NADEZHDA SHARATUNOVA

EVALUATION OF HIGH-SPEED FPGA IO FOR INTER-BOARD COMMUNICATION

Master of Science thesis

Examiner: Prof. Timo D. Hämäläinen
Examiner and topic approved by the
Faculty Council of the Faculty of
Computing and Electrical Engineering
on 6 May 2015

ABSTRACT

NADEZHDA SHARATUNOVA: Evaluation of high-speed FPGA IO for inter-board communication
Tampere University of Technology
Master of Science thesis, 58 pages
April 2015
Master's Degree Programme in Information Technology
Major: Digital and Computer Electronics
Examiner: Prof. Timo D. Hämäläinen
Keywords: FPGA, SoC, high-speed, transceiver, LVDS, board-to-board

Growing demand for computation power requires high speed interconnects between FPGA devices. While there are multiple solutions available it is still challenging to choose one suited for the particular task. It is therefore extremely import for both academic and industrial purposes to have access to real world performance evaluation of high speed interconnect technologies commonly offered on FPGAs.

In this thesis we study the feasibility of high-speed interconnect and find that it is most relevant to evaluate the performance of LVDS and dedicated transceivers for board-to-board communication scenario. To address this requirement we design evaluation of a system implemented in Altera Cyclone V devices and conduct measurements of the transmission performance and resource usage.

LVDS inter-board communication was implemented as point-to-point topology between two FPGA boards. The maximum received data rate is 823 Mbps per channel. On the base of the transceiver interface, the chain topology was created for communication of three devices. The maximum measured speed in the transceiver system is 1822 Mbps. The average logic utilization of the designs is about 3% of the FPGA resources. At the same time, 38% of the global clocks are used in the transceiver design.

On the base of the performed experiments, we conclude that required high-speed interconnection can be implemented by establishing FPGA-to-FPGA communication via LVDS and the dedicated transceivers interfaces.

PREFACE

This Master of Science Thesis was written in the Department of Pervasive Computing at Tampere University of Technology during the fall of 2014 and the spring of 2015.

I would like to thank my supervisor Prof. Timo D. Hämäläinen for giving me opportunity to work in this thesis. I'm grateful for his incredible guidance, help, encouraging and understanding.

I also thank Nokia, especially Dr. Ari Kulmala, for understanding of my research and supporting it along the way.

Very special thanks to my family and friends which always support me and dedicated to my success. Also I'm thankful to my friend Anastasiia Voropaeva for support in hard moments and helping me to believe in myself. And finally I thank Alexander Pyattaev for assisting me in proof reading of this work and helping me in any case.

Tampere, April 2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

ASIC	Application-Specific Integrated Circuit
BER	Bit Error Ratio
CDR	Clock and Data Recovery
CMU	Clock Multiplier Unit
DC	Direct Current
DDR	Double Data Rate
DSP	Digital Signal Processing
EMI	Electro-Magnetic Interface
Gbps	Gigabits per second
FIFO	Fist-In-First-Out
FPGA	Field Programmable Gate Array
Mbps	Megabits per second
GCLK	Global Clock
GPIO	General Purpose Input Output
HPS	Hard Processing System
HS	High-Speed
HSMC	High Speed Mezzanine Card
HSSI	High-speed Serial Interface
IO	Input Output
IP	Intellectual Property
ISI	Inter Symbol Interference
JTAG	Joint Test Action Group
LVDS	Low-voltage differential signaling
M-LVDS	Multipoint LVDS
Mbps	Megabits per second
MSB	Most Significant Bit
NOC	Network-on-Chip
OCT	On-chip termination
OE	Output Enable
OSI	Open Systems Interconnection
PCIe	PCI express
PCLK	Periphery Clock
PCS	Physical Coding Sublayer
PHY	Physical Layer
PLL	Phase Locked Loop
PMA	Physical Medium Attachment

RCLK	Regional Clock
RTL	Register Transfer Level
SDR	Single Data Rate
SERDES	Serialization and Deserialization
SI	Signal Integrity
SoC	System-on-Chip
USB	Universal Serial Bus
VOD	Differential Output Voltage
VHDL	VHSIC Hardware Description Language

1. INTRODUCTION

1.1 Motivation

In today's multi-processor world, it is commonly known that the hardware capabilities of a single chip no longer keep up with the demand for computation power. Some numerical problems require parallel execution on a scale that just can not be achieved on a single die. As a result, it is common for Digital Signal Processing (DSP), bioinformatics and other specialists try to connect several application-specific chips or boards in order to increase system capacity.

In particular applying Field Programmable Gate Array (FPGA) technology in high-performance computing tasks is increasing. FPGA devices are viable solutions for prototyping complex systems supporting high-speed connection with reliable outcome quality. Reconfigurable properties, low costs and high performance are the key benefits of FPGAs.

Applications from the last decades constantly have been increasing the requirements to the data bandwidth. The data rates of the communication interfaces in digital video or color 3D graphic need higher and higher bandwidth. In 2008 the broadband Internet connection in Tampere region was running in the speeds of 5.37 Megabits per second (Mbps), nowadays this value is around 52.67 Mbps [1]. In present technologies there is a tendency that communication bandwidth rises up to terabits range. A high-speed term in the present chip technology refers to the multi gigabit interconnections the requirement to high-speed communication is to establish more then 1 Gbps. Evolution of technologies moved speeds from Mbps to the gigabits range. From the point of view of hardware designs the concept of high-speed is best illustrated as the order of the bandwidth in the interconnection inside chip between FPGA and Hard Processing System (HPS) parts. In particular, the Cyclone V devices used in this thesis work run HPS-FPGA link at a rate of 100 Gbps [36].

One of the solutions for combination processing resources within one chip is a System-on-Chip (SoC). The integrated components usually are processors, memories and specialized intellectual property (IP) blocks [30]. These elements are linked by interconnection inside one hardware device. To implement this communication various Network-on-chip (NOC) architectures have been proposed. The principle of this model is increasing parallelism and introducing communication-centric design methodologies [31].

An another method to set a connection is off-board linking between multi-FPGA devices. That interconnection can be applied if SoC system is too large and can not be placed in one device. For inter-board communications the strict requirements on link reliability, bandwidth and latencies should be considered. For instance, a typical requirement could be 1.0 Gigabit per second (Gbps) link, with latency of exactly 25 ns.

Current FPGA devices offer a number of available high-speed (HS) interfaces and integrated digital transceivers with very high data rates. The implementation of many communication systems based on the FPGA technology are applied in the portable devices, medical devices, wireless, wire line and military markets [2].

In addition the requirements to the quality of the data transfer is high. Transmission of data should guarantee the known level of error.

1.2 Purpose

In this thesis we develop a system of multi-FPGAs for establishing reliable connection with high data rates and evaluate the performance of the design. Performance parameters of interest are effective bandwidth and latency of the inter-board connection.

The objective of the work is to perform experiments with HS interfaces on the FPGAs, explore methods to set up the system, evaluate received results and consider an effort needed for implementation of such systems. During this work, a detailed research of existing HS FPGAs input output (IO) capabilities was also conducted.

1.3 Overview of conducted research

To implement HS inter-board communication the Low Voltage Differential Signaling (LVDS) and transceiver interfaces were selected from the variety of available options, as explained below.

Today Ethernet interface is the most popular solution in computer networking. It is, however, oriented to transmission of large packets over long distances. Long-distance transmissions make Ethernet transceivers drain more power than is necessary for a very short board to board links. Further, when dealing with short payloads Ethernet struggles to provide consistent latency and good efficiency due to headers and synchronization preambles [26].

Another common interface that is often available in the FPGA devices is PCI express (PCIe). PCIe protocol is typically used for interfacing a CPU with its peripherals. PCIe is especially well suited for star-like topologies [28], but not much else. Generally, PCIe provides a good solution for board to board connectivity when available. However, one could view the dedicated transceivers in Altera devices as being a good approximation of the PCIe connection, since they use almost identical physical layer.

1.4 Overview of implemented designs

To evaluate the performance of selected HS interfaces, custom FPGA designs were implemented. Particular measuring of bandwidths and latency for topologies were completed. The snips of observation of signals and the real speeds computations of high-speed interconnections between few FPGAs are presented. Received experimental results were compared with state-of-the-art systems in the literature. The system implementations were tested using Cyclone V FPGA SoC boards. This device contains 17 available LVDS channels and eight embedded HS transceivers.

Two different experimental set ups are carried out in this present thesis. The first of them is design based on the LVDS communication between 2 boards. The series of experiments were run in different bit rates from 400 Mbps to 840 Mbps. The transceiver designs is implemented in chain topologies between three boards and by one transmission lane. Measurements for transceiver topologies were executed in the available data rate range from 800 Mbps to 1800 Mbps. Different cables were applied in experiments with varying length and cabling quality.

In addition this thesis highlights the problems, which were encountered during implementation of the systems. Clocking distribution is a key design issue to provide proper functionality.

The requirement of synchronizing system timing with accurate precision time latency between communication blocks is solved by a synchronization block. Data alignment is discussed in details as one of the most problematic aspects in the systems based on the high speed serial interfaces in the FPGA. Existing various synchronization methods in hardware circuit should be studied to gain better performance results.

In this thesis the systems based on the LVDS interface and Altera's embedded transceivers are compared in terms of measured performance and in complexity of the implementation effort.

Manuals and data sheets were studied and familiarization with the new tools for both hardware and software were completed. The best results in speed in the transceiver design chain topology with the highest archived speed 1.8 Gbps. All together more than twenty Quartus II designs were created.

1.5 Methodology

The evaluation of the LVDS and transceiver communication was executed in three core directions:

- The quality of the documentation, available literature, as well general applicability of the interface in other hardware (non-Altera and non-FPGA/ASIC).
- The achieved speed and stability of the connection and possible topologies.
- Implementation complexity in terms of source code, logical blocks needed, routing and timing constraints, as well as physical wiring required.

Following the outline of the methodology, the evaluation is split over the next three parts of the thesis, such that each part focuses on its own aspect of the evaluating.

1.6 Outline

Chapter 2 includes background information about basics of HS serial interfaces and gives detail description of LVDS and transceiver architectures. Chapter 3 describe designs of the implemented systems respectively using LVDS and transceivers communication interfaces, and explains the applied requirements and parameters in the software tools. Chapter 4 shows the results of the measurements and analyses of archived values. Chapter 5 summarizes the work and provides suggestions for future research.

2. THEORETICAL BACKGROUND

As previously discussed, high-speed interconnection between systems and chips is unavoidable in present technology. From the start of developing the integrated circuit technology the processing performance of a single chip has been continuously increasing, although the data rate between chips has been improved with the less demand [25].

The traditional method of external linking of boards was parallel transmission, when several data signals are sent simultaneously over parallel separate lines [33]. While the communication volumes grow, the parallel buses met negative effects. An approach of serial data communication was introduced instead of parallel transmission. Ethernet, I2C, PCIe, SPI and Serial RapidIO are among the most popular serial communications protocols.

Operating on the high data rates brings problems associated with transmission line effects [34]. The most common way used to transmit electrical signals is single-ending signaling when one wire carrying the transmitted signal cannot guarantee reliable results[17]. Nowadays differential signaling is commonly applied rather than single ended transmission.

Particularly FPGA devices offer a variety of interfaces operating at very high data rates. To implement designs based on the HS communication, FPGA suppliers provide specialized built-in hardware components, as well as IP development cores to utilize them. There are various devices, supported features and configuration options.

In this chapter the serial communication and differential signaling principles are first explained in the context of FPGA implementation. Further FPGA device connectivity options are provided. In particular we give overview how the FPGA fabric is connected with external interfaces, and how the timing information is distributed inside the chip. Finally, the key design requirements for LVDS and Transceiver communication are given on the example of Altera Cyclone V devices.

2.1 Serial communication

Evolution of parallel buses has been totally explored, e.g. in 1975 microprocessor-based systems used 8-bit buses to drive data. As the needs to transmit more data faster grew, the width of the communication bus was increased to 16 bits, then 32 bits, then 64 bits, and so forth [23]. To support higher data speeds a number of the methods were applied in the communication : increasing of the clock frequencies, widening the interface of the bus, and pipelining transmissions [21].

Expanding the width of the bus enlarges number of pins in the board, but because resources can not be expanded infinitely the interface reaches limits of the wires [19].

The communication with greater numbers of bus-based tracks is affected by signal integrity issues as susceptibility to noise and crosstalk. At the same time, larger connectors were needed. The costs of parallel-bus interfaces were increased and as outcome the price of system implementation came unfeasible and less number of interfaces should used. These critical challenges were met in widening of the parallel bus and greatly influenced the performance of the designs. The requirements for reducing interconnect dimension, increase data rate and minimize energy consumption were needed [32].

Serial communication became a successful solution of these issues. Instead of transmitting in parallel, the stream of serial communication transmits one bit per time in each link. The speeds are kept the same as in parallel transmission. The physical problems are removed as the number of used pins is reduced. Other benefits of serial communication are possibility of data transmission on the long distances, scalability and low cost. Serial HS IO interfaces with improved signal integrity and high transmission speeds is a solution for the present concerns of communication. This has greatly evolved in the last decades.

The most popular IO communication serial protocols are Ethernet (1 Gigabit Ethernet and 10 Gigabit Ethernet), XAUI and PCIe that shifted from the parallel bus based interfaces to the serial interfaces to keep up with ever increasing data rates.

Many serial HS IO interfaces contain serialization and deserialization (SERDES) dedicated logic [13]. SERDES circuity provides serial-to-parallel conversions on incoming data and parallel-to-serial conversion on outgoing data. Transmitter is the device outputting data onto a serial link, and receiver the data receiving the data [40]. Figure 2.1 depicted the basic principle of SERDES architecture.

Non-expensive passive multi-gigabit serial cables on the market are available. For

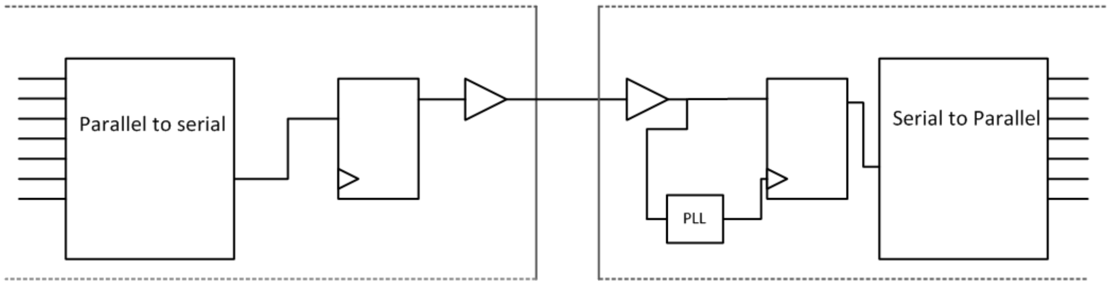


Figure 2.1 *SERDES basic structure.*

the long distances, stating from 20 km, optical cables are widely used, because the low attenuation rate allows for fewer repeaters to be used [27]. As well fiber optic cables are applied to connect server systems [18].

2.2 Differential signaling

The principle of differential signaling is based on transmitting of data via two paired wires. The converting single-ended signal to the differential is described further. Differential driver gate in the transmitting device generates opposite signals (true and complement signals) flowing in the transmission lines. These wires carry signals of the transmitted data. Applying of complementary levels guarantee that signals are less changed by noise from external sources, such as radio interference [22]. As the current flows within the wire pair, thus the current loop area is small and generates lowest amount of electro magnetic interface (EMI). Differential signaling is beneficial in terms of signal integrity and crosstalk to improve the quality of transmission. To get best results, the transmission lines are tightly paired together without thermal issues. Figure 2.2 shows behaviors of the single and differential signaling.

The same amount of noise is applied to the both transmission links. Waveforms cases are generated when logical levels of data are switching from low to high levels and in inverse direction. The picks on the single-ended line can cause a glitch inside the FPGA. While in differential signaling link the spikes of the circuit affect both lines of the pair, the receiver get only the differences between signals and the common mode noise is ignored. That principle allows differential signaling to be more stable and error-free data transmission technology and to operate on the longer distances. The different levels of voltage are supported. There are number of different protocols based on the differential signaling. More detailed information about LVDS standard is provided in the next chapter.

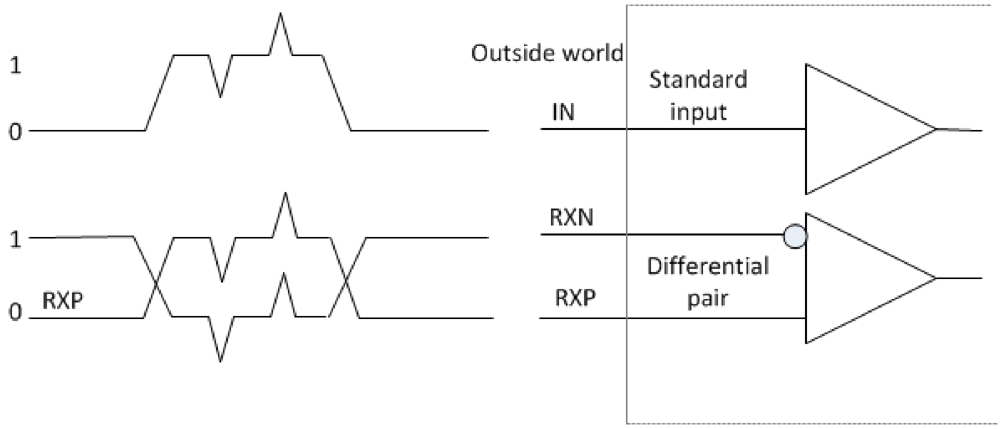


***Figure* 2.2** *Sigle-ending signaling and differential signaling.*

2.3 8b10b encoding

As it was previously mentioned, HS signal tend to be attenuated by board and tracks. That could result in a corrupted signal on the receiver side. The transmitted and received signals on the HS IOs have significant analog characteristics. The sequential bits in transmission stream are grouped to the symbols and could affect downstream symbols and result in the inter symbol interference (ISI) issue.

To reduce this effects in serial data transmission, the signal needs to be Direct Current (DC) balanced. 8b/10b encoding standard is a widely adopted solution. This technique ensures that five identical bits are never sent in row. Each hexes of data are mapped to the 10 bits word on the transmitting side and remapped the word back to 8 bits by the receiver. After transmitting the fifth and eighth bits, two additional bits joins the stream,the values of bits are the opposite and based on the concept of disparity [12]. Figure 2.3 illustrates principle of the 8b/10b encoding flow.

That condition balances number of ones and zeros and referred to DC balance. There are special control characters for synchronization. The 10-bit code groups are generated as valid data code-groups (Dx.y) or special control code-groups (Kx.y) [9].

8b/10b encoding addresses become apparent due the concepts the differential serial communication receivers. This enables applying 8b/10b encoding to HS data links and allows to establish reliable communication, avoid data corruption and transmission errors, support long transmission distances and provide more effective

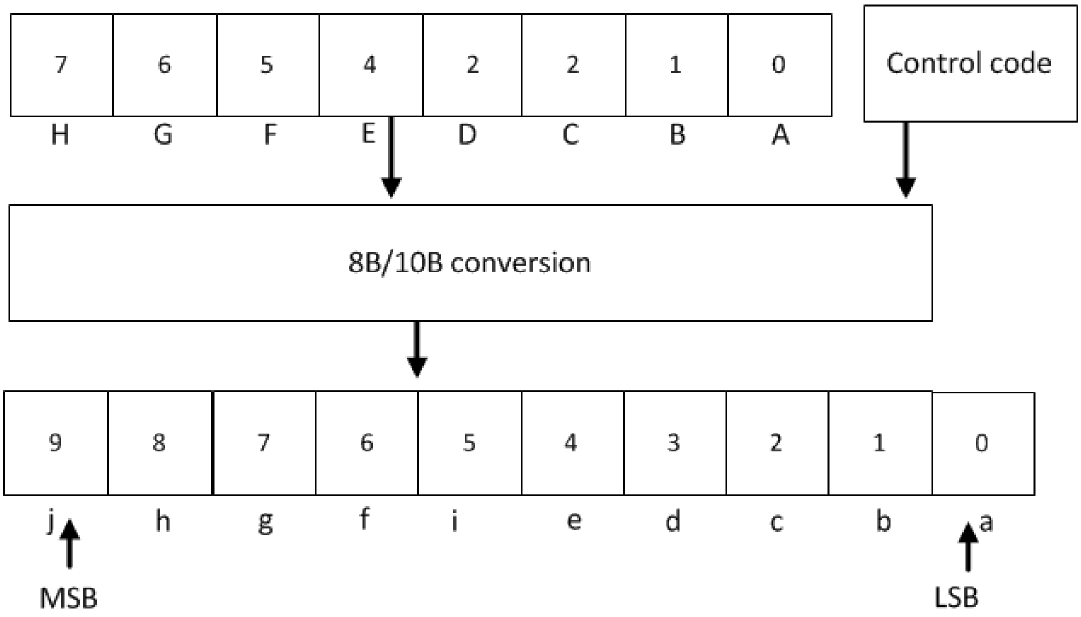


Figure 2.3 *8b/10b encoding.*

error-detection [24].

