International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Design of Fixed One-Bit Latency Serdes Transceiver for High Speed Data Transmissions

Sangeetha N

Assistant Professor, Dr Ambedkar Institute of Technology, Bengaluru, Karnataka, India

Abstract: ***Today's communication world experiences a maximum amount of problems linked with serial interconnects since they occupy the entire communication field, therefore the serializer / deserializer (SerDes) devices make huge changes in the market with large differences in cost and performance. The serial interfaces are generally used for Transfer of information within chips and boards. The parallel interfaces were replaced by serial interconnects which were high speed as the bandwidth grew into multi-gigabit range. SerDes devices play an important role in exchanging information in this range. A SerDes device can compress the information, work in large bandwidth, and enables to exchange the information. It lowers the complexity associated with design, cost, and board space usage and signal strength compared with parallel connectors. The serial architecture in the present design depicts how the GTP transceiver which embeds the high speed SerDeses achieves a constant latency.***

Keywords: Changeable Delay Tuning, Dynamic clock phase shifting, fixed onebit latency, FPGA, Serializer / deserializer.

1. Introduction

Serializer/Deserializer devices embedded in the GTP transceivers though seem to be much advantageous in the field of communication for their high speed transfer capabilities, face much problems when they undergo processes like reset, relock and when powered up. In order to overcome these problems a solution in the form of designing an external dedicated circuitry was introduced. But this design was not actually needed for telecom and Datacom communications. However today's SerDes Devices replace the parallel connectors due to their high speed multi gigabit transfers because of the huge developments involved in bandwidths? Serdeses today are not only meant for their high speeds alone but show great improvements in device parameters like information formatting, device topology, protocol overheads etc. Serdeses are also associated with maintaining clocking, timing, latencies, buffering and logic which increased their cost and performance parameters linked to data acquisition and manipulation. The problem associated with the serdes chips is that they do not actually maintain the same latency after few operations like reset, power up and relock. Therefore a need for extra circuitry had to be implemented in the presently used transceivers in GTP in order to overcome the problem to improve the parameters linked with communication overhead. therefore an extra circuitry named as clock and data slider block is designed and implemented in the present work. The principle goal is layout of a fixed one bit latency serial transceiver based on delay tuning which is changeable and phase transfer of clock technology. As compared with the roulette approach it processes all clock phase offsets produced in serializing and parallelizing conversion. Subsequently, it removes out the reset-relock crisis.

Liu, Qing-Xu Deng, Ze-Ke Wang [1] stresses on usage of a rigid and fast serial transceiver designed on the principle of changeable delay tuning and dynamic clock phase shifting technologies. It facilitates to overcome all the problems posed by buffering and clocking mechanisms involved for transmissions with the development of phase offset values between the clocks of the transmitter and receiver. It also focuses in removing out the reset-relock problem. **A. Aloisio, F. Cevenini et al** [**2**] speaks approximately excessive-speed, constant-latency serial links in dispensed information acquisition and manipulation structures, inclusive of the timing trigger and control (TTC) device for excessive energy physics experiments. **R. Giordano and A. Aloisio et al [3]** describe how to make use of the inner alignment circuit of the SerDes transceivers to put off the clock phase offset, which results in a variation in connection latency. **A. Aloisio, F. Ameli, V. Bocci et al [4]** speaks approximately of utilizing source and data acquisition for transfer of data. It focuses on implementing a high speed transceiver in FPGAs with a clocking scheme and two configurations. It pressurizes on implementing pipelining mechanism for the serial link to improve the latency and performance. **J Adamczewski-Musch et al [5]** paper stress on a vast development in the field of radiation tolerance, limited area for hardware, and synchronization mechanism. CBM network protocol uses fiber connection which is bidirectional and single ended to achieve good latency and synchronization and thus helping to build a new network topology. **P. P. M. Jansweijer and H. Z. Peek [6]** reports on the examine-out device of the future KM3NeT undersea area of numerous synchronized optical detecting nodes. **F. Lemke, D. Slogsnat et al [7]** speaks about the Compressed Baryonic matter (CBM) experiment which were used For identifying the prototypes to improve performance in serial links. The DAQ software discusses about using various data inputs, optical connections reading through USB and Ethernet etc. **Jinhong Wang, Xueye Hu et al [8]** propose a detailed implementation of the fixed latency scheme, as well as simulations of the actual surroundings within the ATLAS forward muon region. **SCAN25100 [9]** tells about SCAN25100 that is a SerDes for high-velocity bidirectional serial data transmission over FR-4 revealed circuit board backplanes, balanced cables, and optical fiber. **Tlk2711 a 1. 6 to 2. 7 Gbps transceiver [10]** tells that TLK2711A is a member of multigigabit transceivers, meant for use in ultrahigh-velocity bidirectional part-to-part data transmission structures. The TLK2711A helps for a construction of powerful serial interface for high bandwidth. **R. J. Aliaga, J. M. Monzo et al [11]** inform that magnitude and control packages are more and more with the use of disbursed device technology. **Clifford E. Cummings and Peter Alfke [12]** the writers tell that An asynchronous FIFO refers to a FIFO design where all the data values are written into the FIFO buffer from one clock region and the data values are checked from similar FIFO buffer in another clock region and the clock regions are asynchronous to each other.

