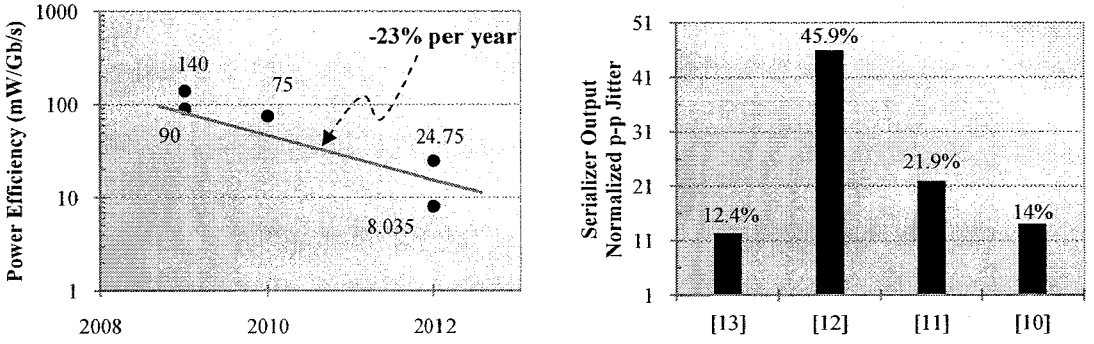


Figure 1.3: Area and power for serial and parallel links versus technology node [5].

However, improving the serial links is always constrained by increasingly strict power budget and noisy on-chip environment. This has prompted extensive research efforts in the direction of improving the power consumption and the jitter performance, to cope with the increasing performance requirement. For instance, the specific power efficiency (expressed in mW/Gb/s) and jitter performance of various recently published serial links [9-13] are shown in Figure 1.4.



(a) (b)

Figure 1.4: (a) Power efficiency and (b) jitter performance of recently published serial links for chip-to-chip communication.

Until recently (as of 2013), the improvement in the power efficiency is approximately only 23% per year. This slow improvement can prevent fully utilizing the on-chip computation capacity of the cutting-edge SoCs (e.g. Ivy Bridge), which require several terabits-per-second (Tb/s) of off-chip communication bandwidth. The designs reported in [9-13] are implemented using either analogue or mixed signal techniques. To mitigate such high power dissipation, the fully-digital implementations have recently emerged as a viable alternative to their analogue counterpart [14].

1.4 Why Less Analogue Content Is Better?

Recently, there has been a growing demand for developing new semi- or fully-digital architectures that obviate the need for any analogue or off-chip components. Indeed, moving towards reducing the analogue content in favour of expanding the digital content becomes more attractive for the following significant advantages:

1.4.1 Less Power Consumption

The CMOS (complementary metal oxide semiconductor) inverter is the basic building block for digital circuit design. As shown in Figure 1.5, the inverter performs the logic operation of A to $\overline{A}$. When the input of the inverter is logic low, the p-channel transistor is on and the n-channel transistor is off, causing the load capacitance to charge to V_{dd} and pulling the output to a logic high, and vice versa.

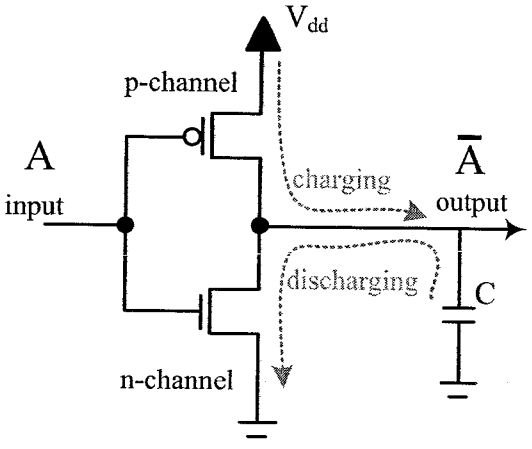


Figure 1.5: Basic CMOS gate.

The power consumption of a digital gate is typically broken down into its static and dynamic power dissipation. The static power occurs during signal transitions, for example from off to on, when both the n-channel and p-channel transistors are on for a very small period of time causing a direct path between the V_{dd} and the ground (short circuit), hence it is significantly negligible [15].

The dynamic power consumption is essentially resulted from charging and discharging the load capacitance. The charge stored by the capacitor (Q) can be derived as:

$$Q = C.V_{dd} \tag{1.1}$$

where C is the channel capacitance being switched per clock cycle, V_{dd} is the supply voltage. The current flow across the channel can be obtained as:

$$I = Q/T = C.V_{dd}.F \tag{1.2}$$

where T is the time required to charge and discharge the capacitor (i.e. the propagation delay) and F is the switching frequency. Then the dynamic power consumption can be broken down as follows:

$$P_{CMOS} = P_{CMOS\_dyn} = I.V_{dd} = C.V_{dd}^2.F \tag{1.3}$$

Similarly, the total power consumption associated with a linear chain of N identical CMOS gates is given by:

$$P_{CMOS} = N.C.V_{dd}^2.F \tag{1.4}$$

To normalize the total power for a given frequency, the power-delay product is given by:

$$PD_{CMOS} = N.C.V_{dd}^2 \tag{1.5}$$

On the other hand, MOS current-mode logic (MCML) are more popular for implementing analogue and mixed-mode circuits [16]. The MCML inverter gate is shown in Figure 1.6. When the input is logic high, the transistor in the left side is on and the transistor in the right side is off. Ideally, all current flows from the load resistance through the left side and drains through the tail current source, pulling down the output to logic low and its complement to logic high. The high output voltage is V_{dd}, while the low output is given by V_{dd}-ΔV, where ΔV is the voltage swing drop across R ($\Delta V = R.I_{tail}$). Normally a swing of 30% of V_{dd} is used [17-19].

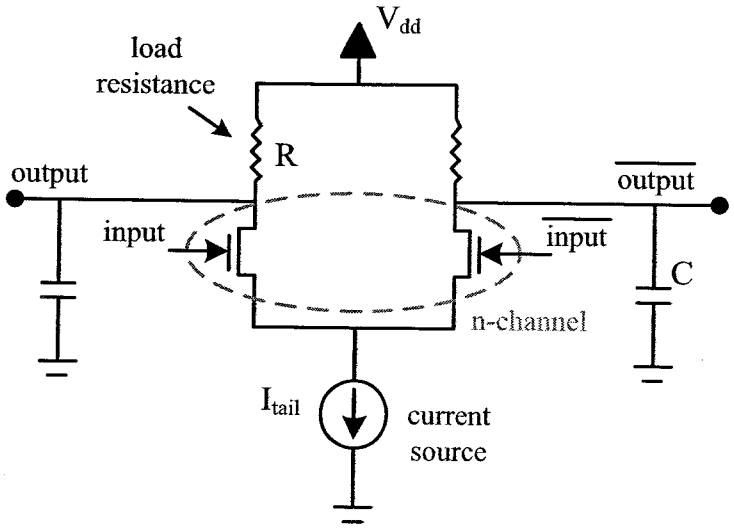


Figure 1.6: Basic MCML gate.

The MCML inverter gate is drawing a constant current over time and independent of the switching activity. Therefore, the amount of power consumption is always the same and hence the sum of the dynamic and static power is constant. This is due to the fact that MCML consumes static power, even when not switching, due to the constant current source. The total power consumption for the same number of identical gates, N, can be obtained as follows [19]:

$$P_{MCML} = N \cdot I_{tail} \cdot V_{dd} \tag{1.6}$$

The total propagation delay of the chain is given by:

$$D_{MCML} = N \cdot C \cdot R = \frac{N \cdot C \cdot \Delta V}{I_{tail}} \tag{1.7}$$

Thus, the power-delay product can be given by:

$$PD_{MCML} = N^2 \cdot C \cdot \Delta V \cdot V_{dd} = 0.3 \cdot N^2 \cdot C \cdot V_{dd}^2 \tag{1.8}$$

Using (1.5) and (1.8), for a given F, C, and V_{dd}, the power consumption of N identical MCML gates is **0.3N** times larger than their counterparts of CMOS gates. This theoretical expression can be verified by the simulation results presented in [19].

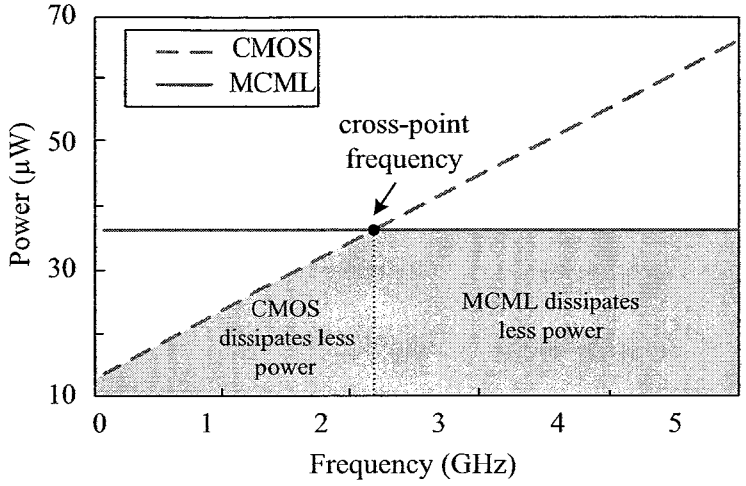


Figure 1.7: Power versus frequency in CMOS and MCML inverters for a CMOS 0.18-μm process, V_{dd} = 1.8 V [19].

1. As shown in Figure 1.7, while the power consumption of CMOS logic increases linearly with the operating frequency, the MCML logic counterpart is irrespective of the frequency.

2. After the cross-point frequency, the power consumption of CMOS logic becomes greater than that of MCML. However, the cross-frequency will increase with the technology scaling down. For example, in 180 nm CMOS process the cross-frequency of the inverter is about 2.2 GHz, while in 45 nm process it is about 9.1 GHz [20].

Thus, the first advantage is that, with the technology scaling down, the fully-digital implementations are ideally suited as the engine for both high performance systems and low power consumption [21].

1.4.2 Less sensitivity to PVT variations and exhibits less phase noise

Second of all, analogue circuits are sensitive to process, supply voltage, and temperature (PVT) variations [22]. Thus integrating such circuits in a noisy digital SoC environment is challenging and must be redesigned for each new technology. On the contrary, the all-digital implementations have less susceptibility to the PVT

variations and offer higher noise immunity [23, 24]. Further, they exhibit better phase noise performance and thus better jitter performance.

Noise is of major concern in serial links. This can be simply understood as a consequence of the effect of introducing even small noise into any part of the link (e.g. oscillator), which leads to dramatic changes in its phase noise (frequency domain) and timing jitter (time domain). A relatively analytical study in [25] derives expressions for phase noise of MOS differential (MCML) and single-ended (inverter chain) ring oscillator. The expressions are validated by simulation and measurement. The phase noise of the inverter-based ring oscillator at an offset frequency of Δf can be expressed as:

$$\mathcal{L}_{CMOS}(\Delta f) \approx \frac{8}{3\eta} \cdot \frac{kT}{P} \cdot \frac{V_{dd}}{V_{char}} \cdot \left(\frac{f_o}{\Delta f}\right)^2 \tag{1.9}$$

where η is a proportionality constant ($\approx$1), k is the Boltzmann's constant (1.38×10^{-23} Joule/Kelvin), T is the absolute temperature in Kelvin, P is the total power dissipation, V_{char} is the characteristic voltage of the device, and f_o is the oscillation frequency. Similarly, the expressing for the phase noise of the differential ring oscillator is:

$$\mathcal{L}_{MCML}(\Delta f) \approx N \cdot \frac{8}{3\eta} \cdot \frac{kT}{P} \cdot \left(\frac{V_{dd}}{V_{char}} + \frac{V_{dd}}{R_L I_{tail}}\right) \cdot \left(\frac{f_o}{\Delta f}\right)^2 \tag{1.10}$$

where N is the number of stages in the ring. Subsequently, the relationship between the phase noise and jitter, for both CMOS and MCML ring oscillators, is derived in [26] and can be expressed as:

$$\sigma^2 = \frac{\mathcal{L}(\Delta f) \cdot \Delta f^2}{f_o^3} \tag{1.11}$$

However, the above expressions for the phase noise and jitter draw the following important conclusions:

3. The jitter and phase noise of an CMOS ring oscillator are found to be independent of the number of stages, N. However, for an MCML ring

oscillator, the phase noise and jitter values grow with the number of stages for a given power dissipation, P, and frequency of oscillation, f_o.

4. For a given P, N, and f_o, the phase noise and jitter of an MCML ring oscillator are approximately $\mathbf{N \cdot [1 + V_{char}/R_L I_{tail}]}$ times larger than their counterparts of an CMOS oscillator.

1.4.3 Cost-effectiveness and size-reduction

The third advantage is that the cost for the digital implementation is much less than the cost for its analogue counterpart. For instance, most of the analogue implementations of phase locked loops (PLLs) and delay locked loops (DLLs) require large capacitors in their loop filters while the all-digital implementations realizes the loop filter as a digital circuit (e.g. counter or finite state machine), without any passive components. This can be achieved by eliminating the need to convert the phase information to voltage or current, and hence it can greatly minimize the required area [27].

1.4.4 Better scalability and seamless programmability

The last but not least, by obviating the need for any analogue or off-chip components (including passive components), all-digital implementations can offer better scalability (portability) across multiple technologies and platforms with inconsiderable modifications. In addition, describing the circuit in hardware description languages gives it a high flexibility to program all design parameters in a very short time compared with the analogue designs which need to be re-designed at transistor level for any parameter change. This can radically reduce cost and time-to-market by saving a significant amount of development time.

1.5 Research Objectives

The general objective of this work is to design, implement, and characterize the performance of a low-power low-noise truly-digital serial link system. Among the folds of this objective, a couple of specific objectives need to be achieved that are as follows:

1. To come up with fully-digital high-resolution frequency synthesizer architecture that has the ability to generate multiple clock frequencies.
2. To implement and validate the functionality of a fully-digital data serializer based on the aforesaid synthesizer circuit.
3. To adopt a proof-of-concept strategy for validating the quarter-rate concept that is previously designed in [28].
4. To identify and design a low-power fully-digital quarter-rate clock and data recovery (CDR) circuit based on the validated phase and frequency detectors.
5. To implement and validate the functionality of a fully-digital data deserializer circuit based on the aforesaid CDR circuit.
6. To validate the proposed serial link system using the target FPGA as a proof-of-concept vehicle to verify the fully-digital functionality using Verilog-HDL.

1.6 Scope of Research

The scope of the thesis is illustrated in Figure 1.8. The reasons for these choices have been explained, as appropriate, throughout the thesis.

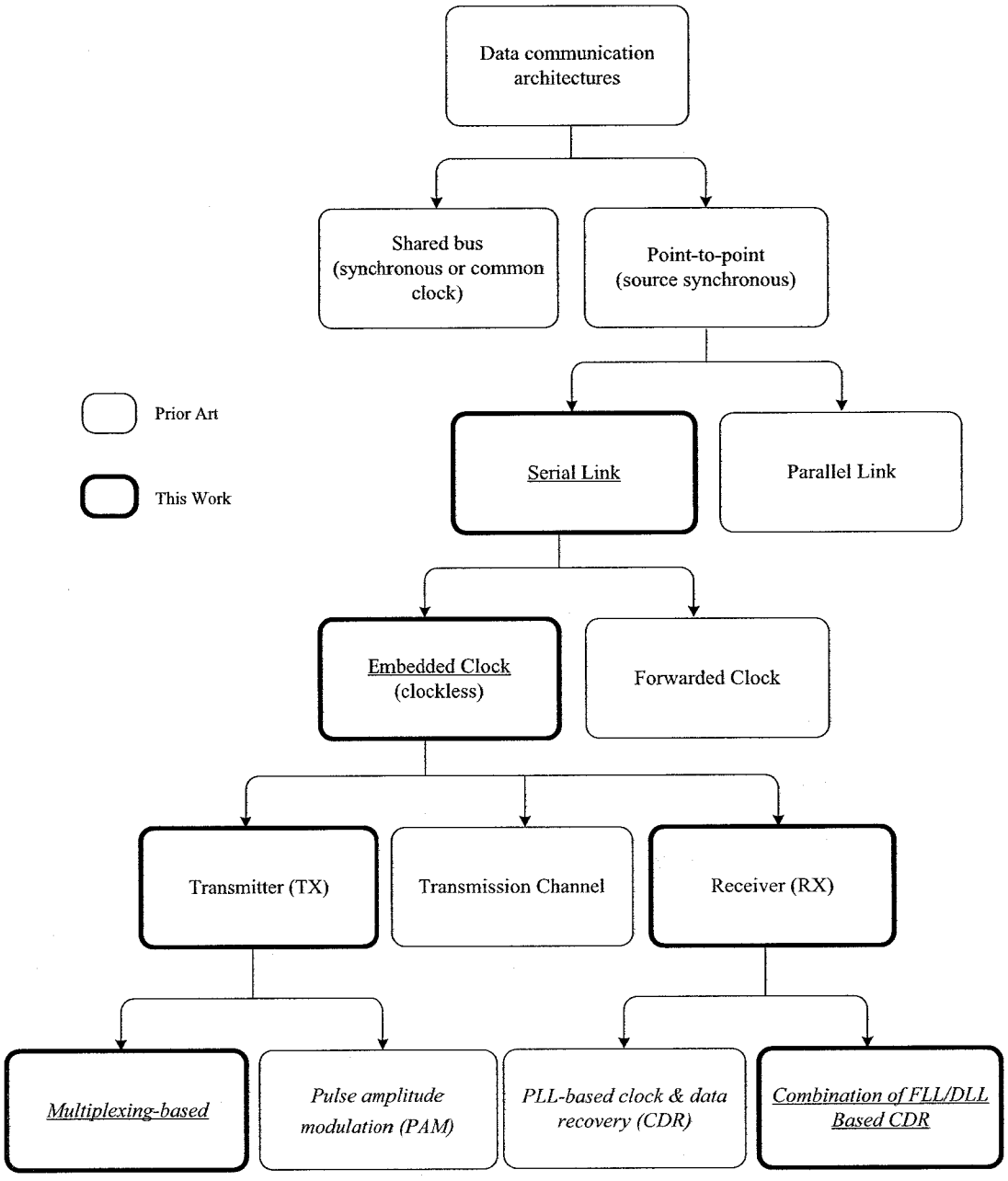


Figure 1.8: Hierarchy and scope of research.

1.7 Research Methodology

This research has evolved through three main stages. The work flow description of each phase is stated as follows:

- First stage:

- Theoretical background on point-to-point link components.
- Literature review on frequency synthesizer circuits.
- Implementing several applicable prior art using Altera Quartus II.
- Proposing and designing a fully-digital frequency synthesizer.
- Integrating the synthesizer circuit as a part of the data serializer and implement it into the target FPGA.
- Characterizing the design for functionality, including measurements for logic utilization, step resolution, peak-to-peak and RMS jitter.
- Authoring of publication for prestigious journal and conference papers.

- Second stage:
  - Literature review on CDR circuits.
  - Validating the functionality of quarter-rate phase and frequency detectors using Altera Quartus II.
  - Designing a fully-digital quarter-rate CDR circuit, including 22.5°-spaced phases generator.
  - Integrating the CDR circuit as a part of the data deserializer and implementing it into the target FPGA.
  - Characterizing the design for functionality, including the same previous measurements.
  - Authoring of publication for prestigious journal and conference papers.

- Third stage:
  - Integrating the proposed serial link components as a two FPGA boards, one for the serializer and the other for the deserializer circuit.

- Thoroughly validating the functionality of the link by sending eight parallel streams to the serializer and compare them to the eight outputs of the deserializer circuit and then measuring the bit-error-rate.
- Carrying out more measurements for logic utilization, power consumption, propagation delay, and jitter performance.
- Authoring of publication for prestigious journal and conference papers.

1.8 Thesis Organization

This thesis is comprised of six chapters including Chapter 1 which is the introduction. The scope of the chapters is summarized in the following text.

Chapter 2 presents different categories of serial link architectures including the main advantages and drawbacks of each category. This chapter also demonstrates a review of literature including the basic link's components and the recent findings of each. At the end of the chapter, a brief description of the serial link performance metrics will be carried out.

Chapter 3 explains the reasons behind the selection of the target hardware platform and its properties. It also provides the design flow and the development tools utilized to design the proposed system.

Chapter 4 introduces proposed fully-digital serial link system in details. This includes a discussion for overcoming the design challenges that arise when considering fully-digital architectures.

Chapter 5 discusses the implementation and experimentally validation of the proposed system. The results obtained from the FPGA implementation are compared with the recent publications after normalization process for a fair comparison.

Chapter 6 summarizes the contribution of the previous chapters. Possible future extensions of the research area are also discussed.

CHAPTER 2

LITERATURE REVIEW

2.1 Chapter Overview

This chapter firstly introduces two of the most important categories of serial link architectures: forwarded clock and embedded clock. The main advantages and drawbacks of each category from an architectural perspective will be highlighted. Secondly, a review of literature including the basic link's components and the recent findings of each component will be briefly demonstrated. The provided background materials are devoted to the development of low-power low-noise fully-digital serial link communications techniques. Finally, a description and estimation of the serial link performance metrics will be carried out. The essential performance metrics are channel data rate, power consumption, bit error rate, and jitter performance.

2.2 A Typical Serial Link

Serial links are a key component in a wide range of applications in modern data communication systems. They are widely used for both on-chip and off-chip data communications [5]. In particular, they are used for on-chip inter-module communication in system-on-chip (SoC) as presented in [29-31]. They are also commonly used for off-chip communications [9-13], such as internet routers, multi-processor systems [32], USB (universal serial bus) [33] and Firewire (IEEE 1394 serial bus standard) [34] that connect peripheral electronic systems to computer, in addition to SATA (serial advanced technology attachment) which communicates the computer motherboard with mass storage devices (e.g. hard disk) and PCI-Express

(peripheral component interconnect) that normally connects cards (sound, video or other) to the motherboard.

Serial links can be generally classified into two categories depending upon the clocking architecture they use, namely, forwarded clock and embedded clock (clockless) architectures. Both classifications typically use source synchronous timing concept, this means that each module transmits the clock information about the transmitted data stream along with it, regardless if the clock information is sent embedded in the serial data stream or separately. Source synchronous concept has the advantage that the clock information and the transmitted data stream are originating from the same source. Thus, transmitter timing variations such as thermal drifts of certain jitter components are present on both, the data content and the clock. This widens the margins to latch the data correctly at the receiving side [1].

In the forwarded clock architecture, the clock information is transmitted along an additional channel, as shown in Figure 2.1(a). Typically, there is delay mismatch between the clock and data stream at the receiver side, due to the impact of driver strength, loading mismatches, and interconnect length mismatches. This adds difficulties to control the clock skew and propagation delay, which requires skew compensation techniques for each channel [35, 36]. Additionally, timing skew between data and a clock signal degrades the maximum data rate [37].

To further increase the data rate, the extra clock channel is eliminated and the clock information is combined with the data stream within the same channel, as shown in Figure 2.1(b). This requires an additional circuit at the receiver side to separate back the clock and data information from the channel and reconstructs the data stream. For an effective extraction method, the transmitted data needs to have enough data transitions, which is usually fulfilled by deploying an appropriate coding or scrambling function [37]. However, the advantages of combining the clock and data information on the same channel can be described as follows: Firstly, this approach allows the data and clock information to experience the exact same noise and jitter during the transmission, and thus the propagation delays and skew are nonissues. This allows embedded clock architecture to transmit data at very high data rates [1, 38]. Secondly, the methodology used to extract data and clock information

has influence on which kind of timing variations can be tolerated on embedded clock architecture and which kind is likely to cause data transmission problems [35]. Thirdly, due to the substantial reduction in the number of interconnects required, the significant saved space allows for better isolation of the channel from its surrounding components leading to integrate more modules and lower crosstalk. However, the price to pay for the aforesaid benefits is the increased complexity of the transmitter and principally the receiver. The main building blocks of serial embedded clock system are extensively explained in the following sections.

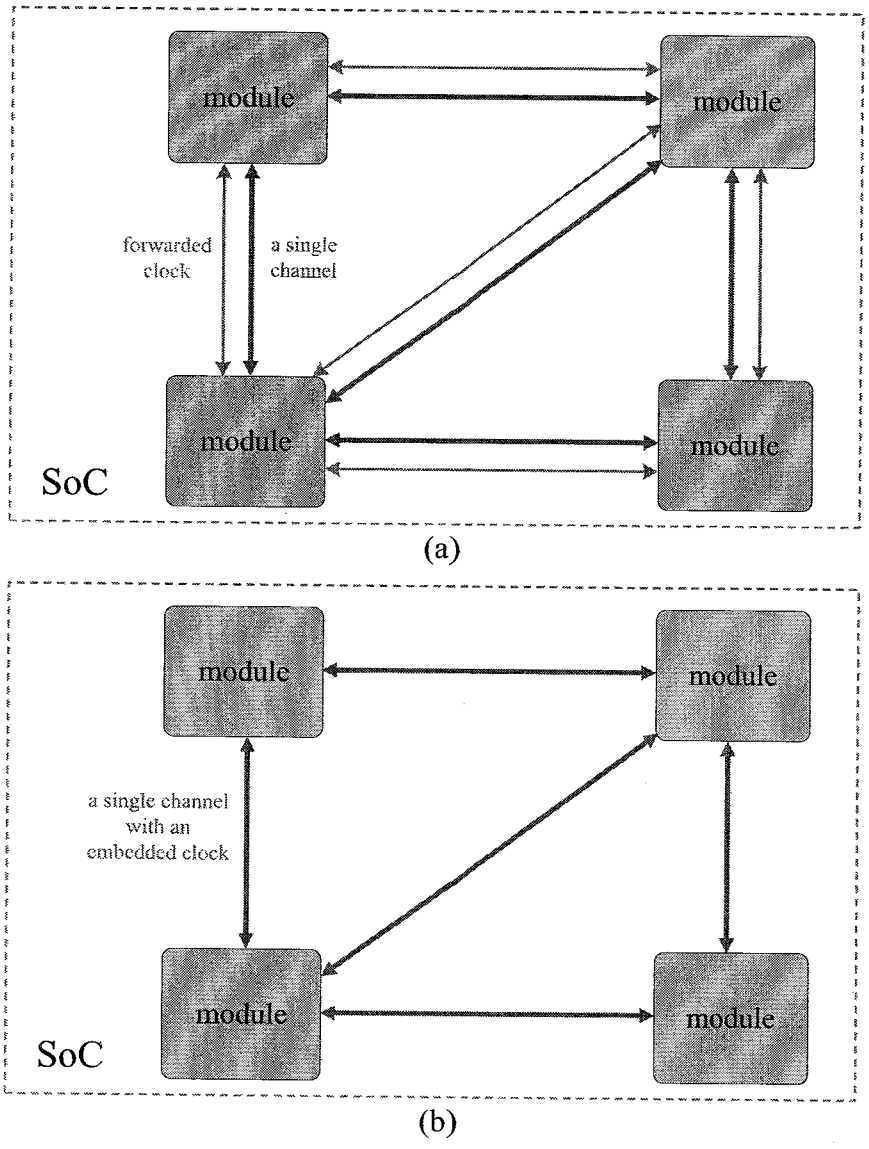


Figure 2.1: Serial link based communication in SoC: (a) forwarded clock architecture and (b) embedded clock architecture.

2.3 The Key Elements of a Link

A high-level representation of an embedded clock serial link is illustrated in Figure 2.2. The serial link system typically comprises, beside the channel, two main circuits, namely, transmitter and receiver.

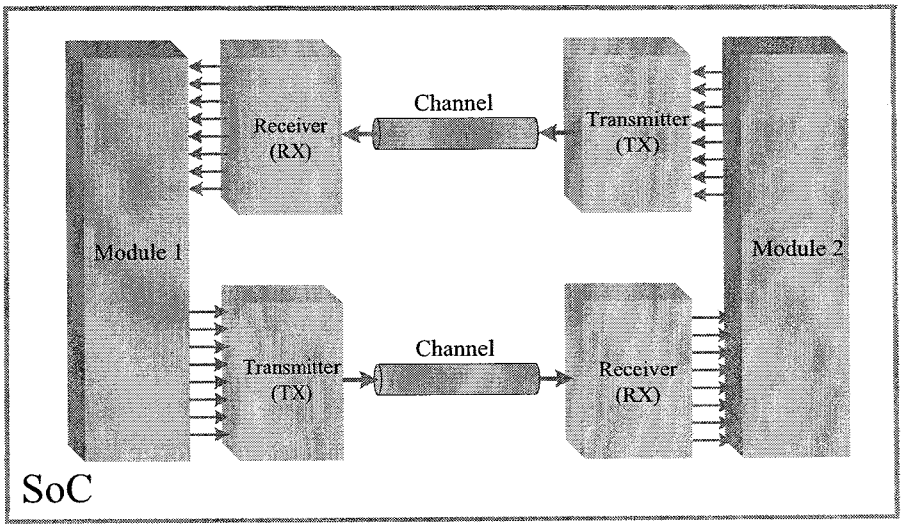


Figure 2.2: Block diagram of an embedded clock serial link.

2.3.1 Transmitter Circuit

The transmitter circuit starts normally with serializing multi parallel data streams into a single serial data stream. The serialized data is then transmitted through the channel to the receiver. The transmitter circuit can be realized using two different methods, namely, pulse amplitude modulation (PAM) and multiplexing.

2.3.1.1 Pulse Amplitude Modulation (PAM) Transmitters

In PAM transmitters, the amplitude of the input clock signal is modulated based on the input bits. For Instance, as illustrated in Figure 2.3, to transmit 2 bits in one symbol time, there are four possibilities (00, 01, 10, and 11), and thus four-level PAM (4-PAM) transmitter is required. This approach reduces both the required channel bandwidth and the maximum required input clock frequency [39]. On the other hand,

increasing the number of bits to be transmitted in one symbol time (e.g. 3 bits) requires applying higher number of PAM levels (e.g. 8-PAM in [40]). This requires more complex circuitry, including digital-to-analogue (DAC) and analogue-to-digital (ADC) converters to generate and detect the multi-level signal, which increase the power consumption and the occupied area. Further, as a result of the technology scaling, the supply voltage continues to drop, which makes it more difficult and challenging to increase the number of PAM levels and hence limits its spreading.

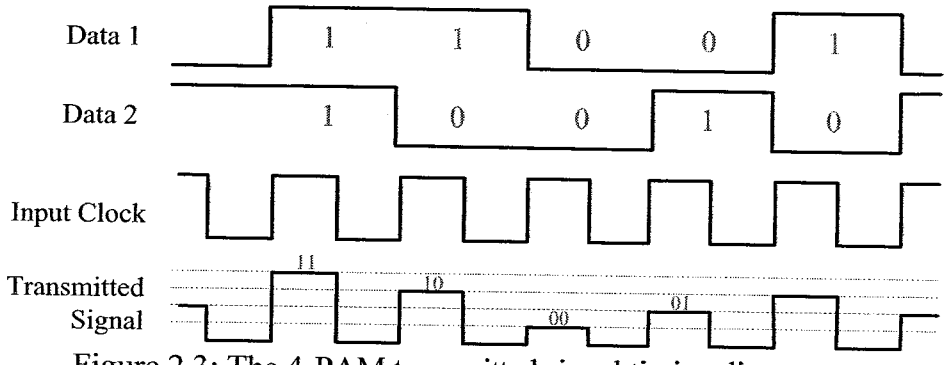


Figure 2.3: The 4-PAM transmitted signal timing diagram.