2.4 FPGA clock networks and hardware specializers distribution principles

To better understand the clocking principles this section describe FPGA clocking distribution. FPGAs are synchronous devices, which are controlled by clock. Designs where only one clock drives all logic or a designs with multiple clocks can be implemented.

A source synchronous interface architecture is a structure where a reference clock is used by both the transmitting and receiving chips, or the receiver gets a clock from the transmitter [29]. Figure 2.4 shows second method of the clock synchronization. SERDES circuity is an example of a source synchronous architecture, this logic enables the FPGA itself to operate at a lower speeds, than the communication data rate requires [11].

2.4.1 Clock domain and clock network

Clock resources in the FPGA device include clock networks. There are global clock (GCLK), regional clock networks (RCLK) and periphery clock networks (PCLK).

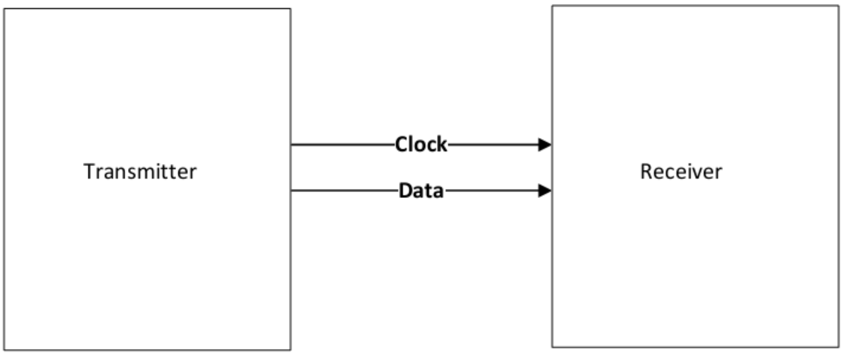


Figure 2.4 *Synchronous interface architecture.*

Specific clock signal driven to the chip forms a clock region.

- GCLK networks can drive throughout the device, signal is driven to this clock forms an entire device clock region;
- RCLK are applicable only to the quadrant they drive into, regional clock region is formed by passing clock source into RCLK. When clock is driven to the two RCLK a dual-region clock region is generated.
- PCLK are used for general purpose routing to drive signals and out of FPGA device.

Clock sources for clock networks are dedicated reference clock pins, Phase Locked Loop (PLL) outputs, HS serial interface (HSSI) outputs and internal logic that can drive the clock networks [9].

GCLK, RCLK and PCLK enable driving of section clock networks (SCLK) , which are clock resources to the core functional blocks, PLL and IO interfaces of the device.

Clock domain means that particular one clock network controls all synchronous elements, such as flip-flop, synchronous RAM blocks, pipelined multipliers [14]. If there are two inputs to the design it means two clock domains exist. Multiple clock domains introduce clock crossing issues, which could cause problems in the system timing. Fist-in-first-out (FIFO) structure is a common way to handle asynchronous clock domains.

A clock skew is the maximum delay from clock input of one register to the clock input of another register. In the multiple clock domain the problems are not easy to founding and it is important to detect and handle interlock interfaces before implementation stage.

2.4.2 What is PLL and why is it necessary

To synthesize clocks of the different frequencies PLL can be used. PLLs provide synchronizing of internal device clocks with an external clock, minimizing of clock delay and adjusting timing issues.

To generate the clocks the PLLs multiply a reference input clock by a ratio. There are two types of PLL :

- fractional PLL synthesizes a clock of any supported frequency
- integer PLL generates an output clock that is an integer multiple or factor of the reference clock.

PLL inside SERDES transceiver synthesizes serial clock - high-speed clock for the serial data and parallel clock - low-speed clock for the parallel data. The tight synchronization between clock and data should be insured in SERDES circuity for reliable communication. For this purposes a lower frequency clock is distributed and multiplied with on-chip PLL [14].

Particularly, Cyclone V PLLs are able to drive out clock outputs through GCLK or RCLK network. The number of IOs standards are supported, including LVDS the standard.

2.4.3 Recovering clock techniques

To support synchronous design, the clocking signals should be transmitted along with the data. The different interfaces apply different recovering approaches. The most popular architectures are parallel clock, Clock and Data Recovery architecture (CDR) and encoding method. First two are described further.

- Parallel clock SERDES recovery principle is based on the fact that transmitter sends data and clock signals to the target receiver [15]. PLL generated slow parallel clock allows to set a correct word boundaries for the received data. Serial clock rate is exact multiple of the clock for the parallel data. Then PLL

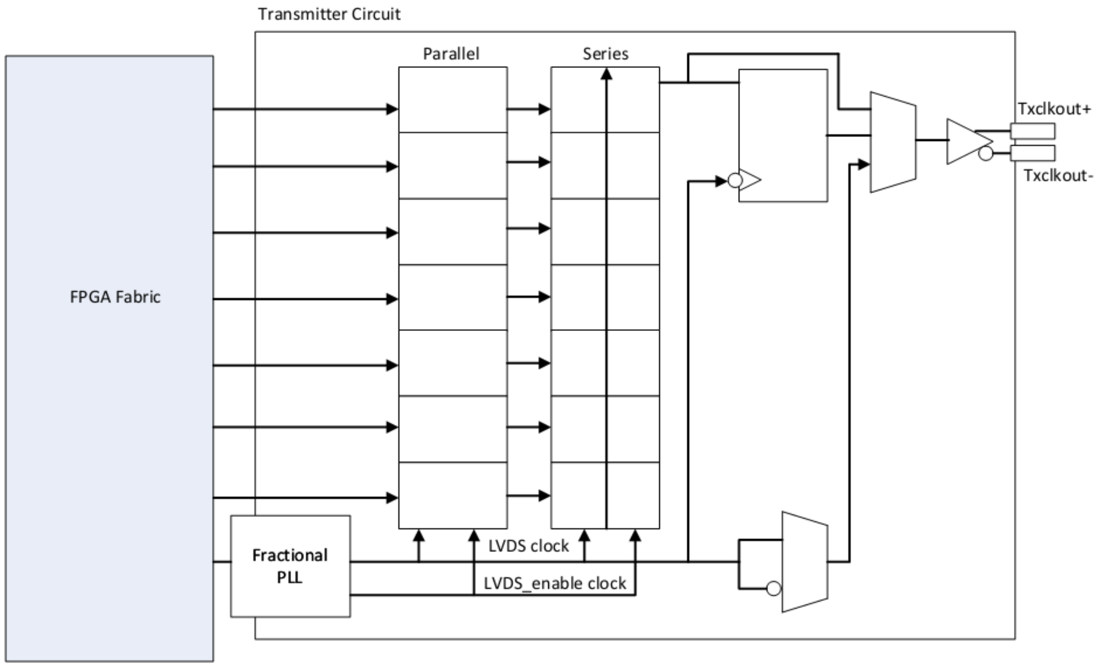


Figure 2.5 *Parallel clock SERDES.*

locks to the rising edge of the reference clock and resets. Thus the deterministic relationship on the default word position in the logic is received. The serial bits of one word are registered on one rising edge on the reference clock and the deserializer continue to set word boundaries in this position.

The physical medium connecting the transmitter and receiver IO peripherals may introduce skew between the serial data and the transmitted source-synchronous clock. This can lead to the errors in the communication. To avoid data corruption, the fractional PLL allows to shift the external clock by different phases to compensate this skew. The phase is chosen depending on the specific device and design. If multiple transmission channels are used the instantaneous skew between each LVDS channel and the clock can be different. The jitter on the data and clock signals as seen by the receiver should be concerned in this case.

This recovery method is applied in the LVDS implementation of Altera LVDS circuity, which is explained in more details in the following. In Figure 2.5 the principle of parallel clock SERDES is shown.

- To ensure data integrity, many HS interfaces include CDR circuit, when the clock information is transmitted along with data without using additional line for the clock signal. In this method, number of used pins for the communication are reduced compared to parallel clock SERDES [14]. Figure 2.6 shows

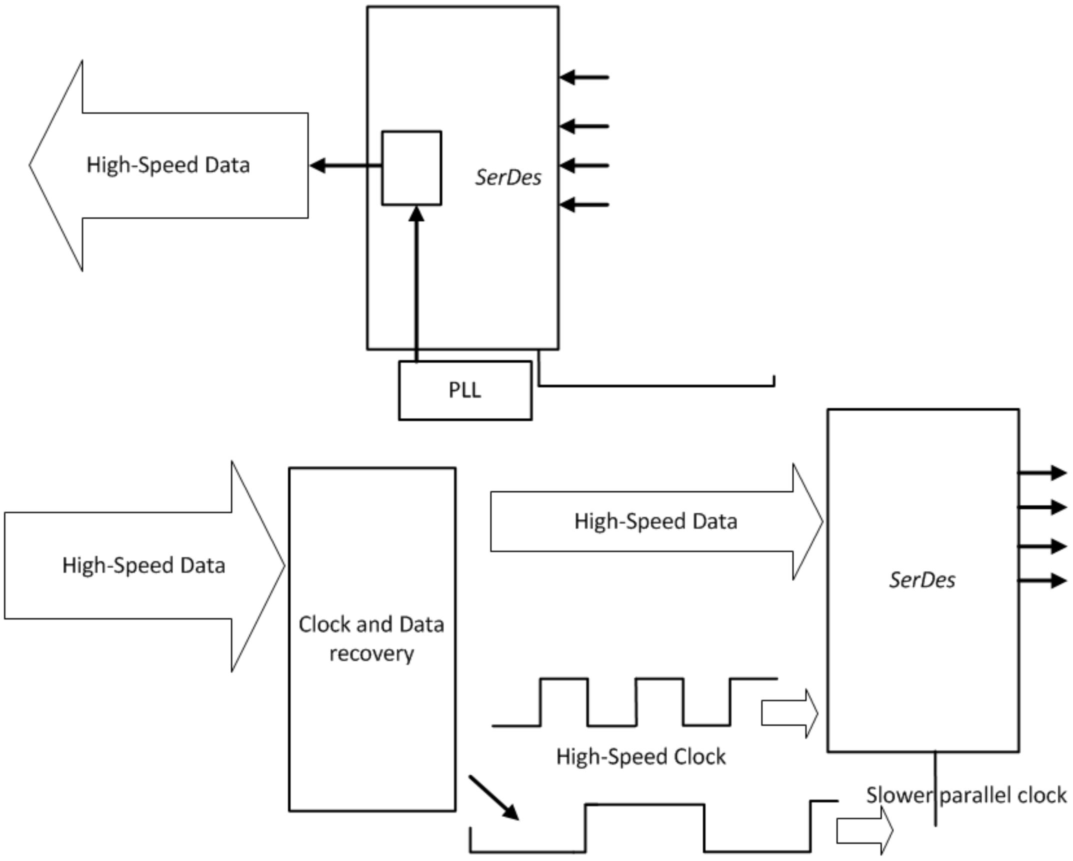


Figure 2.6 *CDR SERDES.*

the principle of SERDES with CDR architecture.

On the receiver side, incoming data from the transmitter recovers the serial clock by CDR approach using PLL. CDR logic generates the bit clock from the stream of the received data and provide re-timing of serial data based on recovered clock. The parallel clock is also recovered in the receiver side. Recovering is processed by dividing the recovered serial clock or parallel clock from the clock divider that is used by the channel's transmitter. For deserialization function of SERDES both clocks are used. Some functionality of the receiver clocking depends on the applied configurations. Cyclone V transceivers use this method of clock recovering.

2.4.4 Word synchronization arrangements

Driving data from transmitter to the receiver can cause the losing of the word boundaries from the serial-to-parallel conversion in the deserializer [20]. Thus the received data should be restored correctly to the initial borders of the parallel data.

The data realignment block is a component created to ensure word synchronization and to compensate a possible skew added on the received data. There are several techniques to implement word aligner logic in HSSI architectures.

- Word alignment with recovered clock is used in Cyclone V LVDS SERDES implementation. In this circuity the received data stream can be realigned by inserting bit latencies into the serial stream. The data slips one bit in the rising edge of the special data alignment port. Timing conditions should be taken in consideration in this case.
- Transceivers apply an extended word aligner logic. First synchronization block runs synchronization stage to receive the predefined alignment pattern for avoiding errors. After this the word boundaries of parallel data from the deserializer could be restored correctly [7]. There are few available word aligner modes, which are listed in the next chapter.

2.5 LVDS standard

LVDS is standardized interface oriented to increase performance, decrease power consumption, and reduce cost [39]. ANSI/TIA/EIA-644 standard and IEEE Std. 1596.3 define physical layer (PHY) of LVDS. Many communication standards and applications use specified PHY LVDS and declare own specific data link layers on top of it, to provide completed OSI model.

LVDS typical applications are high-speed video, graphics, flat panel displays, general purpose computer buses, as well communication applications such as hubs, set-top boxes. Most of the recent FPGAS include the logic for managing LVDS [11].

The ANSI/TIA/EIA-644 standard recommends a maximum data rate of 655 Mbps [39], which is based on a limiting set of assumptions. A theoretical maximum of 1.923 Gbps is based on a loss-less medium. The most typical operating bandwidth of LVDS is 500 Mbps per differential pair. The structure of a LVDS communication is depicted on Figure 2.7.

The LVDS consists of a current source with nominal value 3.5 mA, which drives differential pair line. Depending on the value of currently transmitted bit of data, the driver switches the polarity of the output voltage. Next, input current flows across terminating resistor of 100 Ohm, which results in the input of the receiver a nominal voltage drop of about 350 mV between receiver and receiver input. The receiver detects polarity of differential signals and amplifies them into standard logic

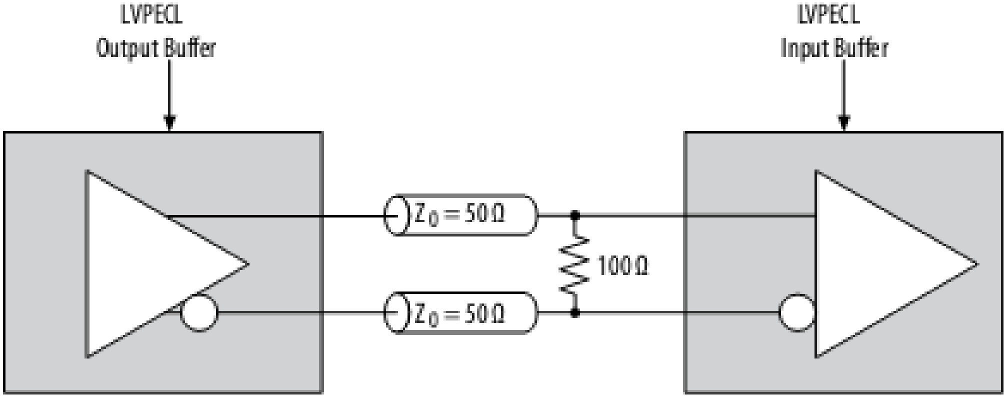


Figure 2.7 *LVDS basic structure [9].*

levels. A high-impedance of the receiver allows operate with low voltage values as 20 mV [39]. A typical driver common-mode voltage is 1.2 V, although this value depends on the vendor. Receiver accepts an input range from ground to 2.4 V. The resulting common-mode voltage difference between the driver and receiver does not influence up to +-1 V. The standard levels for LVDS were designed around a 3.3 V supply systems, however power supply range is not strictly fixed and allow to apply other power values like 2.5 V and 5 V. Described LVDS architecture is point-to-point communication topology with only one transmitter and one receiver. Other LVDS topologies also available with multiple drivers and receivers. Multipoint LVDS (M-LVDS) is standardized in ANSI/TIA/EIA-899 is one of the popular differential signaling technologies apart from LVDS.

As well from LVDS standard reduced swing differential signaling (RSDS) and mini-LVDS standards were invented. These specifications have reduced EMI, power consumption and mainly are applied in flat panel displays communication [16].

2.5.1 LVDS dedicated circuity

This thesis uses Altera Cyclone V devices in the experimental part, thus a dedicated circuity for LVDS implementation in Altera technology is described.

The circuity for LVDS transmitter and receiver consists of true differential buffers, SERDES, fractional PLLs, programmable-emphasis, data realignment block (bit-slip) and on-chip termination (OCT) [4]. LVDS SERDES circuity is presented in Figure 2.8.

The dedicated clock to the LVDS transmitter should be located on the same IO block of the device. Reference clock is driven to fractional PLL, which generates

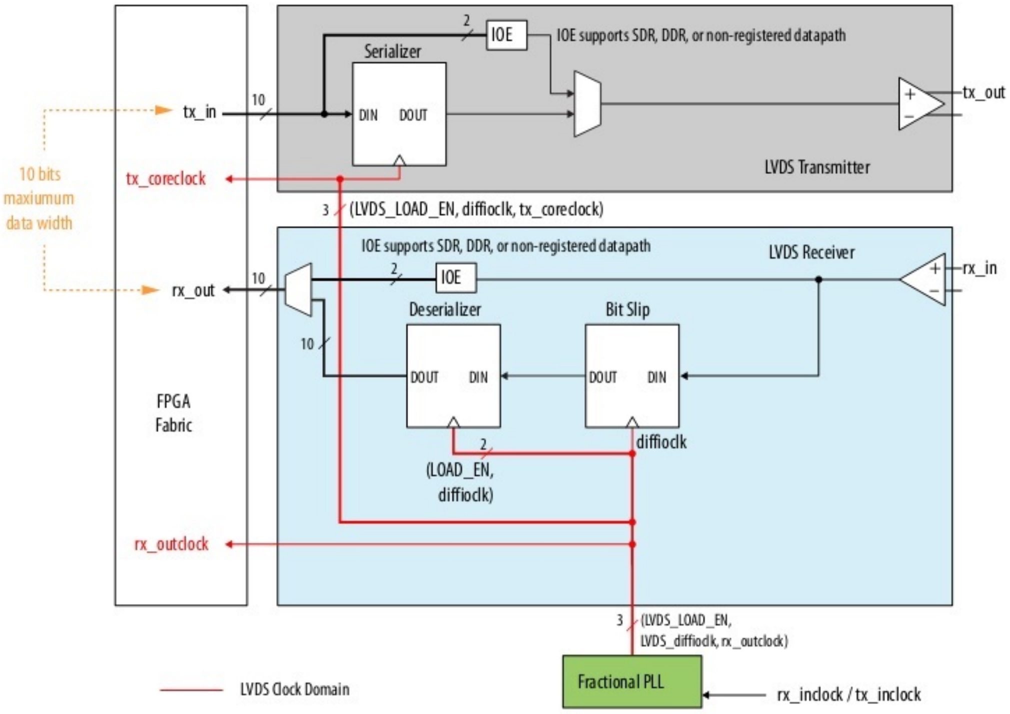


Figure 2.8 *LVDS SERDES dedicated circuity [9].*

clocks of the different speeds for the LVDS circuity. The PLL can be implemented inside the transmitter and receiver blocks, as well external PLL can be used.

The differential transmitter and receiver buffers can drive out and get LVDS, mini-LVDS, and RSDS signals, programmable differential output voltage (VOD) and programmable pre-emphasis. Serialization and deserialization blocks contain parallel load and shift registers.