Serdes devices undergoing latency variations in existing technique

The Xilinx FPGA which embeds a GTX transceiver is used to explain the latency overheads. The simplified structure of this design has a Medium Attachment (PMA) sub layer and a physical Coding Sub layer (PCS). The PMA parallelizes the serial data and serializes the parallel data. PCS takes care of control operations like processing before serialization and after parallelization. The clock which is parallel XCLK, a reference clock for Clock and data recovery (CDR) circuit, and an input serial clock for parallel in /serial out block are generated from the PLL. The serial input/parallel output block needs RXRECCLK AND GETS IT FROM CDR circuit. The phase adjusts FIFO removes out variations in transmitter clock and elastic buffer removes out variations in receiver clock.

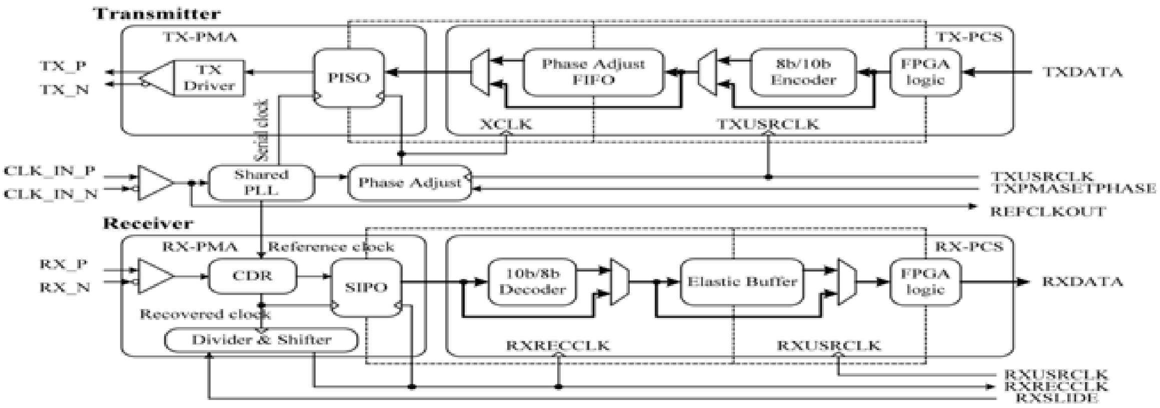

Figure 1: Xilinx FPGA embedding GTX transceiver

Delay problems may arise in GTX transceiver‘s PMA and PCS blocks the delay in PMA are because of division and multiplication of frequencies in serial links. CLK_IN is improved in frequency by a factor four which yields a serial clock. XCLK the parallel clock is same as CLK_IN. XCLK is used to obtain RXECCLK. The problem linked to delay is measured in unit interval UI, the serial symbol length.

The 20 bit information can be obtained by the 16 bit words due to serial exchange. The PLL present in the transmitter performs serial bit rate. The barrel shifter present in the receiver end aligns data parallely. A synchronization mechanism helps for the complete alignment of data in the barrel shifter. 1) The time difference value between any two points can be identified by sending a marking signal to and fro. 2) A time interval is initiated when the master point generates a "start" pulse and declined when the receiver in master point decodes the marking signal thus initiating "stop" pulse.

Proposed design structure

In the below mentioned constant latency transceiver a changeable delay tuning technology and dynamic phase shifting technology is used to design a clock and data sliding block to allow data and clock latencies become lessened.

Complete design architecture

A Design consisting of two GTX transceivers, one RX phase Align, one TX phase Align, one Payload Generator, and one Clock and data Slider (CDS). TXUSRCLK/ TXUSRCLK2 and XCLK donot have a phase coordination, therefore REFCLKOUT is connected to it. thus XCLK is aligned with REFCLKOUT with the help of TX phase align. this ends up with the bypass of phase adjust FIFO in the transmitter end. similarly the RXUSRCLK/RXUSRCLK2 is combined to RXRECCLK bypassing the elastic buffer in the receiver. Thus delays created due to the buffers is reduced.