2.3.1.2 Multiplexing-based transmitters

The most commonly used architecture as a serial link in industry is the serializer/deserializer (SERDES) architecture. The serializer (i.e. transmitter) circuit in this approach is realized using two main blocks, namely, an K-to-1 serializer and a frequency synthesizer, as shown in Figure 2.4. The K-to-1 serializer circuit starts normally with converting the data from parallel to serial by multiplexing multi low-frequency parallel data streams, K, into a single higher frequency serial data stream, in one or more stages. The high-frequency serialized data is then transmitted through the channel to the deserializer (i.e. receiver). In order to properly perform the multiplexing operation, multiple clock frequencies of the input data streams are needed. The required maximum clock frequency is determined by the multiplication of the number of parallel data streams and the input bit rate. This is normally achieved internally using frequency synthesizer circuit. The frequency synthesizer is also required to have minimal skew relative to the system reference clock, minimal phase noise accumulation, and finest frequency resolution.

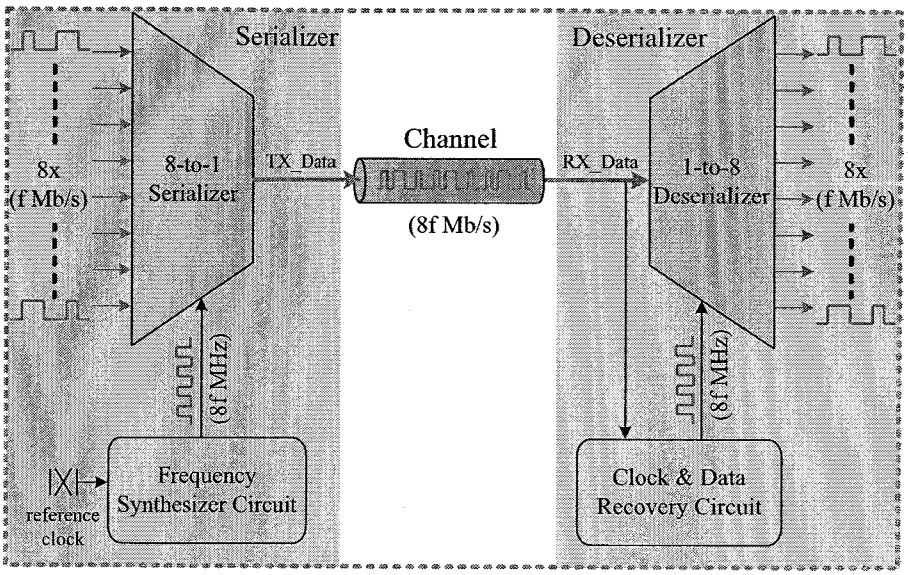


Figure 2.4: Simplified top-level block diagram of a SERDES system.

2.3.2 Frequency Synthesizer Circuit

Frequency synthesizer (also called frequency multiplier) can be shown simply as a local oscillator that has one or more high-frequency output clock signals. Conventional frequency synthesis techniques existing today can be classified as the following two types: Direct-digital frequency synthesizer (DDFS) and PLL/DLL-based frequency synthesizers.

2.3.2.1 Direct-Digital Frequency Synthesizer (DDFS)

This technique involves the generation of a waveform from a series of digital numbers (This block is commonly referred to as a numerically controlled oscillator (NCO)) which are then converted into an analogue signal output (sine or cosine wave). DDFS architecture typically comprises a phase accumulator followed by a ROM look-up table and a linear digital-to-analogue converter (DAC), as shown in Figure 2.5. In some cases, the output signal is filtered by a low-pass filter, which is not normally considered as a part of the DDFS. When compared to the other classifications of frequency synthesizers, DDFS provides the advantages of both fast frequency switching and fine frequency tuning resolution. And hence they are best suited for use in spread-spectrum applications that require tuning capability to different output

frequencies with extremely fine frequency resolution and switching speed of the order of nanoseconds [41]. However, the fast frequency switching is mainly due to its open loop architecture, while enhancing the frequency resolution can be normally achieved using a large accumulator and a huge ROM look-up table. Since the phase to amplitude mapping algorithm is stored in the ROM, the size of the ROM doubles with every 1-bit increment in resolution [42]. Further, the larger accumulator and ROM require a large number of input clock cycles to complete a single output cycle. Together these create two major drawbacks, high power consumption (i.e. several hundreds of mW as reported in [42]) and low maximum clock frequency [43], resulting in reduced usability for portable wireless communication applications.

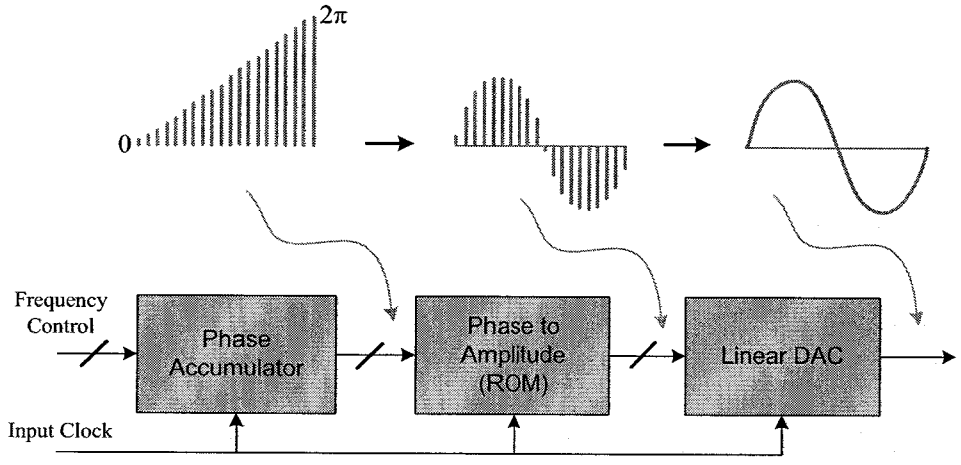


Figure 2.5: Simplified block diagram of a direct-digital frequency synthesizer.

2.3.2.2 PLL and DLL-based Frequency Synthesizers

Over the years, the phase-locked loops (PLLs) and delay-locked loops (DLLs) are widely employed in the data communication systems including, but not limited to: the implementation of the frequency synthesizer, clock synchronization (de-skewing), and clock and data recovery circuits [44, 45]. A PLL is a feedback system that generates an output clock signal whose phase and frequency are simultaneously related to their counterpart of an input reference signal. The PLL is considered to be in the locked state if the phase of these signals is synchronized (they have the same phase, or a constant difference), and hence it is called a "phase-locked loop". Another closely

related circuit is the DLL, which dynamically aligns its output clock signal with a reference clock signal.

Traditionally, PLLs and DLLs are implemented using either analogue or mixed signal techniques with some off-chip components. In particular, these techniques heavily rely on the utilization of analogue voltage-controlled oscillators (VCOs), analogue loop filters, large capacitors, charge pumps, and analogue-to-digital/digital-to-analogue converters (ADCs/DACs). Recently, there has been a growing demand for developing new fully-digital or semi-digital architectures that obviate the need for any analogue or off-chip components.

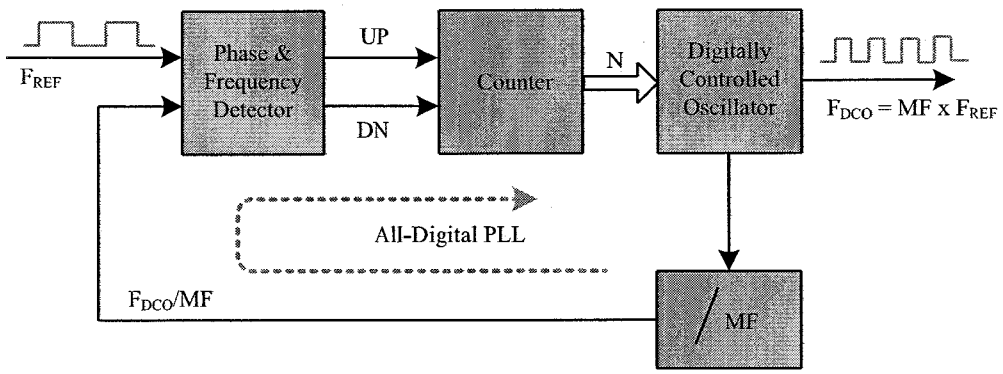


Figure 2.6: Basic block diagram of the all-digital PLL-based frequency synthesizer.

The basic block diagram of a common implementation of an all-digital PLL-based frequency synthesizer is shown in Figure 2.6. Its basic operation requires mainly four blocks: a phase and frequency detector (PFD), a loop filter, a digitally-controlled oscillator (DCO), and a frequency divider. On the other hand, The DLL-based frequency synthesizer can be considered as a PLL that replaces the DCO by a digitally-controlled delay line (DCDL), as shown in the bottom of Figure 2.7. When the loop is in the locked condition, the input and output of the DCDL are in-phase. Whereas the DCDL total delay equals the time period of the reference clock and the output phases of every delay stage are equally spaced [46]. These phases are used by the edge combiner circuit to produce the high-frequency output signal.

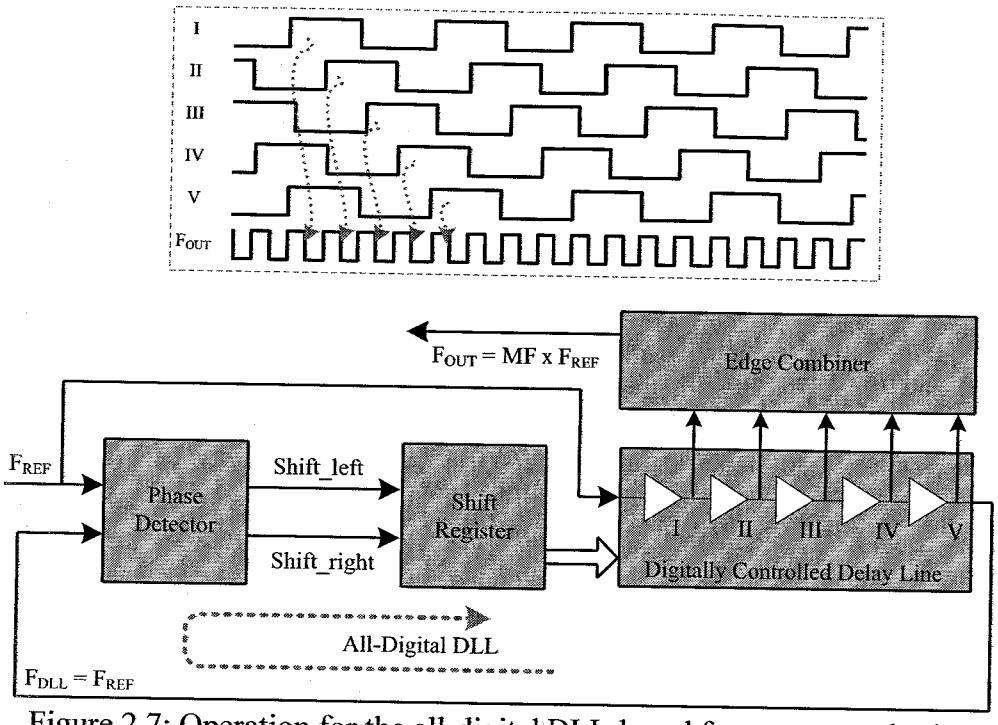


Figure 2.7: Operation for the all-digital DLL-based frequency synthesizer.

At first glance, most of the all-digital PLL and all-digital DLL's components are somewhat similar in topology. But in fact, they have very different properties; given an identical noise environment and circuit components, the all-digital PLL has higher jitter than the all-digital DLL due to phase noise accumulation process [46, 47]. Consequently, several all-digital implementations of the PLL have been proposed to enhance the jitter performance. The implementations could be roughly categorized into two types. The first type is an all-digital cell-based architecture [48, 49] where two digitally controlled oscillators (DCOs) are used to effectively decrease the clock jitter. The inner DCO is used for closing the loop and tracking the reference clock, while the outer DCO is used for generating the output clock based on averaging the output of the inner DCO's controller. However, the power consumption and chip area are greatly increased. The second type [22, 50] utilizes a time-to-digital converter (TDC) as a digital filter to increase the resolution of the phase error measurement and hence decrease the jitter performance. Meanwhile, all-digital implementations of DLLs suffer as well from two major drawbacks. First, the multiplication ratio of the reference clock signal depends mainly on the number of delay cells in the delay line. Second, any mismatch in the edge combining logic will be translated directly into a duty-cycle error and jitter [47]. The aforesaid approaches of enhancing the jitter performance of the

PLLs/DLLs have significantly necessitated performing more analytical studies to analyze the performance of the PLLs, DLLs, and dual-loop based frequency synthesizer architectures in a comparable environment. For instance, the analytical studies in [47, 51] show that while the DLL-based frequency synthesizer outperforms the PLL-based in term of rejecting the on-chip noise, the latter is better suited for rejecting the noise of the input reference clock. According to relatively recent studies in [52-54], the dual-loop architectures have shown a potential in attenuating both the on-chip and input clock noise, and they do not have the accumulated jitter issue.

2.3.3 Receiver Circuit

Conceptually, the deserializer circuit performs the inverse function of the serializer block. The deserializer extracts the embedded clock information from the incoming data stream, re-samples the noisy stream using the extracted clock for jitter removal and synchronization. And finally deserializes the single stream back into low-frequency parallel data streams of similar number of the serializer, K, for performing the desired processing operations. These essential functions can be realized using two main blocks, namely, clock and data recovery (CDR) circuit and 1-to-K deserializer circuit.

2.3.4 Clock and Data Recovery (CDR) Circuit

Since the clock is embedded in the data stream, a CDR circuit is used as a part of the deserializer to extract the associated clock information from the incoming noisy data stream and then recover the transmitted data stream. The CDR circuit represents the most critical and challenging task in modern high performance serial link systems [14, 55]. There are a variety of different architectures and approaches for CDR process. However, the most commonly used CDR architectures for embedded clock serial links are referenceless dual-loop topologies [27, 55, 56]. Typically, the dual-loop designs can be roughly categorized into two types: a PLL-based and a combination of frequency-locked loop (FLL) and DLL based CDR.

2.3.4.1 PLL-Based CDR Circuit without Reference Clock

A simple block diagram of such PLL-based CDR circuit is shown in Figure 2.8. The circuit is composed of frequency tracking and phase tracking loops that operate dependently and share common loop filter and voltage controlled oscillator (VCO) [14, 57-59]. In summary, the frequency tracking loop employing a frequency detector (FD) to provide a frequency comparison between the received data and the VCO output clock, and hence obviating the need for a dedicated external reference clock signal. When the frequency difference is very small, the phase tracking loop takes over and leads to a phase locking between the recovered clock and the received data through a phase detector (PD). Although it has two loops but since it has a VCO block as the central clock source in addition to a single loop filter, it is referred to as PLL-based architecture.

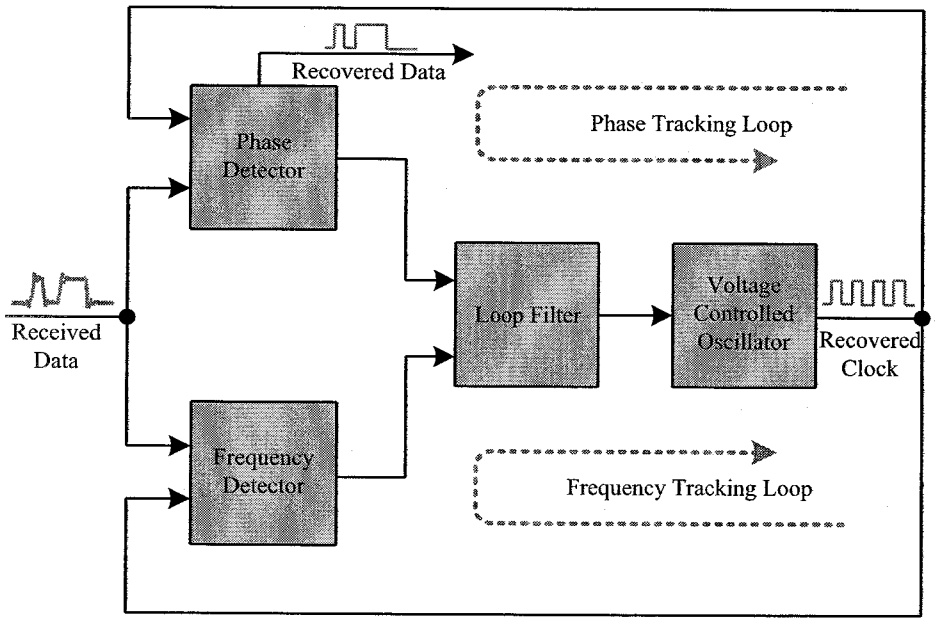


Figure 2.8: Block diagram of a PLL-based CDR without a reference clock.

This architecture entails a possible issue, as the phase loop takes over the control of the VCO from the frequency loop, the two loops may potentially interfere with each other, indeed resulting in a failure to recover the transmitted data stream [60]. Addressing this issue requires a modification in the architecture such that each loop has its own loop filter and operates independently. This introduces the combination of FLL/DLL based CDR architecture.

2.3.4.2 Combination of FLL/DLL Based CDR

It generally consists of a FLL controlling a VCO and a DLL or PLL controlling a phase shifter on either data or clock, operate independently to recover both the clock and the data from the incoming data stream, as shown in Figure 2.9. This can effectively enhance the stability of the system, which derives from the manner in which loop stabilization is accomplished [61, 62]. Further, using a combination of FLL/DLL has more robustness in jitter performance than the PLL-based CDR [27]. It can have a good input jitter rejection as a result of having a short acquisition time for the PLL since the PLL here is only responsibly for a frequency locking [56]. The acquisition time is known as the time interval that is needed for the PLL to achieve the locked condition.

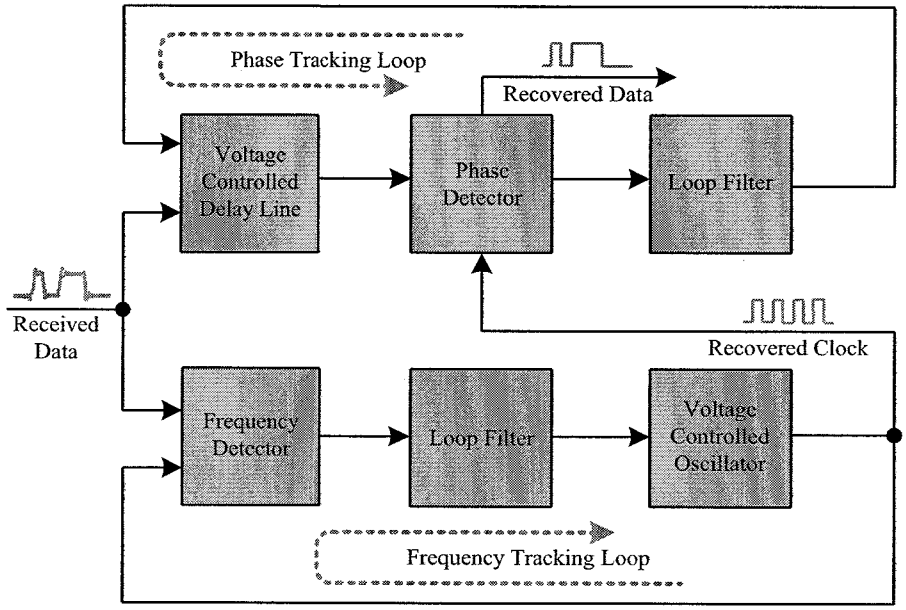


Figure 2.9: Block diagram of a combination of FLL/DLL based CDR architecture.

However, the aforesaid advantages come at the expense of occupying a larger area that is required for the additional loop filter and VCDL. Reducing this area overhead and further increasing the stability of the system require minimizing the analogue content in favour of expanding the digital content, which eventually leads to realize a semi-digital (digitally-assisted) [57, 63] or fully-digital [58, 64] CDR architectures. Nonetheless, the dual-loop CDR designs can be further divided into two major categories depending on whether the data rate and clock frequency are equal or not, namely, full-rate and fractional-rate CDR architectures.

2.3.4.3 Full-Rate and Fractional-Rate CDR Architectures

In full-rate circuits, the location of the data transition is compared with the rising or falling edge of the clock. This requires the clock frequency to be equal to the data rate as illustrated in Figure 2.10(a). Thus, the data recovery process can be achieved using a single flip-flop that operates either on rising or falling edge of the clock signal. On the contrary, a fractional-rate circuit utilizes multiple phases of a clock signal running at a frequency less than the data rate (e.g. half or quarter-rate) in order to properly sample the location of data transition, as illustrated in Figure 2.10(b) and Figure 2.10(c).

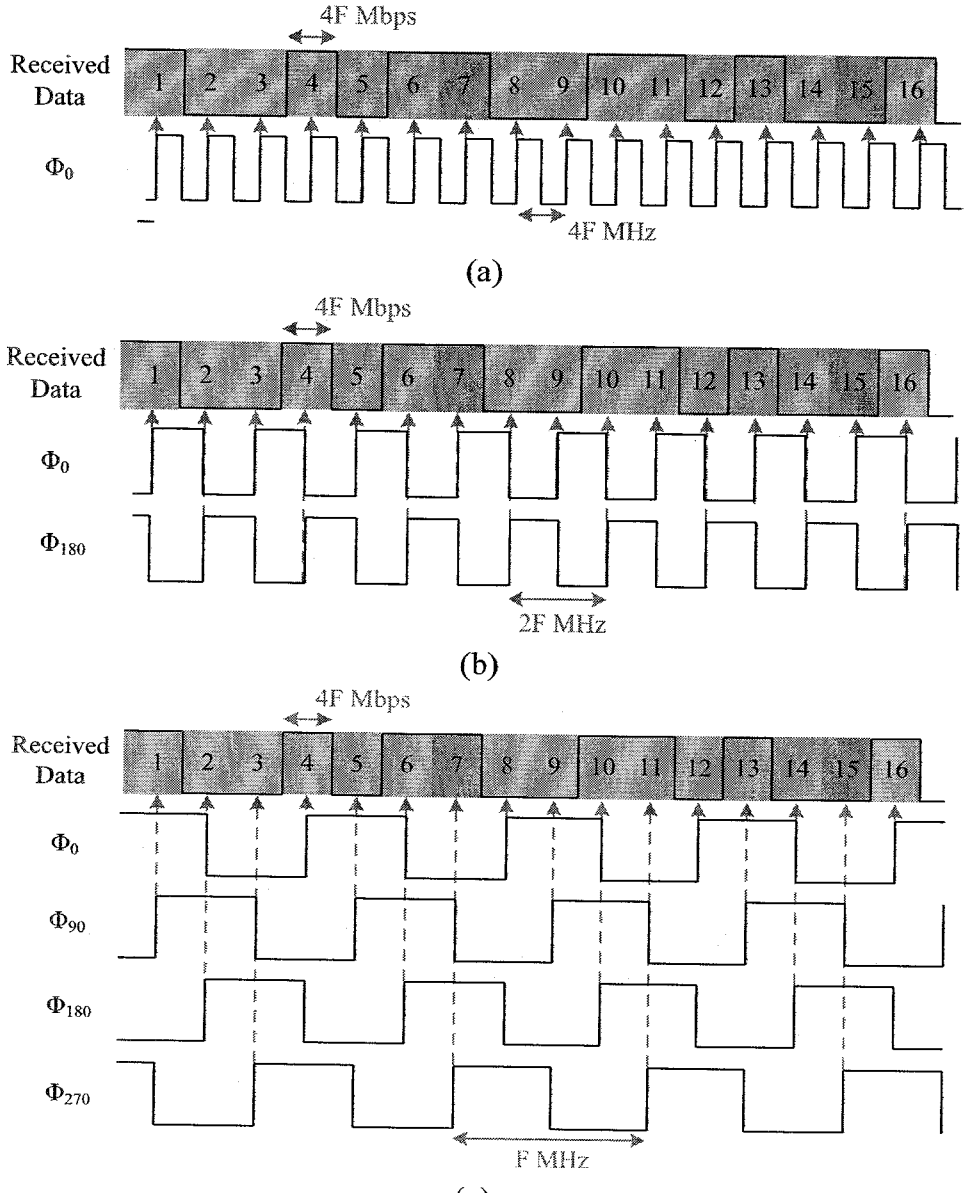


Figure 2.10: Waveforms for (a) full-rate, (b) half-rate, and (c) quarter-rate CDR.

Therefore, its operational principle is to sample the data stream by the rising or the falling edges of the multiple phases using multiple flip-flops. The main advantage of using full-rate CDR circuit is the greater simplicity in the design and the robustness of its functionality. However, it has a disadvantage of requiring a relatively high clock speed that results in a VCO with large power consumption. The CDR circuits generally are the most power-consuming block among other blocks of the serial link [14]. Considering the advantages of serial links, increasing the number of links per chip relies heavily on reducing the power consumption per link. The factors affecting the dynamic power consumption of a digital CMOS circuit can be broken down as follows [65]:

$$P_{CMOS\_dyn} = C \cdot V_{dd}^2 \cdot F \tag{2.1}$$

Where C is the channel capacitance being switched per clock cycle, V_{dd} is the supply voltage and it is a technology dependent (e.g. for CMOS 180 nm and CMOS 130 nm, V_{dd} = 1.8 V and 1.3 V, respectively), and F is the switching frequency. A reduction of each of these parameters results in a reduction of the dissipated power. However, the only practical and realistic way to reduce the dynamic power consumption is reducing the switching frequency without affecting negatively the data rate. This can be achieved by using fractional-rate architectures.

2.3.5 Serial Link Performance Metrics

With the growing number of serial links per chip and greater data rates, testing and characterizing the links have posed significant challenges in terms of testing cost and quality. This section summarizes the key performance metrics used to describe the serial link:

2.3.5.1 Data Rate

The data rate is normally expressed as the total number of transmitted data bits per second over the channel from the serializer to the deserializer.

2.3.5.2 Power Consumption

The power consumption can be broken down into three major components: dynamic, static, and I/O blocks power dissipation.

The dynamic power is the power resulting from the capacitive loads charging and subsequent discharging during the signals toggling, at the operation mode. Thus, the main factors affecting the dynamic power are capacitance charging, the supply voltage, and the clock frequency. On the contrary, the static power is the power consumed when no signals are toggling (at idle mode). It is mainly affected by the number of logic elements (LEs) used in the circuit. More resources usage leads to an increase in the static power consumption.

However, the total power consumption mainly depends on several parameters, including choice of the process technology, environmental conditions (i.e. temperature), and device resource usage.

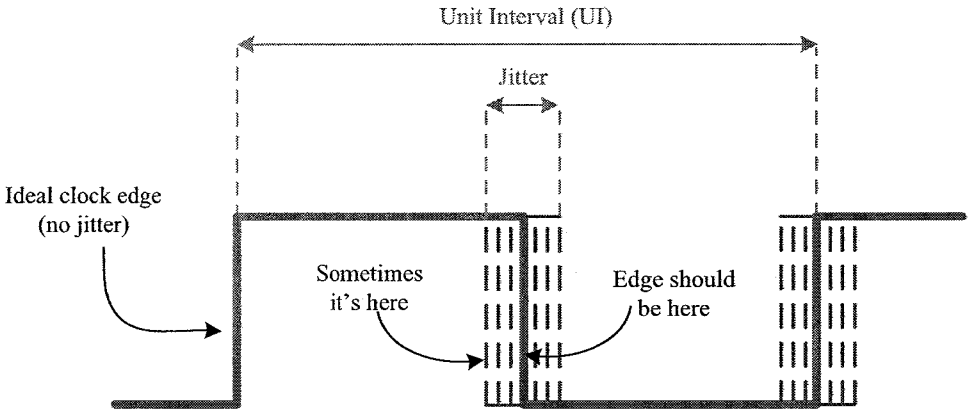
2.3.5.3 Bit Error Rate (BER)

BER is the ratio of the number of bits incorrectly received to the total number of transmitted bits in a given time interval. Most of the communication standards specify a very low target BER of 10^{-12} [66]. This implies that for every one trillion bits transmitted, on average, no more than one error is received. However, recently the concern is no longer only whether there are any errors on the data transmission but also how many bits can be transmitted on average before the first error occurs. This fact requires the use of new techniques to analyze the performance of the serial link such as jitter analysis, data eye diagram, and BER bathtub curve.

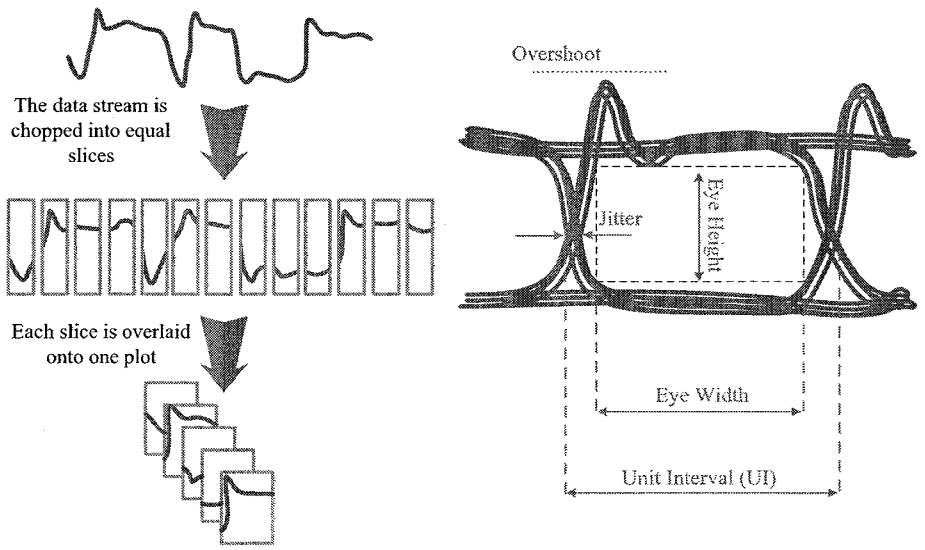
2.3.5.4 Jitter Analysis

Jitter analysis is increasingly significant as it is one of the major potential causes for data being received in error. Jitter is defined as the deviation of a data or clock edge from its ideal position, as illustrated in Figure 2.11. However, more attention must be given to the

data streams, as no transition occurs when the same bit repeats two or more times consecutively. Thus the data stream can be chopped into equal periods and overlaid into one plot, producing the eye diagram as shown in Figure 2.11(b). The eye diagram gives insight into the amplitude behaviour of the waveform as well as the timing behaviour, by measuring the eye height (vertical eye opening) and the eye width (horizontal eye opening), respectively.