If the serialization factor is one, system runs on the single data rate (SDR), if two - double data rate (DDR) is applied, in this configurations data is bypassed, instead of flowing to SERDES circuity[9]. IO elements support this mode. Input path of IO registers handles driving a data to the core, the output path is for moving data from the core to the pin. The output enable (OE) path directs the OE signal to the output buffer. The benefit of using that registers is faster source-synchronous register-to-register transfers and resynchronization.

- Transmitter

  The serialization factor in transmitter can be chosen from 1 to 10. The word is clocked into the load registers and serialized by shift registers. These registers are driven by the serial fast clock. After this the serial data is sent to the differential buffer to be transmitted. The transmission starts from the most significant bit (MSB) of the parallel data. Parallel clock signal is sent to the receiver as well.

- Receiver

  The differential receiver contains hardware blocks of data realignment block and deserializer. Deserializator provides opposite functionality of the serializer. Input bits are registered by deserializer in the rising edge of the serial clock.

2.5.2 LVDS benefits and disadvantages

To summarize, LVDS technology uses HS analog circuits to provide interconnection with coper mediums. It is unidirectional serial communication.

- Advantages

  High data rates with low power consumption are the main benefits of the LVDS. Noise performance is improved compared to signal-ended technologies. The voltage across the terminating resistors is lowered. Slower transition reduce the problem of reflection from transmission-path impedance [38].

  Most of FPGA devices have a number of pins supporting LVDS signaling. Thus LVDS is one of the simplest approach to connect the General purpose IO (GPIO) pins on the FPGA device.

  In the market, point-to point applications are widely applying LVDS for data transmission - twisted-pair copper cables are inexpensive and widely available.

- Disadvantages

  The main disadvantages of LVDS communication are skin effect, dielectric losses and reflections.

  Long parallel links are affected by signal integrity(a cable geometry for a good quality signal) and skew.

  The maximum operating distance for the LVDS communication line is 10 m, but only when low loss cables are used. Thus cables for LVDS interfacing are usually short (centimeters) and must be employ careful construction. This limits the size of cluster that can be built. Available frequency is usually limited.

  LVDS is a suitable solution for the data transfer between boards, modules and box-to-box. Stackable hubs for data communication, wireless base stations, ATM switches in telecommunication and flat-panel displays in automotive market are some examples [10].

In comparison with other communication protocols, LVDS link reduce noise and EMI significantly, have many configurations and offer low-cost implementation. However when data transfers are with strict requirements, additional overheads in implementation occur.

2.6 High-speed transceivers embedded in FPGA

Nowadays multi-channel Gbps transceiver technology is available on FPGA devices. Transceiver block is a combination of a transmitter and a receiver which are able to transmit digital data at very high bandwidths. Flexible architecture allows implementing a variety of protocols on the transceiver base.Typical examples of interfaces using off-board transceiver communication are PCIe, HDMI and Ethernet. The performance of transceivers tends to fall into three ranges, up to 3.125 Gbps, up to 6.5 Gbps and 10 Gbps+ per channel. The communication lanes use differential signaling to accurately send high speed data across a wire, with low power consumption. Reliable communication, low jitter and low Bit Error Ratio (BER) are provided by transceiver technology.

Transceivers on FPGA board communicate over different physical medium. SATA, PCIe, SMA, SFP+ copper and SFP+ optical cabling are popular solutions.

SERDES components are embedded inside the transceiver block. Parallel data from FPGA fabric is sent over layers and transmitted as serial data to the receiver part, it is simplify board layout and design. A number of encoding and electrical criterion should be considerate using transceivers. As well 8B10B encoding, data alignment, channel bonding (up to 8x) and programmable pre-emphasis and equalization are provided in transceiver circuity [9].

In particular we focus on an Cyclone V SX with embedded full duplex serial transceivers with a data rate 3.25 Gbps.

2.6.1 Transceiver clocking

Cyclone V transceivers support non-bonded and bonded transceiver clocking configurations:

- Non-bonded configuration : only the serial clock from the transmit PLL is routed to the transmitter channel. The parallel clocks are locals for each channel, where it is generated by clock divider.

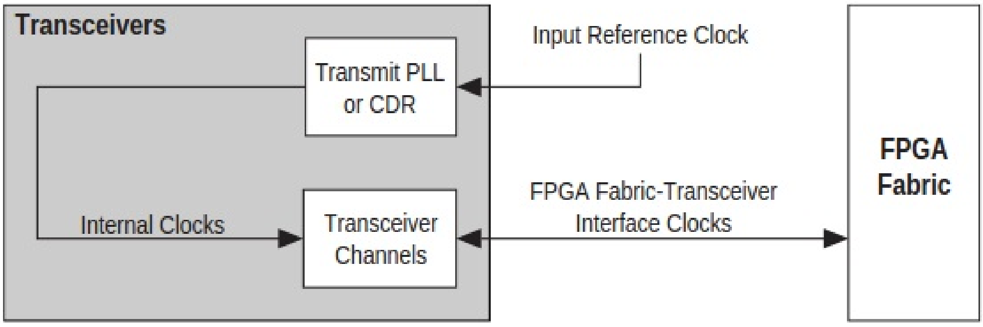


***Figure* 2.9** *Clocking of transceivers [9].*

- Bonded configuration : the serial and parallel clocks are routed from the central clock divider. Depending on the channel location in a transceiver bank clock divider is defined as local and central. The central clock divider can additionally feed the clock lines used to bond channels compared to the local clock divider.

Transceiver channel includes transmitter and receiver paths. Figure 2.9 describes briefly clocking of transceiver architecture provided by Altera.

For each transceiver bank there is a one dedicated reference clock pin. Direct connection from reference clock pin to transceiver channel is available only for one channel in the transceiver bank. Receiver differential pair can be used as an additional input reference clock source when it is not used as the receiver.

In Cyclone V SX devices, there are two transmitter PLL sources: clock multiplier unit (CMU) PLL (or channel PLL) and fPLL.

- Channel PLL in CDR configuration is recovers the clock and serial data stream, each channel PLL independently recovers the clock from the incoming serial data.
- CMU mode is applied then the transceiver channel operates only as transmitter.

The clock networks in the FPGA core are used by clock signals from the FPGA fabric to the transceiver blocks and clock signals from the transceiver blocks to the FPGA fabric. This interface clocks are able to transfer data, control, and status signals between the FPGA fabric and the transceiver channels.

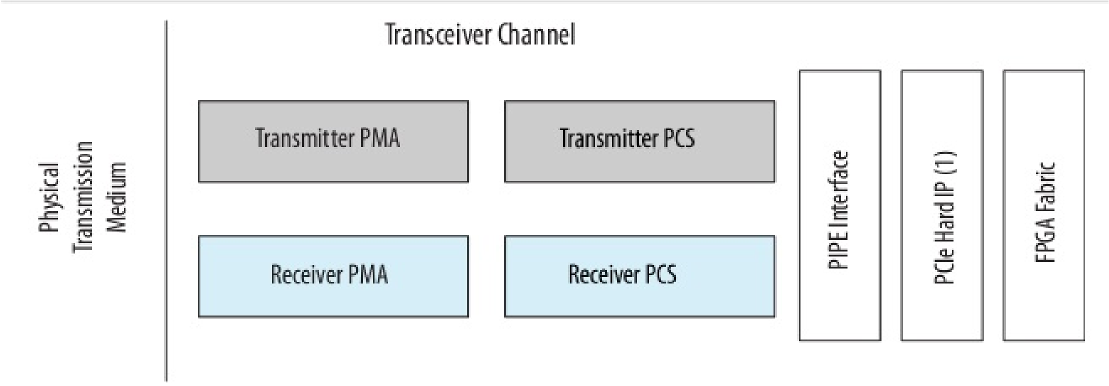


Figure 2.10 *Transceiver channel architecture [9].*

Also transceiver channel forwards the recovered receiver clock (in configurations without the rate matcher) or the transceiver parallel clock(in configurations with the rate matcher) to the FPGA fabric to clock the data and status signals from the receiver into the FPGA fabric. Input reference clocks is forwarded to the FPGA fabric, where it can then clock the data and control signals into the transmitter [9].

2.6.2 Architecture overview transceiver

Architecture of transceiver is following :

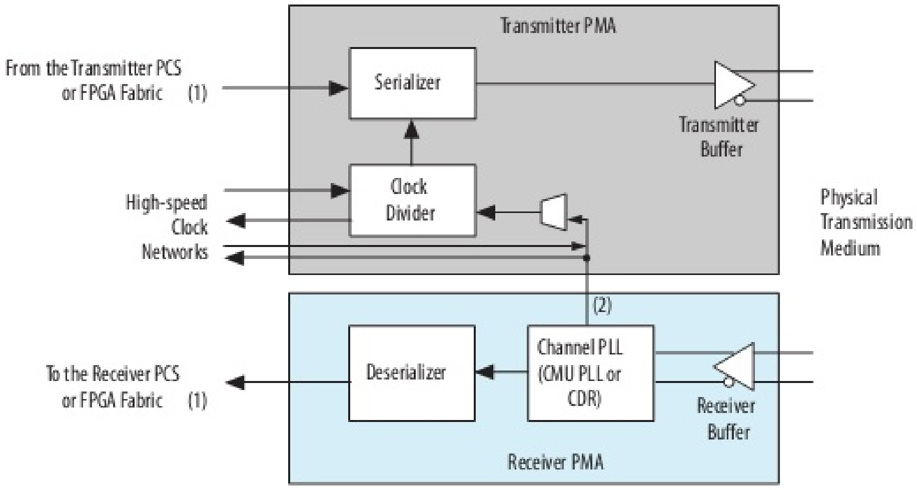
- Physical Media Attachment (PMA) - convert digital data to analog stream and opposite, provide connecting FPGA to physical medium.
- Physical coding sublayer (PCS) - transmitting or receiving data to or from PMA layer and responsible that data is transmitted into appropriate formats
- FPGA logic interface

The Cyclone V device allows variety reconfigurations of PHY components. The Figure 2.10 shows the block diagram of the transceiver architecture.

PMA

PMA consists of the transmitter and receiver paths. Figure 2.11 depicts main components of PMA.

Data stream flow from transmitter and came to receiver by transmitter or receiver buffers with differential OCT. OCT in the PMA requires the calibration block to compensate for process, voltage, and temperature variations [9].


Transmitter PMA
From the Transmitter PCS or FPGA Fabric (1)
Serializer
Transmitter Buffer
High-speed Clock Networks
Clock Divider
Physical Transmission Medium
(2)
To the Receiver PCS or FPGA Fabric (1)
Deserializer
Channel PLL (CMU PLL or CDR)
Receiver Buffer
Receiver PMA

Notes:
1. The channel PLL provides the serial clock when configured as a CMU PLL.
2. The channel PLL recovers the clock and serial data stream when configured as a CDR.

Figure 2.11 *Transceiver PMA [9].*

- Transmitter contains serializer and clock divider. The transmitter serializer support polarity inversion and bit reversal capabilities.
- The receiver includes deserializer and channel PLL.

  The deserializer clocks in serial input data from the receiver buffer by high-speed serial recovered clock. Next data is deserialized by low-speed parallel recovered clock. The deserializer forwards the deserialized data to the receiver PCS. The receiver deserializer support clock-slip.

PCS

PCS transmits or receives data stream to or from the PMA and responsible that data is grouped into appropriate code group.

Transceiver and Receiver phase compensation FIFO interfaces compensate the phase difference between the low-speed parallel clock and the FPGA fabric interface clock when interfacing transmitter and receiver PCS with FPGA fabric.

PCS datapath can be configurations in single-width and double-width modes. In single-width mode PMA-PCS width is 8, then width between FPGA fabric and transceiver PCS are 8 or 16 bits. In double-width mode PMA-PCS width is 16, then width between FPGA fabric and transceiver PCS are 16 or 32 bits.

Each block is reconfigurable to support in single- and double-width modes for device different protocols.

- Transmitter PCS Datapath consists of : Transmitter Phase Compensation FIFO, Byte Serializer, 8B/10B Encoder, and Transmitter Bit-Slip.

  Byte serializer allows the transceiver channel to operate at higher data rates while keeping the FPGA fabric interface frequency within the maximum limit. The datapath clock rate at the output of the byte serializer is twice the FPGA fabric transmitter interface clock frequency.

  To avoid the channel-to-channel skew between multiple transmitter channels transmitter bit-slip allow slipping the data sent to the PMA.

- Receiver PCS Datapath

  Receiver PCS contains word aligner, rate match FIFO, byte deserializer, byte ordering, and receiver phase compensation. Rate match FIFO compensates for the possible clock frequency differences between transmitter and the local receiver clocks.

  Data flows to the 8B/10B decoder after the rate match FIFO, if rate match FIFO dis-enabled the 8B/10B decoder receives data from word aligner. 10-bit data is decoded into an 8-bit data and 1-bit control identifier, indicates if the decoded 8-bit code is a valid data or special control code.

  Byte deserializer is able to reduce the FPGA fabric-transceiver interface frequency to half while doubling the parallel data width.

  When byte deserializer enabled byte ordering can be used. It restores the proper byte order of the byte-deserialized data before forwarding it to the FPGA fabric.

Block diagram in Figure 2.12shows transceiver channel in Native PHY IP Core.

2.6.3 Transceiver benefits and disadvantages

Using Transceiver blocks provide a number of features. As well some restriction and negative limitations are exist.

Benefits.

- Possibility of implementation of extremely complicated interfaces, as PCIe is supported.

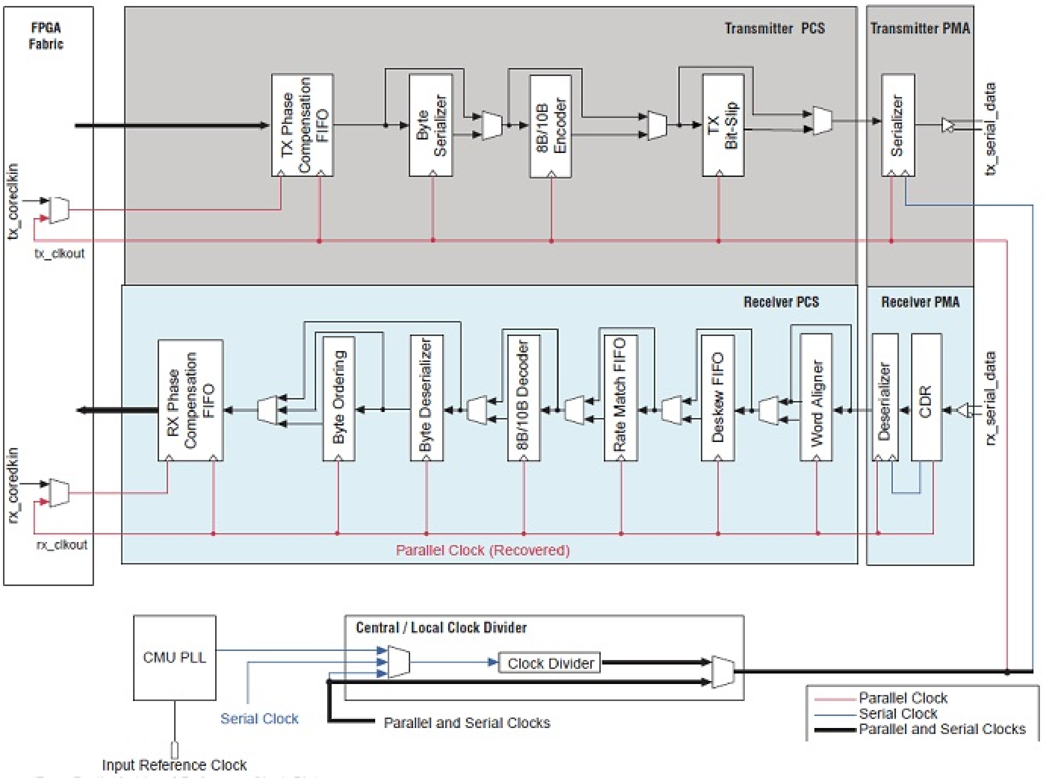


***Figure* 2.12** *Transceiver architecture [9].*

- Dedicated silicon uses lower power consumption, compare to implementation of the same logic blocks within the FPGA.
- Encoding mechanisms applied in transceiver save FPGA resources and can run at a guaranteed high speed.
- Electrical interfaces are handled with minimum external parts.

Disadvantages.

- Limited to a specific external interfaces, while FPGAs are flexible enough and there are a number of high speed interfaces that are becoming common.
- The costs of FPGAs with powerful transceivers are increased.

2.7 Board specification

2.7.1 Cyclone V SX overview

The testing and verification of developed design is done using SoCKit Development Kit from Terasic. This Altera SoC is based on Cyclone V 5CSXFC6D6F31 FPGA

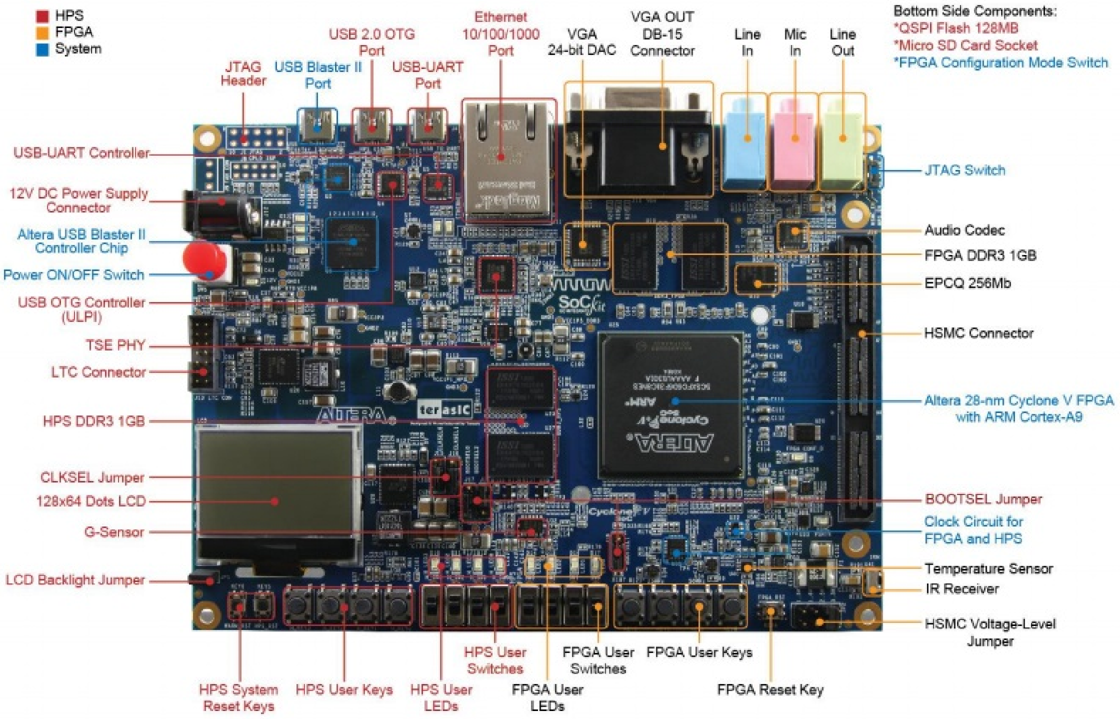


Figure* *2.13 *SoCKit Development Kit [36].*

fabric 28-nm technology. FPGA is combined with dual-core ARM Cortex-A9 HPS. It is low-power low-cost board, which offers wide range of hardware for flexible designs [36]. Figure 2.13 shows the layout of the board, the main parts are pointed in the picture.

HPS and the FPGA can operate independently and are tightly coupled via a high-bandwidth system interconnect built from high-performance ARM AMBA AXI bus bridges.

SoCKit Development Kit supports number of high-speed communication interfaces like Ethernet, GbE, PCIexpress and GPIOs. The detailed specification of those interfaces can be found in [9].

2.7.2 IO resources

Cyclone devices offer four I/O banks around the periphery. Altera Cyclone V devices support differential signaling as LVDS, RSDS, MINI-LVDS, HSUL and SSTL I/O. Particularly groups of pins in each of the four I/O banks (on both rows and columns) support the LVDS interface. For more details [4] lists the IO interfaces pins.