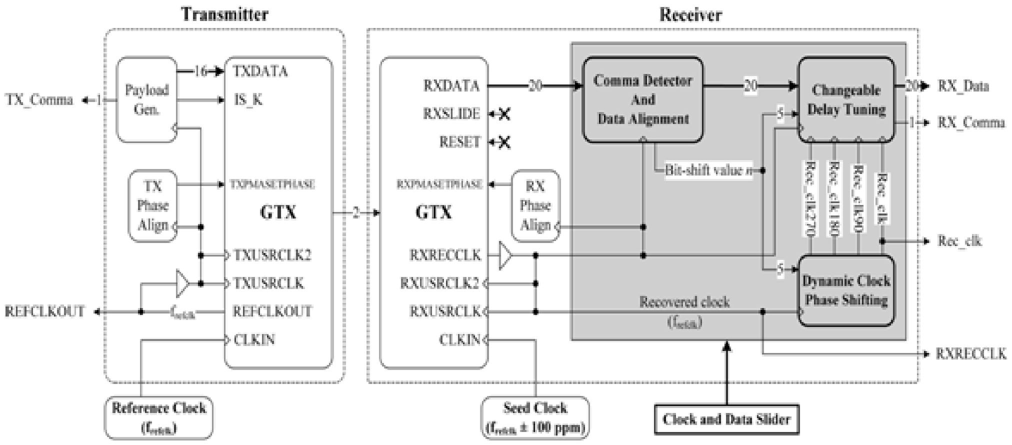

Figure 2: serial transceiver with constant latency

Clock and data Slider

The CDS consists of one Dynamic Clock phase shifting (DCPS) block, one Comma Detector and data Alignment (CDDA) block and one Changeable delay Tuning (CDT) block. Consistently the bit-shift value ‘n’ of RXDATA and the phase offset ΔP among RXRECCLK and the transmitted clock fulfill the following equation:

$$\Delta P = nx360^{\circ}/N \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 1$$

‘n’ value ranges from 0 to N-1

In which N is the internal data-path width. The K28. 5 symbol (one of 8 b/10 b control characters), within the parallel acquired data RXDATA is identified by the CDDA block, To decide the bit-shift value ‘n’.

Logic of Comma Detector and data Alignment

The simple block diagram of a comma detection and data alignment logic in fig. 3 may be depicted.comma indicators (0011111010 or 1100000101) help to transfer the data packets. Initially first 10 bit data is stored in the upper register (d10-d19), the upper register content material is transferred to the lower register (d0-d9), while new 10 bit information arrives as indicated through the shaded area. the comma detector checks only the content of 19 bit registers (d0-d18) to check the comma signals this is carried out to avoid two times comma signal check. d19 content is never checked.

The comma detector consists of ten comma Detection blocks (CDB), a ten registers, and 10 input OR gate to save the selected signals. Each CDB compares ten input data with the comma indicators (0011111010 or 1100000101). while one of the CDB detects a comma signal, the ten input OR gate sends a trigger signal to the ten registers which causes the comma detector to generate the select signal. If for instance the comma signal become that proven through the shaded region d[15:6], the pick out signal would be sel[6]. subsequent data may be extracted through the data alignment cellular a tri-state buffer from the selected registers d[6-15]. The comma detection and data alignment circuit may be operated within one clock at 500 MHZ and the latency of the data alignment logic is about 3 clocks.

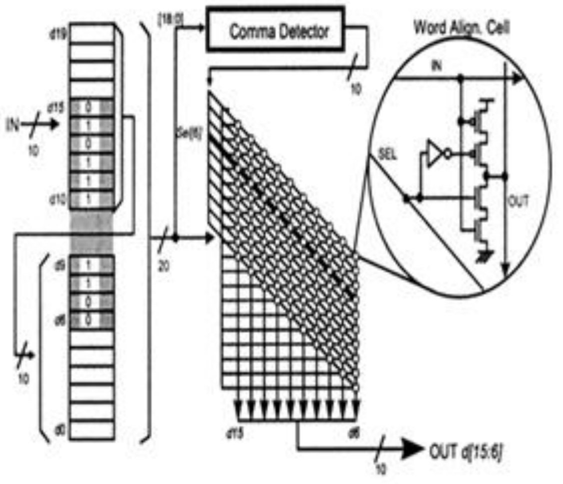

Figure 3: Block representation of comma detection and data alignment logic

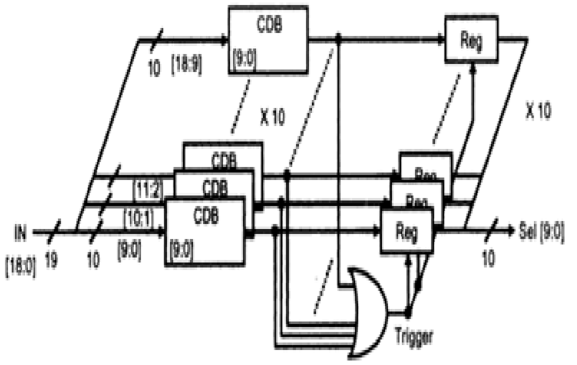

Figure 4: The block diagram of comma detector

Comma Detector and data Alignment block (CDDA)

The simplified structure of CDDA is proven to consist fractional bits of RXDATA and fractional values of RXDATA’s delay register, namely RXDATA_DLY. The bit-shift value ‘n’ can be extracted from RXDATA. Includes The bit- shift value ‘n’ comes to a decision to integrate the two data sources. Once the CDDA obtains the bit-shift value ‘n’, RXDATA_DLY [n-1:0], RXDATA [N-1:n] is selected as its result.