(a)



(b)

Figure 2.11: Effect of jitter on (a) clock signal and (b) data stream.

Jitter is usually expressed in picoseconds (ps), but is more often specified in terms of percent of the unit interval (UI). A UI is the ideal time duration of a single bit or clock period. Generally, the origin of this timing deviation has many sources which can either be deterministic or random [67, 68], as listed in Figure 2.12.

Random jitter (RJ) is the accumulation of jitter through the system due to the thermal noise, which cannot be predicted. As the primary source of RJ is the thermal noise that is known to have a Gaussian distribution, RJ can be also modeled by the Gaussian probability density function (PDF) [69]. For instance, an eye diagram for a signal with only random jitter is shown in Figure 2.13. Ideally, its two tails extending away from the center of the histogram asymptotically approach zero, but never fully reach zero. Thus its peak-to-peak value has no limit, and hence, RJ is considered to be unbounded.

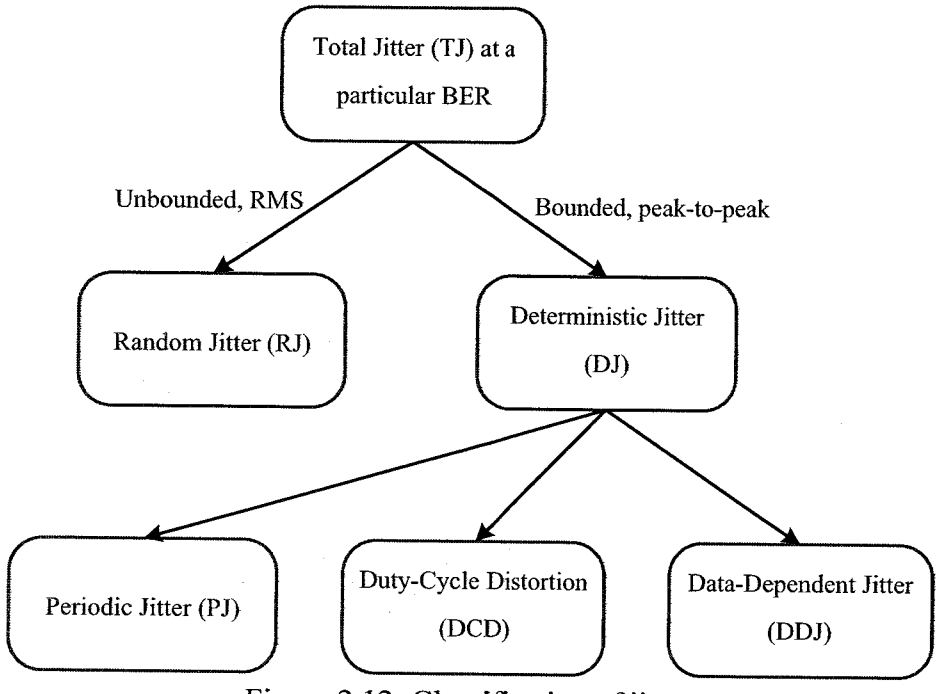


Figure 2.12: Classification of jitter.

Therefore, RJ can be characterized by its standard deviation value (oftentimes called the RMS value). It is the amount of time that extends one sigma (σ) from the mean (μ) on each side. One sigma is equivalent to 34.1% of the total number of samples in the histogram. The mean value is simply the crossing point of the eye diagram or the starting point of its UI.

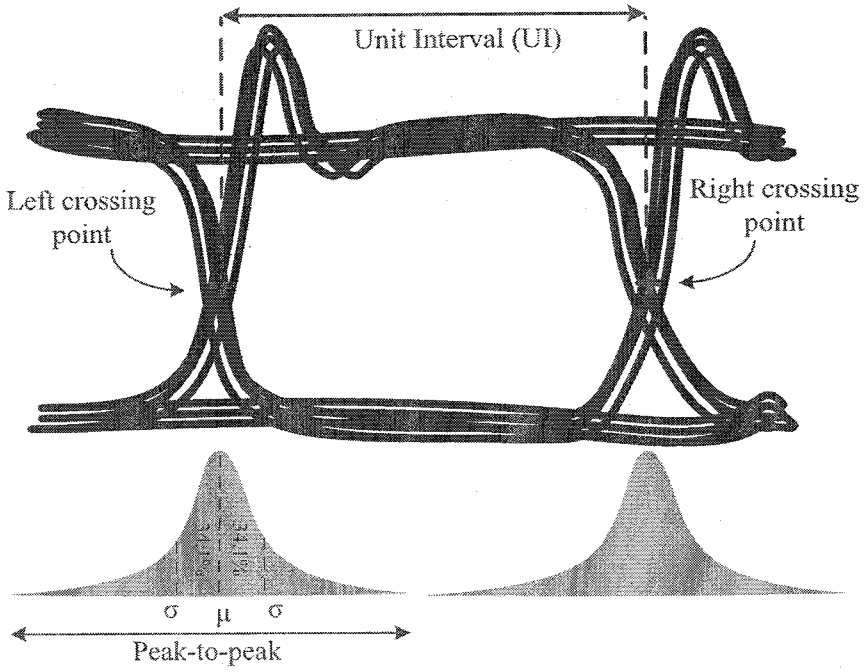
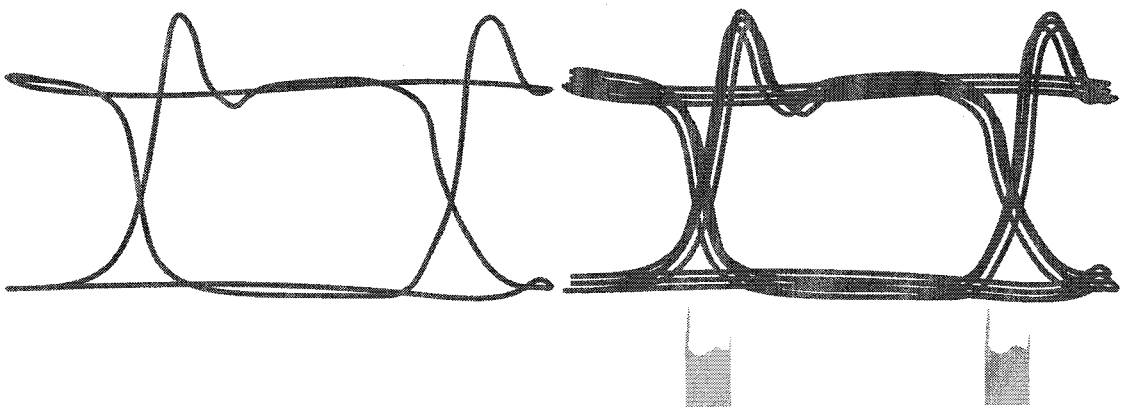
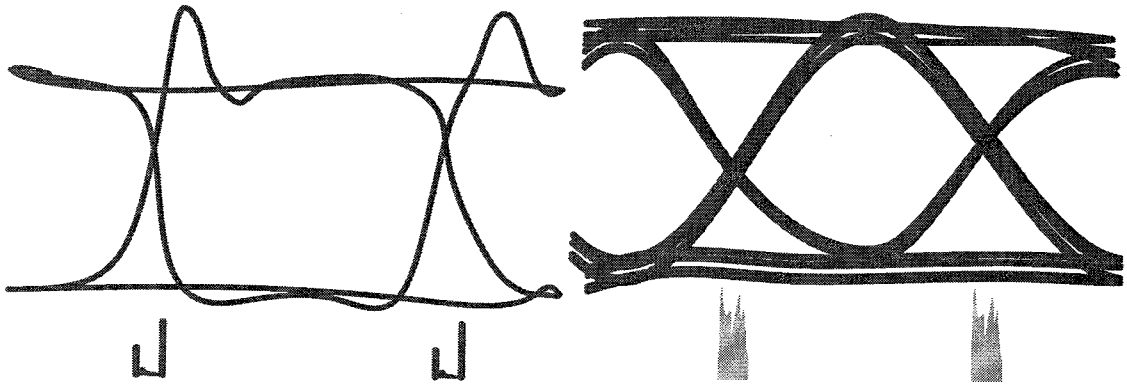


Figure 2.13: Characterization of random jitter in an eye diagram by Gaussian statistics.

Deterministic jitter (DJ) is called "deterministic" because it can be accurately predicted for each edge since it is caused by non-random sources. Unlike the RJ, the DJ can be well characterized by its peak-to-peak value, since it follows a non-Gaussian bounded PDF. As exemplified in Figure 2.14, the DJ PDF can take any form depending on the jitter source [1, 70], namely, periodic jitter (PJ), duty-cycle distortion jitter (DCD), and data-dependent jitter (DDJ). PJ is caused by switching power supply noises. DCD is generated due to the difference in the slew rate of the rising edges from that of the falling edges DDJ is caused by the bandwidth limitation of the transmission channel.



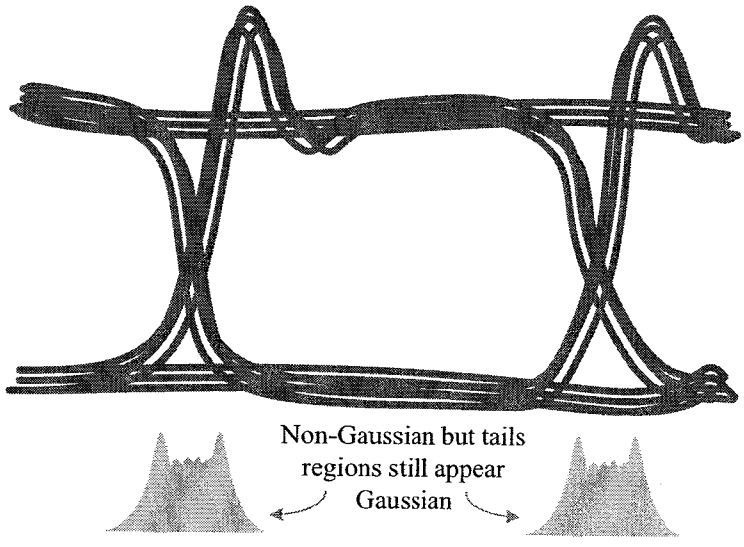
(a) (b)



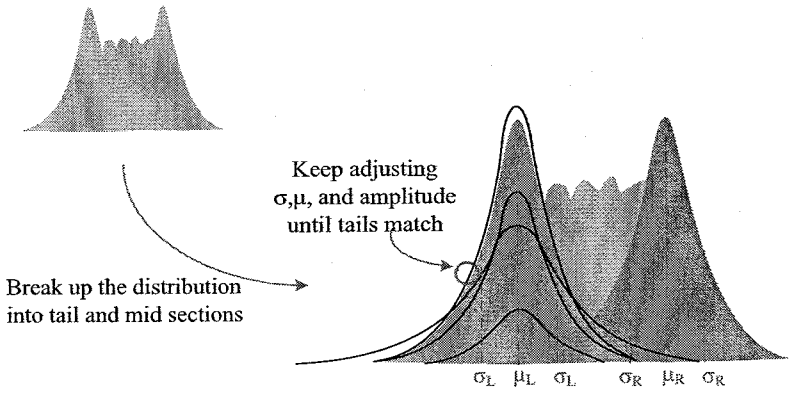
(c) (d)

Figure 2.14: Example of (a) a reference signal data eye and the influence of (b) periodic jitter, (c) duty-cycle distortion jitter, and (d) data-dependent jitter on it.

The common objective of jitter measurement and analysis is to determine if the total jitter (TJ) will cause the serial link to meet or exceed a given BER. In order to obtain the TJ's PDF, the convolution of its RJ and DJ components should be calculated first [68]. An example of the resulting histogram with the presence of DJ and RJ is shown in Figure 2.15(a).



(a)



(b)

Figure 2.15: (a) Characterization of total jitter in an eye diagram. (b) Fitting the histogram with Gaussian curves.

A further processing technique is applied on the resulting PDF to get a useful value of TJ. This technique is normally known as a histogram tail fitting [71]. It seeks to find well-defined Gaussian curves that are the best fit for the tail regions, as illustrated in Figure 2.15(b). The right and left regions are fitted separately and in general have different values of sigma and magnitude. After finding the best fit, the peak-to-peak values of DJ and RJ can be calculated. The peak-to-peak value of DJ is the difference between the two means of the histogram tail fit. In contrast, as the peak-to-peak value of RJ naturally has no limit, it is commonly calculated with respect to a certain BER. Thus the TJ is also calculated at a particular BER. However, the RJ peak-to-peak value is equivalent to the multiplication of the average of the left and right sigma values with the RJ multiplier. The latter is a function of the desired BER, which is given in [71]. Finally, the TJ peak-to-peak value will be the sum of the DJ and RJ peak-to-peak values as a function of BER. The resulting values of TJ are commonly plotted as BER versus UI, which is referred to as bathtub curve.

The TJ at a particular BER can be calculated from the bathtub curve as follows:

$$\textit{Total Jitter (BER)} = \textit{UI-Eye width (BER)} \tag{2.2}$$

Obviously, the RJ is proportional to the BER, so as the BER increases the eye width becomes narrower. The eye width at a particular BER represents the time available for the CDR circuit to recover the received data without a bit error.

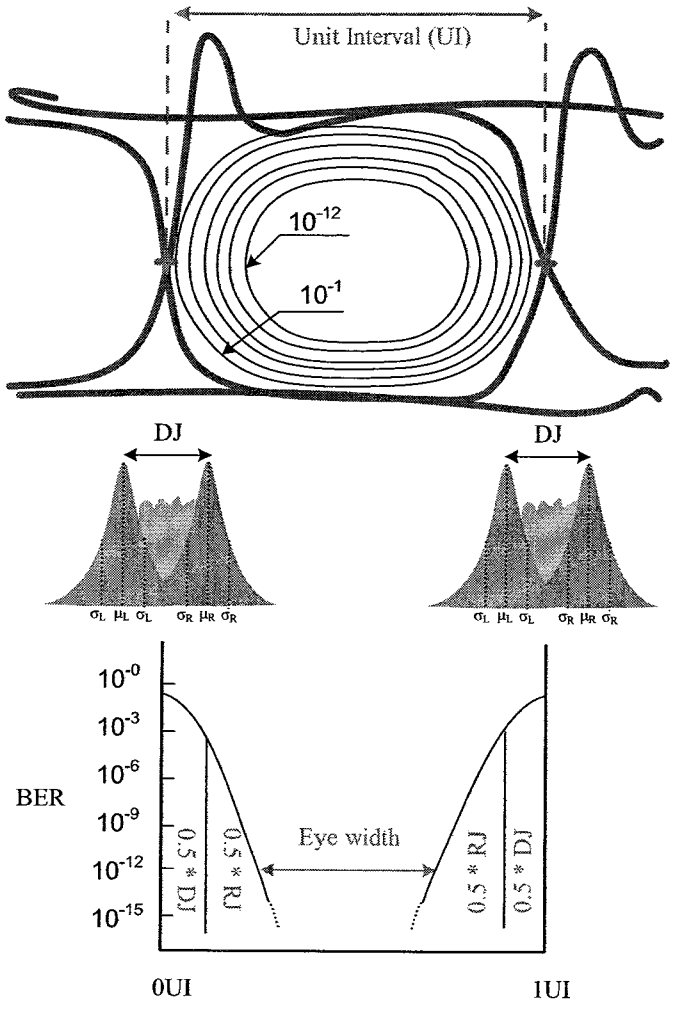


Figure 2.16: Bathtub curve and the total jitter from a bit-error-rate perspective.

In the context, in order to ensure reliable communication, the receiver must tolerate or compensate the presence of jitter up to a certain amount specified by the standard associated to a particular type of communication system. Thus, three important concepts involved in the characterization of a CDR are jitter generation, jitter transfer, and its jitter tolerance functions. Jitter generation refers to the jitter produced by the CDR circuit itself, under locked condition, when the input data stream is jitter free. Jitter transfer refers to the relationship between the jitter applied at the input of the CDR circuit to the resulting jitter at the output of the circuit. Jitter tolerance is defined as how much input jitter a CDR circuit must tolerate without increasing the BER.

CHAPTER 3

FPGA HARDWARE PLATFORM

3.1 Chapter Overview

The intent of this chapter is to explain and justify the reasons behind the selection of the target hardware platform. The required fundamentals and properties of the selected platform with a short description of the platform structure will also be provided. Finally, the design flow and the development tools utilized to design the proposed system will be explained to complete this chapter. Those already familiar with these topics can skip this chapter without loss of continuity.

3.2 Choice of Technology (Target Hardware Platform)

Serial link is a building block for larger SoC systems and most SoC systems are fabricated in CMOS (complementary metal oxide semiconductor) technologies. However, FPGA (field programmable gate array) has become recently the mainstream and key implementation media for almost any kind of digital system. It can be defined as a general-purpose integrated circuit that is configured by the designer rather than the device manufacturer. This means that its functionality can be changed even after fabrication process. It can be normally configured using a hardware description language (HDL). As compared with single-purpose ASIC (application specific integrated circuit) technologies, FPGAs feature several significant advantages that can be outlined as follows [72]:

- ASICs typically require several months to fabricate the desired device. FPGAs on the other hand can be configured or programmed in less than a minute and

can be repeatedly reconfigured if a mistake is made without any additional development costs, and hence they have lower risk.

- ASICs development and fabrication costs are rising aggressively (e.g. from hundreds of thousands to millions of dollars), while FPGAs costs are far much cheaper (e.g. from a few hundred dollars to a few thousand dollars).

- FPGAs offer the ability to realize high-performance and high-speed signal processing due to the inherently parallel processing capability.

- FPGAs significantly achieve faster time-to-market.

3.3 Altera DE2-70 FPGA Development Board

Implementing a designed digital system normally requires selecting a suitable FPGA development board. The selected board features an FPGA chip connected to a variety of interesting peripheral devices such as switches, LEDs, seven segment and LCD displays, and RS-232 serial communication. Although they are not necessary for some applications, they are really helpful for both FPGA user interface and for debugging through the hardware design process.

The most suitable FPGA development board to our needs is selected as the Altera DE2-70 board that is also used widely in many of the universities in the world for both education and research purposes. This is mainly due to the fact that the DE2-70 board is equipped with a sufficiently large number of logic elements (LEs) at the lowest cost and lowest power consumption compared to the competing 90-nm FPGAs [73]. The top view of the board is given in Figure 3.1.

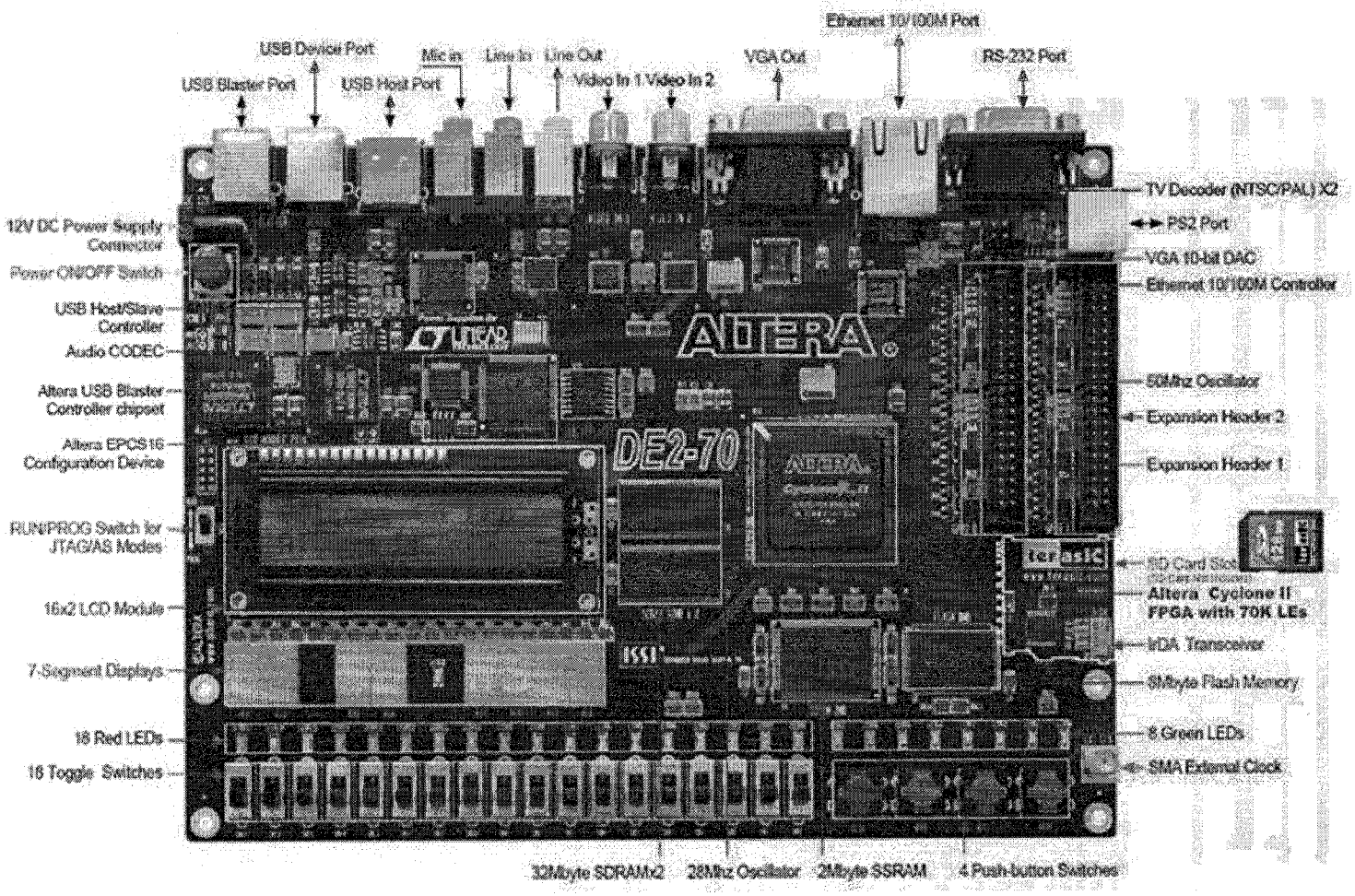


Figure 3.1: Altera DE2-70 FPGA development board [74].

The Altera DE2-70 development and education board is equipped with the following hardware [74]:

- Altera Cyclone II EP2C70F896C6 FPGA device.
- Altera Serial Configuration device - EPCS16.
- USB Blaster (on board) for programming; both JTAG and Active Serial programming modes are supported.
- 2-Mbyte SSRAM.
- Two 32-Mbyte SDRAM.
- 8-Mbyte Flash memory.
- SD Card socket.
- 4 pushbutton switches.
- 18 toggle switches.
- 18 red user LEDs.
- 9 green user LEDs.
- 50-MHz oscillator and 28.63-MHz oscillator for clock sources.
- 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks.
- VGA DAC (10-bit high-speed triple DACs) with VGA-out connector.
- 2 TV Decoder (NTSC/PAL/SECAM) and TV-in connector.
- 10/100 Ethernet Controller with a connector.
- USB Host/Slave Controller with USB type A and type B connectors.

- RS-232 transceiver and 9-pin connector.
- PS/2 mouse/keyboard connector.
- IrDA transceiver.
- 1 SMA connector.
- Two 40-pin Expansion Headers with diode protection.

3.4 Functioning of FPGAs

FPGAs normally consist of a large number of identical LEs surrounded by programmable routing resources that allow these LEs to be connected based on the logic functions defined by the user. This structure is further surrounded by programmable I/O blocks that allow the off-chip connection with any other component on the board, as shown in Figure 3.2.

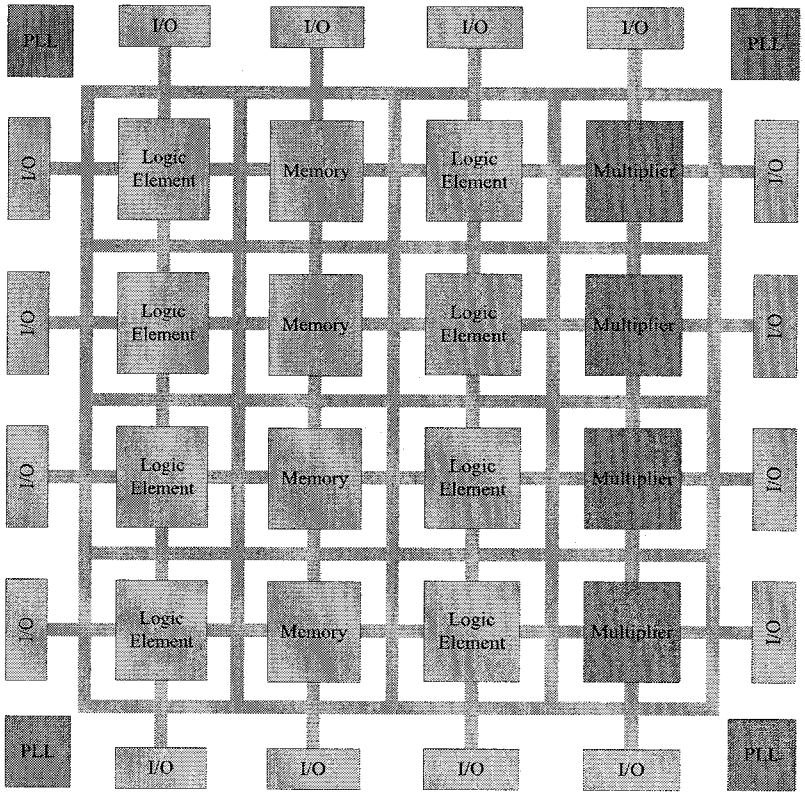


Figure 3.2: Simplified Cyclone II EP2C20 FPGA block diagram.

At power up, the FPGA reads a preloaded configuration bitstream either from a flash memory (i.e. active serial programming) that is permanently programmed or from an SRAM (static random access memory) (i.e. JTAG programming) that requires configuration stream to be loaded each time the FPGA powers up. The bitstream controls the function of the corresponding LE and the routing resources. The LE is the smallest unit of logic in the Cyclone II EP2C70F896C6 architecture. The latter has a total of 68,416 LEs. Each LE features [73]:

- A four-input look-up table (LUT), which is a function generator that can implement any function of four variables.
- A programmable register.
- A carry chain connection.
- A register chain connection.
- The ability to drive all types of interconnects.
- Support for register packing.
- Support for register feedback.

3.5 FPGA Design Flow and Development Tools

The typical design flow of FPGA-based systems is illustrated in Figure 3.3. It consists of five major procedures, namely, design entry, synthesis, place & route, timing analysis, and finally programming the desired FPGA device. After specifying the design requirements, the designer normally defines the required functionality of the system using schematic entry, hardware description language (such as Verilog or VHDL), or combination of both. In this design flow, a combination of Verilog and schematic entry is used.

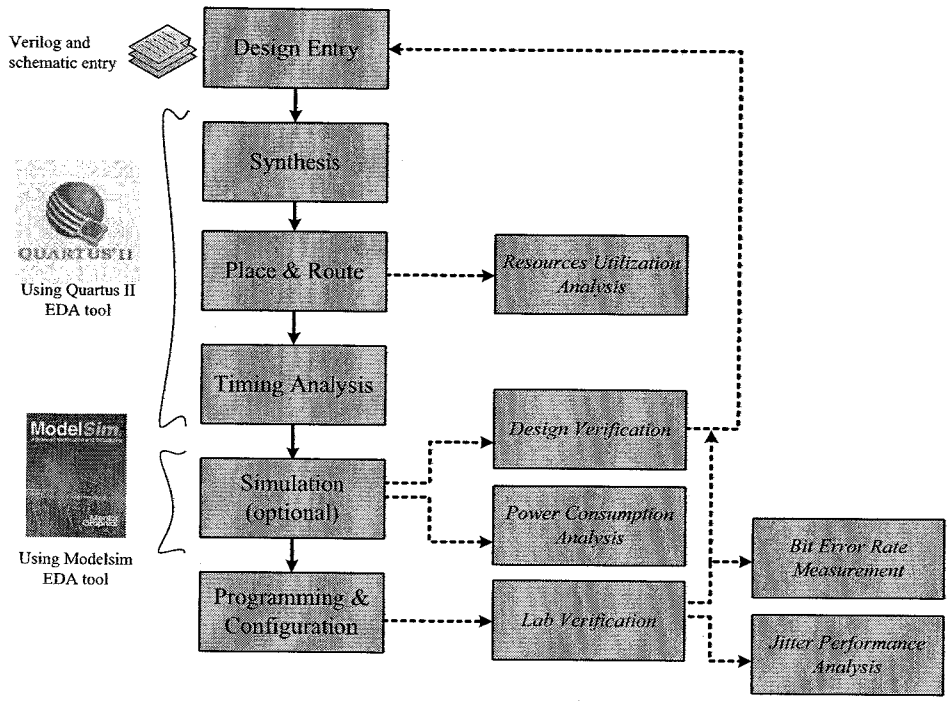


Figure 3.3: Overall FPGA design flow.

From this point onward, the design flow is totally accomplished by the assistance of EDA (electronic design automation) tools. A synthesis tool is used to map the output of the design entry process into logic gate implementations. The next step is to fit the resulting configuration into a specific FPGA device using a fitter (place and route) tool. The timing analysis tool helps the fitter tool to meet any previously defined timing requirements and generate bitstream. In this design flow, synthesis, fitter, and timing analysis tools are automatically performed using Altera Quartus II tool, which is only used for designs targeting Altera FPGA devices.

At this point, the functional and timing simulation can be optionally performed to ensure functional correctness of the design, using Altera Modelsim tool. Finally, the generated bitstream is directly loaded into the target FPGA device to acquire the desired functionality. Further experimental verifications can be performed after this stage including measuring the performance metrics.

CHAPTER 4

PROPOSED HARDWARE ARCHITECTURE

4.1 Chapter Overview

The literature review conducted earlier in this thesis has motivated us to come up with true fully-digital complete serial link architecture. This chapter firstly highlights the design challenges that arise when considering fully-digital architectures. Lastly, further details of the proposed link components design and functionality will be described.

4.2 Fully-Digital Serial Link Design Challenges

The fully-digital serial link architectures offer the ability to achieve low-power consumption, low phase noise, and better scalability (portability) across multiple technologies and platforms with inconsiderable modifications. In addition, describing the circuit in hardware description languages gives it a high flexibility to program all design parameters in a very short time compared with the analogue designs which need to be re-designed at transistor level for any parameter change. This can radically reduce cost and time-to-market by saving a significant amount of development time. However, beside these considerable advantages, the fully-digital architecture poses several design challenges which can be described as follow:

At the serializer side, major issues are focused on designing a fully-digital frequency synthesizer circuit that avoids any off-chip components, especially the DCO and loop filters. The main difficulty is to synthesize a fully-digital DCO into an FPGA platform. The DCO basically consists of an odd number of cascaded inverters

connected in a feedback loop. The synthesis tool used to map the DCO design into logic gate implementations will likely eliminate the redundant inverters in the feedback loop completely although they are used deliberately. To prevent this, a special constraint/attribute is used in the HDL that is vendor specific (not standard HDL features). This attribute is */* synthesis keep */* that is on the same line, before the semicolon, as the declaration of the input and output of each inverter you want the synthesis tool to keep. The other difficulty is how to control or minimize as possible the delay of each inverter in order to create a fine frequency resolution. Since the inverter characteristics in any FPGA are fixed and unchangeable, another approach should be sought to realize a small frequency step.