Cyclone V contains 8 transceiver channels with operating speed in the range 614 Mbps and 3.124 Gbps. Tree transceiver banks are available, each bank is comprised of three channels.

Cyclone V device has access to low-level hardware of transceivers. Implemented are by PHY in the Open Systems Interconnection (OSI) model.

On-board High Speed Mezzanine Card (HSMC) connector has programmable bi-directional pins, which can be used to communicate by differential signaling IO standards [8]. Total number of LVDS resources are 19 full-duplex channels for data and clock transmissions. Transceiver class pins also located on on-board HSMC connector.

IO resources of Cyclone V 5CSXFC6D6F31 :

- Total number of FPGA I/Os is 288 pins.

  HSMC contains 107 pins. Up to 71 GPIO and 14 input-only pins, with digital de-bounce and configurable interrupt mode.

- The number of available LVDS resources : receiver - 72, with 875 Mbps data rate, transmitter - 72, with 840 Mbps.

- I/O standards : single-ended, non-voltage-referenced and voltage-referenced, LVDS, RSDS, mini-LVDS, HSTL, HSUL, and SSTL I/O.

- Supporting receiver and transmitter OCT.

2.8 Development area

For implementation the system software Quartus II version 13.1 was used. This tool is one of the most popular solutions for SoC systems. It is powerful software providing all stages of the design and programming the device. Megawizard is a tool in Quartus II that allows to access hard core IPs implemented by manufacturing vendors. Thus, to design LVDS and Transceiver communication Megafunctions are used. MegaWizard GUI is a fast and easy tool to create and configure IP cores. There are number of advantages and as well some disadvantages in the present implementations of the MegaWizard tool.

- IPs may be applied only to specific defined FPGA families and contain limited configurations. Some IPs are in ongoing development, or there are not updated versions for new devices. This may require additional effort to integrate IP in the design or prohibit using a newer FPGA.

- Requirements to the device fitting could be strict as particular clock frequencies, PLLs or clock routing.
- Supporting of some protocols is limited. For example SerialLite and Aurora are only supported by one FPGA manufacturer.
- Licensing is required. Accessing IPs should be licensed from vendors, that condition could be high cost or can results in difficult evaluation. As a simulation of a link does not capture physical effects and so a license may be required for evaluation on a physical FPGA.
- The configuration constraints are flexible , user could apply needed parameters.
- Bonded links feature is useful on a custom PCB with skew-free parallel lanes between FPGAs. Bonding can reduce the dimensions of the cluster to compare with single links, adding hops and latency.

SiganlTapII is an embedded analyzer which is used for the tracing of signals in the design. We observe data flowing through HS links by SignalTapII. More information about the principle of this tool is provided in Chapter 4.

3. IMPLEMENTATION

This chapter describes the implementation of inter-board communication designs between multiple FPGAs through LVDS and transceiver interfaces. For each of designs, the hardware settings are first explained. Then MegaFunction IP core blocks are introduced and configurations of IPs are listed. Next we describe custom logic blocks used for transmitting and receiving of data and for measuring of the bandwidth. The timing constrains and pin mapping considerations are explained.

3.1 LVDS communication design

3.1.1 LVDS point-to-point topology

LVDS inter-board design connected to the host PC consists of two SoC Kit boards connected between each other by ribbon wires through HSMC interface.

HSMC connector is prearranged to support HSSI as well single-ended applications. For improving signaling the host board FPGA device and HSMC connector are intended to be DC coupled. To get access to the LVDS pins located on the HSMC connector of the board there are interface-compatible GPIO-HSTC adapters attached. That allows to extend the peripherial-set of the FPGA board.

GPIO-HSTC daughtercard consists of three expansion Prototype Connectors: J1, J2 and J3. Connectors J2 and J3 includes 8 LVDS channels, altogether 16 channels for data and clock transferring.

Choosing connectors and pins for communication should be carried out carefully of the boards' linking. More information can be found in [35]. The design shown in this thesis applies in LVDS communication for one data channel and one channel for clock on the J3 connector.

Implemented system includes Master Board, which starts the transmission and sends signals to Slave board. This board receives transmitted signals, as well as the clock. Assembled system is depicted in Figure 3.1.

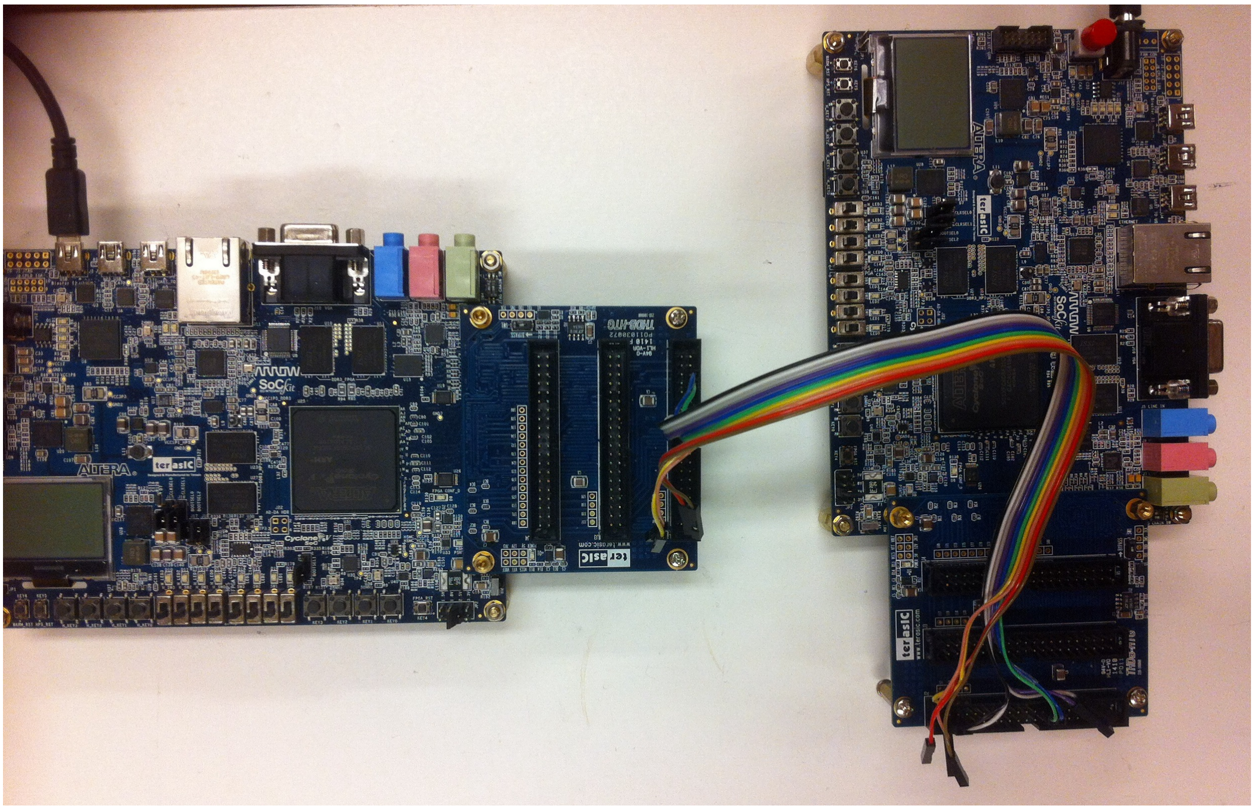


***Figure 3.1** LVDS communication design.*

The ribbon wires are 16 cm in length. High-speed operating bandwidth requires that physical the medium characteristics would be in precise attention as its properties affects stability of the whole system. Thus differential pairs of the channels should be closely coupled to gain EMI advantages of the differential signals. Figure 3.2 shows connection of wires to the HSMC adapter.

3.1.2 Structure of the system

The LVDS communication design provides the following functionality :

- Transmitting and receiving differential LVDS data is controlled by ALTLVDSTX and ALTLVDSRX SERDES from Altera.
- Custom logic generates data in the Master board and directs it to the transmitter SERDES IP block, which sends data and clock to the transmitting pins.
- Receiver on the Slave board gets data and clock signals. A custom logic in the receiver side counts number of packets during the certain time.

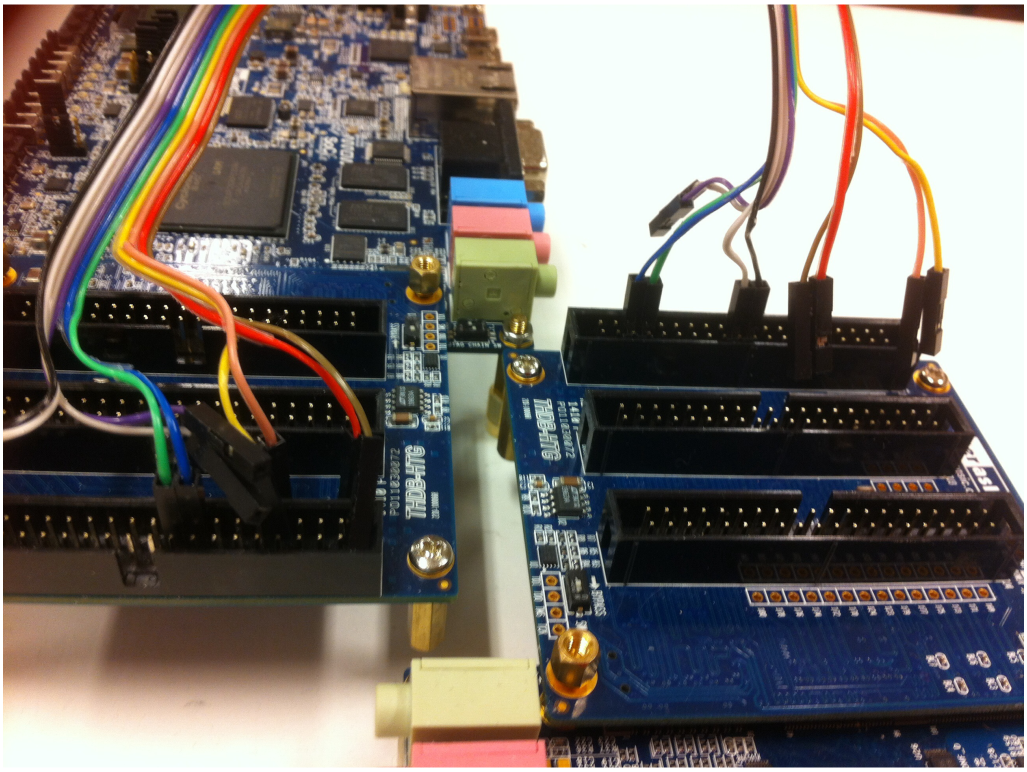


***Figure* 3.2** *HSMC daughtercard.*

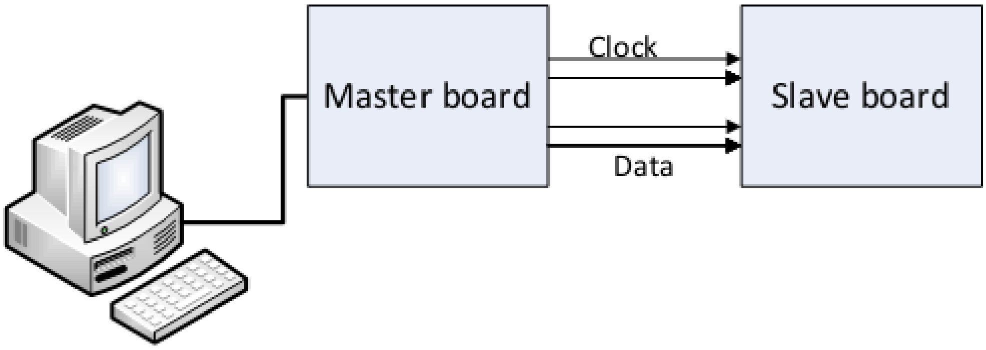


***Figure* 3.3** *Simplified LVDS design*

Figure 3.3 shows the design with Master and Slave devices.

3.1.3 ALTLVDS IP core configurations

Reconfiguration parameters of ALTLVDS MegaFunction IPs can be set in MegaWizard PlugIn Manager and depend on the implemented design. This IP core described in [4] Altera documentation. The available features of IPs varies by programmed device. For example, the clock data recovery(CDR) is not supported in Cyclone series chips, however in Arria and Stratix Altera devices CDR can be applied.

As example, further are listed main configurations of ALTLVDSTX and ALTLVDSRX IPs of one of the executed designs. Transceiver and receiver SERDES instances apply several common parameters, when running in one system.

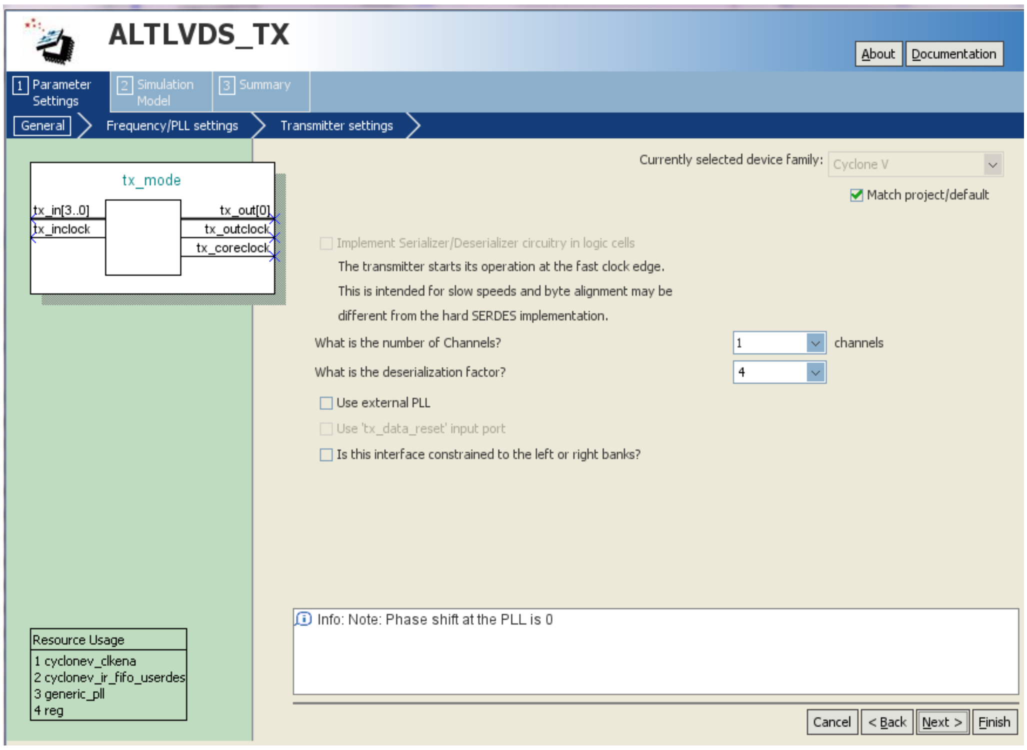
- Number of channels : 1.
- Serialization factor (available from 1 to 10, except 3): 4.
- Data rate: 500 MBps.
- Frequency : 125 MHz.
- External PLL : off.
- Shared PLL : off.

To optimize used resources the compiler can perform merging of PLLs in the compilation stage. Shared PLL(s) function can be enabled in Receiver and Transmitter blocks to share a PLL. This can be applied if the transmitter and receiver are driven by identical input clock sources and have identical settings.

Figure 3.4shows the example of configuration of ALTLVDSTX Mega function.

The ALTLVDSTX specific settings :

- The phase alignment of the data transmitted by the core logic array with respect to the reference clock.
- Using the parallel clock from transmitter signal to register the data input before it feeds the SERDES. This provides optimal phase position for registering the data with respect to the high-speed clock that drives the SERDES.
- Out-clock divide factor (B) to specify the frequency of the parallel output clock signal as the transmitter output data rate divided by divide factor. To provide the same transmitted serial closk as coreclock B factor is set to 4.



Currently selected device family: Cyclone V

☑ Match project/default

□ Implement Serializer/Deserializer circuitry in logic cells

The transmitter starts its operation at the fast clock edge.
This is intended for slow speeds and byte alignment may be
different from the hard SERDES implementation.

What is the number of Channels? 1 channels

What is the deserialization factor? 4

□ Use external PLL

□ Use 'tx_data_reset' input port

□ Is this interface constrained to the left or right banks?

Info: Note: Phase shift at the PLL is 0

Resource Usage
1 cyclonev_clkena
2 cyclonev_ir_fifo_userdes
3 generic_pll
4 reg

Figure 3.4 ALTLVDS Megafunction interface.

- Transmitter core clock signal is enabled to the registers of all the logic that feeds the LVDS transmitter logic.

The ALTLVDSRX specific settings :

- The phase alignment of the received data with respect to receiver clock is based on the number of resets to reach the synchronization. The values for this option are device dependent.

- Register outputs : the outputs of the receiver are registered by the receiver slow signal.

- Number of pulses, when the circuitry restores the serial data latency to 0. This value should be equal to the deserialization factor or larger. The maximum is 11 bit-times of insertion before a rollover occurs.

- To synchronize channel-by-channel port alignment implements control of the the word boundaries of the incoming data. The data slips one bit for every pulse on the align port. In the present design we do not use this feature, because we set deterministic relationship on the default word position in the SERDES during PLL resetting. The reference clock is equal to the data rate

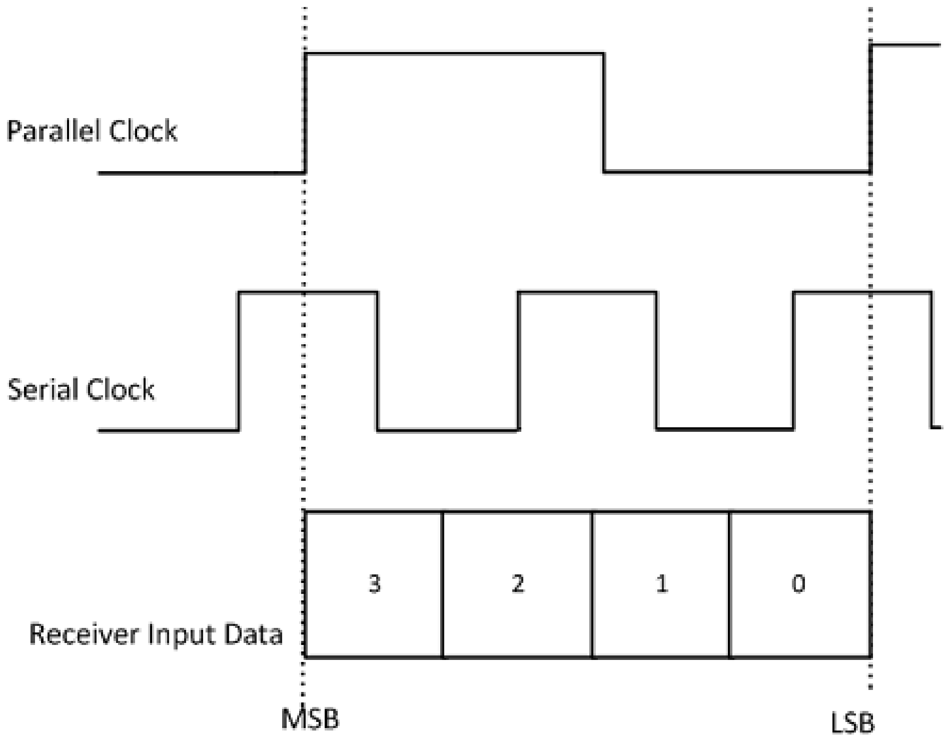


Figure 3.5 LVDS RX clocking.

divided by the deserialization factor. In this configuration PLL locks to the rising edge of the reference clock. As the serial word is registered on one rising edge on the reference clock, the deserializer continue to set word boundaries in this position. Figure 3.5 illustrates clocking of the receiver based on this configurations.

3.1.4 Timing constraints

To meet timing requirments the Quartus II software automatically places the SERDES logic at the best location. This feature provides placement constraints on the ALTLVDS IP core logic. The LVDS transmitter and receiver functions with the ALTLVDS IP core are characterized and guaranteed to function correctly within the LVDS system specification.