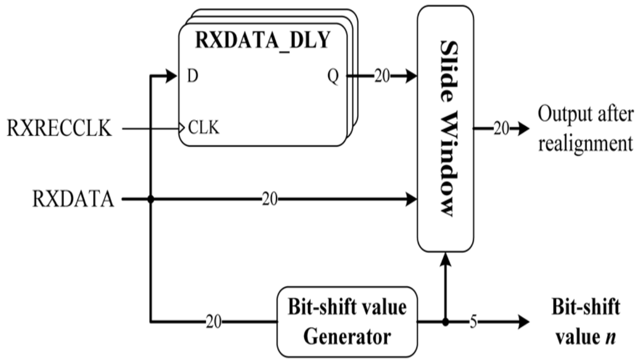

Figure 5: Pictorial representation of Comma Detector and data Alignment (CDDA) block

Dynamic Clock phase shifting (DCPS) block

A digital Clock manager (DCM), a phase-Locked Loop (PLL), and a phase Shift control Unit (PSCU) are used to design the DCPS block. The coarse and fine grained phase shift is obtained by DCM. In the “VARIABLE_POSITIVE” shift mode the DCM is used to operate in any of the 5 possible operating modes. The phase-shift value ‘P’ defines the equation given as:

$$P = p * tclk/256 \ldots\ldots\ldots\ldots\ldots\ldots\ldots 2$$

Where p is the integer parameter. DCM timing parameter gives the variety of the integer.

tclk is the frequency of input clock.

CLK180 output of the DCM enables to extend the phase-shift value While fine-grained phase varying is in evolution, since all the clock outputs of the DCM are adjusted.

Coarse-grained phase transferring is achieved by PLL. The PLL outputs CLK0, CLK90, CLK180, and CLK270 which are phase-shifted by using a quarter of clock input length relative to one another. With the aid of the PSEN, PSINCDEC, PSCLK, and PS- DONE ports, the phase Shift control Unit allows the DCM to execute the phase-shifting feature. two steps are considered for this operation:

1) It first resets the DCM and waits for it to relock. The initial rate of p is zero, so the phase shift value P is zero ns.
2) If the phase offset value is lesser than a 180° then, clk0 is selected because the input clock of PLL, and if its value is greater than 180° then, clk180 is selected as the input clock of PLL. CLK_SEL signal driven by way of the phase shift control unit selects the clocks amongst those clocks of PLL.

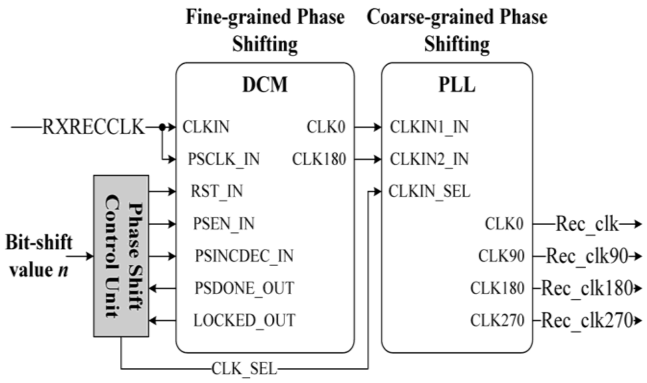

Figure 6: Dynamic Clock phase shifting (DCPS) block

Asynchronous FIFO

An asynchronous FIFO is a mechanism wherein Data from one clock domain to another clock domain are passed safely. The width and depth of the asynchronous FIFO can be selected accordingly by the designer. One clock signal helps the asynchronous FIFO to write information into the FIFO where as another clock signal helps to read the data from the FIFO.