At the deserializer side, the concerns are related to minimize the power consumption resulted from the switching activities and reduce the complexity of the circuit by replacing all the off-chip components by embedded components. First, reducing the power consumption requires reducing the frequency of the internally generated clock signal without affecting negatively the data rate. The challenging idea is the use of multiple phases of a clock running at a frequency less than the data rate (e.g. half and quarter-rate). In this work, a quarter-rate CDR circuit which operates using eight phases of a clock running at a frequency equal to one-fourth the data rate is designed. As a result, the proposed architecture is an appropriate solution to partially reduce the switching frequency to one-fourth, and hence the total dynamic power consumption is reduced to one-fourth. Second, detecting various data rates normally requires an externally supplied reference oscillator and a frequency divider block [19], [20]. Avoiding the need for these components necessitates the designing of a DCO that is able to generate a wide-range of frequencies. The latter is normally controlled by a stable loop controller which operates dependently with the frequency detector. Using a digital stable controller can minimize the loop latency, the recovered clock jitter, and the loop complexity.

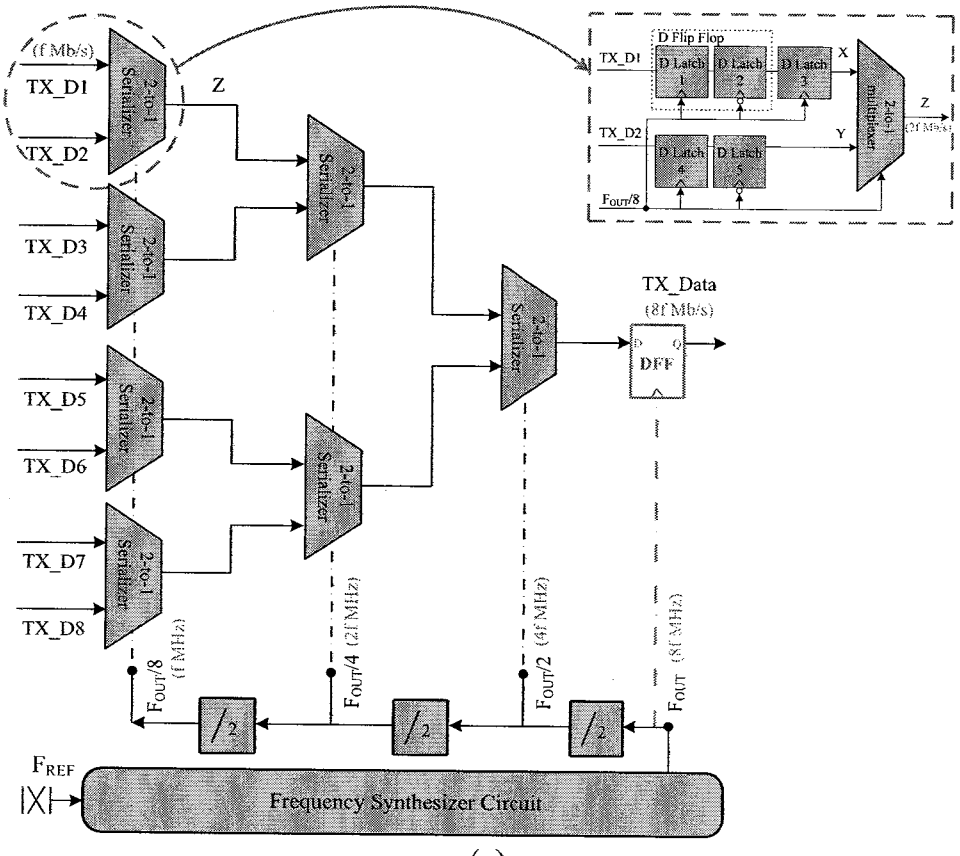
4.3 Serializer Design

The serializer circuit starts normally with converting the data from parallel to serial by

multiplexing multi low-frequency parallel data streams into a single higher frequency serial data stream. The high-frequency serialized data is then transmitted through the channel to the receiver. The serializer circuit is realized using two main blocks, namely, an 8-to-1 serializer and a frequency synthesizer circuit.

4.3.1 8-to-1 Serializer

As shown in Figure 4.1(a), the 8-to-1 serializer is a tree-like structure of seven units of 2-to-1 serializer. Each 2-to-1 serializer comprises 5 latches and a 2-to-1 multiplexer. The latch1 and latch2, as well as latch4 and latch5, constitute a single D flip-flop. The two D flip-flops guarantee the synchronization of the two input data edges to the clock edges so that they guarantee sufficient setup and hold time for the two input signals of the 2-to-1 selector. As a result, the 2-to-1 multiplexer output is less sensitive to the input noise or timing jitter [75]. Moreover, the latches provide a half cycle difference between the two input data streams for a proper operation of the multiplexer.



(a)

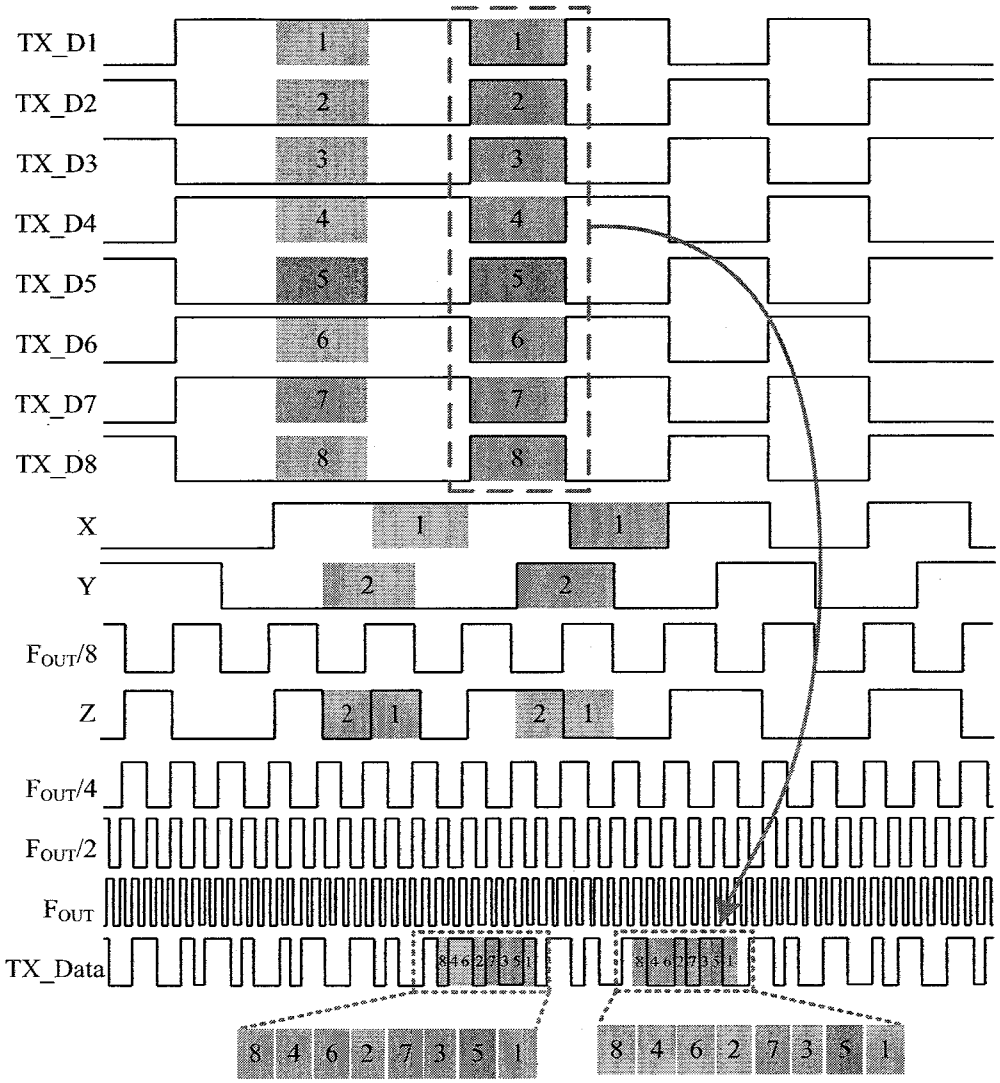


Figure 4.1: (a) Block diagram and (b) timing diagrams of the 8-to-1 serializer.

4.3.2 Frequency Synthesizer Circuit

The proposed frequency synthesizer circuit performs a critical function in the data transmission circuits. Its primary task is to generate multiple clock frequencies of the input data streams, with minimal skew relative to the system reference clock, minimal phase noise accumulation, and finest frequency resolution. The multiple clock frequencies are needed in order to properly perform the multiplexing operation of the 8-to-1 serializer. As shown in Figure 4.2, the circuit is composed of FLL and DLL that share a common reference clock signal (F_{REF}). Their basic operations require seven important building blocks to provide frequency and phase locking.

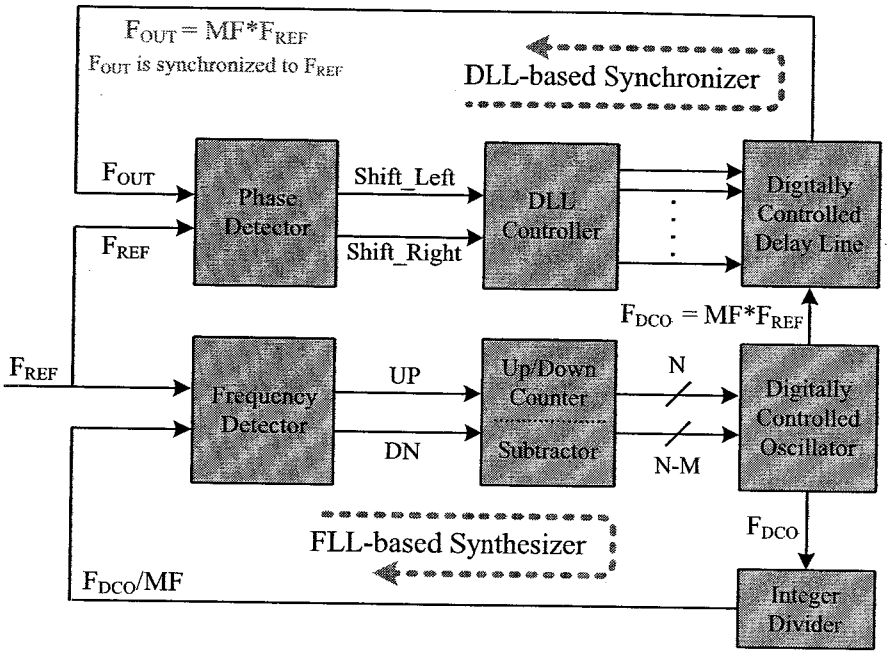


Figure 4.2: Block diagram of the proposed frequency synthesizer circuit.

In summary, at the heart of the FLL lies a DCO which deliberately avoids any analogue tuning controls. This gives the ability to realize the frequency locking mechanism as a fully digital architecture. The output clock signal of the DCO (F_{DCO}) is scaled down by an integer divider and connected to the frequency detector (FD). The integer divider allows the divided output clock (F_{DCO}/MF) to be relatively convergent with the frequency of F_{REF}. This provides also the ability to select an integer multiplication factor (MF) of the F_{REF} signal frequency (e.g. MF= 2, 4, 8, 16, 32, or 64). The FD detects the frequency difference between the F_{REF} and the F_{DCO}/MF signals and generates an up (UP) or down (DN) signal to indicate that the DCO should be speeded up or slowed down respectively. Then both up/down counter and full subtractor update the DCO control words to adjust the output frequency of the DCO. Meanwhile, the DLL provides a phase locking between the F_{DCO} and the F_{REF} signals through the phase detector (PD). The PD generates a shift right (shift_right) signal or shift left (shift_left) signal to adjust the delay of the digitally controlled delay line. The frequency synthesizer circuit will generate an output signal (F_{OUT}) that is synchronized with respect to the F_{REF} signal as well as MF times the F_{REF} frequency. The F_{OUT} signal is connected to the 8-to-1 serializer to properly perform the multiplexing operation.

4.3.2.1 Digitally Controlled Oscillator

A digitally controlled oscillator previously proposed in [76] is used in the proposed FLL design that has the ability to generate multiples of the F_{REF} signal frequency. The selection of this architecture is mainly based on both, the portability of the architecture (technology independency) and the ability to generate finest frequency resolution compared to the existing fully-digital designs [77-81]. The DCO consists of two main blocks: ring oscillator and fractional divider, as shown in Figure 4.3. The ring oscillator consists of one NAND gate which enables/disables the oscillation and a chain of AND-OR delay elements.

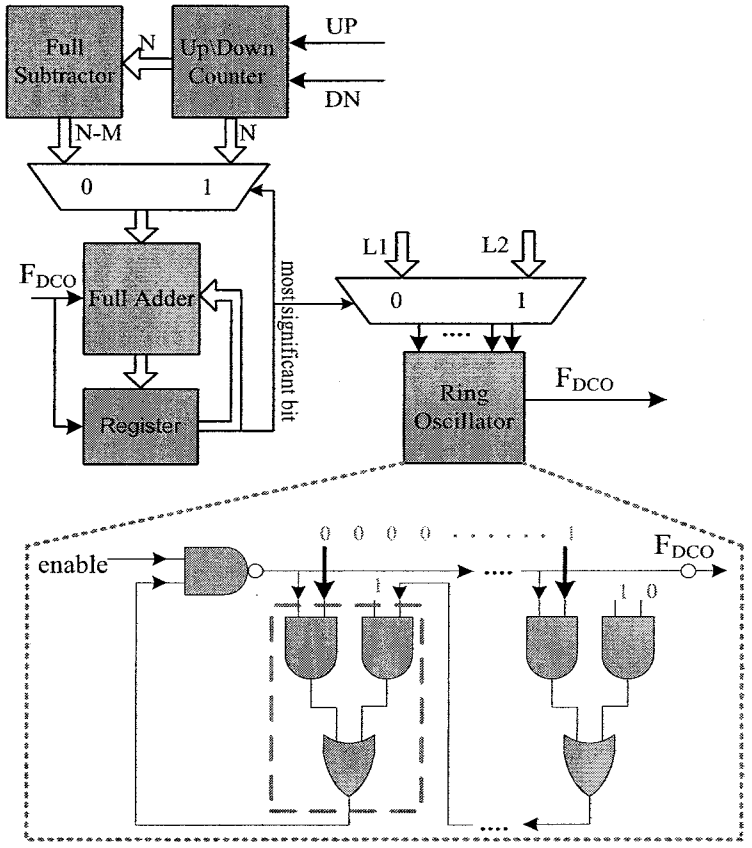


Figure 4.3: Functional block diagram of the digitally controlled oscillator.

The ring oscillator produces a clock signal (F_{OSC}) whose frequency is proportional to the number of the delay elements in the ring. The F_{OSC} is given by

$$F_{OSC} = 1/(2 \cdot L \cdot t_{de}) \tag{4.1}$$

where t_{dc} is the time delay for each delay element and L is the chain length that is defined by a one-hot coded control word. Consequently, reducing the number of the

delay elements in the ring gives higher frequency and vice versa. Moreover, changing the ring oscillator chain length via a one-hot coded word provides a coarse frequency resolution as shown experimentally in Figure 4.4.

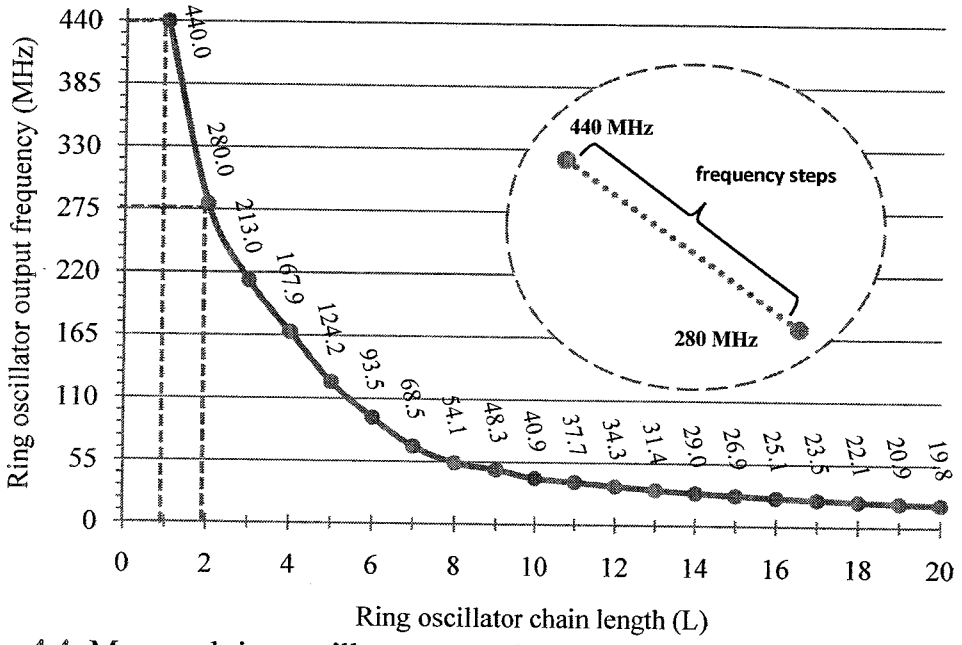


Figure 4.4: Measured ring oscillator output frequency F_{OSC} versus chain length. The number of bits of N defines the number of frequency steps between the two extreme limits (440 and 280 MHz).

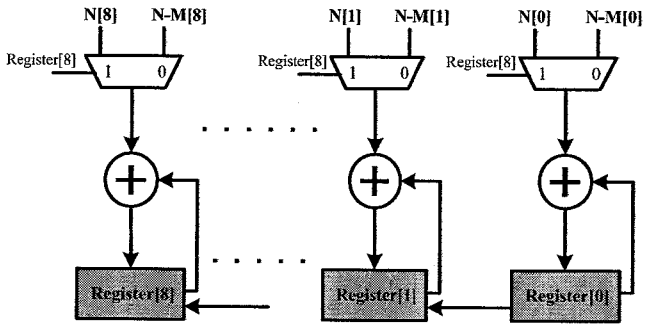
The fractional divider comprises an adder-accumulator as shown in Figure 4.5. The most significant bit (MSB) of the accumulator signed register is used to switch the input of the adder between signed integer number N and its two's complement N-M. The fractional divider is also used to switch between two adjacent ring oscillator chain lengths, (L1) and (L2). The digitally controlled oscillator output clock frequency, F_{DCO} is given by:

$$F_{DCO} = M/\big((N/F_{OSC}(L1)) + (M - N/F_{OSC}(L2))\big) \tag{4.2}$$

Accordingly, switching between two adjacent chain lengths L1 and L2 (L2 = L1 + 1) provides on average fine frequency resolution. The DCO output frequency step is extracted from the simulation results and can be calculated by equation (4.3). Increasing the number of bits of M realizes a higher frequency resolution.

$$F_{DCO\_STEP} = (F_{OSC}(L1) - F_{OSC}(L2))/M \tag{4.3}$$

Typically, the DCO must be able to provide a high frequency resolution as well as very good frequency stability. Good frequency stability can be achieved by designing a stable and fast controller to control the DCO, whereas a high frequency resolution is achieved by increasing the number of bits of the accumulator signed register.




```
begin
if (Register == 0)
   Register <= Register + N-M;
else if (Register [8] == 0)
   Register <= Register + N-M;
else if (Register [8] == 1)
   Register <= Register + N;
End
```

Figure 4.5: Block diagram of the fractional divider and behavioural Verilog code.

4.3.2.2 Integer Divider

The integer divider consists of a chain of divide-by-2 circuits, as shown in Figure 4.6. Each circuit is a single D flip-flop. The presence of the integer divider block in the FLL is to scale down the output clock signal of the DCO to be relatively convergent with the frequency of F_{REF} and allows the latter to run at a low frequency. The divider provides the ability to select an integer MF of the F_{REF} signal frequency (e.g. MF= 2, 4, 8, 16, 32, or 64).

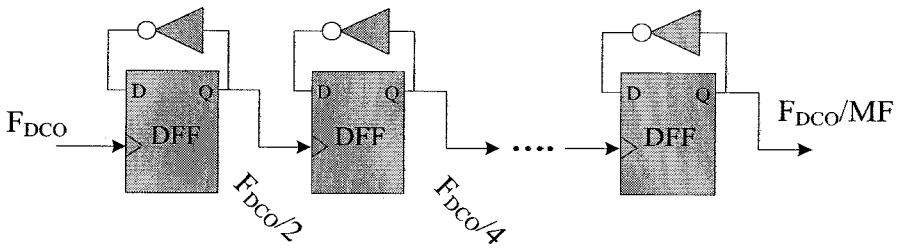


Figure 4.6: Block diagram of integer frequency divider.

4.3.2.3 Frequency Detector (FD)

The block diagram of the rotational frequency detector is given in [82]. The rotational FD has three inputs, the F_{REF} signal and the in-phased I and the quadrature Q signals of the F_{DCO}/MF signal. As shown in Figure 4.7, I and Q signals are sampled by the

transitions of the reference clock at the four D flip-flops. The DFF1 and DFF2 store the current sampled output, whereas DFF3 and DFF4 store the previous sampled output. Thus, using the combinational logic shown in Figure 4.7(a) the frequency difference can be detected. As illustrated in Figure 4.7(b), the logic works as follows: an UP pulse is produced when the current and previous state of Q_1 and Q_2 signals are 01 and 00, respectively. An DN pulse is generated when the current and previous state of Q_1 and Q_2 signals are 00 and 01, respectively. The frequency of UP or DN signal is equal to the difference between the frequency of I and the reference clock frequency.

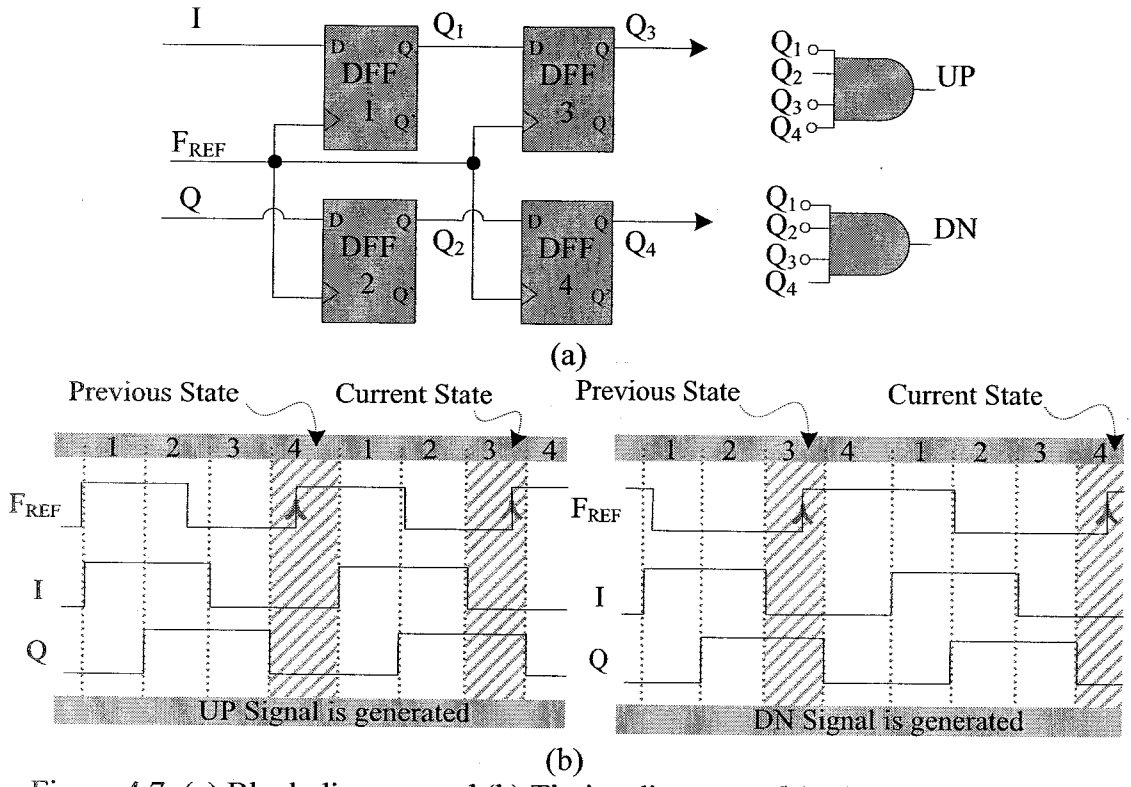


Figure 4.7: (a) Block diagram and (b) Timing diagrams of the frequency detector.

The rotational frequency detector becomes ineffective when the frequency of I exceeds 30% of the reference clock frequency. However, the integer divider in the frequency locked loop scales down the frequency of I to be relatively convergent with the frequency of F_{REF}. As a result, the integer divider scales down the difference in frequencies to less than 30%.

4.3.2.4 Up/Down Counter and Full Subtractor

The up/down counter and full subtractor are used to control the output frequency of the DCO, by generating the signed integer number N and its two's complement N-M, respectively. First of all, the counter used is a normal nine-bit synchronous up/down counter. It has two input signals, Up/$\overline{\text{Down}}$ and clk. However, the Up/$\overline{\text{Down}}$ and clk signals are formulated by:

$$Up/\overline{Down} = UP + \overline{DN} \tag{4.4}$$

$$clk = \left((UP + DN) \cdot \left(\overline{UP} + \overline{DN}\right)\right) \cdot \overline{F_{REF}} \tag{4.5}$$

where UP and DN signals are the output of the frequency detector. The up/down counter generates nine bits output signal N based on the received up (UP) or down (DN) signal from the frequency detector. For each decision the counter updates N value by adding or removing one from the current N value. Second of all, the subtractor used is a nine-bit full subtractor. As listed in Table 4.1, it generates nine bits output signal (N-M) based on N and M values, where all bits of M value are set to be 1 except the MSB.

Table 4.1: Fractional divider input and output signals

N[8:0]		N-M[8:0]		Register[8:0]	
56	000111000	-199	100111001	0	000000000
56	000111000	-199	100111001	-199	100111001
56	000111000	-199	100111001	-143	101110001
56	000111000	-199	100111001	-87	110101001
56	000111000	-199	100111001	-31	111100001
56	000111000	-199	100111001	25	000011001
56	000111000	-199	100111001	-174	101010010
55	000110111	-200	100111000	-119	110001001

4.3.2.5 Digitally Controlled Delay Line (DCDL)

In this work, the phase tracking mechanism is separated from the frequency tracking loop. This approach adds an essential benefit to the design which is the ability to synchronize the output clock signal with the input reference signal. The success of the phase locking process is based on the presence of a linear relationship between the DLL controller output and the DCDL output delay, thus a chain of linear delay elements (DE) is employed in the structure of the DCDL. As illustrated in Figure 4.8, each DE consists of three NAND gates [83]. One of them is used to activate the selected DE, while the other two gates are used to delay/advance the F_{DCO} signal. An additional inverter gate is added to the end of the delay line chain to produce the original signal without inversion.

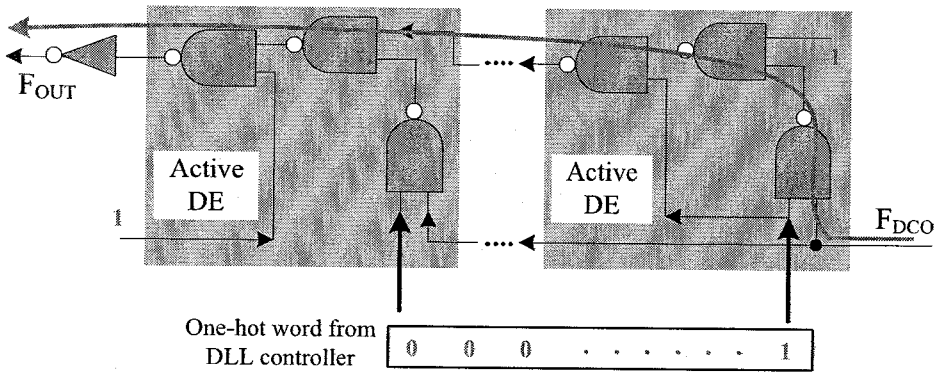


Figure 4.8: Block diagram of the DCDL.

4.3.2.6 DLL Controller

The controller in [83] is replaced by a one-hot finite state machine (FSM). This gives the ability to eliminate the need for a number of inverters that equals to the number of DEs utilized in the DCDL, and hence saves the occupied area. The DLL controller is responsible for controlling the DCDL chain length based on the received shift right (shift_right) or shift left (shift_left) signal from the phase detector. For each decision the DLL controller updates the number of the active DEs in the chain. A shift_right signal decreases these DEs and thus decreases the delay of the input clock of the DCDL while a shift-left does the opposite function.

4.3.2.7 Phase Detector (PD)

The phase detector in [83] is used with modifications as shown in Figure 4.9. It generates shift_right or shift_left regardless of the frequency difference between F_{REF} and F_{OUT}. As a result, a frequency divider block is not needed in the delay locked loop. This can be verified by simulating the entire delay locked loop system. The resulting timing waveforms are shown in Figure 4.10.

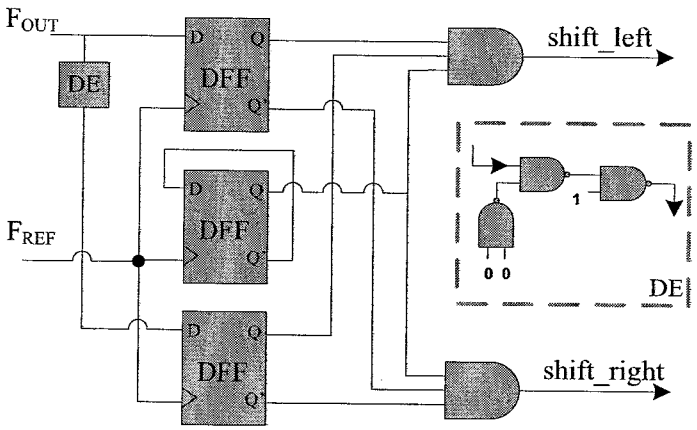


Figure 4.9: Block diagram of the phase detector.

It is illustrated from the shift_left and the shift_right signals that F_{OUT} and F_{REF} signals become synchronized and in-phase after 6-cycles of F_{REF} signal. The delay element of the detector is normally set to be identical to the DE of the delay line. This governs that the phase detector will not contribute to the final DLL phase error. Generating shift-left or shift-right once every two cycles of the reference clock provides stability for the DLL controller.