3.1.5 Send/receive data logic

Master board is programmed with the design, which contains a simple data generator implemented in VHDL IP block. This logic produces digits from 0 to 15 and drives

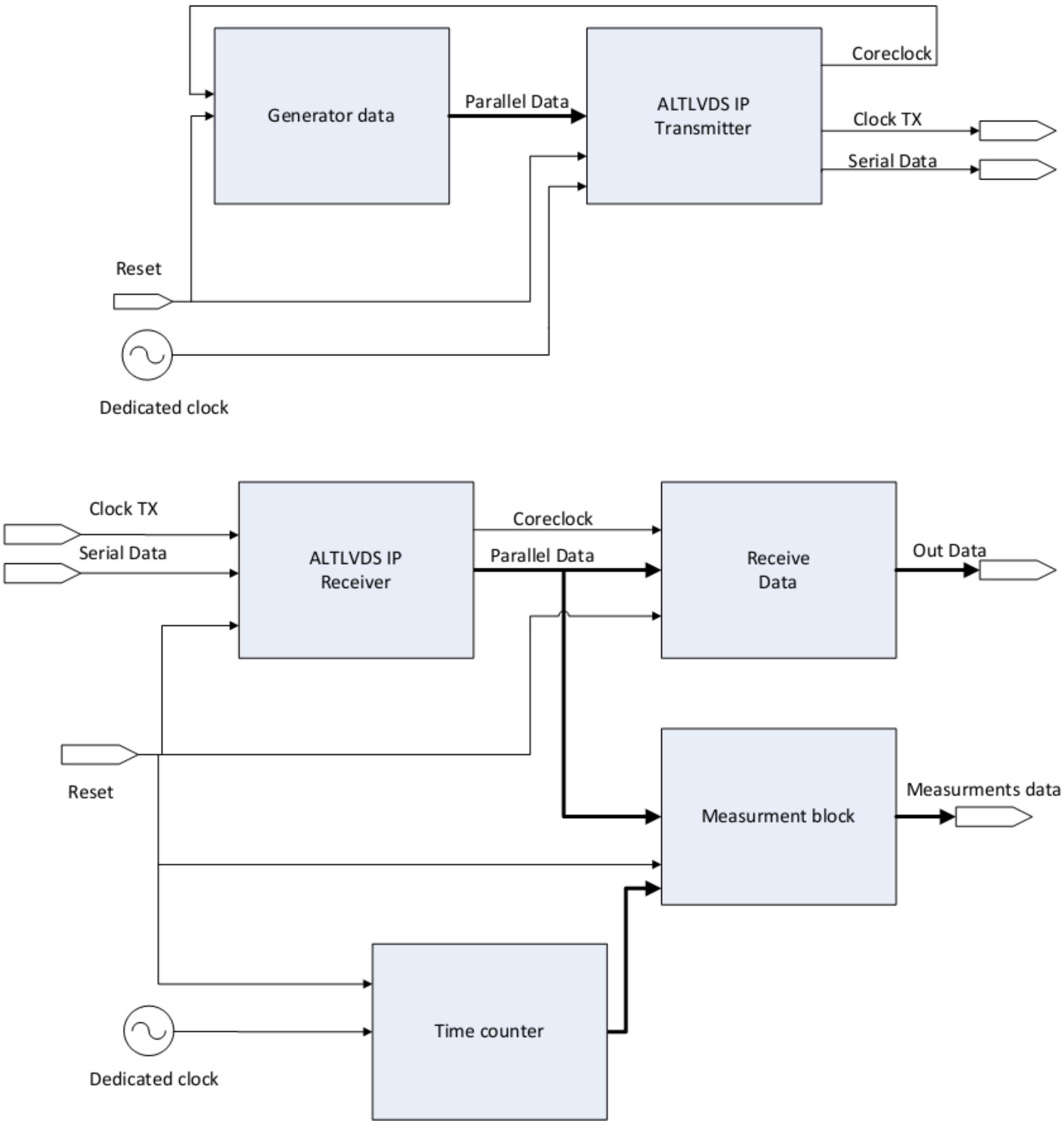


Figure 3.6 *LVDS transmitter structure.*

these values onto the data input port of the transmitter SERDES. A parallel clock from the transmitter SERDES clocks the user logic.

The blocks created on the receiver side contribute to receiving data and measuring logic. To calculate a bandwidth of the transmission link we want to get a number of the received packets, then calculate the overall transmitted bits and finally to divide this by the time of execution. The IP block of measuring logic counts clock cycles until predefined value, once as it reached the calculation of received packets is stopped. Figure 3.6 shows the top-level of design including Master and Slave designs.

3.1.6 Pin assignments

Assignment of the input and output LVDS signals are based on the Cyclone V data sheet specification [5]. As the dedicated SERDES are implemented in LVDS transmitter/receiver the output of the dedicated logic cannot be assigned to single-ended IO standards.

Pins assignments were made in the Quartus II Assignment Editor. LVDS standard should be chosen for the output and input pins. For receiver's pins on-chip differential termination should be set.

In implemented system the out pins of the transmitter are channel number 11 connected to pins A4 and A3 for data and out clock two A11 and A10 for clock signals. The receiver respectively gets data in the positive and negative pins E12 and D12 of the channel number 11, and data is received in input clock 2 H15 and J15 pins.

The transmitter design requires dedicated clock input, which is master reference clock according to the Figure design. As it was pointed earlier, the locations of LVDS channels and reference clock should be in one bank of the FPGA. The input clock for the bank 8A where LVDS pins are located dedicated clock is 50 MHz clock K14.

The placement of clock and data channels should be chosen with respect to reduce the skew between the channels.

3.2 Transceiver communication

To establish high-speed communication between FPGA boards via transceivers, a design based on the transceiver interface was implemented in this work.

In this section we explain how to apply transceiver design with using dedicated MegaFunction Transceiver IP core in Quartus II tool. MegaWizard tool offers transmitter-receiver Transceiver IP cores with few available modes. Native PHY Megacore mode of Transceiver is handled in the present configuration.

3.2.1 Transceiver hardware setup

The developed design consists of three FPGA boards, XTS daughter boards, SMA cables and host PC.

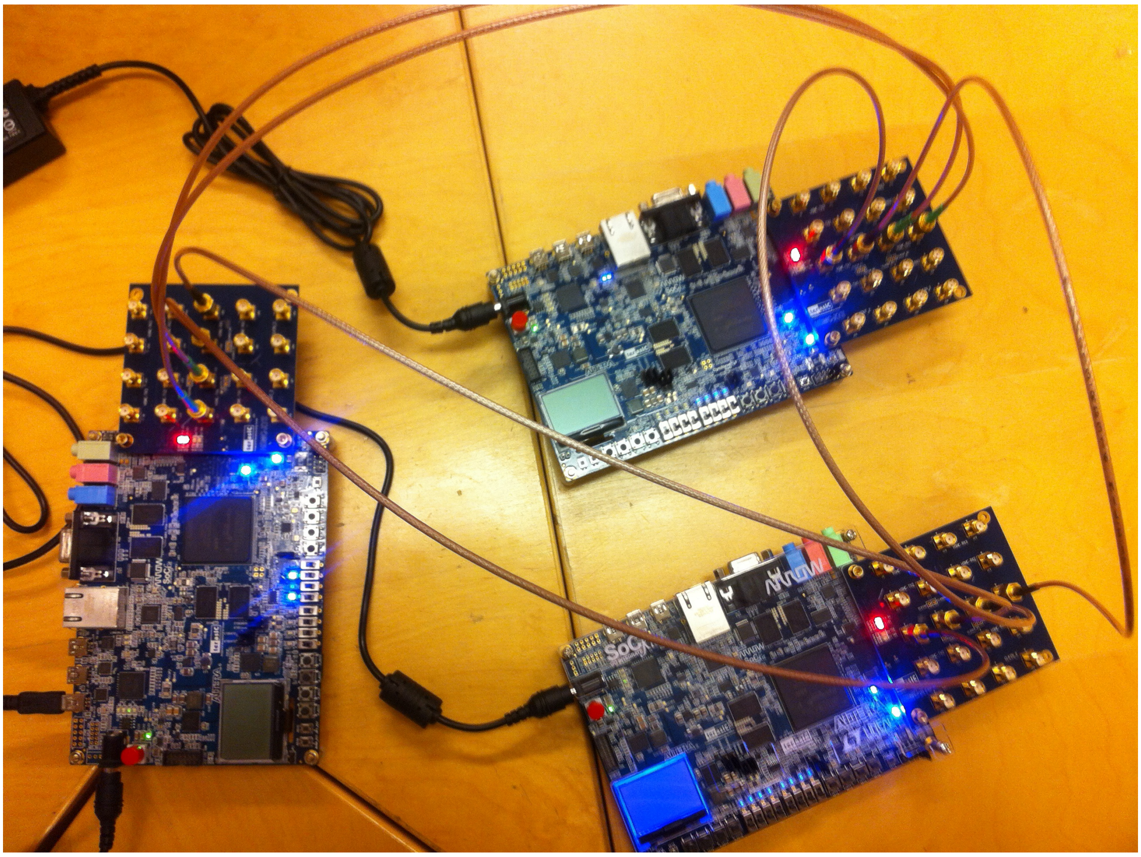


Figure 3.7 *Transceiver hardware setup*

XTS daughter boards are attached to HSMC connectors of each SoC Kit. This card converts transceiver channels through a HSMC inteface to SMA connectors. Four transceiver channels are supported by XTS daughter board. More information can be found in the XTS [37]. Present design applies SMA connectors on the XTS board to establish transceiver interconnection between FPGA boards via SMA cables, which are 40 cm length. Figure 3.7 shows how assembled design looks in life . Figure 3.8 introduces interface of the XTS.

In this topology, one of the boards is master board and two are slave boards. The master board generates data and two others by pass it further. In the master boards the sent and received values of control data are compared. The principle of topology is shown in Figure 3.9.

3.2.2 Structure of the system

The presented design perform following actions : at first of the initializing transceiver design is activated, then data generator send n-bit count value or synchronization word for receiver alignment. User logic in the receiver side processes incoming data

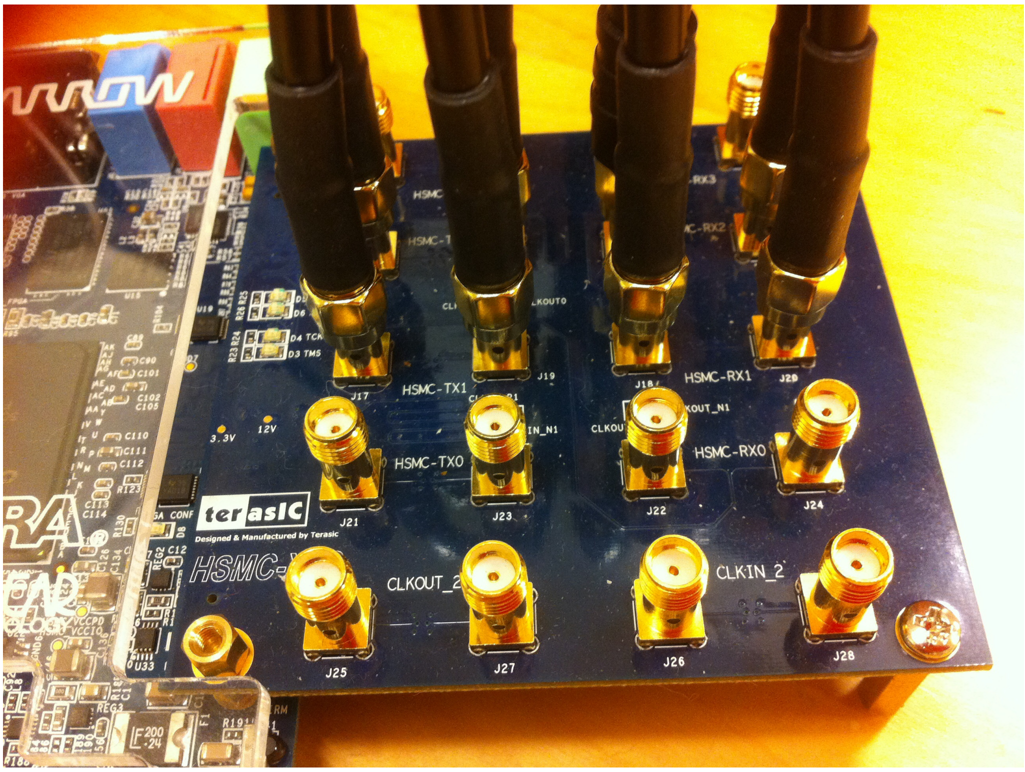


***Figure* 3.8** *XTS daughter board.*

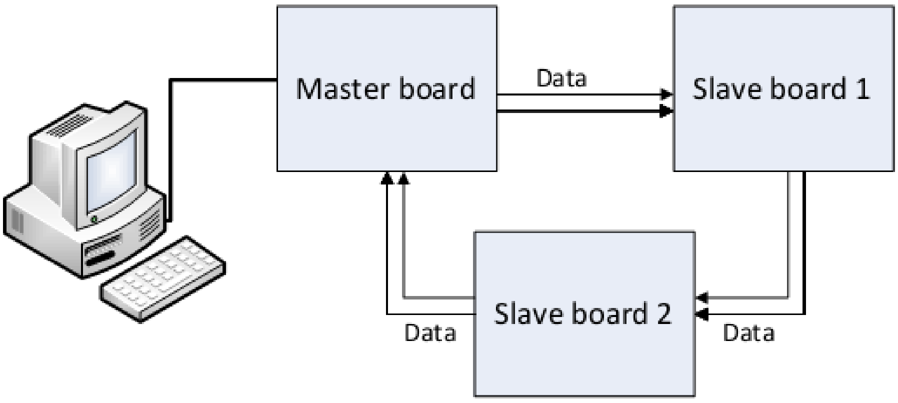


***Figure* 3.9** *Simplified transceiver design*

and measures the bandwidth and latency parameters.

The implemented design requires single ended clock input, which is a master reference clock. The Cyclone V SX has on-chip 50 MHz crystal oscillator. This clock drives PLL IP block which generates clocks in 50 MHz to 600 MHz range. The gener-

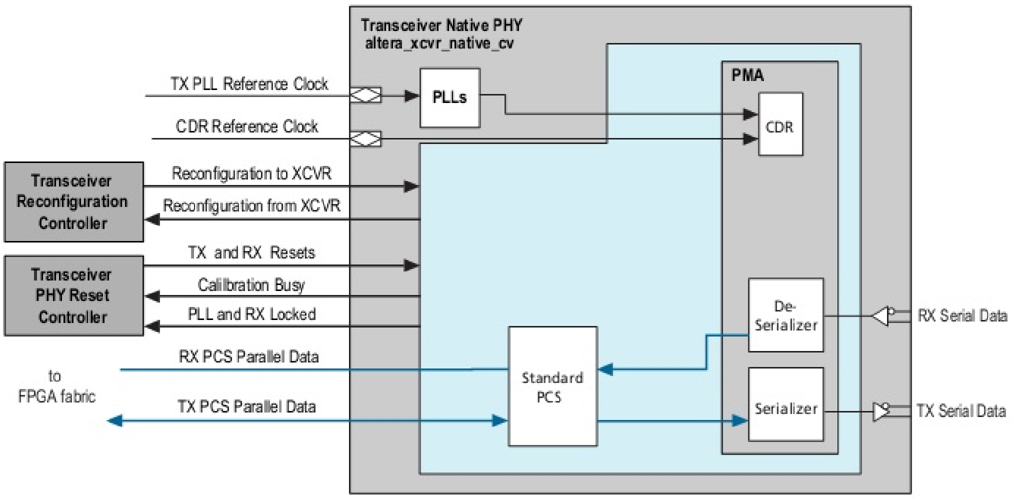


Figure 3.10 *Transceiver PHY Native architecture.*

ated clock are driven to the inputs of the clocks of Transceiver PHY IP, Transceiver Reconfiguration Controller and Reset Controller IP blocks. This Mega cores from Altera are used for implementation transceiver functionality.

Cyclone V Transceiver PHY IPs cores

Transceiver PHY IP, Transceiver Reconfiguration Controller and Reset Controller Megafunctions provided by Altera are required for implementation transceiver functionality. Figure 3.10 shows three blocks with appropriative connections.

- Cyclone V Transceiver Native PHY IP core parameters

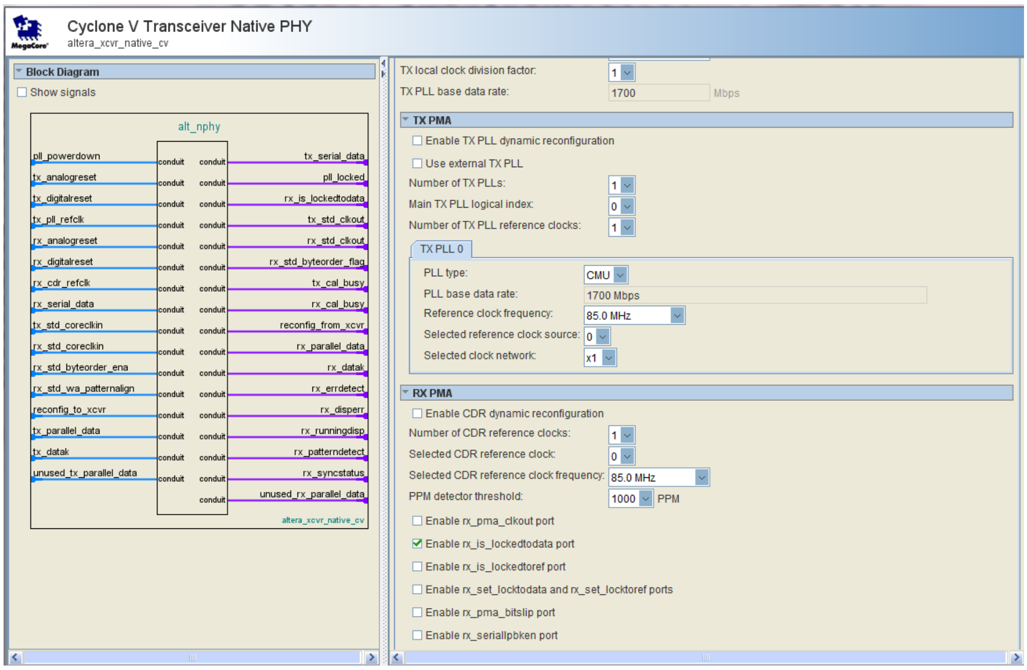
  Cyclone V Transceiver Native PHY IP is one of the available transceiver configurations. All signals are connected directly to the ports, without memory-mapped interfaces, as it implemented in other transceiver IP cores.

  Standard data path is available for the Cyclone V device : PMA and PCS.

  The general configurations for one of the test of executed designs are listed below.

  Number of data channels : 1.

  - PMA configurations.
  - Data rate: 1700 MBps.
  - Number of TX PLL : 1.



TX local clock division factor: 1
TX PLL base data rate: 1700 Mbps

TX PMA

□ Enable TX PLL dynamic reconfiguration

□ Use external TX PLL

Number of TX PLLs: 1
Main TX PLL logical index: 0
Number of TX PLL reference clocks: 1

TX PLL 0

PLL type: CMU
PLL base data rate: 1700 Mbps
Reference clock frequency: 85.0 MHz
Selected reference clock source: 0
Selected clock network: x1

RX PMA

□ Enable CDR dynamic reconfiguration

Number of CDR reference clocks: 1
Selected CDR reference clock: 0
Selected CDR reference clock frequency: 85.0 MHz
PPM detector threshold: 1000 PPM

□ Enable rx_pma_clkout port

☑ Enable rx_is_lockedtodata port

□ Enable rx_is_lockedtoref port

□ Enable rx_set_locktodata and rx_set_locktoref ports

□ Enable rx_pma_bitslip port

□ Enable rx_seriallpbken port

***Figure* 3.11** *Transceiver Native PHY IP configuration.*

- Reference clock frequency : 85 MHz.

Standard PCS configurations :

- Standard PCS protocol mode : basic.
- Standard PCS/PMA interface width : 10.
- FPGA fabric/ Standard TX PCS interface width : 8.
- TX/RX FIFO mode : low latency.

The word aligner operates in one of the following modes.