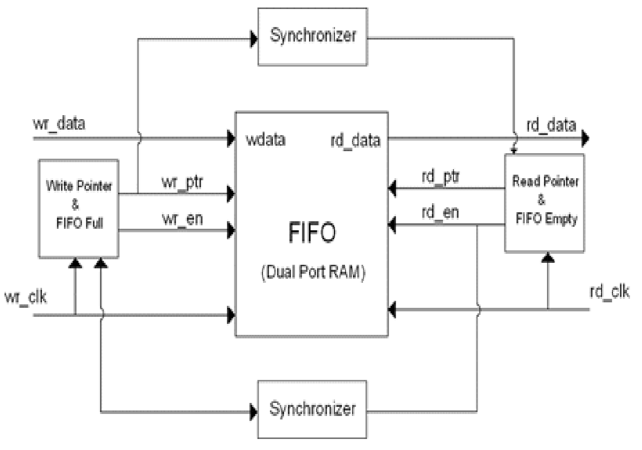

Figure 7: Illustrative figure for Asynchronous FIFO

A. Passing more than one signals Asynchronously

A problematic approach in an attempt to synchronize a pair of changing signals within clock domains is always not acceptable as it results in data transfers. Multi-bit information words are successfully transferred by FIFOs from one clock area to some other.

FIFO memory buffer array stores the data words by manipulating the signal in one clock area, and the data words are removed from another port of equal FIFO buffer with the help of Control signals. The decision making with respect to full and empty conditions in a FIFO memory design is the challenging task.

B. Asynchronous FIFO pointers

FIFO guidelines work as stated underneath. There are two pointers which carry out the entire pointing operations in the FIFO. Upon reset, both pointers are set to 0, which makes the write pointer to point towards the next FIFO word area. When a FIFO write signal is asserted the data shall be written to the location onto which the write pointer was pointing previously.

The read pointer always points to a region from where the next data has to be fetched. Upon reset again both pointers point towards zero. If in any case the FIFO is empty the read pointer will point towards an invalid data since it is an indication that the FIFO is empty and empty flag is set. As soon as the first information data is written into the FIFO, the write pointer increments, and the empty flag is cleared. Now the read pointer points towards the first data of the FIFO and when the read flag is set this data is fetched from the FIFO.

An important observation has to be made that when write signal is asserted the stack of the FIFO memory should be empty and when read signal is asserted The FIFO memory should be full.

Changeable delay tuning Block

The CDT block includes one asynchronous FIFO, one Comma Comparator, and one delay Tuning Unit, as shown. In the Comma Comparator, the parallel input data realigned with the help of CDDA block is compared whether it consists the comma values which are pre defined. If this comma comparator block obtains the comma values in the parallel data then, the comparator result is high. This high signal goes as a Write enable signal to the asynchronous FIFO block and also as the input signal to the first flip-flops of all the delay units. Upon high signal on write enable the parallel data queued is written into asynchronous FIFO. Now it is the choice of the multiplexer to select any one of the delay unit blocks based on phase offset value and bit shift value. once the multiplexer is done with its task, a flip flop stores the result as to which delay unit is activated. thus the parallel Data which was stored in FIFO will be read by device. At this time the comma out signal is high indicating the detection of comma word.

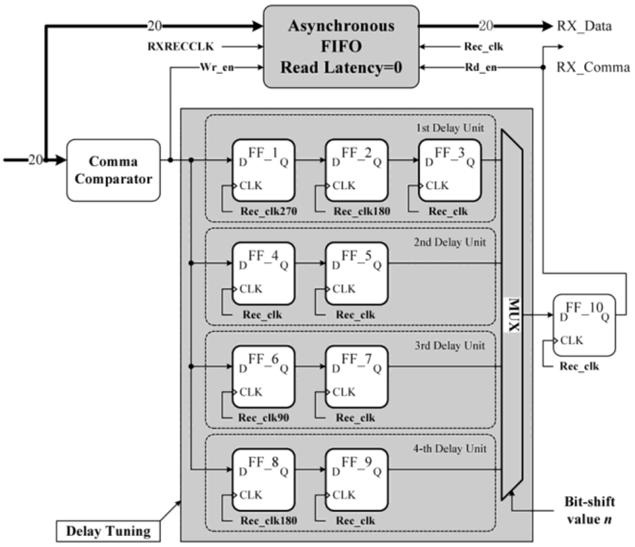

Figure 8: Changeable delay tuning Block

Detailed Operation of the delay tuning Block

The Changeable delay Tuning block gets the phase offset ΔP from the parallel received data stream. According to the phase offset ΔP, the DCPS block generates four multi-phase clocks: Rec_clk, Rec_clk90, Rec_clk180, and Rec_clk270. Here, the data-path width N of the GTX transceiver is set to 20, so there are 20 exclusive values for ΔP.

1. If ΔP is 342º, 324º, 306º, 288º, or 270º the first delay block is active and this output is selected as the result of the Changeable Delay Tuning circuit.
2. If ΔP is 252º, 234º, 216º, 198º, or 180º the second delay block is active and this output is selected as the result of the Changeable Delay Tuning circuit.
3. If ΔP is 162º, 144º, 126º, 108º or 90º the third delay block is active and this output is selected as the result of the Changeable Delay Tuning circuit.
4. If ΔP is 72º, 54º, 36º, 18º, or 0º the fourth delay block is active and this output is selected as the result of the Changeable Delay Tuning circuit.