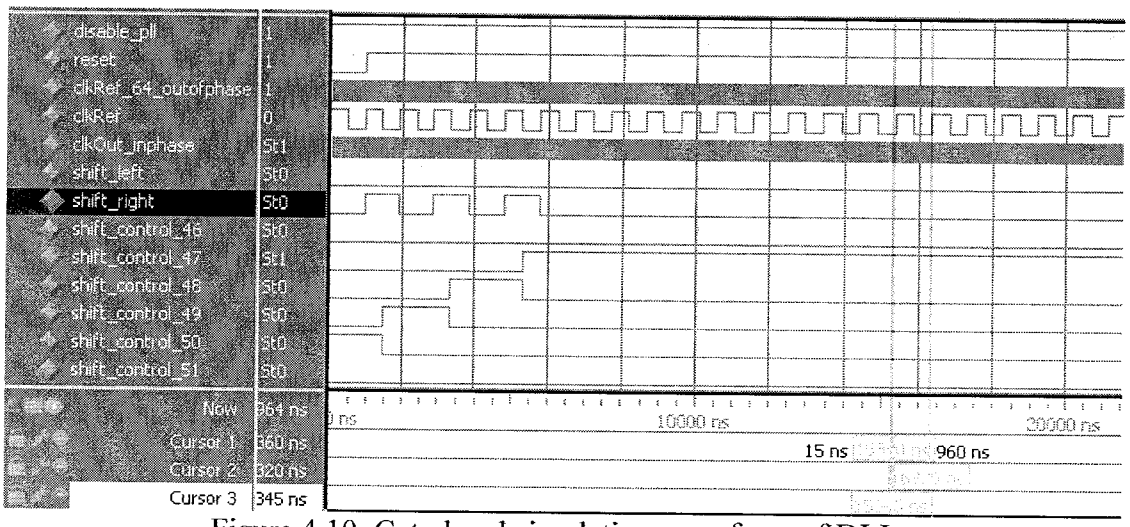


Figure 4.10: Gate-level simulation waveform of DLL.

4.3.3 Built-In PRBS Generator

Pseudorandom bit sequences (PRBS) are widely employed in verifying the functionality of serial links [84, 85]. The response of the link for different length of PRBS patterns can reveal essential hints about its performance including jitter generation and bit error rate (BER) analysis. For instance, generally CDR generates higher jitter when a longer length test pattern is fed in. Thus, it is more flexible to build in a PRBS generator with selectable sequence length [85, 86]. For a convenient and cost efficient way of testing, a programmable and multi-pattern PRBS generator circuit is designed and integrated with the link.

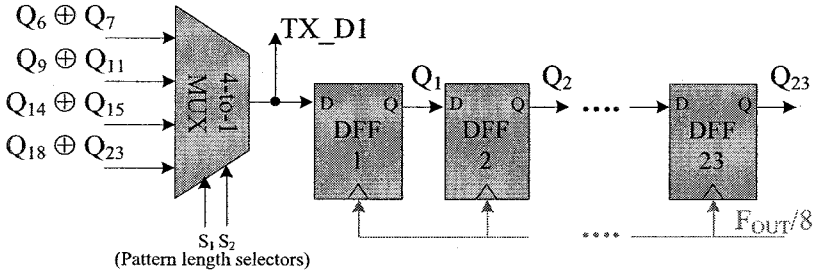


Figure 4.11: Block diagram of the built-in programmable PRBS generator.

As shown in Figure 4.11, generating a serial PRBS is typically realized using a linear feedback shift register (LFSR), where the length of the register (n) and the feedback function determine the length of the sequence (2^n-1). The multi-pattern generator is capable of producing the most commonly used PRBS lengths based on ITU-T recommendation O.150 [87], which are 2^7-1, 2^{11}-1, 2^{15}-1, and 2^{23}-1. Such an arrangement provides sufficient randomness while saving significant area.

4.4 Deserializer Design

Conceptually, the deserializer circuit performs the inverse function of the serializer block. The deserializer extracts the embedded clock information from the incoming data stream, re-samples the noisy stream using the extracted clock for jitter removal and synchronization, and deserializes the single stream back into low-frequency parallel data streams of similar number to the serializer. These essential functions can be realized using two main blocks, namely, a CDR circuit and a 4-to-8 deserializer.

4.4.1 Clock and Data Recovery (CDR) Circuit

In this design, a fully digital implementation of a referenceless mixed FLL/DLL quarter-rate CDR circuit is proposed and utilized. The proposed architecture is shown in Figure 4.12. Its basic operation requires six important blocks, a quarter-rate frequency detector, an up/down counter and subtractor, a DCO-based eight-phase generator, an early-late phase detector, a delay line controller, and a digitally controlled delay line.

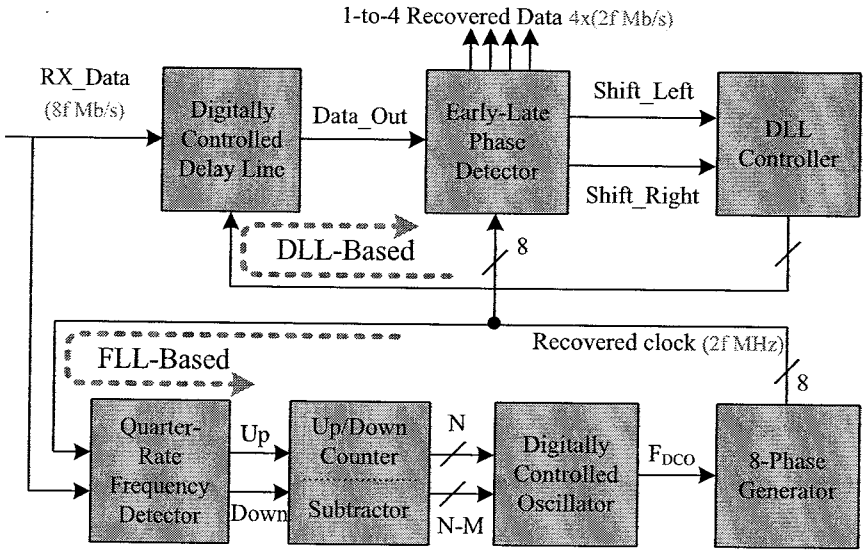


Figure 4.12: Block diagram of the proposed CDR circuit.

4.4.1.1 Early-Late Phase Detector (ELPD)

The phase detector used [28] is an early-late, quarter-rate non-linear design. As shown in Figure 4.13, the ELPD samples the synchronized data stream (Data_Out) using the clock phases 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°, and producing eight signals D_0, D_{45}, D_{90}, D_{135}, D_{180}, D_{225}, D_{270}, and D_{315} at the D flip-flop outputs. The aforesaid eight output signals are used to generate the Shift_Right and Shift_Left signals as follow:

$$\text{Shift\_Right} = (D_0 \oplus D_{45}) + (D_{90} \oplus D_{135}) + (D_{180} \oplus D_{225}) + (D_{270} \oplus D_{315}) \quad (4.6)$$

$$\text{Shift\_Left} = (D_{45} \oplus D_{90}) + (D_{135} \oplus D_{180}) + (D_{225} \oplus D_{270}) + (D_{315} \oplus D_0) \quad (4.7)$$

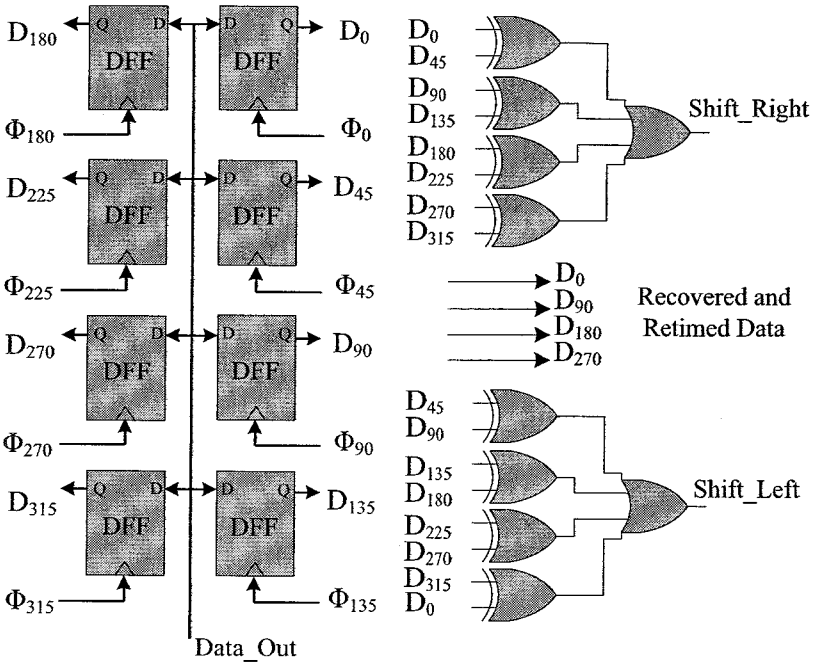


Figure 4.13: Block diagram of the ELPD.

The Shift_Right and Shift_Left signals indicate the relative clock edge positions with respect to the data edges. If the ELPD is in the locked state as illustrated in Figure 4.14(a), the edges of the half-quadrature phases (Φ_{45}, Φ_{135}, Φ_{225}, and Φ_{315}) are aligned with the data transitions.

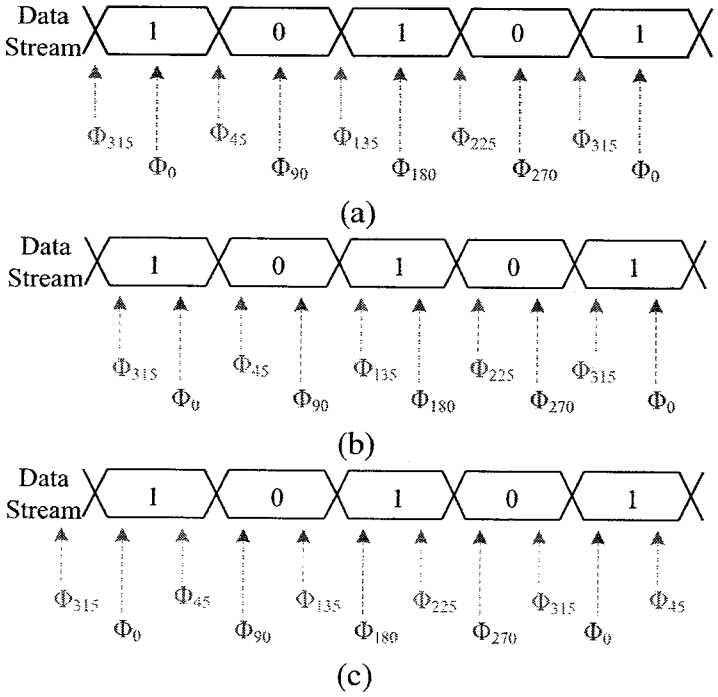


Figure 4.14: Detecting conditions of the ELPD (a) Locked state (b) Late state (Shift_Right is high) (c) Early state (Shift_Left is high).

And hence D_0, D_{90}, D_{180}, and D_{270} will be the recovered and 1-to-4 demultiplexed data. In contrast as illustrated in Figure 4.14(b), when the data transition is situated between D_0 and D_{45}, a Shift_Right signal that indicates a late state will be generated, whereas Figure 4.14(c) illustrates an early state where the data transition is situated between D_{45} and D_{90}. The Shift_Right and Shift_Left signals will be used by the delay line controller to properly adjust the mid-point of the data bits as close as possible to the clock edges for optimum sampling.

4.4.1.2 Setup/Hold Time Violation in ELPD

To guarantee a reliable operation of the ELPD, the input signals must meet the D flip-flop's timing requirements. The Data_Out signal must be stable (no transition should occur) for a setup time (T_{SU}) before the recovered clock edge and a hold time (T_H) after the recovered clock edge. Accordingly, the flip-flop changes its output signal after the recovered clock edge by the clock to output time (T_{CO}). Violating the flip-flop's timing requirements may cause its output to become metastable. Metastable outputs are neither high nor low for an amount of time before it resolves to a state of logic high or logic low. This may cause system failure.

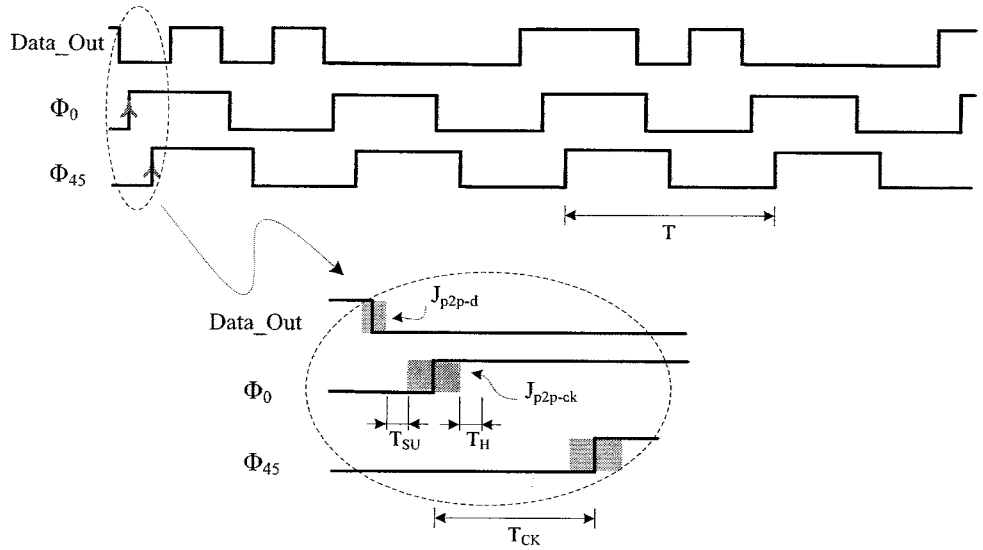


Figure 4.15: Timing diagrams showing the condition to avoid setup/hold time violation in the ELPD.

As illustrated in Figure 4.15, the timing diagram showing the Data_Out data stream and only 2 phases of the recovered clock (i.e. Φ_0 and Φ_{45}) in which the clock period is four times the data bit width (i.e. clock is quarter-rate) and assume that the

time difference between the 2 above phases is T_{ck}. In order to guarantee that the T_{ck} is sufficient to avoid any setup/hold time violation, the following equation has to be respected at all time:

$$T_{ck} \geq T_{CO} + J_{p2p\text{-}ck} + J_{p2p\text{-}d} + T_{SU} + T_H \tag{4.8}$$

where $J_{p2p\text{-}ck}$ is the peak-to-peak jitter of the recovered clock and $J_{p2p\text{-}d}$ is the peak-to-peak jitter of the recovered data. Specific values for T_{SU}, T_H, and T_{CO} are provided in the device data sheet [73]. However, $T_{ck} = T/8$, where T is the recovered clock period. Thus to avoid any setup/hold time violation, the recovered clock frequency and the input data rate should be:

$$\Phi_0 \leq \frac{1}{8 \cdot (T_{CO} + J_{p2p-ck} + J_{p2p-d} + T_{SU} + T_H)} \tag{4.9}$$

$$Data\_Out \leq \frac{4}{8 \cdot (T_{CO} + J_{p2p-ck} + J_{p2p-d} + T_{SU} + T_H)} \tag{4.10}$$

4.4.1.3 Delay Line Controller (DLC)

A delay line controller implemented as an FSM is proposed; its stability feature is desired in order to keep the loop in the locked state and to reduce the noise added to the data stream from the delay line. The DLC is responsible for adjusting the delay line chain length via one-hot coded word. As listed in Table 4.2, a specific coded word is generated based on the controlling Shift_Left or Shift_Right signal issued from the ELPD to delay or advance the data stream respectively.

Table 4.2: Delay line controller operations

Shift_Right	**Shift_Left**	**Action**
High	Low	**Late state:** shift the RX_Data to the right by decreasing the delay of the delay line.
Low	High	**Early state:** shift the RX_Data to the left by increasing the delay of the delay line.
High	High	**Locked State:** No Action.
Low	Low	No Action.

4.4.1.4 Digitally Controlled Delay Line (DCDL)

The DCDL has an essential function, which is adjusting (i.e. delay or advance) the mid-point of the data bits as close as possible to the clock edges for optimum sampling. Therefore, the delay line is implemented as a chain of linear delay elements (DEs) [88]. As shown in Figure 4.16, each DE consists of three NAND gates. One of them is used to activate the selected DE, while the other two gates are used to delay/advance the data stream. An additional NAND gate is added to the delay line chain to produce the original signal without inversion.

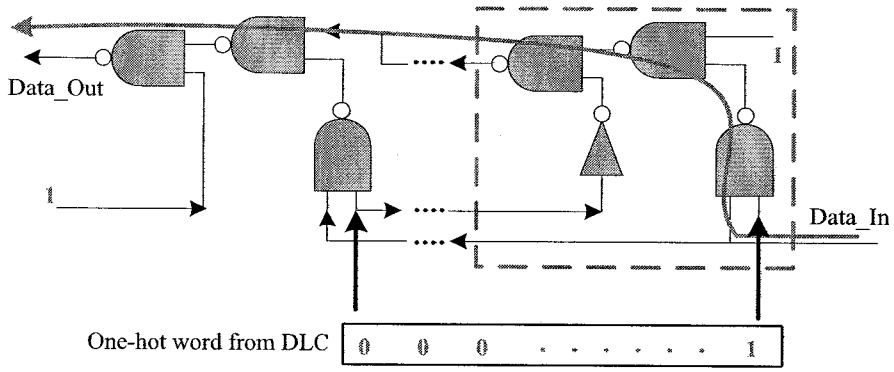


Figure 4.16: Block diagram of the DCDL.

4.4.1.5 Quarter-Rate Frequency Detector (QRFD)

The QRFD performs a critical function in the proposed CDR architecture. Its primary task is to track and detect the frequency difference between the incoming data stream (RX_Data) and the generated eight phases, whereas the latter's frequency should be equal to one-fourth of the data rate in order to achieve the concept of the quarter-rate CDR and hence partially reduce the power consumption.

The previously proposed QRFD in [28] is used with slight modifications in the double-edge triggered flip-flop architecture [89], as shown in Figure 4.17(a). Its operational principle comprises three main steps. Firstly, the clock phases 0°, 22.5°, 45°, and 67.5° are sampled by the transitions of the RX_Data signal at DFF1, DFF2, DFF3 and DFF4, respectively. The output signals of these DFFs indicate the position of each data transition with respect to the relative clock phases according to Table 4.3.

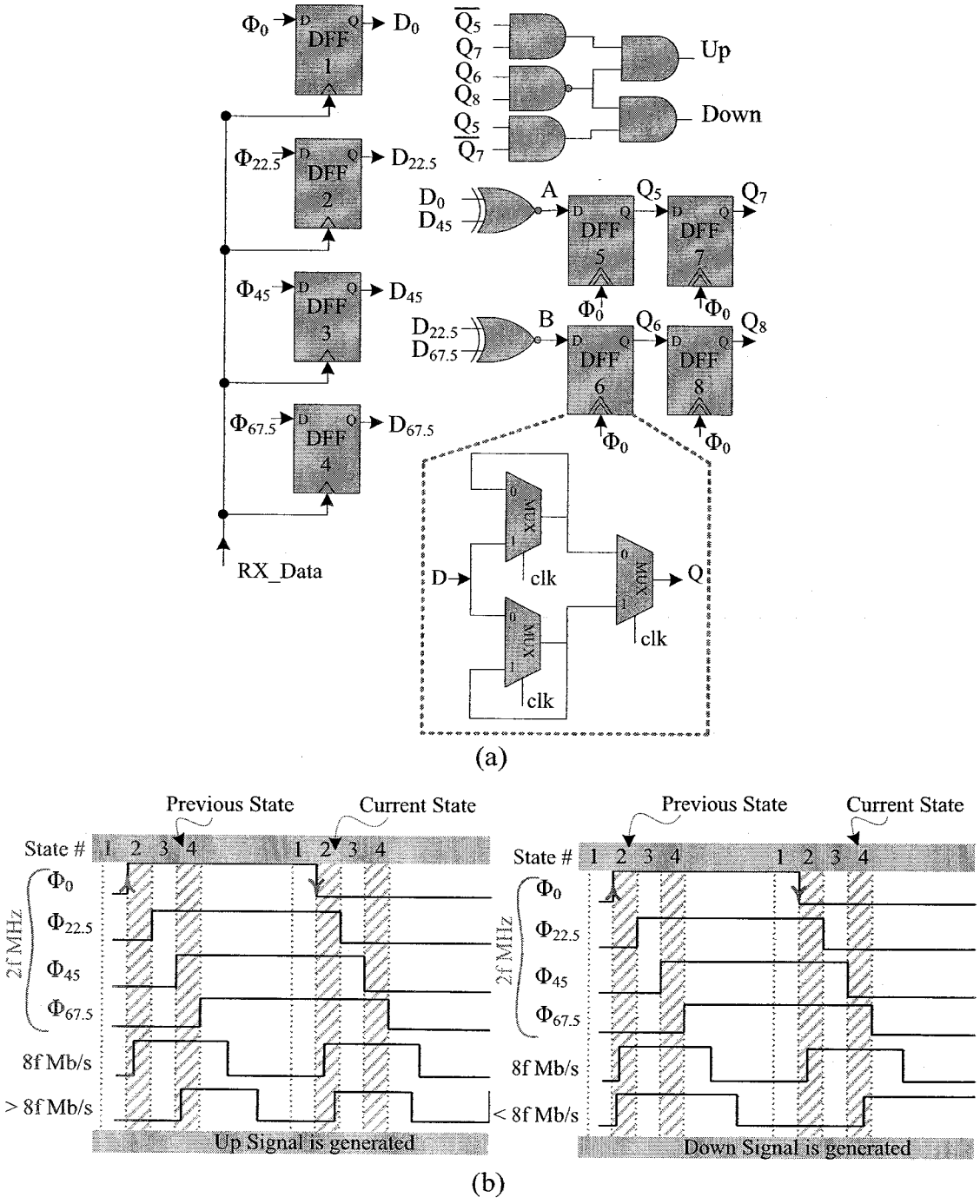


Figure 4.17: (a) Block diagram and (b) Timing diagrams of the QRFD architecture.

Table 4.3: QRFD internal signals (A and B) states

D_0	$D_{22.5}$	D_{45}	$D_{67.5}$	**A** $D_0 \oplus D_{45}$	**B** $D_{22.5} \oplus D_{67.5}$	**State Number**
Low	Low	Low	Low	**Low**	**Low**	**1**
High	Low	Low	Low	**High**	**Low**	**2**
High	High	Low	Low	**High**	**High**	**3**
High	High	High	Low	**Low**	**High**	**4**

Secondly, the detected state in form of (*A*, *B*) is stored in the DFF5 and DFF6, whereas the state of the previous data transition was stored in the DFF7 and DFF8. Finally, using the logic shown in Figure 4.17(a) two signals are generated, namely, Up and Down based on the two stored states as shown in Table 4.4. These two signals will be used by the up/down counter to properly adjust the frequency of the eight phases to be equal to one-fourth of the data rate.

Table 4.4: QRFD truth table

		Current State Number			
		1	**2**	**3**	**4**
Previous State Number	**1**	-	Up	Up	-
	2	Down	-	-	Down
	3	Down	-	-	-
	4	-	Up	-	-

4.4.1.6 Up/Down Counter and Full Subtractor

The architecture of the up/down counter and full subtractor used in the CDR design is exactly typical to the block used in the design of the frequency synthesizer circuit at the serializer side. Down signal indicates that the recovered clock needs to be reduced. Up signal indicates that the recovered clock needs to be increased.

4.4.1.7 DCO and Eight-phase Signal Generator

For proper operations of the proposed design, eight 22.5°-spaced phases and their complements are required, the latter are normally implemented as a DLL or shift register/finite state machine (FSM) [90]. Although the FSM-based phase generator seems more attractive due to its wide working frequency range and better output jitter performance [90], it requires a 16 times higher clock frequency to generate the aforementioned 16 phases. In this design, a new way of generating the required phases is designed using only 7 IQ (in-phase and quadrature-phase) signal generators as shown in Figure 4.18. The IQ-based phase generator requires only 8 times higher clock frequency instead of 16 times. Another DCO, which is separated from the serializer circuit, is used to generate the input clock signal for the phase generator circuit. The wide-range frequency of the DCO gives the CDR circuit the ability to determine the required centre frequency to accommodate for different data rate without modifying the CDR circuit design, which makes it a fully-tuneable design.

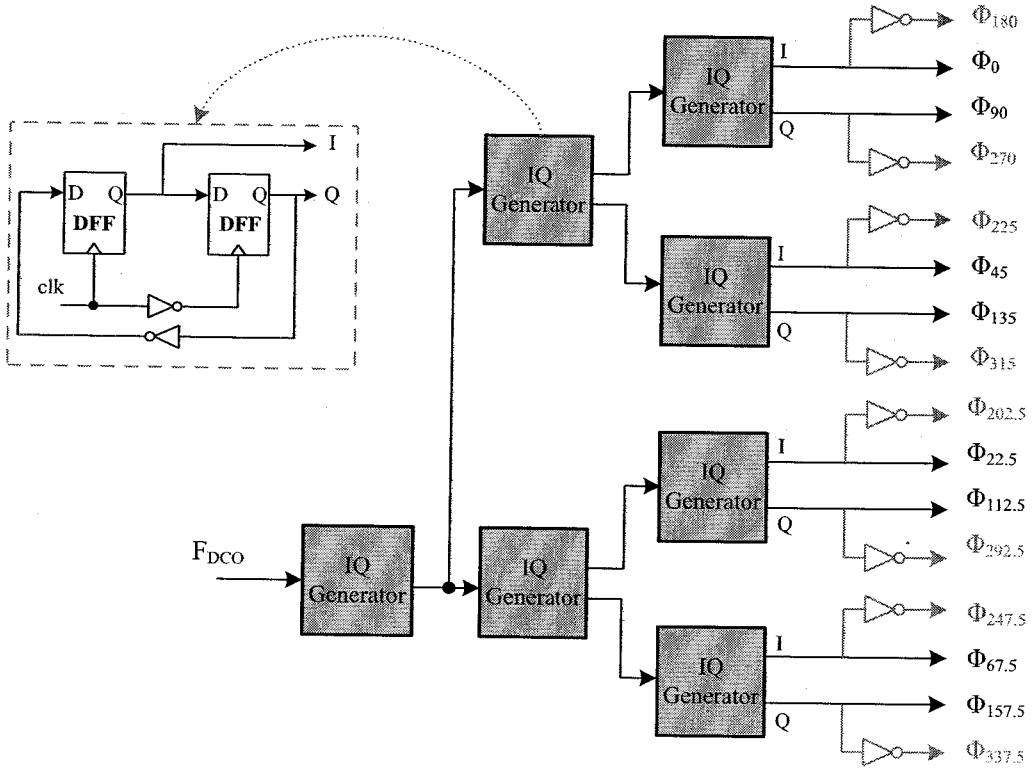
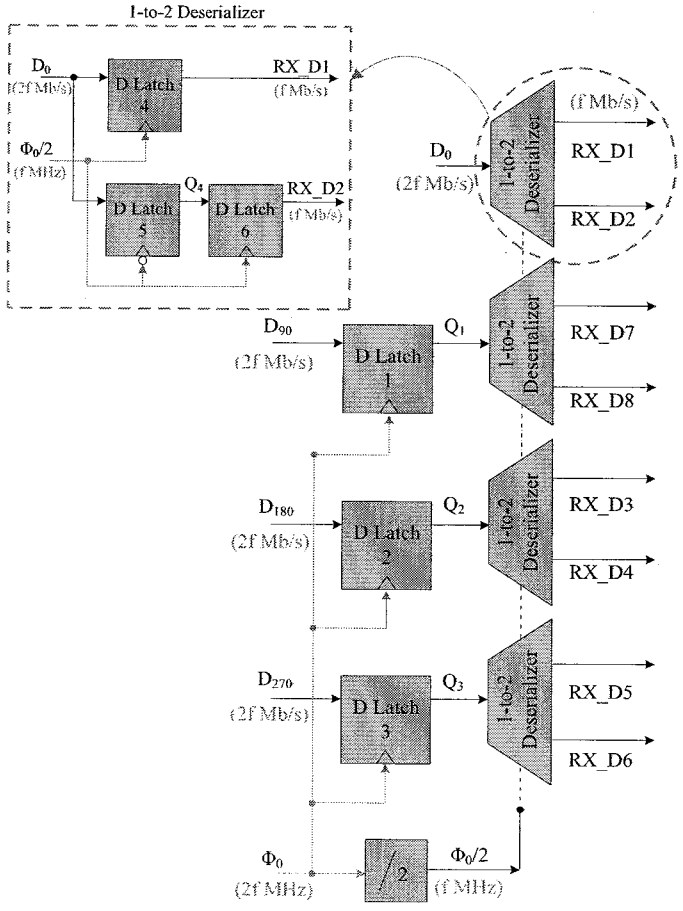


Figure 4.18: (a) Block diagram of the proposed 8-phase generator.

4.4.2 4-to-8 Deserializer

Since the designed quarter-rate CDR circuit is capable of recovering and automatically demultiplexing (1-to-4) the incoming data stream, an additional circuit is required to retime the four quadrature data streams and also function as a 4-to-8 demultiplixer. This is due to the need for comparing the 8 inputs of the serializer to the 8 outputs of the deserializer to thoroughly verify the functionality of the serial link. The block diagram of the 4-to-8 deserializer circuit is shown in Figure 4.18(a) and it is composed by 17 latches (including the divide-by-2 block). The 4-to-8 deserializer is integrated with the ELPD block by connecting its inputs to the outputs of the ELPD which are D_0, D_{90}, D_{180}, and D_{270}, and the recovered clock Φ_0. The first stage of the 4-to-8 deserializer circuit synchronizes the input data streams with the rising edge of the Φ_0 clock. Consequently, the following stage demultiplexes the outputs based on $\Phi_0/2$ clock and generates the eight recovered data streams, RX_D1-RX_D8, as shown in Figure 4.18(b).