- Manual alignment - when user control word alignment.
- Bit-slip- the word boundaries shifted by inserting latencies.
- Automatic synchronization state machine - programmable state machine control word.

- RX word aligner mode : manual.
- RX word aligner pattern length : 10.
- RX word aligner pattern (hex) : 17C, K28.5 negative comma is used.
- The Figure 3.11 shows the example of configuration of Transceiver Native PHY IP.

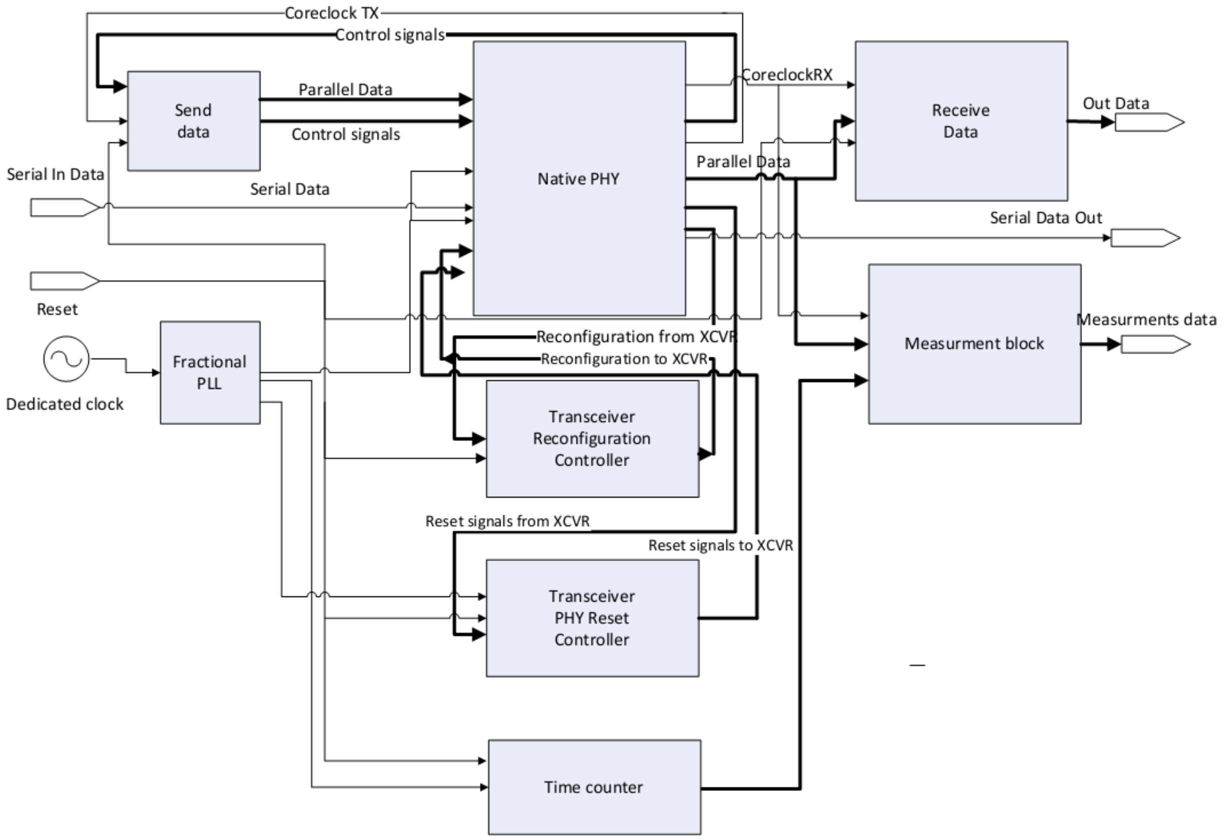


Figure 3.12 *Transceiver Master board design.*

- Transceiver Reconfiguration Controller IP core allows to collaborate and reconfiguration the signals PHY IP core. In the configurations of this IP block the total number of reconfigurable interfaces needed to provide.
- Transceiver PHY Reset Controller IP core makes power up and initializing of reset sequences for correct transceiver design.

3.2.3 Send/receive data logic

- Master board. The design programmed to the Master device supports all function to process a data: generating, controlling, sending and receiving. Figure 3.12 provides implemented design for Master board.

  Send data block generates a data and is driven by clock output from the Native PHY. The divide factor is assigned to one, thus clock frequency is the same as reference clock.

  First of all, data generator sends control characters to the data channel to establish a synchronization of data to receiver. This pattern is known by receiver part and compares with the received characters with predefined value in the Transceiver Native PHY IP Core. If incoming data is corrupted and is not equal to the specified one, then signal is delayed until sampling clock edge is placed exactly in the middle of two transitions. This process guarantees the correct byte alignment of the received data. For each of the channels calibration is required [7].

Pattern and error detect, disparity control and type of data signals from MegaFunction ports deliver information about current status of the data from the receiver. When pattern-detect and control signals are asserted then alignment pattern BC is correctly received by the receiver side. Data generator block observes continuously status information and ensures that alignment is completed to continue generate and send packets of the data.

The transmitted data is driven to the input of the Native PHY.

Data stream flows through the boards and is received back to the input of the receiver part of the Master device. Receive data block gets data from Transceiver Native PHY IP and processes to measure the performance of the system.

- Slave board. In the design programmed to the slave device the data is driven from the Native PHY IP receiver to Receive data block. Figure 3.13 shows main components of this design Also received values are transmitted to the Data from RX block, which passes data back to Transceiver Native PHY IP transmitter.

3.2.4 Measuring logic

The implemented system is designed to get experimental values of latency and bandwidth of the data. For evaluation of the speed the systems handle values from this IPs. Measurements of latency and bandwidth are implemented in the logic of blocks, which enables transmission and receiving of data to/from Transceiver IP block.

For evaluating of latency, the number of clock cycles needed for loop transferring of the data to the same FPGA board are counted. Data flows through chain of devices. To calculate the latency in seconds we multiply the measured number of clock cycles to the length of clock period of clock, which drives this logic block. Measurment block handles this measurement.

To calculate the bandwidth we count the number of packets, which are sent in the fix period of time. Size of the packets are predefined to the 2048 bits.

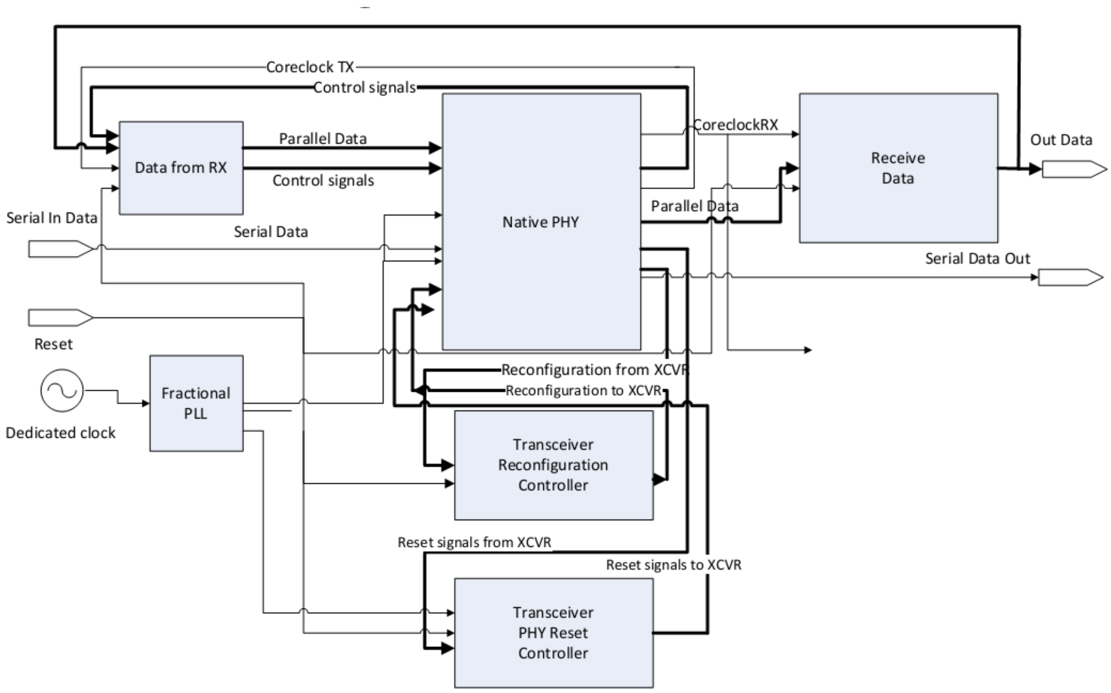


Figure 3.13 *Transceiver Slave board design.*

When the reasonable period of time is decided, the counter measures the number of received packets during that interval.

To control the time of measuring the additional logic block was implemented. The needed time of experiment is divided to the duration of clock cycle, which drives the logic; the resulting value is a maximum value of the counter. Time counter block makes this measurement.

3.2.5 Pin mapping

To port the implemented design to tested hardware the input and output ports of the implementation should be directed to the correct physical pins on the FPGA board. Mapping is done similarly as in the LVDS design. IO standard of transceiver pins, slew rate and output strength, pull-up or pull-down registers should be considered.

The clock is assigned to 50 MHz crystal oscillator input. Resets are connected to the active-low reset of the board. The transceiver's transmitter and receiver ports are mapped to expansion HSMC header's pins. In this work we use one channel design configurations. The transmitter signals are driven to the channel number 1 to the AB4 and AB3 pins. The receiver respectively gets data in the AC2 positive and AC1 negative pins of the channel 1 on the Cyclone V device. The used transceiver channel should be accurately chosen due clock distributing limitations.

A number of design requirements in terms of nominal voltage and current supply is recommended to be considered. Power supply integrity is important for all high-speed transceivers, because they rely on PLLs that contains VCOs [7]. The reference information is provided in guidelines [5].

3.2.6 Timing constraints

Running on the clocks enabled hardware designs to satisfy required timing closures [8]. For reliable operations timing constrains should be ensured when data goes through the registers.

To correctly capture data a common consideration should be applied to the registers' timing:

- Setup time : the time when input to a register must be stable before the clock edge.
- Hold time : during this time input to a register must be stable after the clock edge [3].

This relationships are also pointed as following statements.

- Duration of propagation delay and setup time are less or equal to the clock period.
- Duration of propagation delay is more or equal to the hold time [30].

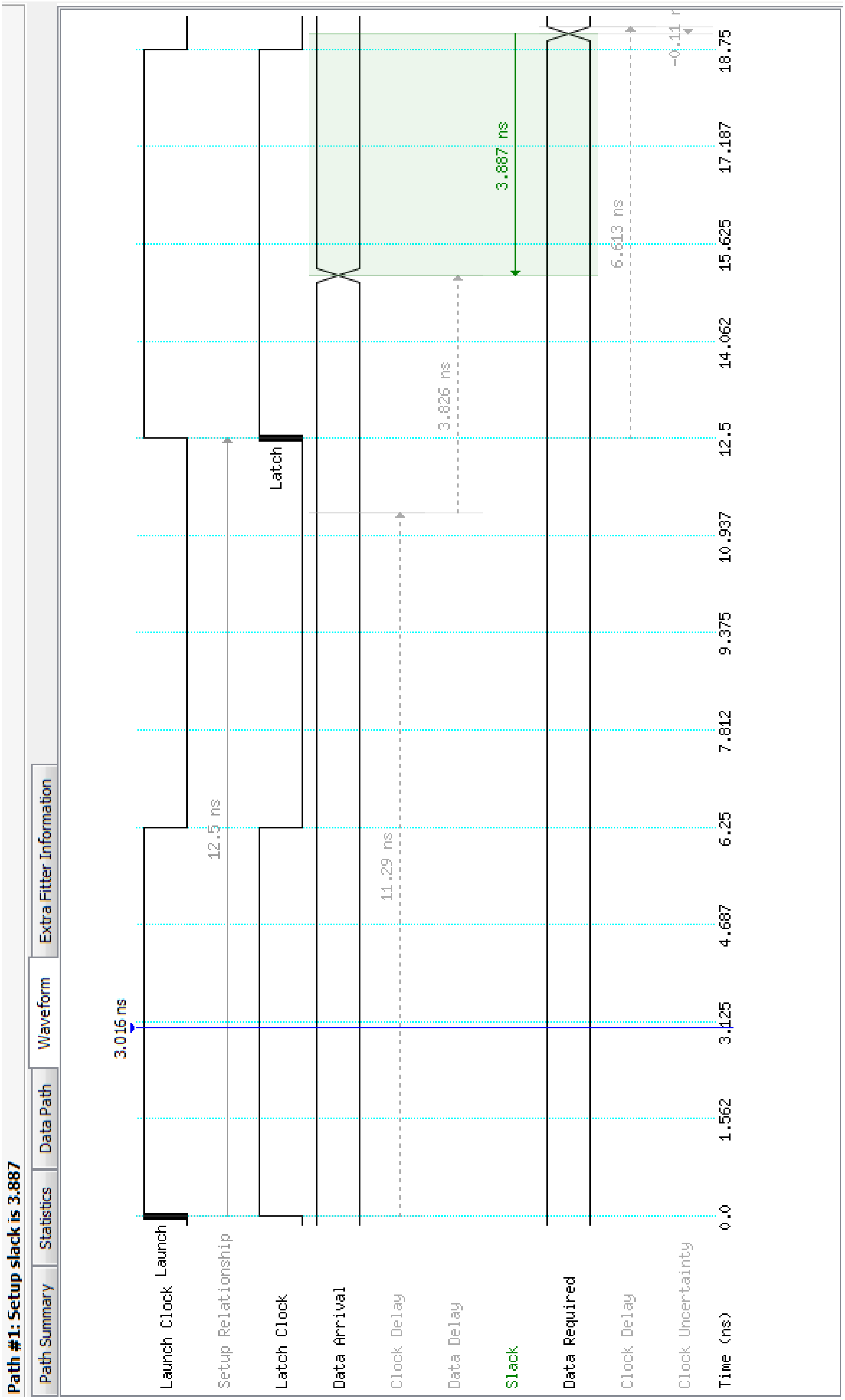
To avoid timing failures and problems in design this conditions should be archived for all paths within the chip.

In the implementation of design timing constraints, Synopsys Design Constraints (.sdc) files describe the timing for specific FPGA board, for example, the target frequency of the device and the timing to external peripherals.

Quartus II provides access to the Timing Analysis GUI Tool, which enable graphically set a timing constrains by setting multiple pat and multiple clocks. The common control of timing analyzes in this software consists of checking the timing multiple times with different timing modes. For example the maximum and minimum propagation delays for the temperature and manufacturing variation are considered together.

Figure 3.14 shows the Timing Analysis GUI and timing diagrams of one of the Transceiver designs.

Slack describes by how much the setup and hold times are overfilled. Slack is positive, thus timing requirements are overfilled. In comparison if the slack is negative the timing requirements are not met. In this case additional considerations should be applied and debug process provided.



Path #1: Setup slack is 3.887

Figure 3.14 *Timing Analysis GUI.*

4. PERFORMANCE EVALUATION

One of the purpose of this thesis is to evaluate the created designs in real conditions. The hardware designs are assembled according to the described systems in the previous chapter.

For obtaining the results we use SignalTapII included in Quartus II software. This on-chip Logic Analyzer debugging tool offers real-time observation in the circuit signal changes in the various points.

This chapter provide description of basic settings of SignalTapII, as well as the outcomes of measurements for every topology, and finally the results are analyzed.

4.1 SignalTapII logic analyzer

Altera's SignalTapII tool provides a high-resolution view of the data being transferred on a number of signals.

At first all devices are programmed by Joint Test Action Group (JTAG) interface Universal Serial Bus (USB) Blaster download cable from host computer and then one device is connected to observe signals inside FPGAs.

Via this communication link the memory blocks of the FPGA store capture data and transfer the data to the Quartus II software waveform display [9]. Figure 4.1 shows the principle of SignalTapII work.

To monitor signals we need to create and configure instance in SignalTapII GUI interface. There are an option to select signals, to define triggering condition and number of samples to be recorded before triggering. When triggering condition is earned the recording of the data stream is finished.

For instances it is necessary to assign a clock signal to control the acquisition of the data by the SignalTapII. Driving clock is an important parameter in the instance configuration. Sampling of the data is processes on every rising edge of the acquisition clock. The logic analyzer does not support sampling on the falling edge of

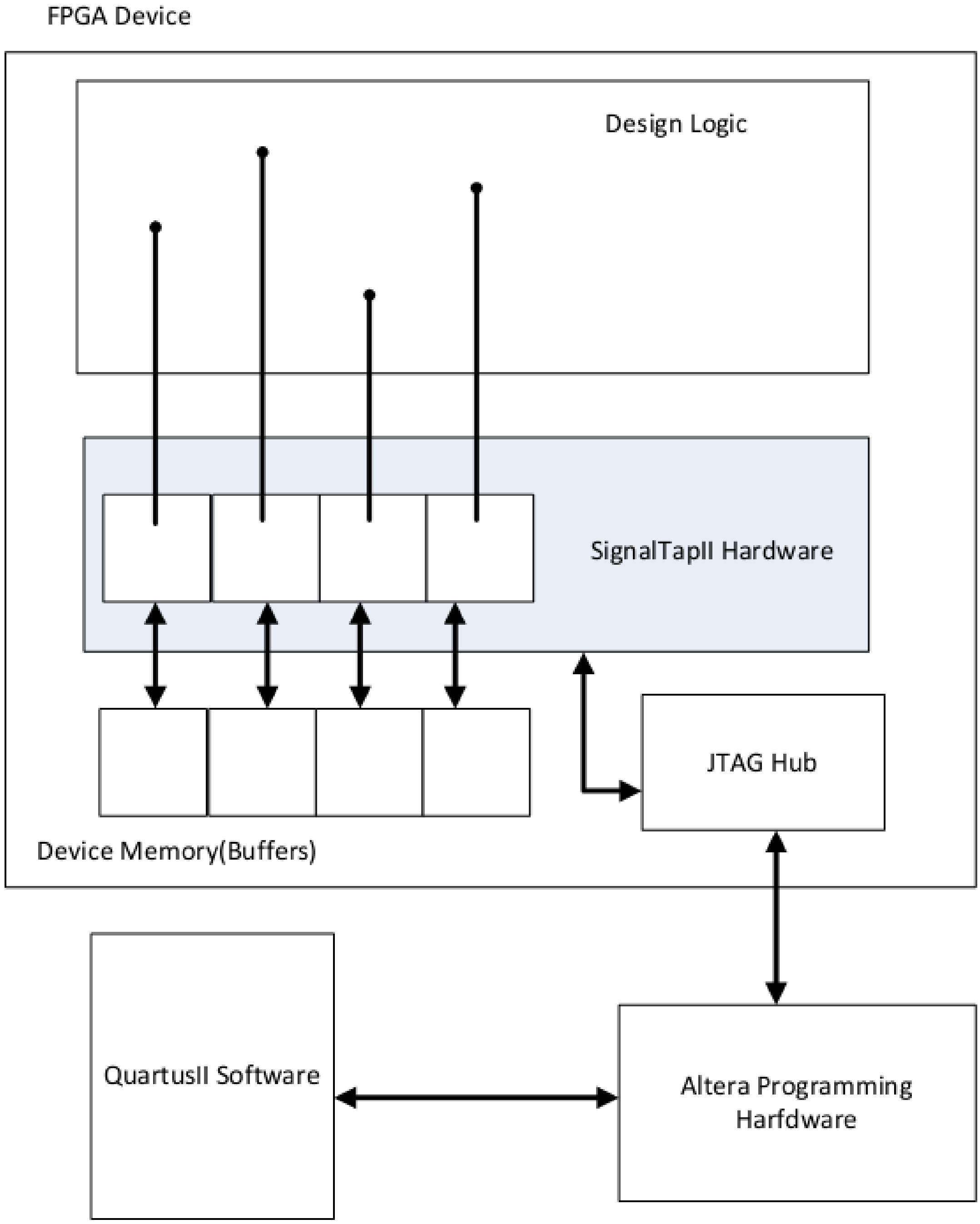


Figure 4.1 *SignalTapII scheme*

the acquisition clock. In Altera guidelines [6] it is recommended to use a global, non-gated clock synchronous to the signals under test for data acquisition. Because a gated clock as acquisition clock can not guarantee that behavior of design would be totally kept. The Quartus II static timing analysis tool shows the maximum acquisition clock frequency at which the design can be run. The maximum frequency of the logic analyzer clock can be found in the previously described Timing Analysis tool.