2. Implementation and Results

The RTL Schematic of top module of main block illustrates the input and output requirements for the complete block.

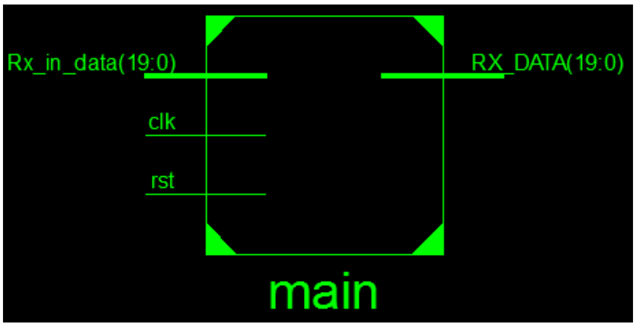

Figure 9: RTL Schematic of Top module of main block

The above RTL schematic clearly depicts that the entire unit depends on clk, rst and Rx_in_data which acts as input and RX_DATA acts as output. The input as well as the output is a stream of 20 bit information. The input data stream depend on clk and rst signals to get received at the receiver end. According to my design the input data stream is given at the comma detector and data alignment block, The data will be entering the sliding window and delay register as well. The bit shift value 'n' is responsible for combining both the data. This Data is realigned by the sliding window depending on the sliding window. At this stage only the data is aligned. This aligned Data enters the changeable delay tuning block.

The clocking for the entire system is obtained by the Dynamic clock phase shifting block, which consists of a phase shift control unit, a DCM and a PLL. The phase shift control unit is responsible to help the DCM to perform phase shifting function. Thus, the DCM generates two clks with phase shift values, clk0 and clk180. These two clocks enter as inputs to the PLL block which generates a coarse grained phase shifting to give out four output clocks clk0, clk90, clk180 and clk270 which are each phase shifted by a quarter of input clock period relative to each other. At this stage four different clocks are obtained which enter as input to the changeable delay tuning block along with the data obtained from CDT block.

In the CDT block, the 20 bit data stream identifies the presence of comma character and if comma is detected the data is queued at the asynchronous FIFO and a write enable signal is asserted. The output from the comma comparator makes the D flip flops to get activated depending on the respective clocks present at the delay tuning units. There are totally nine D flipflops at the delay tuning units, each getting activated at different phase shift values of clocks obtained from the PLL. These D flip flops give out the result to a 4:1 MUX which consists of bit shift value 'n' as its select line, to select the result from any one of the delay units. The output of MUX is given to another D flip flop which throws away a result indicating the presence of an comma out, and the same signal acts as a read enable signal to the asynchronous FIFO. Upon the activation of the clock the output is obtained at the asynchronous FIFO. This output is phase adjusted as well as synchronized. This is achieved because of the phase offsets which is different for the entire 20 bit data stream.

Simulation Results of CDDA Block

The CDDA block consists of RXRECCLK receiver recovered clock and 20 bit input data stream RXDATA as inputs. The output of this block are the bit shift value 'n' and the 20 bit realigned data. The realignment is done depending on the bit shift value.

The simulation results shown in fig. 10 depicts the RXRECCLK, RXDATA as input data and the output from this block is the 5 bit Bit_ shift_ val and the 20 bit out_align data. As shown in the result, the 5 bit result 'n' depicts the positioning of value 1 in the entire 20 bit data stream. The delta value indicates the phase offset value. for e.g. when the 20th bit is 1 in the entire 20 bit data stream, then the phase offset value is as calculated from formula 3 is 342°. Similarly when the 19th bit is 1 in the entire 20 bit data stream, then the phase offset value is as calculated from formula 3 is 324°. and this repeats for the entire 20 bit data stream and thus total of 20 different phase offset values are generated for 20 bit positions of input data. This phase offset value is helpful for the DCPS block to generate the four multi phase clocks to activate the CDT block.