(a)

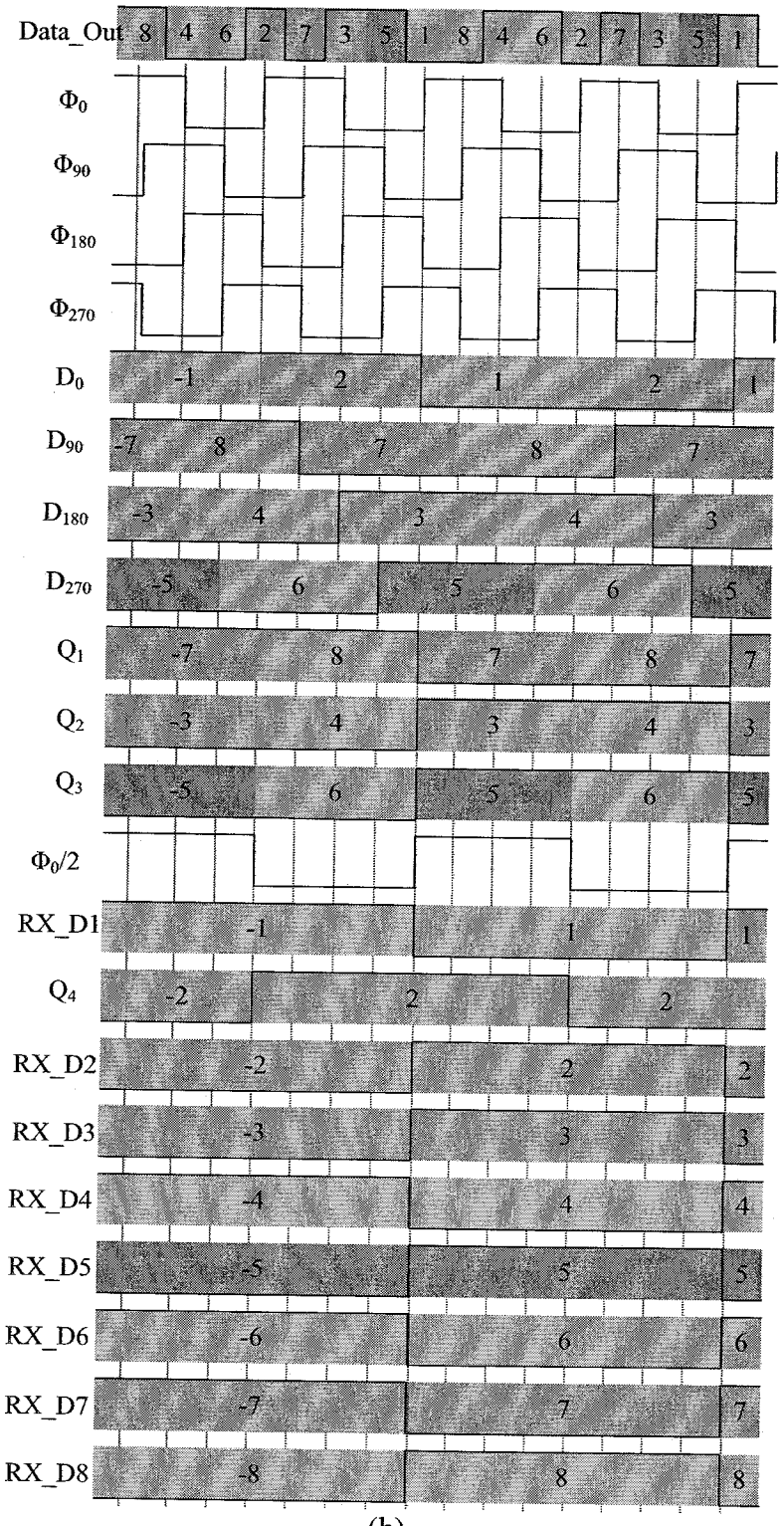


Figure 4.19: (a) Block diagram and (b) timing diagrams of the 4-to-8 deserializer architecture.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Chapter Overview

In the previous chapter, the fully-digital serial link architecture is proposed and designed. This chapter discusses the implementation of the link design into the target FPGA board as a proof-of-concept vehicle to verify its fully-digital status. It starts with the analysis of the resources utilization and the power consumption of the design. The next stage is now set to experimentally validate the functionality of the system as well as to evaluate its characteristics and performance metrics. Finally a comparative performance analysis will be carried out between the proposed link components and various prior art work.

5.2 FPGA Synthesis and Implementation

Both the serializer and the deserializer circuits are totally designed using Verilog-HDL language and synthesized using Quartus II software for Altera Cyclone II FPGA. The fact that it is implemented on an FPGA is a confirmation of its fully-digital status; hence it can be implemented on various platforms, such as FPGAs and ICs.

5.2.1 Resources Utilization Analysis

As listed in Figure 5.1, the Quartus II software yields the following FPGA resources statistics: the proposed architecture size (excluding the PRBS generator) is 1,094 out

of 68,416 (2%) LEs, 0 out of 1,152,000 (0%) memory bits, 0 out of 300 (0%) multipliers and 0 out of 4 (0%) PLLs.

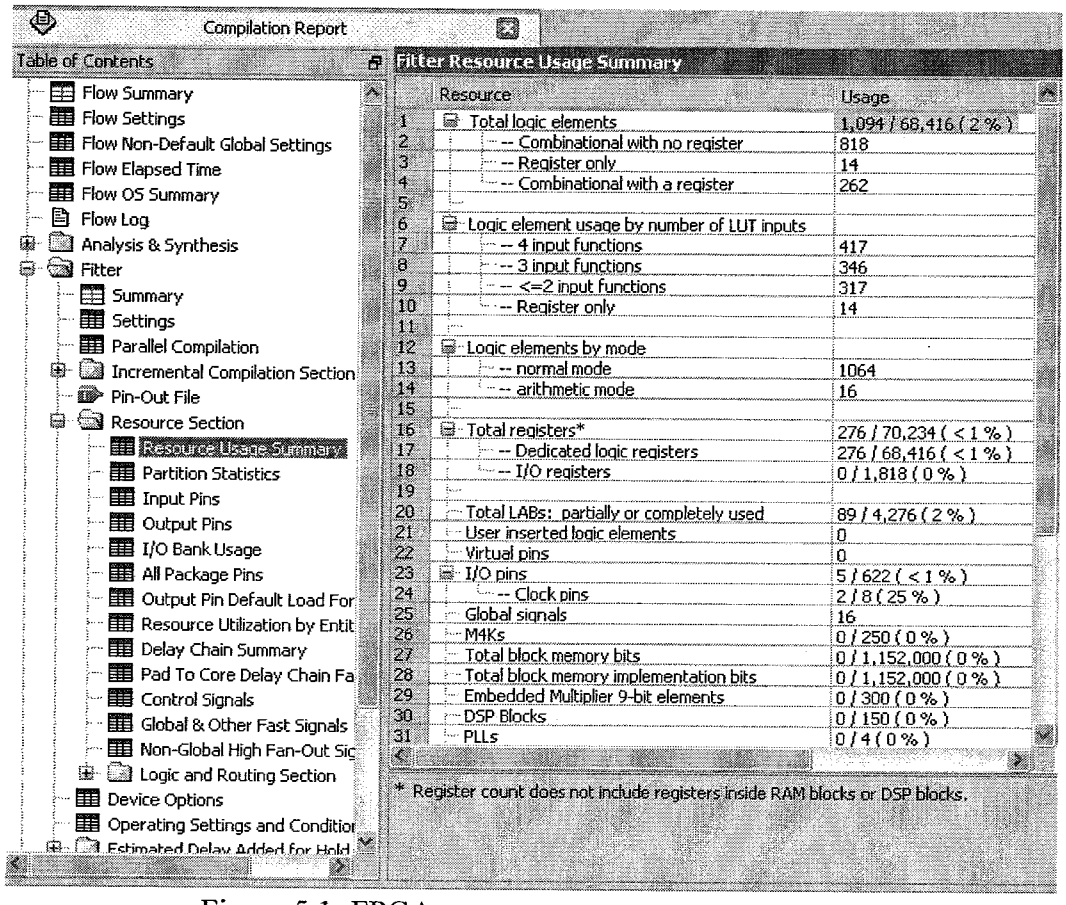


Figure 5.1: FPGA resources usage summary report.

This greatly proves the portability of the proposed architecture and indicates that it is independent of any vendor specific components. The layout of the FPGA chip after the programming process is given in Figure 5.2, using the Chip Planner tool. It shows the utilized LEs and IO blocks among the total chip resources. At this stage, the FPGA board is ready for lab verification process and conducting various experimental measurements.

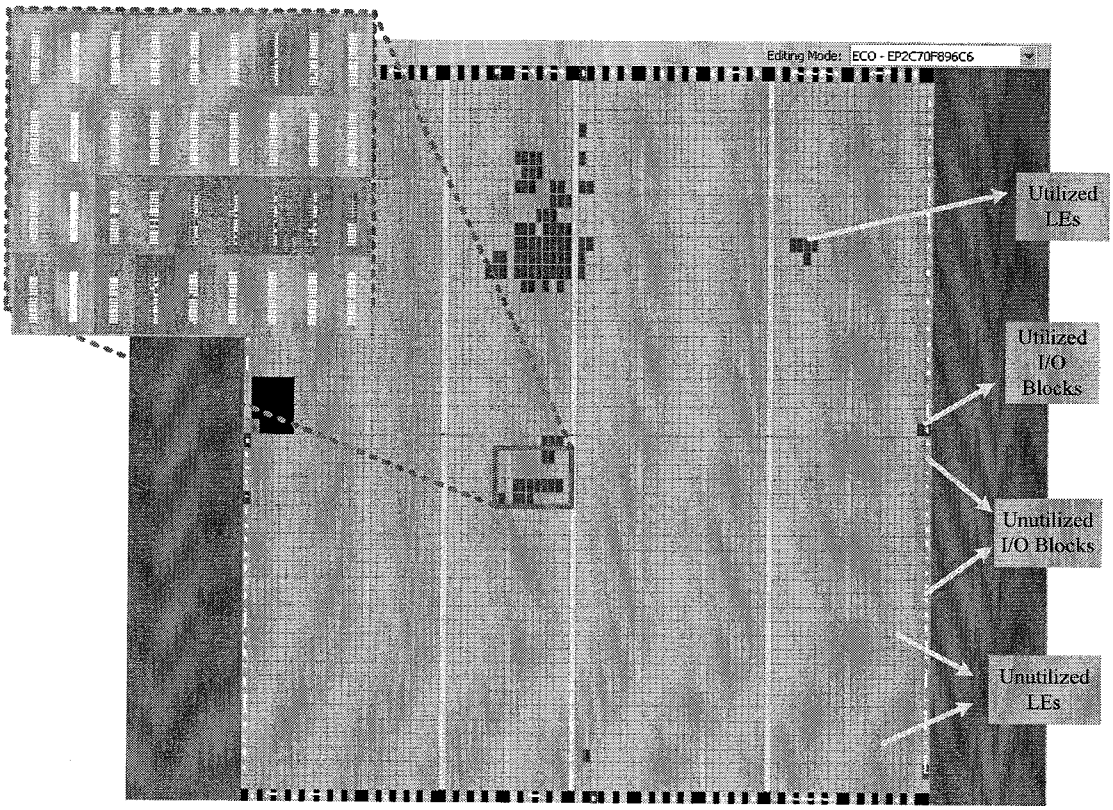


Figure 5.2: Layout of the Cyclone II FPGA chip.

5.2.2 Power Consumption Analysis

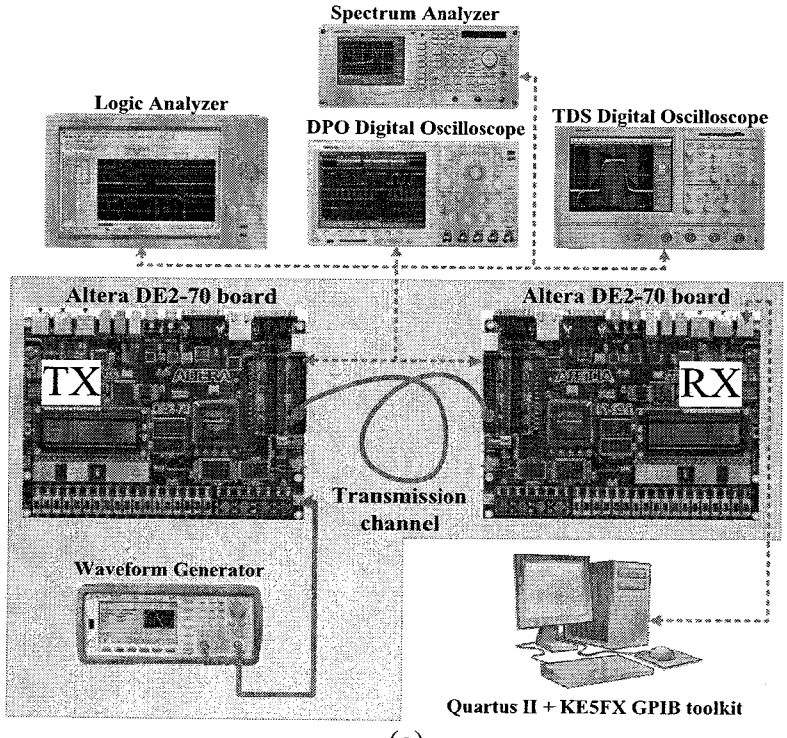
Using Altera power analyzer tool (PowerPlay), the serial link (excluding the PRBS generator) consumes a power of 0.97 mW at 167.36 Mb/s. The reported power estimation confidence level is high. This reflects that the toggle rate provided by simulation is reliable and sufficient [91]. The power efficiency of the design is 5.79 mW/Gb/s.

5.3 Experimental Results

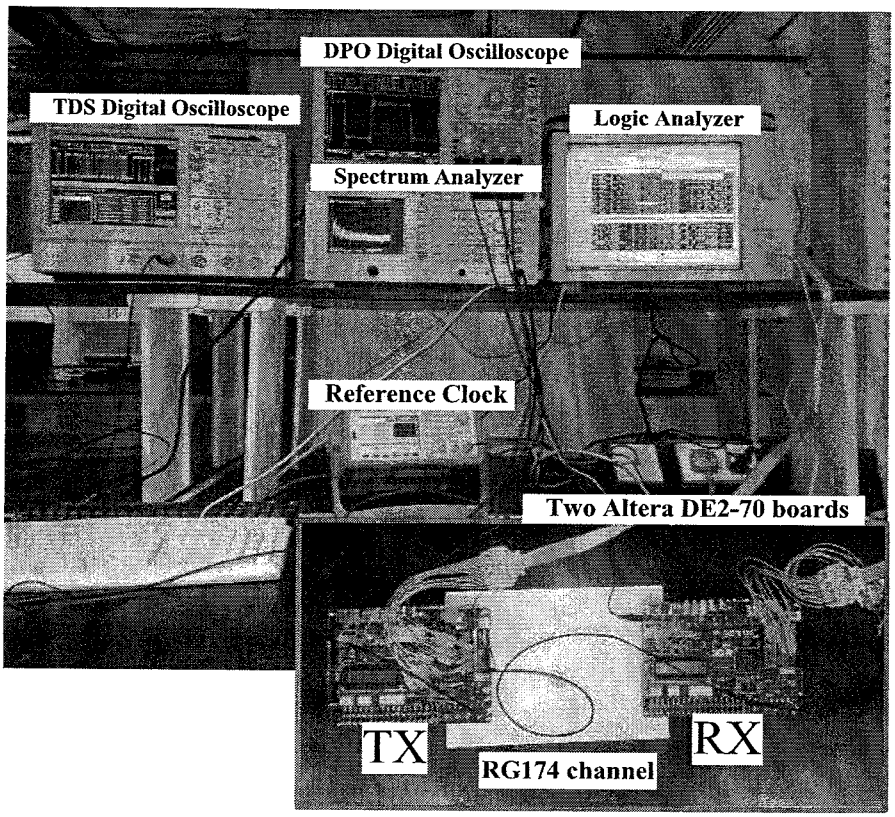
The experimental setup is illustrated in Figure 5.3 and it consists of the following equipments:

- Two Altera DE2-70 boards.

- PC, equipped with Quartus II software (for designing purposes, programming the FPGA chip, and analyzing the resources utilization and the power consumption) and KE5FX GPIB toolkit (for phase noise measurements).
- Agilent 16821A logic analyzer (for watching the input and output signals behaviour and measuring the bit-error-rate).
- Agilent 33522A function/arbitrary waveform generator (as a reference clock source).
- Tektronix DPO4104B digital phosphor oscilloscope (for eye diagram and signal histogram measurements).
- Tektronix TDS-5104 digital phosphor oscilloscope, equipped with TDSJIT3 software (for analyzing the jitter and the bathtub plot).
- Advantest R3132 spectrum analyzer (for measuring the spectrum).



(a)



(b)

Figure 5.3: Measurement setup used to validate the serial link.

5.3.1 Serializer Performance

The configurations of the frequency synthesizer circuit variables are as follow: the two ring oscillator lengths L1 and L2 are set to be first and second active delay elements, respectively, M and N are set to be nine bits numbers, whereas M is equal to 255 and N varies from 0 to 255, the MF is adjusted to be equal to 64. Changing the frequency range of the F_{REF} signal from 5.23 MHz to 6.05 MHz, allows the generated F_{OUT} signal to be in the range of 334.64 MHz to 387.20 MHz, with a frequency step (i.e. resolution) of 206 kHz.

The frequency synthesizer circuit performance is illustrated in Figure 5.4. The F_{OUT} signal has an RMS and peak-to-peak jitter of 28.7 ps and 258.1 ps, respectively. Figure 5.5(a) shows the output signal of the PRBS generator. Figure 5.5(b) shows the waveforms of the 8-to-1 serializer when the locked state is reached under which the F_{OUT} signal frequency is 64 times the F_{REF} signal frequency. As a result, each one bit

period of the output stream of the serializer, TX_Data, is equal to a one clock cycle of F_{OUT} signal. Figure 5.6 shows the waveform and eye diagram of the serializer output, TX_Data, when transmitting a PRBS 2^{23}-1 pattern. The eye diagram shows RMS and peak-to-peak jitters of 52.84 ps and 392.1 ps, respectively.

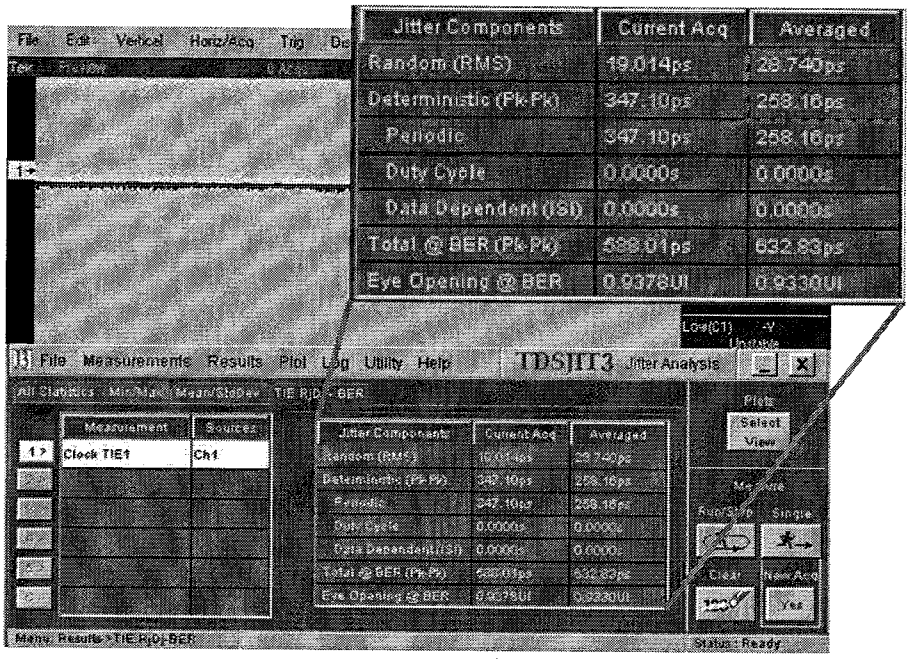
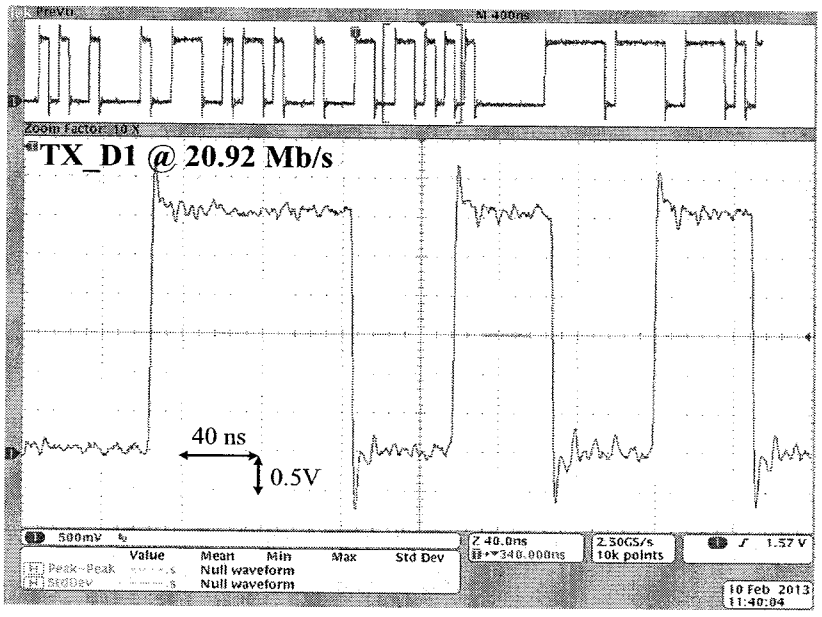
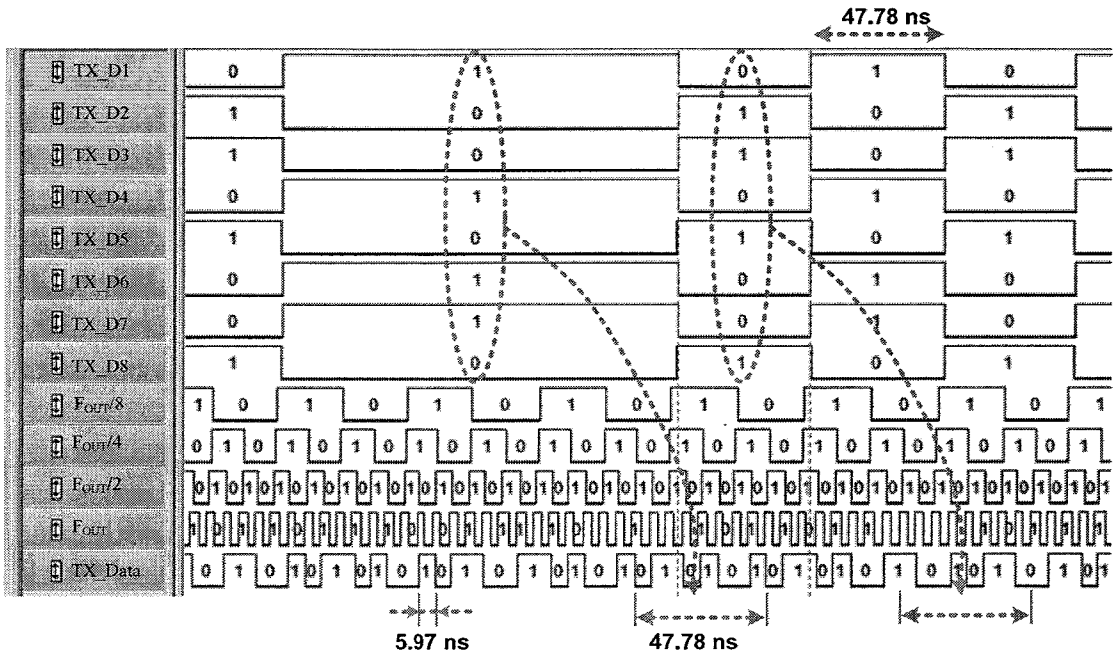


Figure 5.4: Measured RMS and peak-to-peak jitters of the frequency synthesizer output.

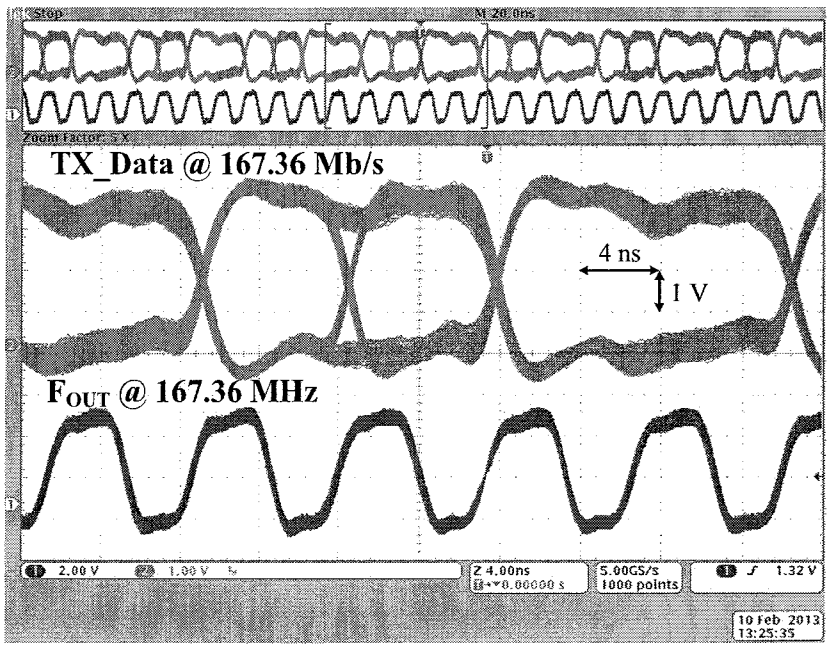


(a)

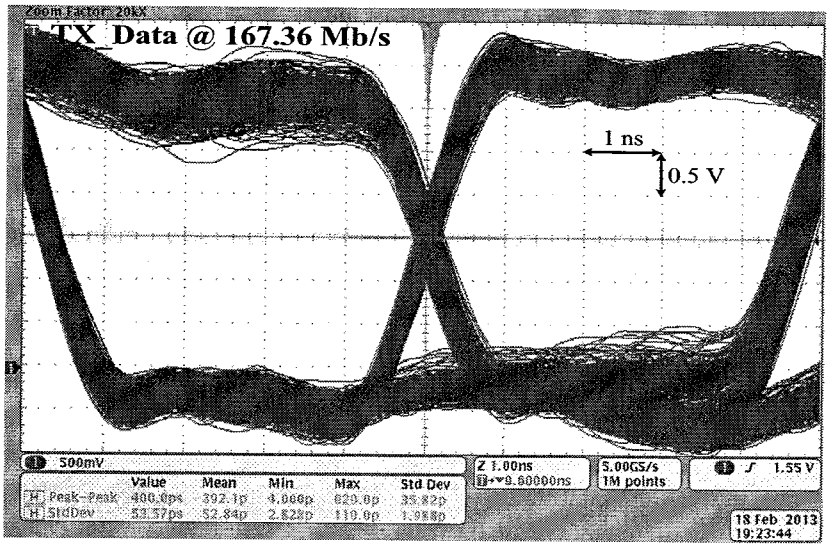


(b)

Figure 5.5: Waveforms of (a) the PRBS generator, (b) the 8-to-1 serializer at the Locked state.



(a)



(b)

Figure 5.6: Measured serializer output: (a) waveform and (b) eye diagram.

5.3.2 Deserializer Performance

The deserializer operates over a limited range of 41.83 Mb/s to 48.4 Mb/s, since the 8-phase generator block divide the maximum output clock of the DCO by 8. The locking range of the serial link is from 167.32 to 193.6 Mb/s, which is four times the recovered clock frequency. Figure 5.7 shows the received data stream affected by the RG174 channel.

Figure 5.8(a) presents the measured spectrum of the recovered clock at 41.85 MHz. Figure 5.8(b) shows the measured jitter histogram of the recovered clock. It shows RMS and peak-to-peak jitters of 49.8 ps and 295.3 ps for a 2^{23}-1 PRBS input data stream, respectively. Figure 5.8(c) shows the measured phase noise of a -100 dBc/Hz at 1 MHz with the carrier frequency of 41.8 MHz. Figure 5.9(a) and Figure 5.9(b) present the CDR circuit when the locked state is reached under which the recovered clock quadrature phases are aligned with the mid-point of the data bits. Figure 5.10(a) shows the measured eye diagram for the 1-to-4 recovered data at 41.8 Mb/s. Figure 5.10(b) shows the measured eye diagram for the 1-to-8 recovered data, RX_D1, at 20.9 Mb/s. It shows RMS and peak-to-peak jitters of 238.3 ps and 800 ps.

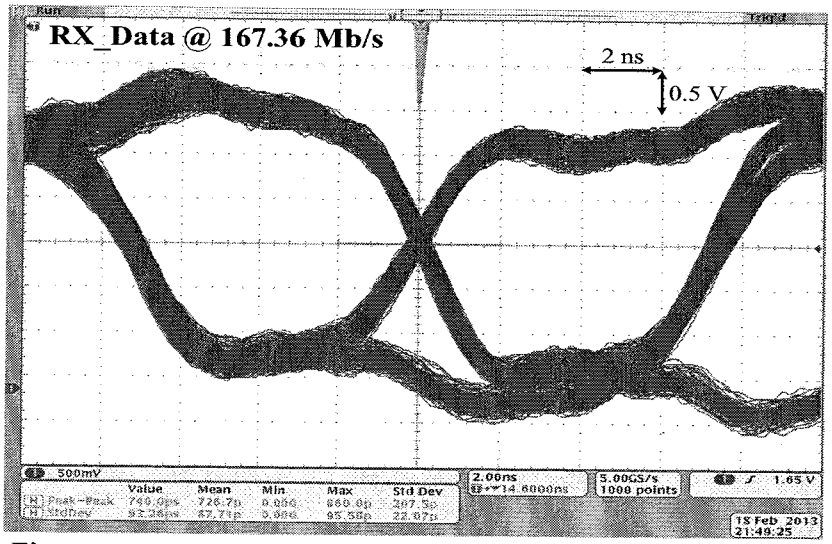
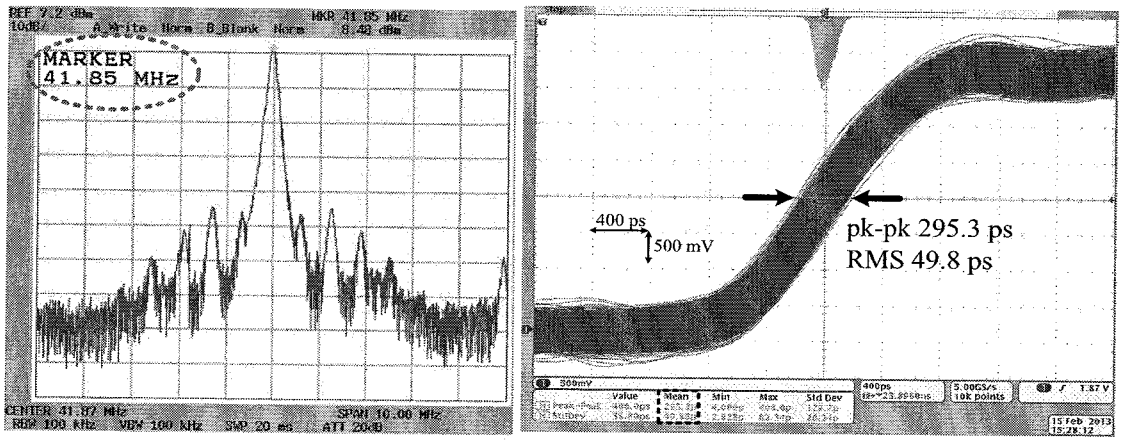
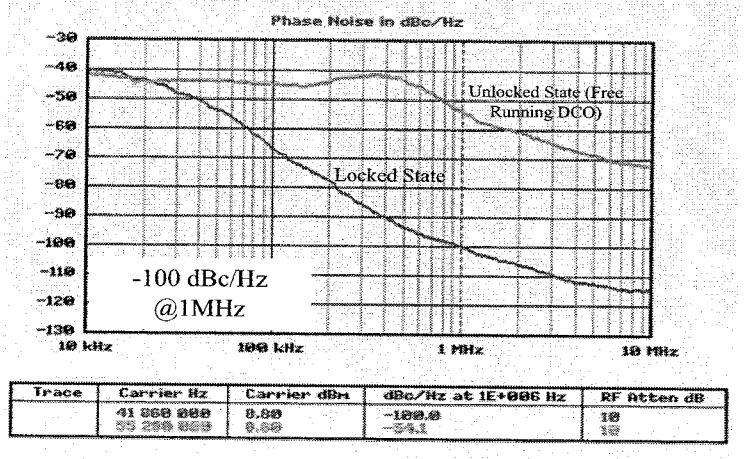


Figure 5.7: Measured eye diagram of the received data stream.