SignalTapII enables debugging an FPGA design by probing the state of the internal signals in the design without the use of external equipment. Custom trigger-condition logic provides accuracy and reduce possible problems. More information about tool SignalTapII Tool can be found in [6].

The observed signal in the implemented the design is the counter of received times of the transmitted data. On the base of received results the bandwidth is calculated.

Also the SignalTapII allows to check the values of data on the source and destination. The signal of the generated source data needs be observed to detect presence of the control character. As the predefined pattern is displayed in the diagram it indicates that the character is generated and sent.

In the current testing in SignalTapII two instances are created : in one we observe the data and number of transmission per fixed time; another shows the latency of the design in the counted clock cycles.

To check working of the data alignment algorithm we manually reset the slave programmed board by the reset switch in the device. When receiver is not able to get data the data generator starts to send control character continuously, waiting for acknowledgment signals from the receiver side. Active-low reset allows to receive data and transmitter starts further transmission.

4.2 Measurement results

4.2.1 LVDS test

As previously described the top-level design of LVDS test includes two boards, which are connected by one data channel. The designs programmed on Master device and Slave device are shown in Figure 4.2 and Figure 4.3. All executed tests were run in the same conditions.

Seven data rates from 400 Mbps to 860 Mbps were chosen to perform the tests.The transmitted packet size is 2048 bits, 30 s is a timing interval.

To measure a bandwidth of the transmission link the following steps were produced.

- Get a number of the received packets.

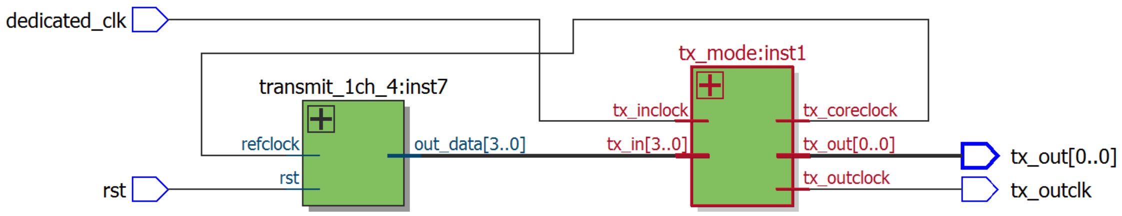


Figure 4.2 *Master LVDS design. RTL view.*

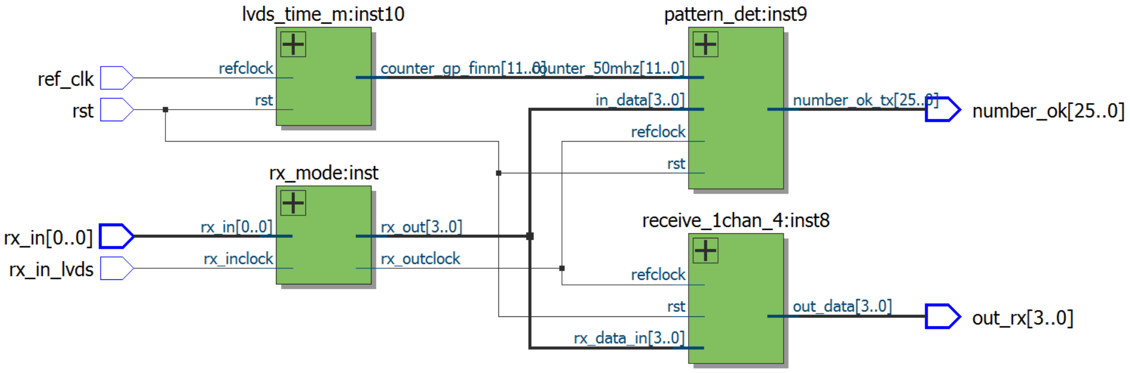


Figure 4.3 *Slave LVDS design. RTL view.*

- Calculate total transmitted bits.
- Divide the result by the time of execution.

Table 4.1 provides output of the LVDS communication measurements. Received results for each data rate are listed. As the bandwidth in the bounds of one data rate do not change significantly, thus three experiments are enough to show the performance of the system.

The SignalTapII instance captures the incoming serial stream,the number of the received packets and the number of clock cycles, which is a trigger condition in the implemented logic. Figure 4.4 shows the snap of the SignalTapII.

4.2.2 Transceiver test

The top-level design consists of three boards, communicating via transceiver interface. Figure 4.5 and Figure 4.6 show the RTL design of the designs programmed to Master and Slave boards.

Table 4.1 *LVDS measurment results*

Data rate, Mb/s	Frequency, MHz	Data rate, b/s			
		Experiment 1	Experiment 2	Experiment 3	Average
400	100	387494312	387494775	387366537	387451875
500	125	484505908	500010315	493454464	494839073
600	150	581407089	581411732	581411732	581410184
700	175	678306465	678012148	677837002	678051872
760	190	752172576	752172576	750145403	751692136
800	200	776446667	776653333	775144667	776081556
840	210	840100800	813867180	813873690	822613890

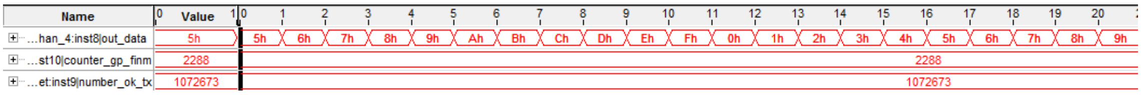


Figure 4.4 *SignalTapII diagram.*

The data rate range of experiments includes speed from minimum 400 Mbps to maximum 1860 Mbps available in Transceiver Native PHY IP MegaFunction. Number of data rates is 14. The measured latency is about 80 ns - 100 ns.

To compute the experimental value of bandwidth of the system, the number of transmitted packets are multiplied by the size of the data packet in bits and divided by the time of execution in seconds. The chosen data packet size is 2048 bits, executed time for measurement is 30 s. Nine experiments for each data rate were performed. The results of measurements are shown in the Table 4.2,Table 4.3 and Table 4.4.

4.3 Analyzing of the results

- The received results from the measurement show that expected data rates are archived in LVDS and transceiver communication designs. The losses are low and do not affect the performance.

- The Transceiver Native PHY IP operating data rate is limited to a maximum data rate for one link by 1.86 Gbps instead of advertised 3.25 Gbps. The results of experiments shows that system running in data rate 1840 Mbps performs best results, thus it is archived maximum for the presented system. This reduces overall performance of that current mode of transceiver interface. However applying all 4 available channels in the communication would result in 7440 Mbps (1860Mbps x 4) total bandwidth and Transceiver Native PHY IP mode can be applied in applications required high speeds.

Table *4.2 Transceivers design measurement results*

Data rate, Mb/s	800	1000	1200	1300	1400
Frequency, MHz	100	100	100	81.25	87.5
Number of experiment	Data rate, b/s				
1	798286099	996795427	1195136539	1288957336	1391761564
2	798190467	996826645	1196106017	1289051291	1391772959
3	798222115	996844877	1196024317	1289125638	1391921309
4	798276940	996720306	1196062071	1288796559	1392103328
5	798286185	996867409	1196043710	1288792388	1391907033
6	798221814	996719016	1196046204	1288947188	1391875041
7	798254365	996772207	1196114058	1289068878	1391956010
8	798193778	996798265	1196041044	1288761600	1391876804
9	798232865	996798652	1195975469	1289039595	1391817808
Average	798240514	996793645	1195949937	1288948941	1391887984

Table *4.3 Transceivers design measurement results (continue 1)*

Data rate, Mb/s	1500	1600	1700	1780	1800
Frequency, MHz	75	100	85	81.25	87.5
Number of experiment	Data rate, b/s				
1	1492211671	1575162111	1659847730	1723345787	1777585858
2	1492126875	1574630459	1660367987	1723666911	1773176896
3	1492142312	1575224633	1659781123	1723631221	1777153407
4	1492125370	1575209411	1660332039	1723471519	1779381495
5	1492245168	1575288918	1658847550	1723314956	1778004506
6	1492182173	1575462810	1659763751	1722928730	1777765985
7	1492094711	1575555819	1659383373	1722720997	1778492427
8	1492115781	1575567128	1659804515	1722494172	1778408147
9	1492124123	1575535781	1659776221	1723243232	1778422251
Average	1492152020	1575293008	1659767143	1723201947	1777598997

***Table 4.4** Transceivers design measurement results (continue 2)*

Data rate, Mb/s	1820	1840	1850	1860
Frequency, MHz	90	92	92.5	77.5
Number of experiment	Data rate, b/s			
1	1798749340	1820792731	1807015918	1813712910
2	1799354479	1821253304	1807495454	1813600379
3	1799151777	1820809759	1807423128	1813410620
4	1799417689	1822808915	1807357338	1813658343
5	1799326443	1822242691	1807369980	1813845092
6	1799313113	1821754727	1807279809	1813675242
7	1799430417	1821277771	1807295891	1813544823
8	1799430417	1822899086	1807258696	1813662987
9	1799362821	1822212075	1807315929	1813544909
Average	1799255106	1821783451	1807312460	1813628367

- To analyze statistically received results we computed confidence interval of a standard deviation for the experimental data rates. The principle of this interval to observe differences of the received value in the defined confidence level. Confidence level is used in calculations of the confidential intervals and commonly chosen 95%. The measured values of the data rates produce intervals which are too small be notable in the gigabits order. The graphical representations of confidence intervals are depicted in Figure 4.7 for LVDS and Figure 4.8 for transceiver designs. Minimum - is a lower limit of confidence interval, maximum - is a higher limit.

4.4 Footprint and development costs

The implemented designs have a simple structure and used resources are minimal. The sources for fitting created systems on the FPGA device are listed in Quartus II Compilation Fitter report. Figure 4.9 shows main PLL resources, which are applied during the compilation of the Transceiver design.

As well we summarize used boards' resources in implementing of LVDS and transceiver based systems.

Table 4.5 includes the report of the usage of the sources in LVDS design. Both

Table 4.5 *LVDS resource usage*

Number of data channels	1	2	4
Logic utilization (ALMs needed)	522	611	623
Global signals	4	4	4
M10K blocks	2	2	2
Total block memory bits	3328	6144	6400
Total memory implementation bits	2048	2048	2048
Fractional PLL	1	1	1
Global clocks	4	4	4
Qudrant clocks	0	0	0
Spine clocks	8	16	12
IO pins	11	16	35

Table 4.6 *Transceivers resource usage*

Number of data channels	1	2
Logic utilization (ALMs needed)	1203	1216
Global signals	7	10
M10K blocks	5	5
Total block memory bits	23808	17480
Total memory implementation bits	2048	2048
Fractional PLL	1	1
Global clocks	6	7
Qudrant clocks	1	2
Channel PLL	2	3
IO pins	11	16

required logical elements and clock resources are minimal for establish the communication. We see that increasing of the number the data channels for transmission enlarges consumed board resources, but not significantly.

Table 4.6 lists the main applied resources for one and two data channels. The results show that in transceiver design the number of the required clocking resource is notable larger compare with LVDS SERDES interface. For example, number of the global signals and the global clocks for LVDS design with four transmission lines are less then the number of such resources in transceiver interface with one communication channel.