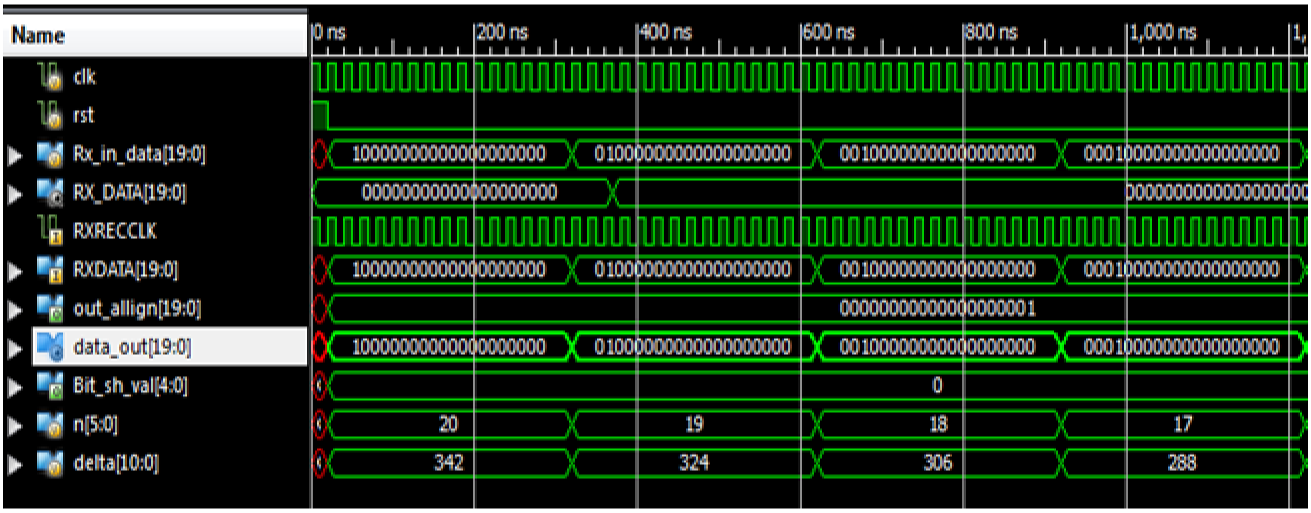

Figure 10: Simulation Results of CDDA block

DCPS Block Implementations

The PSCU block takes the input Bit_shift_val from the CDDA block, clk, PSDONE_OUT and LOCKED_OUT as input from DCM block and PSEN_IN, PSINCDEC_IN as output to the DCM block. This result states that the PSEN_EN signal should be high in order to give output to the DCM block.

Figure 11: Simulation Results of PSCU block

The PSCU block in fig. 11 takes the input Bit_shift_val from the CDDA block, clk, PSDONE_OUT and LOCKED_OUT as input from DCM block and PSEN_IN, PSINCDEC_IN as output to the DCM block. This result states that the PSEN_EN signal should be high in order to give output to the DCM block.

2. 1. Simulation Results of of DCM block

The DCM block as an IP consists of various inputs and output as in schematic, but for my design.

I concentrate on the output signals coming from PSCU block as inputs to DCM, and the two clock signals clk0 and clk180 are given out as output signals for the PLL block. The simulation results in fig. 12 show that whenever clock signal arrives and reset is low, and the PSDONE_OUT signal makes transition the outputs CLK0_OUT and CLK180_OUT which are 180° out of phase with each other emerge out as output from DCM block and arrive as input to PLL block.

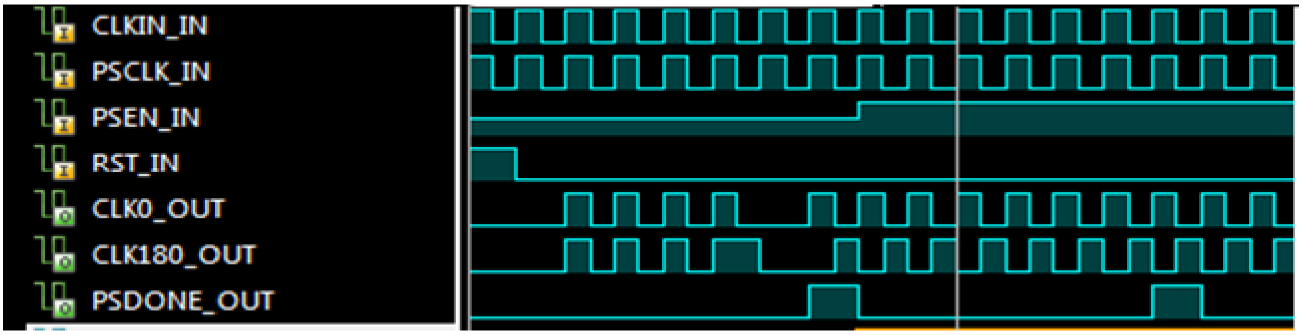

Figure 12: Simulation Results of DCM block

The RTL Schematic of PLL block indicate the two input signals CLKIN1_IN and CLKIN2_IN represent the signals CLK0 and CLK180 respectively. There are four multi phase clock outputs from this block each with a quarter phase shifts. CLKOUT0_OUT represent CLK0 i.e.

rec_clk, CLKOUT1_OUT represent CLK90 i.e. rec_clk90, CLKOUT2_OUT represent CLK180 i.e. rec_clk180, CLKOUT3_OUT represent CLK270 i.e. rec_clk270. These multi phase clock goes as input to the CDT block.