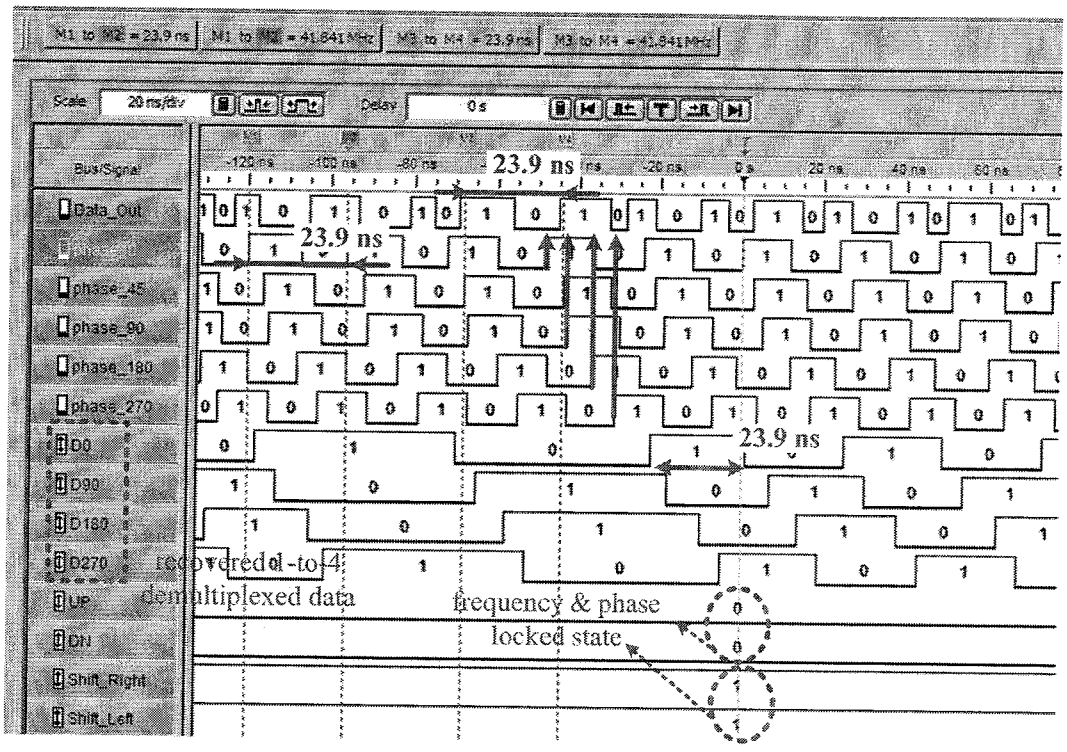


(a) (b)

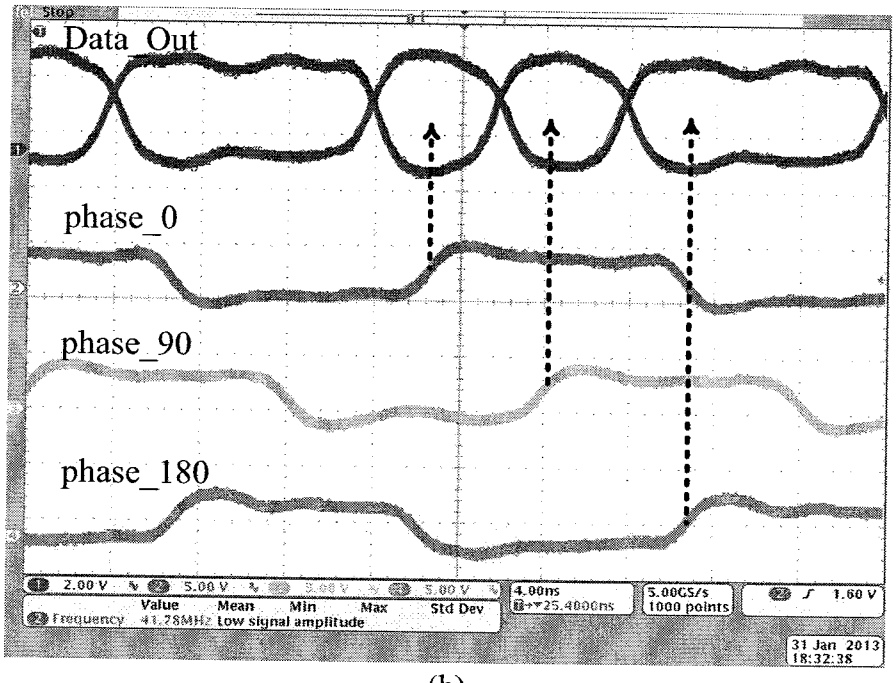


(c)

Figure 5.8: Measured (a) spectrum, (b) jitter histogram, and (c) phase noise of the recovered clock for 2^{23}-1 PRBS input data stream.

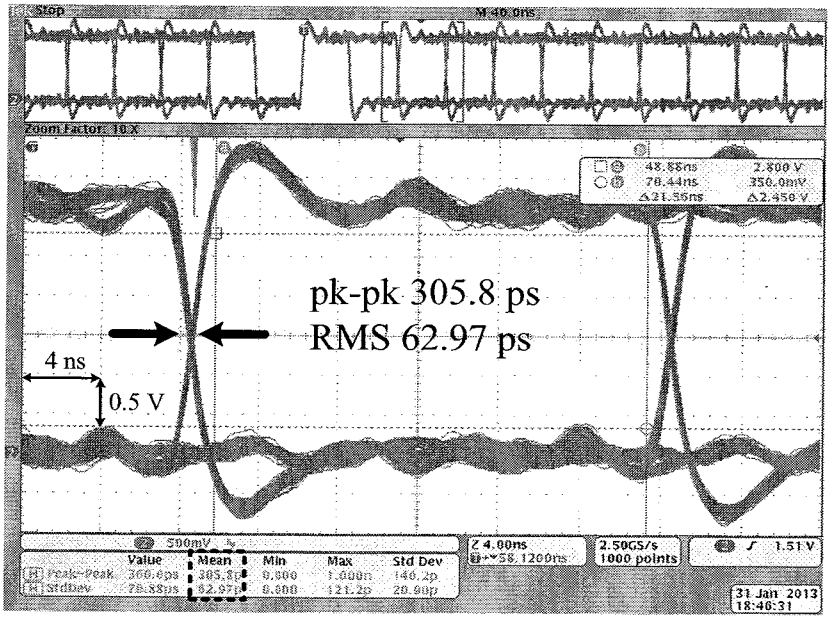


(a)

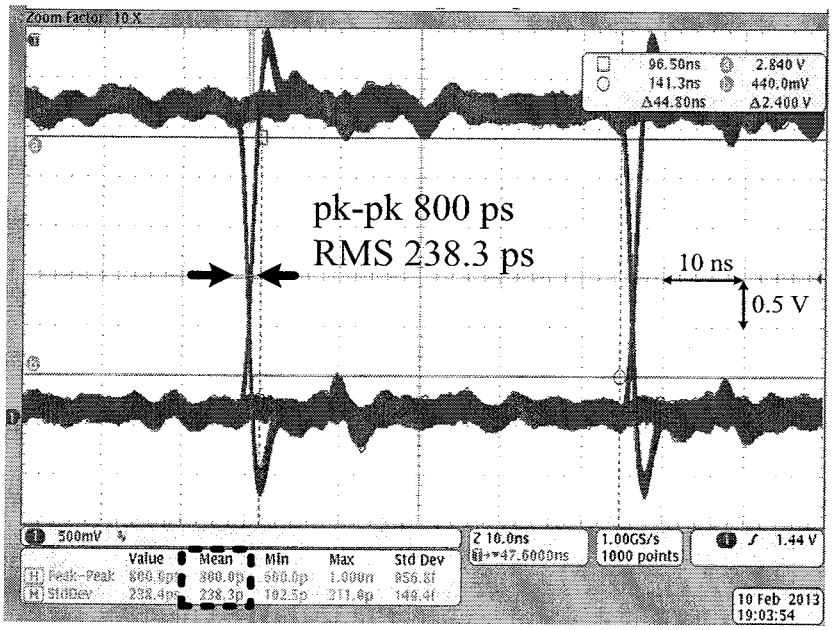


(b)

Figure 5.9: (a) Waveforms of the CDR internal signals and (b) A screen-shot of an oscilloscope for the locked state behaviour.



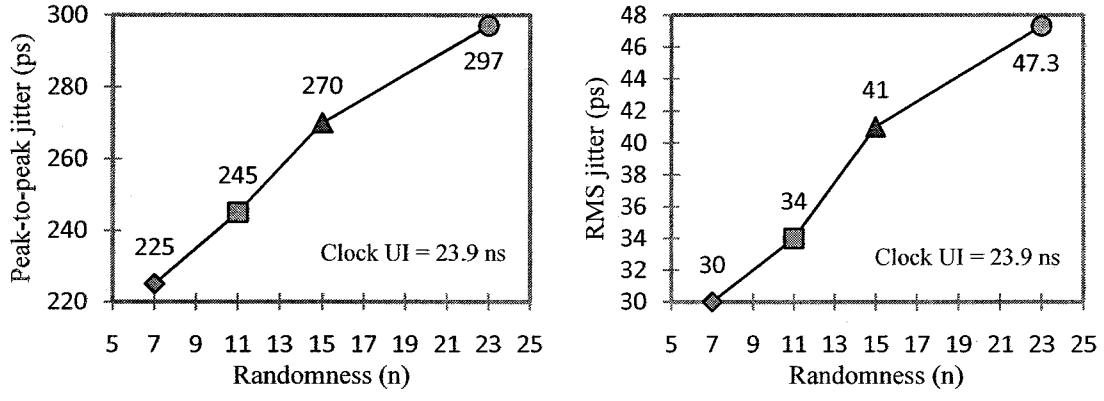
(a)



(b)

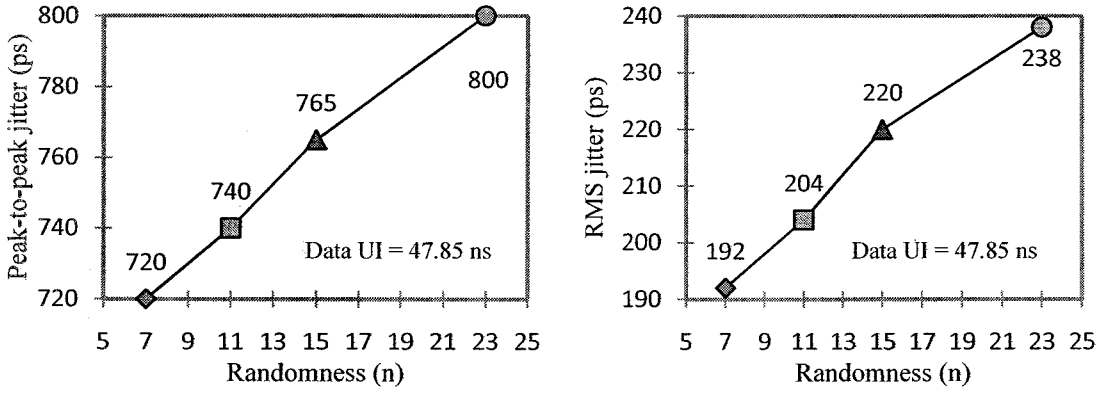
Figure 5.10: Measured eye diagrams of the (a) 1-to-4 and (b) 1-to-8 recovered data, for 2^{23}-1 PRBS input data stream.

As presented in Figure 5.11 and Figure 5.12, a jitter increase is observed when different PRBS pattern lengths (different number of consecutive 1s or 0s) are transmitted over the link. Figure 5.13 shows the BER bathtub curve of the recovered data, RX_D1. The eye opening for a BER of 10^{-12} is 0.97 UI for a 2^{23}-1 PRBS input data stream. This can confirm an error-free operation [92] for BER $<10^{-12}$.



(a) (b)

Figure 5.11: (a) Peak-to-peak jitter and (b) RMS jitter of the recovered clock versus bit length of PRBS (2^n-1) at 41.85 Mb/s.



(a) (b)

Figure 5.12: (a) Peak-to-peak jitter and (b) RMS jitter of the recovered data versus bit length of PRBS (2n-1) at 20.9 Mb/s.

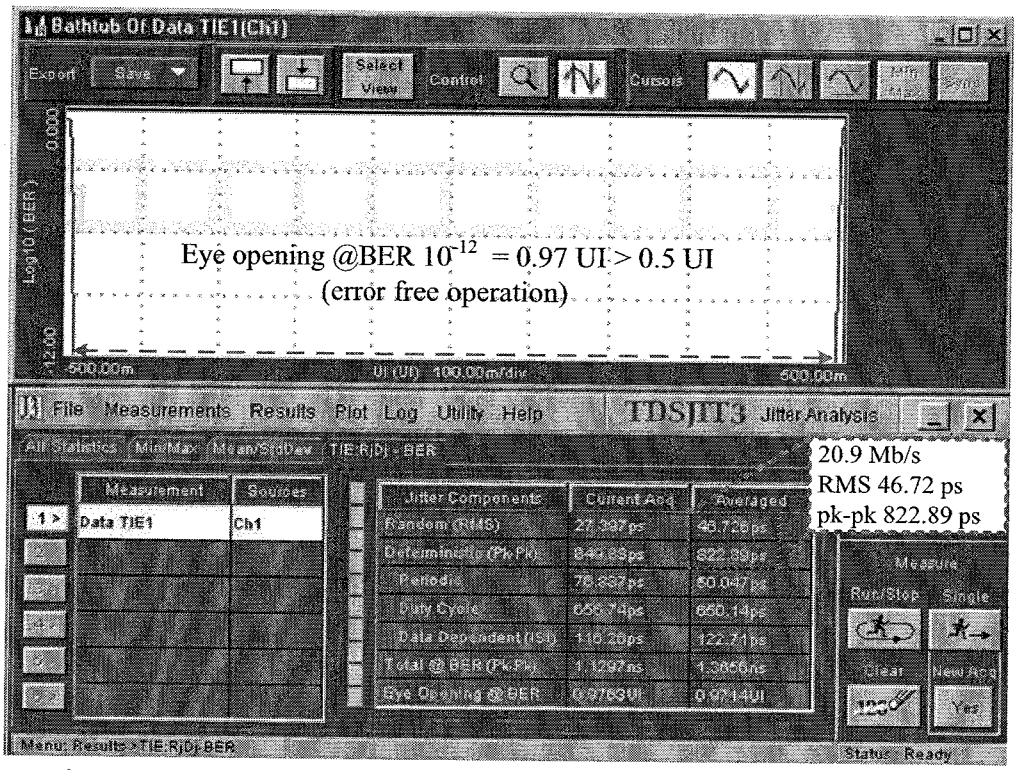


Figure 5.13: Measured BER bathtub curve of the 1-to-8 recovered data.

5.4 Performance Comparison

Although the literature review covered several state of the art works on semi- and fully-digital architectures of the frequency synthesizer and CDR circuits for both platforms, CMOS and FPGA, it is challenging to fairly compare measurements between the two different platforms. This is mainly due to the performance gap between the aforementioned platforms, especially in their operation speed [93]. For a comparable environment, the power consumption and jitter performance should be normalized as a power efficiency (i.e. mW/Gb/s) and percent of the unit interval (i.e. UI%), respectively.

5.4.1 Frequency Synthesizer Performance Comparison

Table 5.1 presents a performance comparison for the proposed frequency synthesizer with the previous designs. The indicated synthesizability and scalability comparison term is basically based on the implementation of the architecture. It is considered as a synthesizable and scalable if the circuit is totally implemented using HDL code or synthesizable cell-based without any extra components and vice versa.

Most of the all-digital frequency synthesizers presented below do not provide true frequency synthesis, for instance the work in [42, 78-80]; rather they require a high-frequency input clock source. This constraint mainly restricts their applications. As discussed in the literature review, the DDFS architectures are able to provide higher frequency resolution (smaller step) than the other architectures, at the expense of their size and maximum output frequency.

Some of the listed architectures (e.g. [78], [94], [22], [95], and [48]) are either dependent on parallel tri-state gates or made full custom to improve the performance (e.g. jitter performance) and hence they are not fully synthesizable architectures. The synthesizability feature can offer better portability and scalability across multiple technologies and platforms with inconsiderable modifications.

5.4.2 CDR Performance Comparison

A performance comparison for the state of the art works on CDR circuits with the proposed circuit is summarized in Table 5.2. The indicated data rate comparison term corresponds to the data speed at the CDR input (i.e. data speed over the channel).

The existing FPGA-based designs presented in [96, 97] basically require extra off-chip components such as the Altera numerically controlled oscillator (NCO) mega core function which constricts their input data rate to few tens of Mb/s. Further, the use of external clean oscillator improves their jitter performance significantly. The analogue architectures presented in [14, 55, 59, 62] occupy a relatively large area of silicon, due to the analogue loop filters that contain large capacitors. The CDR circuits that operate on a single input data rate could show a potential in improving the jitter performance of the recovered clock.

5.4.3 Serial Link Performance Comparison

Table 5.3 presents a performance comparison for the proposed complete serial link with the prior art. These circuits are fundamentally selected because they are suited for chip-to-chip communication.

Table 5.1: Performance comparison for the proposed frequency synthesizer with existing architectures

Ref & Year	Technology	Architecture Type	Maximum Output Frequency	Size/Area	Externally Supplied Clock
This work	*FPGA (Altera Cyclone II)*	*All-Digital dual-loop FLL/DLL*	*387.2 MHz*	*587 LEs*	*NO*
[42] 2010	*90 nm CMOS*	*DDFS*	*612 MHz*	*2 mm^2*	*1300 MHz*
[78] 2010	*FPGA (Altera cyclone III)*	*DDFS*	*60 MHz*	*4341 LEs, 53728 memory bits, 2 DACs, 2 ADCs and 11 multipliers*	*120 MHz*
[79]* 2010	*FPGA (Xilinx)*	*DDFS*	*625 KHz*	*124 slices*	*200 MHz*
[49] 2010	*180 nm CMOS*	*All-Digital cell-based PLL*	*226 MHz*	*0.16 mm^2*	*NO*
[94] 2008	*350 nm CMOS*	*All-Digital DLL*	*450 MHz*	*0.22 mm^2*	*NO*
[80] 2006	*FPGA (Xilinx virtex)*	*All-Digital PLL*	*125 MHz*	*Not Provided*	*YES*
[22] 2004	*350 nm CMOS*	*All-Digital PLL*	*366 MHz*	*0.07 mm^2*	*NO*
[95] 2003	*250 nm CMOS*	*Dual-loop FLL/DLL*	*500 MHz*	*0.21 mm^2*	*NO*
[48] 2003	*350 nm CMOS*	*All-Digital cell-based PLL*	*510 MHz*	*0.71 mm^2*	*NO*
[76] 2003	*FPGA (Xilinx V400BG432)*	*All-Digital PLL*	*40 MHz*	*672 slices*	*NO*

* Simulation Results

Table 5.1: Performance comparison for the proposed frequency synthesizer with existing architectures (Continued)

Ref & Year	Jitter	Normalized Jitter	Step Resolution	Multiplication Factor	Description (Synthesizability and Scalability)
This work	*@167.36 MHz p-p 258.1 ps RMS 28.7 ps*	*UI=5.97 ns p-p 4.3% RMS 0.48%*	*206 kHz*	*64*	*Fully synthesizable and scalable*
[42] 2010	*Not Provided*	*N/A***	*77.5 Hz*	*1/2*	*Hybrid*
[78] 2010	*Not Provided*	*N/A*	*117.2 kHz*	*1/2*	*Partially synthesizable VHDL-code, and requires off-chip components*
[79]* 2010	*Not Provided*	*N/A*	*3.05 kHz*	*1/320*	*Synthesizable Verilog-code and requires high-frequency input clock*
[49] 2010	*@187.5 MHz p-p 130 ps RMS N/A*	*UI=5.33 ns p-p 2.4%*	*127.4 kHz*	*Not Provided*	*Synthesizable cell-based and contains two DCO.*
[94] 2008	*@450 MHz p-p 37.8 ps RMS 4 ps*	*UI=2.22 ns p-p 1.7% RMS 0.18%*	*Not Provided*	*15*	*Synthesizable cell-based except the delay elements*
[80] 2006	*@125 MHz p-p 490 ps RMS N/A*	*UI =8 ns p-p 6.125%*	*Not Provided*	*25*	*Synthesizable code and requires high-frequency input clock*
[22] 2004	*@5 MHz p-p 1.2 ns RMS N/A*	*UI=200 ns p-p 0.6%*	*1020 kHz*	*64*	*Synthesizable VHDL-code except the DCO (parallel tri-state gates)*
[95] 2003	*@ 500 MHz p-p 40 ps RMS 5.84 ps*	*UI=2 ns p-p 2% RMS 0.29%*	*Not Provided*	*4*	*Analogue*
[48] 2003	*@450 MHz p-p 70 ps RMS 22 ps*	*UI=2.22 ns p-p 3.15% RMS 0.99%*	*1300 kHz*	*40*	*Synthesizable cell-based except the two DCO.*
[76] 2003	*Not Provided*	*N/A*	*Not Provided*	*Not Provided*	*Synthesizable VHDL-code and scalable design*

* Simulation Results

** Not Applicable

Table 5.2: Performance comparison for the proposed CDR with the prior art

Ref & Year	Tech.	Architecture Type	Clocking Architecture	Size/ Area	Inherent DEMUX	Input Data Rate	Recovered Clock Frequency
This Work	*FPGA (Altera Cyclone II)*	*Combination of FLL/DLL based*	*Quarter-rate*	*507 LEs*	*1 to 4*	*167.32 to 193.6 Mb/s*	*41.83 to 48.4 MHz*
[14] 2012	*180 nm CMOS*	*PLL-based*	*Quarter-rate*	*4.37 mm^2*	*NO*	*5 Gb/s*	*1.25 GHz*
[57] 2011	*130 nm CMOS*	*All-digital PLL-based*	*Half-rate*	*0.074 mm^2*	*NO*	*1 to 4 Gb/s*	*0.4 to 2.1 GHz*
[58] 2011	*65 nm CMOS*	*All-digital PLL-based*	*Full-rate*	*0.022 mm^2*	*NO*	*480 Mb/s*	*480 MHz*
[96] 2011	*FPGA (Altera Stratix)*	*All-digital PLL-based without frequency loop*	*Full-rate*	*Not provided*	*NO*	*1 to 12 Mb/s*	*1 to 12 MHz*
[55] 2008	*180 nm CMOS*	*Combination of FLL/DLL based*	*Quarter-rate*	*0.8 mm^2*	*NO*	*0.2 to 4 Gb/s*	*50 to 1000 MHz*
[97] 2007	*FPGA (Xilinx Virtex4)*	*All-digital PLL-based*	*Full-rate*	*249 Slices*	*NO*	*25 Mb/s*	*25 MHz*
[59] 2006	*180 nm CMOS*	*PLL-based*	*Full-rate*	*0.88 mm^2*	*NO*	*155 to 3125 Mb/s*	*155 to 3125 MHz*
[98] 2006	*180 nm CMOS*	*Combination of FLL/DLL based*	*Full-rate*	*0.042 mm^2*	*NO*	*1.25 Gb/s*	*1.25 GHz*
[56] 2005	*350 nm CMOS*	*Combination of FLL/DLL based*	*Full-rate*	*9 mm^2*	*NO*	*12.5 to 2700 Mb/s*	*12.5 to 2700 MHz.*
[62] 2003	*120 nm CMOS*	*Combination of FLL/DLL based*	*Full-rate*	*1.94 mm^2*	*NO*	*10 Gb/s*	*10 GHz*

Table 5.2: Performance comparison for the proposed CDR with the prior art (Continued)

Ref & Year	Recovered Clock		Power Consumption		Externally Supplied Clock	Description (Synthesizability and Scalability)
	Jitter	Normalized Jitter	mW	mW/Gb/s		
This Work	@41.8 MHz p-p 295.3 ps RMS 49.8 ps	UI=23.92 ns p-p1.24% RMS 0.21%	0.59 @167.3 Mb/s	3.52	Referenceless	Fully synthesizable and scalable
[14] 2012	@1.25 GHz p-p 30.4 ps RMS N/A	UI= 0.8 ns p-p3.8%	71.9 @5 Gb/s	14.38	Requires reference clock	Synthesizable cell-based except charge pump, LPF and DCO.
[57] 2011	@1.5 GHz p-p 29.2 ps RMS 3.58 ps	UI= 0.67 ns p-p4.38% RMS 0.53%	14.9 @4 Gb/s	3.73	Requires reference clock	Synthesizable cell-based except phase detector, deserializer and DCO.
[58] 2011	Not Provided	N/A	1.75 @480Mb/s	3.65	Referenceless	Synthesizable cell-based
[96] 2011	@1 MHz p-p 10 ns RMS N/A	UI =1000 ns p-p1%	Not provided	N/A	200 MHz	Partially synthesizable code and requires off-chip ADC, DAC, and Altera NCO mega core function as a DCO.
[55] 2008	@500MHz p-p 232 ps RMS 28 ps	UI= 2 ns p-p11.6% RMS 1.4%	14 @4 Gb/s	3.5	Requires reference clock	NOT synthesizable and requires passive elements
[97] 2007	@25 MHz p-p 372.7 ps RMS 51 ps	UI= 40 ns p-p0.93% RMS 0.12%	Not provided	N/A	100 MHz	Synthesizable code and requires a 100MHz input clock to generate multi-phase signals
[59] 2006	@3.125 GHz p-p 48.9 ps RMS 6.4 ps	UI= 0.32 ns p-p15.28% RMS 2%	95 @3.12Gb/s	30.4	Referenceless	Partially synthesizable cell-based
[98] 2006	@1.25GHz p-p 154 ps RMS 16.89 ps	UI= 0.8 ns p-p19.25% RMS 2.11%	17.8 @1.25Gb/s	14.24	156.25 MHz	Synthesizable loop controllers only
[56] 2005	Not provided	N/A	Not provided	N/A	Referenceless	NOT synthesizable and requires passive elements
[62] 2003	Not provided	N/A	550 @10 Gb/s	55	Referenceless	NOT synthesizable and requires passive elements

Table 5.3: Performance comparison for the complete serial link with the prior art

Ref & Year	Applications	Technology	Channel Data Rate	Power Consumption	Normalized Power *mW/Gb/s*
This Work	*Chip-to-Chip (wireline)*	*FPGA (Altera Cyclone II)*	*167.32 to 193.6 Mb/s*	*0.97 mW*	*5.79*
[9] 2012	*Chip-to-Chip (optical)*	*40nm CMOS*	*28 Gb/s*	*225 mW*	*8.035*
[10] 2012	*Chip-to-Chip (wireline)*	*32nm CMOS*	*28 Gb/s*	*693 mW*	*24.75*
[11] 2010	*chip-to-chip (optical)*	*65nm CMOS*	*39.8 to 44.6 Gb/s*	*3 W @40 Gb/s*	*75*
[12] 2009	*Chip-to-Chip (wireline)*	*130nm CMOS*	*40 Gb/s*	*3.6 W*	*90*
[13] 2009	*Chip-to-Chip (wireline)*	*65nm CMOS*	*40 Gb/s*	*5.6 W*	*140*
[99] 2005	*Chip-to-Chip (wireline)*	*180nm CMOS*	*3.125 Gb/s*	*178 mW*	*56.96*
[100] 2005	*Chip-to-Chip (wireline)*	*130nm CMOS*	*4.8 to 6.4 Gb/s*	*310 mW @6.4 Gb/s*	*48.44*
[101] 1998	*Chip-to-Chip (wireline)*	*500nm CMOS*	*4 Gb/s*	*1.5 W*	*375*

Table 5.3: Performance comparison for the complete serial link with the prior art (Continued)

Ref/ Year	Jitter		Normalized Jitter		Clocking Architecture
	TX Output Stream	RX Recovered clock	TX Output Stream UI%	RX Recovered clock UI%	
This Work	*@167.3 Mb/s p-p 392.1 ps RMS 52.84 ps*	*@41.8 MHz p-p 295.3 ps RMS 49.8 ps*	*UI=5.97 ns p-p 6.57% RMS 0.89%*	*UI=23.89 ns p-p 1.24% RMS 0.21%*	*Clockless Quarter-rate*
[9] 2012	*@28 Gb/s RMS 350 fs*	*Not Provided*	*UI = 35.7 ps RMS 0.98%*	*N/A*	*Clockless Quarter-rate*
[10] 2012	*@28 Gb/s p-p 5 ps RMS 450 fs*	*Not Provided*	*UI = 35.7 ps p-p 14% RMS 1.26%*	*N/A*	*Forwarded Clock (Shared PLL)*
[11] 2010	*@22 Gb/s p-p 9.96 ps*	*Not Provided*	*UI = 45.45 ps p-p 21.9%*	*N/A*	*Clockless Half-rate*
[12] 2009	*@39 Gb/s. p-p 11.76 ps RMS 1.85 ps*	*@1.25 GHz p-p 15.55ps RMS 1.77 ps*	*UI = 25.6 ps p-p 45.9% RMS 7.22%*	*UI = 800 ps p-p 1.94% RMS 0.22%*	*Clockless Half-rate*
[13] 2009	*@39.8 Gb/s p-p 3.1 ps RMS 570 fs*	*@20 GHz RMS 15mUI*	*UI = 25.1 ps p-p 12.35% RMS 2.27%*	*UI = 50 ps RMS 1.5%*	*Clockless Half-rate*
[99] 2005	*@3.125Gb/s p-p 46 ps RMS 5.05 ps*	*@3.125GHz p-p 64 ps RMS 11.36 ps*	*UI = 320 ps p-p 14.38% RMS 1.58%*	*UI = 320 ps p-p 20% RMS 3.55%*	*Forwarded Clock (Shared PLL)*
[100] 2005	*@6.4 Gb/s p-p 41 ps RMS 6 ps*	*@6.4 GHz p-p 114 ps RMS 20 ps*	*UI = 156.25 p-p 26.24% RMS 3.84%*	*UI = 153.25 p-p 74.39% RMS 13.05%*	*Forwarded Clock (Shared PLL)*
[101] 1998	*@4.8 Gb/s, p-p 90 ps RMS N/A*	*Not Provided*	*UI = 208.3 ps p-p 43.2%*	*N/A*	*Using Oversampling (24 phases)*

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Chapter Overview

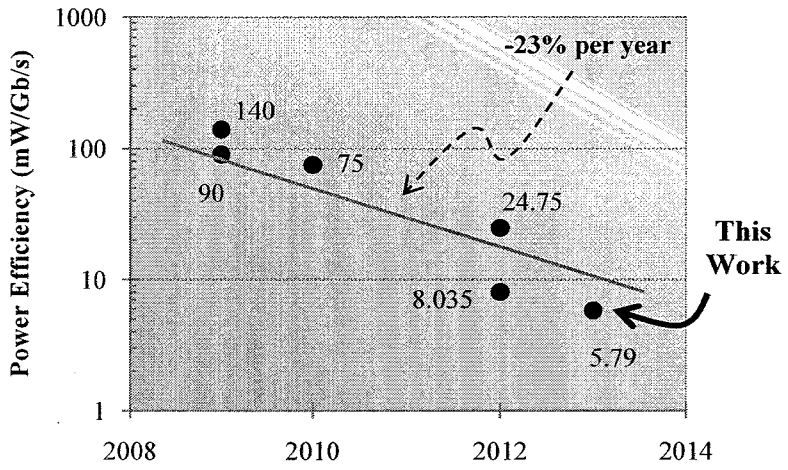
This chapter reviews the main points in the thesis. The contributions from this thesis are listed hereafter.