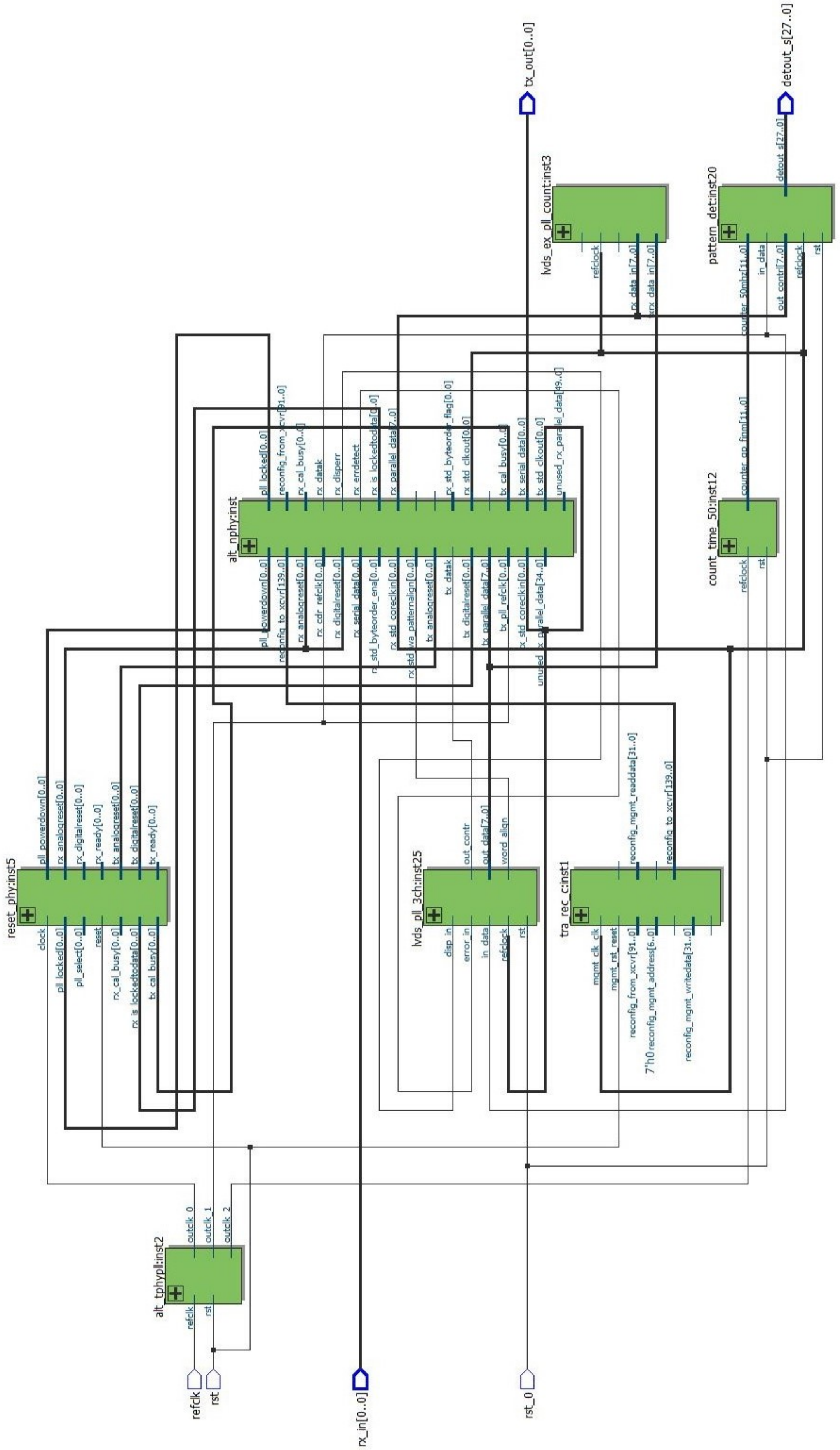


Figure 4.5 *Master Transceivers design. RTL view.*

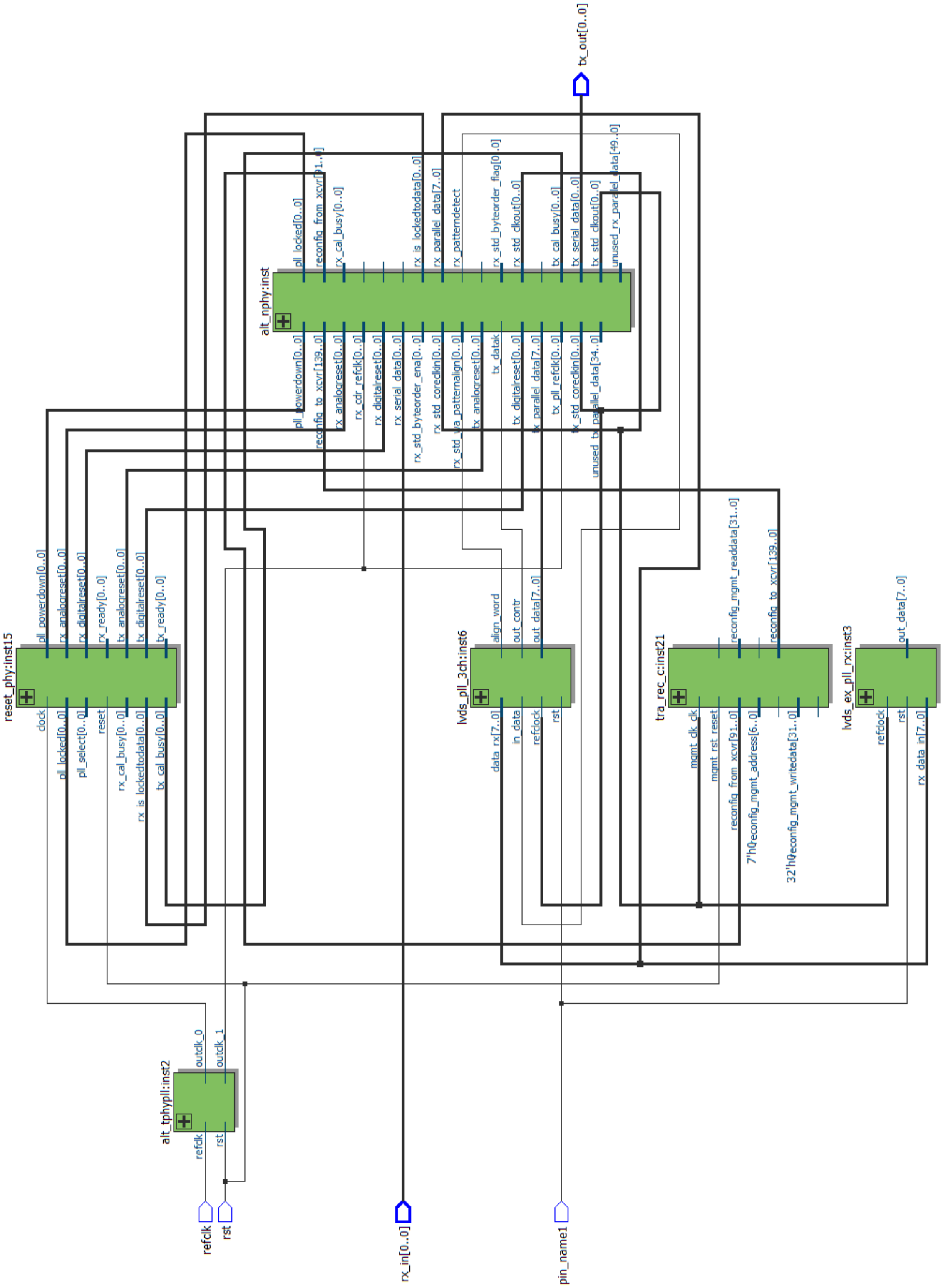


Figure 4.6 *Slave Transceivers design. RTL view.*

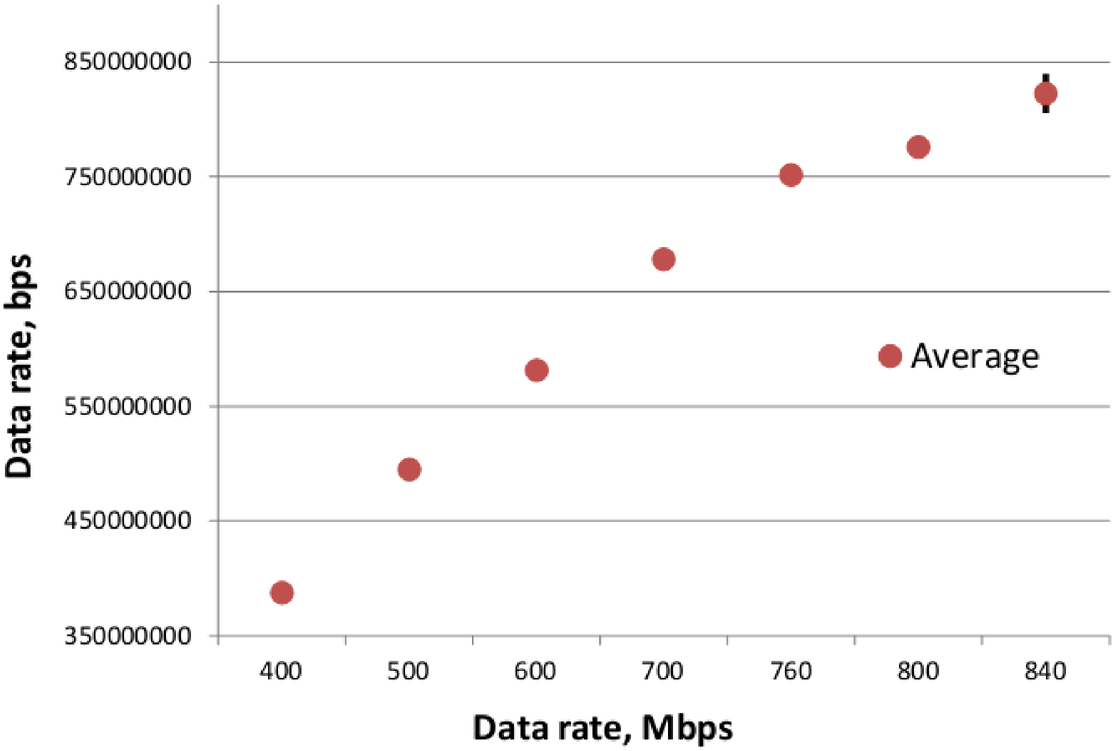


Figure 4.7 *Confidence interval for LVDS measurments*

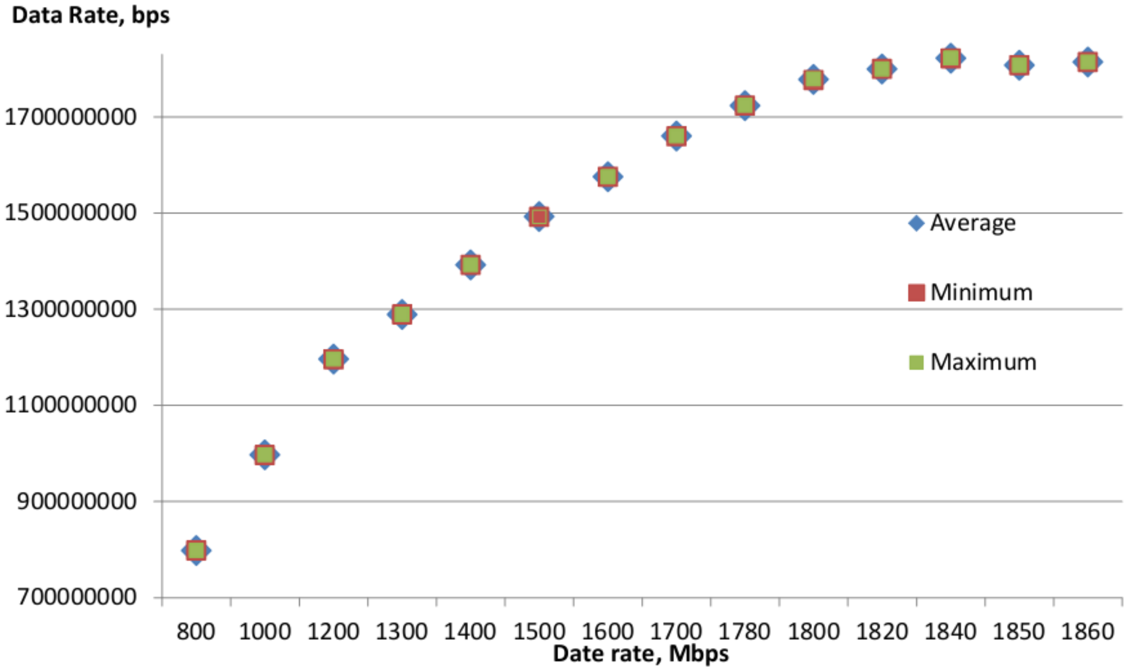
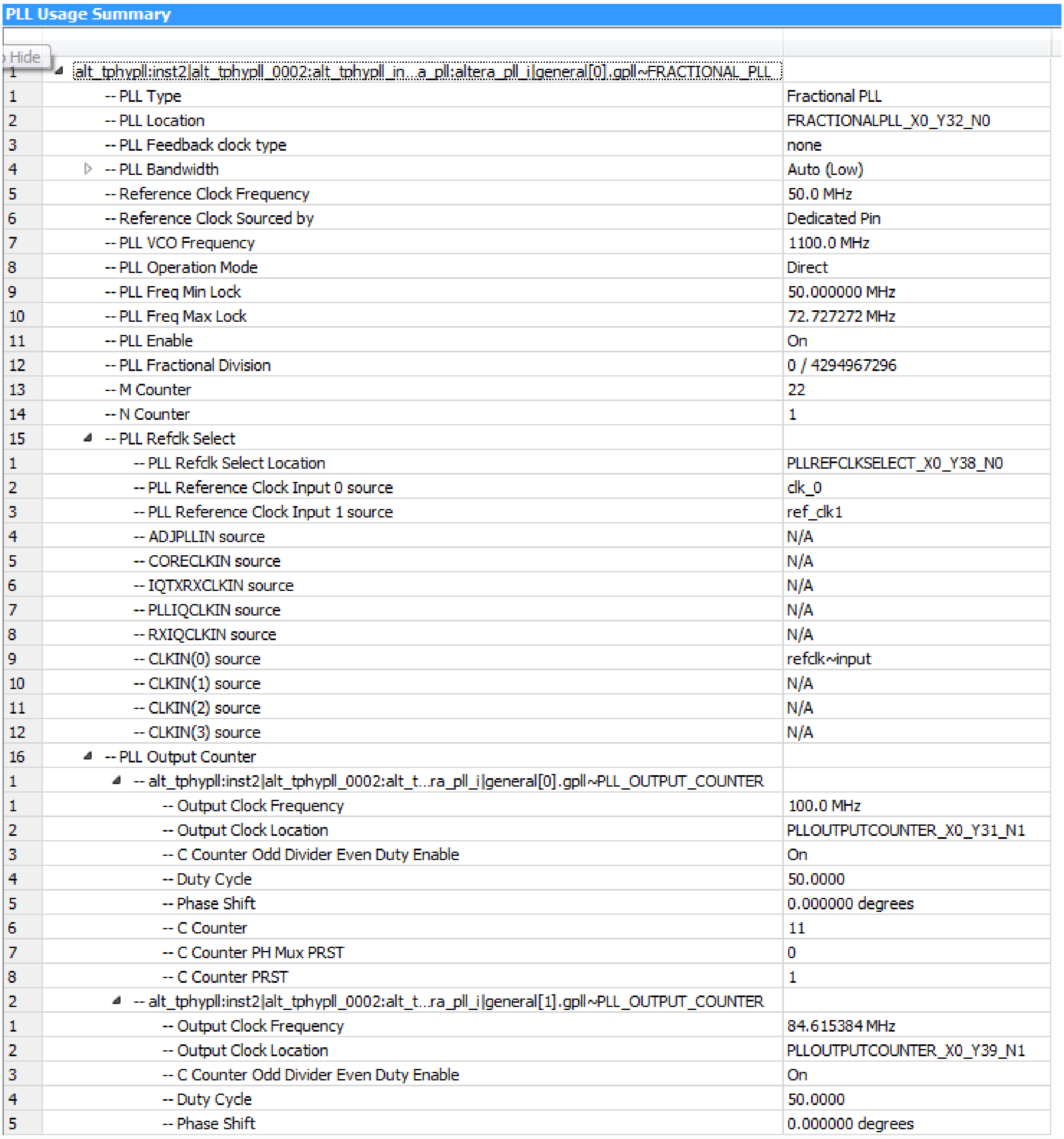


Figure 4.8 *Confidence interval for Transceiver measurements*



PLL Usage Summary

1	alt_tphypll:inst2|alt_tphypll_0002:alt_tphypll_in...a_pll:altera_pll_i|general[0].gpll~FRACTIONAL_PLL	
1	-- PLL Type	Fractional PLL
2	-- PLL Location	FRACTIONALPLL_X0_Y32_N0
3	-- PLL Feedback clock type	none
4	-- PLL Bandwidth	Auto (Low)
5	-- Reference Clock Frequency	50.0 MHz
6	-- Reference Clock Sourced by	Dedicated Pin
7	-- PLL VCO Frequency	1100.0 MHz
8	-- PLL Operation Mode	Direct
9	-- PLL Freq Min Lock	50.000000 MHz
10	-- PLL Freq Max Lock	72.727272 MHz
11	-- PLL Enable	On
12	-- PLL Fractional Division	0 / 4294967296
13	-- M Counter	22
14	-- N Counter	1
15	-- PLL Refclk Select	
1	-- PLL Refclk Select Location	PLLREFCLKSELECT_X0_Y38_N0
2	-- PLL Reference Clock Input 0 source	clk_0
3	-- PLL Reference Clock Input 1 source	ref_clk1
4	-- ADJPLLIN source	N/A
5	-- CORECLKIN source	N/A
6	-- IQTXRXCLKIN source	N/A
7	-- PLLIQCLKIN source	N/A
8	-- RXIQCLKIN source	N/A
9	-- CLKIN(0) source	refclk~input
10	-- CLKIN(1) source	N/A
11	-- CLKIN(2) source	N/A
12	-- CLKIN(3) source	N/A
16	-- PLL Output Counter	
1	-- alt_tphypll:inst2|alt_tphypll_0002:alt_t...ra_pll_i|general[0].gpll~PLL_OUTPUT_COUNTER	
1	-- Output Clock Frequency	100.0 MHz
2	-- Output Clock Location	PLLOUTPUTCOUNTER_X0_Y31_N1
3	-- C Counter Odd Divider Even Duty Enable	On
4	-- Duty Cycle	50.0000
5	-- Phase Shift	0.000000 degrees
6	-- C Counter	11
7	-- C Counter PH Mux PRST	0
8	-- C Counter PRST	1
2	-- alt_tphypll:inst2|alt_tphypll_0002:alt_t...ra_pll_i|general[1].gpll~PLL_OUTPUT_COUNTER	
1	-- Output Clock Frequency	84.615384 MHz
2	-- Output Clock Location	PLLOUTPUTCOUNTER_X0_Y39_N1
3	-- C Counter Odd Divider Even Duty Enable	On
4	-- Duty Cycle	50.0000
5	-- Phase Shift	0.000000 degrees

Figure 4.9 PLL resource usage summary of Transceiver design

5. CONCLUSIONS

In conclusion, the most important findings of this study are following.

- When it is necessary to connect several FPGA devices, one typically considers the usage of either LVDS or built-in transceiver modules. While other solutions such as PCIe and Ethernet can also be used, they are much more complex and thus avoided. Both LVDS and dedicated transceivers provide very high data rates with minimal implementation effort, however in practice there are a lot of issues associated with both of them. During literature research, we found that there is very little information on the real-world performance of HS interconnect interfaces for FPGA to FPGA communications. It is, therefore, very important to understand the exact practical limitations of both approaches.

- To address the missing information on the real-world implementation costs for HS interconnects, a measurement setups were designed that emulates real-world topologies that are typical for multi-FPGA computation setup. Further, appropriate testing units were made to make use of both hardware technologies, as well as the measurement methodology, all of which are presented in Chapter 3. One could expect the presented testing setup to be applicable well outside of this work for evaluation of other interconnect technologies in the future.

- Further, an in-depth evaluation of performance of the selected embedded IP hard cores was completed. Throughput values have been collected to reflect the exact achievable rates. All data was statistically analyzed as well as to confirm the accuracy of the measurements. Based on the measurement data, we could conclude that none of the advertised data rates are actually achieved in a practical setup. In fact, both LVDS and dedicated transceivers struggle to achieve their advertised maximal rates when wiring imperfections are introduced. The performance gap is especially high for transceivers reaching as much as 15% at maximal rate.

- During the design and the measurements, significant amount of data was collected on the hardware cost of both interconnect options considered. Data has been collected on the number of required logical cells and other resources. In general, transceivers consume significantly more logic cells and PLL resources than similar LVDS solutions, and are thus more suited towards applications that really require all or most of the services they provide on top of what LVDS already does. Another side of the practical implementation revealed that there is a lot of inconveniences and not clearly stated issues in the specifications and documentation for both protocols, significantly contributing to the development times.

To implement a HS interconnect, the following points are important to take in consideration from this thesis:

- The documentation on clock networks, available IO resources and dedicated hardware limitations should be carefully studied before implementation of such kind of designs. Otherwise there are a risk of implementing a system that would never be able to run on a specific device due to nature of the high-frequency interfaces involved.
- Following up on the previous note, care must be used when dealing with transmission lines. Strict constraints on the length, impedance and uniformity of the cabling must be observed, as well as grounding and static electricity protection.
- The received numerical results clearly indicate that one should not expect the maximal physical layer bit rate to be achieved. Further, reliable transmission of messages must be ensured with checksums, because probability of error in links is high and the applications in hardware are not designed to handle packet errors.

The direct connection interfaces studied in this thesis are reasonably reliable way to connect multiple FPGA boards to each other. Unlike alternative technologies oriented at data transfers, they may lack proper error correction or built-in addressing. On the other hand, however, their footprint in the FPGA and on the circuit board is minimal, and the knowledge required to use them is much less than that needed to set up an Ethernet network. In the future one could expect many new types of board to board interconnect formats.

BIBLIOGRAPHY

[1] Ookla. [Online]. Available: http://explorer.netindex.com/maps

[2] Y. C. L. Albert Y. Zomaya, *Energy Efficient Distributed Computing Systems*. Wiley-IEEE Computer Society Press, 2012. [Online]. Available: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470908750.html

[3] Altera, *Understanding Metastability in FPGAs*, 2009. [Online]. Available: https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01082-quartus-ii-metastability.pdf

[4] ——, *AN 479 : Design Guidelines for Implementing LVDS Interfaces in Cyclone*, 2013. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/an/an479.pdf

[5] ——, *Cyclone V Device Family Pin Connection Guidelines*, Altera, 2013. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/dp/cyclone-v/PCG-01014.pdf

[6] ——, *Design Debugging Using the SignalTapII Logic Analyzer*, 2013. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/hb/qts/qts_qii53009.pdf

[7] ——, *Altera Transceiver PHY IP Core User Guide*, 2014. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/ug/xcvr_user_guide.pdf

[8] ——, *AN 584: Timing Closure Methodology for Advanced FPGA Design*, 2014. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/an/an584.pdf

[9] ——, *Cyclone V Device Handbook*, 2014. [Online]. Available: https://www.altera.com/en_US/pdfs/literature/hb/cyclone-v/cyclone5_handbook.pdf

[10] A. Athavale, *High-Speed Serial I/O Made Simple A Designers Guide, with FPGA Applications*, 2005. [Online]. Available: http://www.xilinx.com/publications/archives/books/serialio.pdf

[11] D. G. Bailey, *Design for Embedded Image Processing on FPGAs*, 2011. [Online]. Available: http://www.wiley.com//legacy/wileychi/bailey/

[12] M. Balch, *Complete Digital Design: A Comprehensive Guide to Digital Electronics and Computer System Architecture*, 2003.

[13] J. H. Choi, *High-Speed Devices and Circuits with THz Applications*, 2014. [Online]. Available: http://www.crcnetbase.com/isbn/9781466590120

[14] J. A. M. David Ricketts, *The Designer's Guide to Jitter in Ring Oscillators*, 2009. [Online]. Available: http://link.springer.com/book/10.1007%2F978-0-387-76528-0

[15] M. M. Dennis Derickson, *Digital Communications Test and Measurement: High-speed Physical Layer Characterization.* Prentice Hall, 2008.

[16] A. Dhir, *A Comprehensive Guide to Devices, Standards, Future Directions, and Programmable Logic Solutions*, 2004. [Online]. Available: http://www.sciencedirect.com/science/book/9780750678155

[17] M. Di Paolo Emilio, *Data Acquisition Systems : From Fundamentals to Applied Design.*, 2013. [Online]. Available: http://www.springer.com/gp/book/9781461442134

[18] U. H. P. Fischer-Hirchert, *Photonic Packaging Sourcebook: Fiber-Chip Coupling for Optical Components.*, 2015.

[19] I. Grout, *Digital Systems Design with FPGAs and CPLDs.* Newnes, 2008.

[20] F. R. John Watkinson, *Digital Interface Handbook.* Focal Press, 2003.

[21] W. P. Lane, "Rapidio: An embedded system component network architecture architecture and systems platforms motorola semiconductor product sector," 2000.

[22] C. Maxfield, *The Design Warrior's Guide to FPGAs*, 2004. [Online]. Available: http://www.sciencedirect.com/science/book/9780750676045

[23] ——, *FPGAs: Instant Access.* Newnes, 2008. [Online]. Available: http://www.sciencedirect.com/science/book/9780750689748

[24] R. S. Nancy Chan, "Method and apparatus for transporting and interoperating transition minimized differential signaling over differential serial communication transmitters," Patent US20 060 123 177 A1, 2006.

[25] M. S. Narula, "Giga bit per second Differential Scheme for High Speed Interconnect," *International Journal of VLSI Design & Communication Systems*, vol. 3, no. 1, pp. 35–50, 2012. [Online]. Available: http://airccse.org/journal/vlsi/papers/3112vlsics04.pdf

[26] M. Norris, *Gigabit Ethernet Technology and Applications*, Mark Norris. [Online]. Available: http://www.worldcat.org/title/gigabit-ethernet-technology-and-applications/oclc/475974092/editions?referer=di&editionsView=true

[27] K. P. Pahlavan Kaveh, *Principles of wireless networks : a unified approach.*, 2013.

[28] PCI Express, *PCI Express Base Specification*, 2010. [Online]. Available: http://komposter.com.ua/documents/PCI_Express_Base_Specification_Revision_3.0.pdf

[29] J. D. Rockrohr, A. Mohammad, C. R. Ogilvie, K. Dramstad, M. a. Sorna, J. T. Mechler, and D. R. Stauffer, *High Speed Serdes Devices and Applications*, 2009. [Online]. Available: http://www.springerlink.com/index/10.1007/978-0-387-79834-9

[30] C. Rowen, *Engineering the Complex SOC: Fast, Flexible Design with Configurable Processors.* Prentice Hall, 2004.

[31] E. Salminen, "On Design and Comparison of On-Chip Networks," Ph.D. dissertation, 2010. [Online]. Available: http://dspace.cc.tut.fi/dpub/bitstream/handle/123456789/6543/salminen.pdf?sequence=3&isAllowed=y

[32] C. Sharma, Rohit, *Compact Models and Measurement Techniques for High-Speed Interconnect*, 2009. [Online]. Available: http://link.springer.com/book/10.1007%2F978-1-4614-1071-3

[33] R. S. L. Shuangbao Paul Wang, *Computer Architecture and Security: Fundamentals of Designing Secure Computer Systems*, 2014. [Online]. Available: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-111816881X.html

[34] N. Storey, *Electronics: A Systems Approach.* Pearson Education, 2009.

[35] Terasic, *GPIO-HSTC Card User Manual*, 2011. [Online]. Available: http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=67&No=322&PartNo=3

[36] ——, *SocKit User Manual*, 2013. [Online]. Available: http://www.rocketboards.org/pub/Documentation/ArrowSoCKitEvaluationBoard/SoCKit_User_manual.pdf

[37] ——, *Terasic XTS Daughter Card User Manual*, 2013. [Online]. Available: http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=67&No=129&PartNo=3

[38] TI, *mini-LVDS Interface Specification*, 2003. [Online]. Available: http://www.ti.com/lit/an/slda007a/slda007a.pdf

[39] ——, "SN65LVDxx High-Speed Differential Line Drivers and Receivers," 2014. [Online]. Available: http://www.ti.com/lit/ds/symlink/sn65lvds2.pdf

[40] T. Wilmshurst, *Designing Embedded Systems with PIC Microcontrollers*, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780750667555500180