6.2 Conclusion

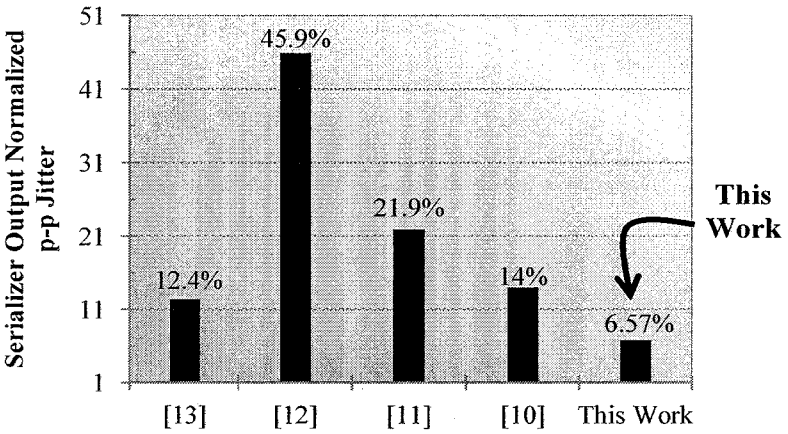
This thesis has presented a fully-digital low-power low-jitter complete serial link system synthesized into Altera Cyclone II FPGA, as illustrated in Figure 6.1. Two fundamental strategies have been used to achieve low-power low-noise design.

The first is eliminating the need for any analogue or off-chip components and completely realizing the serial link as a Verilog-based circuit. This gives the ability to use only rail-to-rail CMOS logics (i.e. no DC biasing-current) instead of power-hungry low-swing current-mode logics, which consume constant current and generate high phase noise.

Being a verilog-based provides the scalability (portability) feature across multiple technologies and platforms with inconsiderable modifications. Further, it gives a high flexibility to program all design parameters (e.g. the link data rate and frequency step resolution) in a very short time. This strategy provides a favourable tradeoff between the maximum data rate achieved and power efficiency. The second strategy is achieving a quarter-rate CDR operations.



(a)



(b)

Figure 6.1: (a) Power efficiency and (b) Jitter performance of the proposed system and recently published serial links.

6.3 Thesis Contributions

This research proposes a truly-digital quarter-rate clockless serial link system to be used for inter/intra-chip communication in SoC. Regardless of the standard for which the system is intended to be used, the fully-digital implementation shows a potential in both minimizing the power consumption and improving the jitter performance. Furthermore, it reduces the cost due to the reduction in the circuit size and improves the scalability and programmability of the design. The contributions of this thesis can be summarized as follows:

- A fully-digital high-resolution dual-loop frequency synthesizer architecture has been proposed, which has the ability to generate up to 64 times higher than the system reference clock.

- A proof-of-concept of referenceless quarter-rate phase and frequency detectors has been adopted.

- A new way of generating eight 22.5°-spaced phases and their complements has been proposed. It exhibits wide working frequency range and requires less input clock frequency than the FSM-based.

- A complete clockless quarter-rate serial link system has been designed, implemented, and experimentally validated.

Additionally, the research work in this thesis has been published/presented in the following (SCOPUS and ISI indexed) journals and conferences:

- M. H. Alser and M. Assaad, "An HDL-based Serial Link for Point-to-Point Communication in NoC," **IEEE Embedded Systems Letters**, March 2013 [submitted].

- M. Assaad, M. H. Alser, and A. Bermak, "Design and Characterization of Low Power and Low Noise Truly All-Digital Clock and Data Recovery Circuit for SERDES Devices," **Journal of Low Power Electronics**, ASP, vol. 9, no. 1, April 2013.

- M. H. Alser, M. Assaad, and F. A. Hussin, "A Wide-Range Programmable Frequency Synthesizer Based on a Finite State Machine Filter", **International Journal of Electronics**, pp. 1-11, 2012/12/29.

- M. Assaad and M. H. Alser, "Design of an All-Digital Synchronized Frequency Multiplier Based on a Dual-Loop (D/FLL) Architecture", **VLSI Design**, vol. 2012, Article ID 546212, 7 pages, 2012.

- M. Assaad and M. H. Alser, "An FPGA-Based Design and Implementation of an All-Digital Serializer for Inter Module Communication in SoC," **IEICE Electronics Express**, vol. 8, no. 23, pp. 2017-2023, 2011.

- M. H. Alser, M. Assaad, F. Hussin, and I. Y. Bayou, "A Novel Low-Power Clock and Data Recovery Circuit using a Quarter-Rate Phase Detector for Inter-Module Communication in SoC," **4th IEEE International Conference on Intelligent and Advanced Systems (ICIAS)**, Kuala Lumpur, Malaysia, 2012.

- M. H. Alser and M. Assaad, "Design and Modeling of Low-Power Clockless Serial Link for Data Communication Systems," **3rd IEEE National Postgraduate Conference (NPC)**, pp.1-5, 2011.

6.4 Future Work

Serial link architectures are constantly evolving due to technological progress and the ever-increasing communication bandwidth requirements. This thesis provides detailed information of FPGA-based system development. It can be used as a base for the ongoing and future projects conducted in Electronic Materials and Devices Research Cluster. The recommended possible future research directions are as follows:

- Due to the lack of equipments, the jitter performance of the quarter-rate CDR circuit cannot be characterized. As a future work, further measurements including the jitter generation, jitter transfer, and jitter tolerance can be verified using equipment such as the J-BERT N4903A or other.

- Investigating the possibility of improving the jitter performance by modelling each component of the serial link using Matlab to estimate the contribution of each one to the total jitter.

- Exploring the use of the proposed serial link in inter-module communication for Network-on-Chip (NoC). This includes designing a fast network router to control the traffic between the modules.

REFERENCES

[1] J. Moreira and H. Werkmann, "An Engineer's Guide to Automated Testing of High-Speed Interfaces," Artech House, Boston, 2010.

[2] S. Sidiropoulos, "High Performance Inter-Chip Signaling," Ph.D. Dissertation, Stanford University, April 1998.

[3] F. Bo and P. Ampadu, "Exploiting Parity Computation Latency for On-Chip Crosstalk Reduction," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 57, no. 5, pp. 399-403, May 2010.

[4] Z. Khan, T. Arslan, and A. T. Erdogan, "Low power system on chip bus encoding scheme with crosstalk noise reduction capability," *IEE Proceedings - Computers and Digital Techniques,* vol. 153, no. 2, pp. 101-108, 2006.

[5] R. R. Dobkin, A. Morgenshtein, A. Kolodny, and R. Ginosar, "Parallel vs. serial on-chip communication," in *international workshop on System level interconnect prediction*, 2008, pp. 43-50.

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

[7] E. Yeung, "Design of high-performance and low-cost parallel links," Ph.D. Dissertation, Stanford University, Jan. 2002.

[8] A. Morgenshtein, I. Cidon, A. Kolodny, and R. Ginosar, "Comparative analysis of serial vs parallel links in NoC," in *International Symposium on System-on-Chip*, 2004, pp. 185-188.

[9] M. Harwood, S. Nielsen, A. Szczepanek, R. Allred, S. Batty, M. Case, S. Forey, K. Gopalakrishnan, L. Kan, and B. Killips, "A 225mW 28Gb/s SerDes in 40nm CMOS with 13dB of analog equalization for 100GBASE-LR4 and

optical transport lane 4.4 applications," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2012, pp. 326-327.

[10] J. F. Bulzacchelli, C. Menolfi, T. J. Beukema, D. W. Storaska, J. Hertle, D. R. Hanson, H. Ping-Hsuan, S. V. Rylov, D. Furrer, D. Gardellini, A. Prati, T. Morf, V. Sharma, R. Kelkar, H. A. Ainspan, W. R. Kelly, L. R. Chieco, G. A. Ritter, J. A. Sorice, J. D. Garlett, R. Callan, M. Brandli, P. Buchmann, M. Kossel, T. Toifl, and D. J. Friedman, "A 28-Gb/s 4-Tap FFE/15-Tap DFE Serial Link Transceiver in 32-nm SOI CMOS Technology," *IEEE Journal of Solid-State Circuits,* vol. 47, no. 12, pp. 3232-3248, December, 2012.

[11] N. Nedovic, A. Kristensson, S. Parikh, S. Reddy, S. McLeod, N. Tzartzanis, K. Kanda, T. Yamamoto, S. Matsubara, and M. Kibune, "A 3 Watt 39.8–44.6 Gb/s Dual-Mode SFI5. 2 SerDes Chip Set in 65 nm CMOS," *IEEE Journal of Solid-State Circuits,* vol. 45, no. 10, pp. 2016-2029, 2010.

[12] J. K. Kim, J. Kim, K. I. M. Gyudong, and D. K. Jeong, "A Fully Integrated 0.13-μm CMOS 40-Gb/s Serial Link Transceiver," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 5, pp. 1510-1521, 2009.

[13] S. Kaeriyama, Y. Amamiya, H. Noguchi, Z. Yamazaki, T. Yamase, K. Hosoya, M. Okamoto, S. Tomari, H. Yamaguchi, and H. Shoda, "A 40 Gb/s multi-data-rate CMOS transmitter and receiver chipset with SFI-5 interface for optical transmission systems," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 12, pp. 3568-3579, 2009.

[14] Y. S. Tan, K. S. Yeo, C. C. Boon, and M. A. Do, "A Dual-Loop Clock and Data Recovery Circuit With Compact Quarter-Rate CMOS Linear Phase Detector," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 59, no. 6, pp. 1156-1167, 2012.

[15] P. R. Panda, B. V. N. Silpa, A. Shrivastava, and K. Gummidipudi, "Power-efficient System Design," Springer, 2010.

[16] A. Tajalli and Y. Leblebici, "Extreme Low-Power Mixed Signal IC Design: Subthreshold Source-Coupled Circuits," Springer, 2010.

[17] E. Morifuji, T. Yoshida, M. Kanda, S. Matsuda, S. Yamada, and F. Matsuoka, "Supply and threshold-voltage trends for scaled logic and SRAM MOSFETs," *IEEE Transactions on Electron Devices,* vol. 53, no. 6, pp. 1427-1432, 2006.

[18] S. Thompson, P. Packan, and M. Bohr, "MOS scaling: Transistor challenges for the 21st century," *Intel Technology Journal,* vol. Q3, 1998.

[19] H. Hassan, M. Anis, and M. Elmasry, "MOS current mode circuits: analysis, design, and variability," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 13, no. 8, pp. 885-898, 2005.

[20] Y. Wu and J. Hu, "Low-voltage MOS Current Mode Logic for Low-Power and High Speed Applications," *Information Technology Journal,* vol. 10, no. 12, pp. 2470-2475, 2011.

[21] M. Anis, M. Allam, and M. Elmasry, "Impact of technology scaling on CMOS logic styles," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 49, no. 8, pp. 577-588, 2002.

[22] T. Olsson and P. Nilsson, "A digitally controlled PLL for SoC applications," *IEEE Journal of Solid-State Circuits,* vol. 39, no. 5, pp. 751-760, 2004.

[23] D. Siprak, M. Tiebout, N. Zanolla, P. Baumgartner, and C. Fiegna, "Noise reduction in CMOS circuits through switched gate and forward substrate bias," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 7, pp. 1959-1967, 2009.

[24] L. Ding and P. Mazumder, "On circuit techniques to improve noise immunity of CMOS dynamic logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 12, no. 9, pp. 910-925, 2004.

[25] A. Hajimiri, S. Limotyrakis, and T. H. Lee, "Jitter and phase noise in ring oscillators," *IEEE Journal of Solid-State Circuits,* vol. 34, no. 6, pp. 790-804, 1999.

[26] A. A. Abidi, "Phase noise and jitter in CMOS ring oscillators," *IEEE Journal of Solid-State Circuits,* vol. 41, no. 8, pp. 1803-1816, 2006.

[27] M. Hsieh and G. Sobelman, "Architectures for multi-gigabit wire-linked clock and data recovery," *IEEE Circuits and Systems Magazine,* vol. 8, no. 4, pp. 45-57, 2008.

[28] M. Assaad, "Design and modelling of clock and data recovery integrated circuit in 130 nm CMOS technology for 10 Gb/s serial data communications," Ph.D. Dissertation, University of Glasgow, Jan. 2009.

[29] M. Ghoneima, Y. Ismail, M. M. Khellah, J. Tschanz, and V. De, "Serial-link bus: A low-power on-chip bus architecture," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 56, no. 9, pp. 2020-2032, 2009.

[30] Q. Sun, C. Hu-Guo, K. Jaaskelainen, I. Valin, X. Fang, Y. Zhang, M. Winter, and Y. Hu, "The Development of On-Chip Serial Link Transmitter for MAPS," *IEEE Transactions on Nuclear Science,* vol. 57, no. 2, pp. 543-549, 2010.

[31] E. Nigussie, S. Tuuna, J. Plosila, J. Isoaho, and H. Tenhunen, "Semi-Serial On-Chip Link Implementation for Energy Efficiency and High Throughput," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 20, no. 12, pp. 2265-2277, 2012.

[32] R. Nair, N. Y. Borkar, C. S. Browning, G. E. Dermer, V. Eriaguntla, V. Govindarajulu, A. Pangal, J. D. Prijic, L. Rankin, and E. Seligman, "A 28.5 GB/s CMOS non-blocking router for terabit/s connectivity between multiple processors and peripheral I/O nodes," in *IEEE International Solid-State*

Circuits Conference Digest of Technical Papers (ISSCC), 2001, pp. 224-225, 450.

[33] S. J. Jou, S. H. Kuo, J. T. Chiu, C. King, C. H. Lee, and T. Liu, "A serial link transceiver for USB2 high-speed mode," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2001, pp. 72-75.

[34] S. G. Servent, M. Duzynski, P. Nauber, M. Scholles, and U. Schelinski, "Novel bridge for networking IEEE 1394 clusters via UWB over coaxial cable for HANA & AV/C digital multimedia networks," *IEEE Transactions on Consumer Electronics,* vol. 54, no. 2, pp. 507-512, 2008.

[35] B. Casper and F. O'Mahony, "Clocking analysis, implementation and measurement techniques for high-speed data links - a tutorial," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 56, no. 1, pp. 17-39, 2009.

[36] G. Torralba, V. Angelov, V. GonzÃ¡lez, V. Lindenstruth, and E. Sanchis, "A VLSI for deskewing and fault tolerance in LVDS links," *IEEE Transactions on Nuclear Science,* vol. 53, no. 3, pp. 801-809, 2006.

[37] K. Yamaguchi, Y. Hori, K. Nakajima, K. Suzuki, M. Mizuno, and H. Hayama, "A 2.0 Gb/s Clock-Embedded Interface for Full-HD 10-Bit 120 Hz LCD Drivers With 1/5-Rate Noise-Tolerant Phase and Frequency Recovery," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 12, pp. 3560-3567, 2009.

[38] A. Ragab, Y. Liu, K. Hu, P. Chiang, and S. Palermo, "Receiver jitter tracking characteristics in high-speed source synchronous links," *Journal of Electrical and Computer Engineering,* vol. 2011, Article ID 982314, 15 pages, 2011.

[39] R. Farjad-Rad, C. K. K. Yang, and M. A. Horowitz, "A 0.3-μm CMOS 8-Gb/s 4-PAM serial link transceiver," *IEEE Journal of Solid-State Circuits,* vol. 35, no. 5, pp. 757-764, 2000.

[40] D. J. Foley and M. P. Flynn, "A low-power 8-PAM serial transceiver in 0.5-μm digital CMOS," *IEEE Journal of Solid-State Circuits,* vol. 37, no. 3, pp. 310-316, 2002.

[41] F. F. Dai, W. Ni, S. Yin, and R. C. Jaeger, "A direct digital frequency synthesizer with fourth-order phase domain $\Delta\sum$ noise shaper and 12-bit current-steering DAC," *IEEE Journal of Solid-State Circuits,* vol. 41, no. 4, pp. 839-850, 2006.

[42] H. C. Yeoh, J. H. Jung, Y. H. Jung, and K. H. Baek, "A 1.3-GHz 350-mW hybrid direct digital frequency synthesizer in 90-nm CMOS," *IEEE Journal of Solid-State Circuits,* vol. 45, no. 9, pp. 1845-1855, 2010.

[43] A. Bellaouar, M. S. O'Brecht, A. M. Fahim, and M. I. Elmasry, "Low-power direct digital frequency synthesis for wireless communications," *IEEE Journal of Solid-State Circuits,* vol. 35, no. 3, pp. 385-390, 2000.

[44] J. Choi, S. T. Kim, W. Kim, K. W. Kim, K. Lim, and J. Laskar, "A low power and wide range programmable clock generator with a high multiplication factor," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 19, no. 4, pp. 701-705, 2011.

[45] B. Mesgarzadeh and A. Alvandpour, "A low-power digital DLL-based clock generator in open-loop mode," *IEEE Journal of Solid-State Circuits,* vol. 44, no. 7, pp. 1907-1913, 2009.

[46] R. C. H. van de Beek, E. A. M. Klumperink, C. S. Vaucher, and B. Nauta, "Low-jitter clock multiplication: a comparison between PLLs and DLLs," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing,* vol. 49, no. 8, pp. 555-566, 2002.

[47] O. Casha, I. Grech, F. Badets, D. Morche, and J. Micallef, "Analysis of the spur characteristics of edge-combining DLL-based frequency multipliers,"

IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 56, no. 2, pp. 132-136, 2009.

[48] C. C. Chung and C. Y. Lee, "An all-digital phase-locked loop for high-speed clock generation," *IEEE Journal of Solid-State Circuits,* vol. 38, no. 2, pp. 347-351, 2003.

[49] G. N. Sung, S. C. Liao, J. M. Huang, Y. C. Lu, and C. C. Wang, "All-digital frequency synthesizer using a flying adder," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 57, no. 8, pp. 597-601, 2010.

[50] B. M. Helal, M. Z. Straayer, G. Y. Wei, and M. H. Perrott, "A highly digital MDLL-based clock multiplier that leverages a self-scrambling time-to-digital converter to achieve subpicosecond jitter performance," *IEEE Journal of Solid-State Circuits,* vol. 43, no. 4, pp. 855-863, 2008.

[51] B. Kim, T. C. Weigandt, and P. R. Gray, "PLL/DLL system noise analysis for low jitter clock synthesizer design," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 1994, pp. 31-34.

[52] Y. C. Bae and G. Y. Wei, "A mixed PLL/DLL architecture for low jitter clock generation," in *International Symposium on Circuits and Systems (ISCAS)*, 2004, pp. IV-788-91 Vol. 4.

[53] M. Sayfullah, "Jitter analysis of mixed PLL-DLL architecture in DRAM environment," in *16th International Conference Mixed Design of Integrated Circuits & Systems (MIXDES)*, 2009, pp. 445-449.

[54] P. O. L. de Peslouan, C. Majek, T. Taris, Y. Deval, D. Belot, and J. B. Begueret, "A new frequency synthesizers stabilization method based on a mixed Phase Locked Loop and Delay Locked Loop architecture," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2011, pp. 482-485.

[55] P. K. Hanumolu, G. Y. Wei, and U. K. Moon, "A wide-tracking range clock and data recovery circuit," *IEEE Journal of Solid-State Circuits,* vol. 43, no. 2, pp. 425-439, 2008.

[56] D. Dalton, K. Chai, E. Evans, M. Ferriss, D. Hitchcox, P. Murray, S. Selvanayagam, P. Shepherd, and L. DeVito, "A 12.5-Mb/s to 2.7-Gb/s continuous-rate CDR with automatic frequency acquisition and data-rate readback," *IEEE Journal of Solid-State Circuits,* vol. 40, no. 12, pp. 2713-2725, 2005.

[57] H. Song, D. S. Kim, D. H. Oh, S. Kim, and D. K. Jeong, "A 1.0–4.0-Gb/s all-digital CDR with 1.0-ps period resolution DCO and adaptive proportional gain control," *IEEE Journal of Solid-State Circuits,* vol. 46, no. 2, pp. 424-434, 2011.

[58] C. C. Chung and W. C. Dai, "A referenceless all-digital fast frequency acquisition full-rate CDR circuit for USB 2.0 in 65nm CMOS technology," in *International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2011, pp. 1-4.

[59] R. J. Yang, K. H. Chao, S. C. Hwu, C. K. Liang, and S. I. Liu, "A 155.52 Mbps-3.125 Gbps continuous-rate clock and data recovery circuit," *IEEE Journal of Solid-State Circuits,* vol. 41, no. 6, pp. 1380-1390, 2006.

[60] B. Razavi, "Challenges in the design high-speed clock and data recovery circuits," *IEEE Communications Magazine,* vol. 40, no. 8, pp. 94-101, 2002.

[61] T. H. Lee and J. F. Bulzacchelli, "A 155-MHz clock recovery delay- and phase-locked loop," *IEEE Journal of Solid-State Circuits,* vol. 27, no. 12, pp. 1736-1746, 1992.

[62] W. Rhee, H. Ainspan, S. Rylov, A. Rylyakov, M. Beakes, D. Friedman, S. Gowda, and M. Soyuer, "A 10-Gb/s CMOS clock and data recovery circuit

using a secondary delay-locked loop," in *IEEE Custom Integrated Circuits Conference*, 2003, pp. 81-84.

[63] M. H. Perrott, Y. Huang, R. T. Baird, B. W. Garlepp, D. Pastorello, E. T. King, Q. Yu, D. B. Kasha, P. Steiner, and L. Zhang, "A 2.5-Gb/s Multi-Rate 0.25-μm CMOS Clock and Data Recovery Circuit Utilizing a Hybrid Analog/Digital Loop Filter and All-Digital Referenceless Frequency Acquisition," *IEEE Journal of Solid-State Circuits,* vol. 41, no. 12, pp. 2930-2944, 2006.

[64] H. Terng-Yin, S. Bai-Jue, and L. Chen-Yi, "An all-digital phase-locked loop (ADPLL)-based clock recovery circuit," *IEEE Journal of Solid-State Circuits,* vol. 34, no. 8, pp. 1063-1073, 1999.

[65] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE Journal of Solid-State Circuits,* vol. 32, no. 7, pp. 1079-1090, 1997.

[66] S. Erb and W. Pribyl, "Design Specification for BER Analysis Methods Using Built-In Jitter Measurements," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 20, no. 10, pp. 1804-1817, 2012.

[67] Tektronix, "Understanding and Characterizing Timing Jitter," Tektronix Application Note 55W-161146-0, 2007.

[68] N. Ou, T. Farahmand, A. Kuo, S. Tabatabaei, and A. Ivanov, "Jitter models for the design and test of Gbps-speed serial interconnects," *IEEE Design & Test of Computers,* vol. 21, no. 4, pp. 302-313, 2004.

[69] M. Azadeh, "Fiber optics engineering," Springer, 2009.

[70] S. Palermo, "ECEN689: Special Topics in High-Speed Links Circuits and Systems (Lecture 10: Jitter)," Texas A&M University, 2012.

[71] M. S. McClure, "Digital jitter measurement and separation," MSc Thesis, Texas Tech University, August, 2005

[72] I. Kuon, R. Tessier, and J. Rose, "Fpga architecture: Survey and challenges," *Foundations and Trends® in Electronic Design Automation,* vol. 2, no. 2, pp. 135-253, 2008.

[73] "Cyclone II Device Handbook," *CII5V1-3.3, Altera Corporation,* 2007.

[74] "Altera DE2-70 Development and Education Board User Manual," *v1.01 edition, Terasic Technologies,* 2007.

[75] W. Y. Tsai, C. T. Chiu, J. M. Wu, S. H. Hsu, and Y. S. Hsu, "A novel MUX-FF circuit for low power and high speed serial link interfaces," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2010, pp. 4305-4308.

[76] R. Stefo, J. Schreiter, J. U. Schlussler, and R. Schuffny, "High resolution ADPLL frequency synthesizer for FPGA-and ASIC-based applications," in *IEEE International Conference on Field-Programmable Technology (FPT)*, 2003, pp. 28-34.

[77] R. E. Best, "Phase-Locked Loop: Design, Simulation, and Applications," Fifth Edition, McGraw-Hill, 2003.

[78] M. Kumm, H. Klingbeil, and P. Zipf, "An FPGA-based linear all-digital phase-locked loop," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 57, no. 9, pp. 2487-2497, 2010.

[79] K. R. Rekha, B. S. Nagabushan, and K. R. Nataraj, "FPGA implementation of low power digital frequency synthesizer," *International Journal of Engineering Science,* vol. 2, no. 10, pp. 5385-5394, 2010.

[80] M. Gude and G. Mueller, "Mixed signal IP: fully digital implemented phase locked loop," in *IP Based SoC Design Conference (IP-SoC)* France, December 2006.

[81] S. Moorthi, D. Meganathan, D. Janarthanan, P. P. Kumar, and J. R. P. Perinbam, "Low jitter all digital phase locked loop based clock generator for high speed system on-chip applications," *International Journal of Electronics,* vol. 96, no. 11, pp. 1183-1189, 2009.

[82] D. H. Wolaver, "Phase-locked loop circuit design," Prentice Hall, 1991.

[83] F. Lin, "Research and design of low jitter, wide locking-range all-digital phase-locked and delay-locked loops," Ph.D. dissertation, University of Idaho, March 2000.

[84] E. Laskin and S. P. Voinigescu, "A 60 mW per Lane, 4 X 23-Gb/s 27-1 PRBS Generator," *IEEE Journal of Solid-State Circuits,* vol. 41, no. 10, pp. 2198-2208, 2006.

[85] W. Z. Chen and G. S. Huang, "Low-Power Programmable Pseudorandom Word Generator and Clock Multiplier Unit for High-Speed SerDes Applications," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 55, no. 6, pp. 1495-1501, 2008.

[86] S. B. Anand and B. Razavi, "A CMOS clock recovery circuit for 2.5-Gb/s NRZ data," *IEEE Journal of Solid-State Circuits,* vol. 36, no. 3, pp. 432-439, 2001.

[87] "ITU-T recommendation O.150."

[88] F. Lin, J. Miller, A. Schoenfeld, M. Ma, and R. J. Baker, "A register-controlled symmetrical DLL for double-data-rate DRAM," *IEEE Journal of Solid-State Circuits,* vol. 34, no. 4, pp. 565-568, 1999.

[89] M. Pedram, Q. Wu, and X. Wu, "A new design of double edge triggered flip-flops," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 1998, pp. 417-421.

[90] X. Gao, B. Nauta, and E. A. M. Klumperink, "Advantages of shift registers over DLLs for flexible low jitter multiphase clock generation," *IEEE Transactions on Circuits and Systems II: Express Briefs,* vol. 55, no. 3, pp. 244-248, 2008.

[91] "PowerPlay Early Power Estimator User Guide For Cyclone II FPGAs," *Altera Corporation,* May 2006.

[92] K. S. Kundert, "Verification of Bit-Error Rate in Bang-Bang Clock and Data Recovery Circuits," *Available Online: http://www.designers-guide.org/analysis/bang-bang.pdf,* May 2010.

[93] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 26, no. 2, pp. 203-215, 2007.

[94] C. K. Liang, R. J. Yang, and S. I. Liu, "An all-digital fast-locking programmable DLL-based clock generator," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 55, no. 1, pp. 361-369, 2008.

[95] G.-Y. Wei, J. T. Stonick, D. Weinlader, J. Sonntag, and S. Searles, "A 500MHz MP/DLL clock generator for a 5Gb/s backplane transceiver in 0.25μm CMOS," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2003, pp. 464-465.

[96] N. Tall, N. Dehaese, S. Bourdel, and B. Bonat, "An all-digital clock and data recovery circuit for low-to-moderate data rate applications," in *IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2011, pp. 37-40.

[97] E. Kilada, M. Dessouky, and A. Elhennawy, "FPGA implementation of a fully digital CDR for plesiochronous clocking systems," in *Internatonal Conference on Microelectronics (ICM)*, 2007, pp. 299-302.

[98] C.-K. Seong, "A 1.25-Gb/s Digitally-Contolled Dual-Loop Clock and Data Recovery Circuit with Enhanced Phase Resolution," MSc Thesis, Yonsei University, 2006.

[99] J. Kim, J. Yang, S. Byun, H. Jun, J. Park, C. S. G. Conroy, and B. Kim, "A four-channel 3.125-Gb/s/ch CMOS serial-link transceiver with a mixed-mode adaptive equalizer," *IEEE Journal of Solid-State Circuits,* vol. 40, no. 2, pp. 462-471, 2005.

[100] V. Balan, J. Caroselli, J. G. Chern, C. Chow, R. Dadi, C. Desai, L. Fang, D. Hsu, P. Joshi, and H. Kimura, "A 4.8–6.4-Gb/s serial link for backplane applications using decision feedback equalization," *IEEE Journal of Solid-State Circuits,* vol. 40, no. 9, pp. 1957-1967, 2005.

[101] C. K. K. Yang, R. Farjad-Rad, and M. A. Horowitz, "A 0.5-μm CMOS 4.0-Gbit/s serial link transceiver with data recovery using oversampling," *IEEE Journal of Solid-State Circuits,* vol. 33, no. 5, pp. 713-722, 1998.