The simulation results of PLL block in fig. 13 indicate the two input signals CLKIN1_IN and CLKIN2_IN represent the signals CLK0 and CLK180 respectively. There are four multi phase clock outputs from this block each with a quarter phase shifts. CLKOUT0_OUT represent CLK0 i.e.

rec_clk, CLKOUT1_OUT represent CLK90 i.e. rec_clk90, CLKOUT2_OUT represent CLK180 i.e. rec_clk180, CLKOUT3_OUT represent CLK270 i.e. rec_clk270. These multi phase clock goes as input to the CDT block.

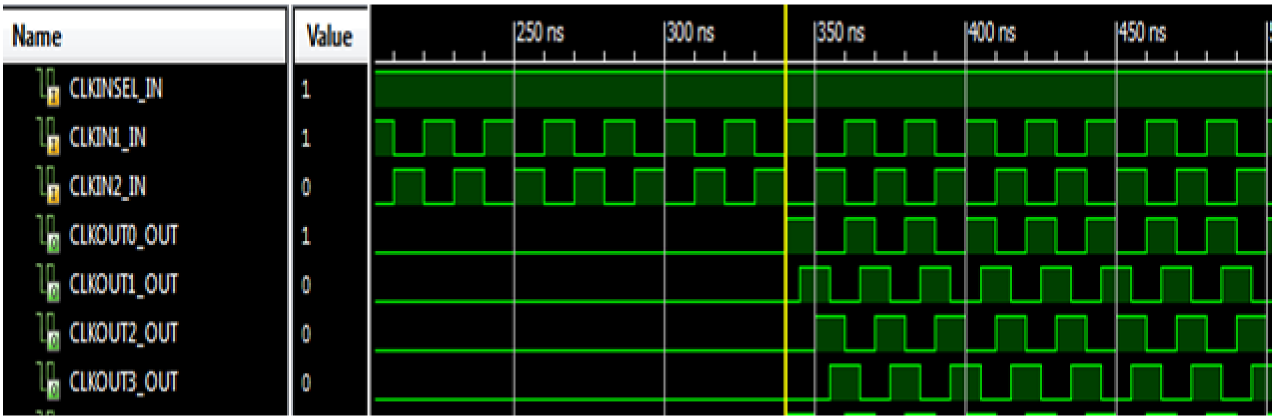

Figure 13: Simulation Results of PLL block

3. Changeable Delay Tuning Block

The Changeable Delay tuning block consists of four clock signals with 90° phase shifts each named as Rec_clk, Rec_clk90, Rec_clk180, and Rec_clk270, which are generated as outputs from the PLL. Another clock RXRECCLK is also an input clock. The five bit select line is nothing but the bit shift value which is necessary to select the various delay units with the help of a multiplexer. The 20 bit input data, data_in and the bit shift value are extracted from the CDDA block. From this we can infer that the clock alignment is achieved by the DCPS block and Data alignment is achieved by the CDDA block. The phase offset ΔP obtained from the CDDA Block is extracted from the parallel received data. The DCPS block generates four multiphase clocks depending on these phase offset values. Therefore for 20 bit input there are 20 different Phase offset values. These phase offset values are obtained from the formula,

$$\Delta P = nx360º/N, n = 0, 1 \ldots\ldots..N - 1$$

The above equation indicates that N is internal data path width of GTX transceiver and is 20 bit wide.

'n' is an integer ranging from 0 to 19. The CDT block consists of various components such as Delay tuning unit, comma comparator, asynchronous FIFO, multiplexer and D flip-flop. Initially the Parallel Data obtained from the CDDA block is queued at the asynchronous FIFO and comma comparator block. This data is compared with a constant predefined comma character. Whenever a high signal is obtained across the comma comparator output, the asynchronous FIFO receives it as Wr_en signal and the parallel data is stored in the FIFO. The four Delay units are activated depending on the phase offset values ΔP. Each delay unit can be selected for five different phase offset ts usage percentage details. It is proven that this design uses less amount of resource.

values. The multiplexer then with the help of its select line bit shift value 'n' selects any one delay units and gives it as output to the D flip-flop. The output of D flip-flop goes as the Rd_en signal to the asynchronous FIFO and upon input clock Rec_clk the output data is emerged out from FIFO.

3. 1 Simulation Results of CDT block

The simulation results in fig 14. show the detailed outcome of the entire design. Here we can observe that upon clock and low on reset signal the 20 bit parallel input data Rx_in_data is given to the CDDA block and the result of CDDA block is 20 bit output out_align. This block is also responsible for generating the 5 bit, bit shift value. The bit shift value is used as the select line from the multiplexer. From the result it can be inferred that for every five phase offset values the bit shift value changes. It is depicted from the result that for every change in position of '1' in the input parallel data stream, there is a different phase offset value generated. For example: if the 20th bit in the parallel input data stream is high then, the phase offset is 342°, if the 19th bit in the parallel input data stream is high then, the phase offset is 324°and so on. The above result shows the phase offsets upto 180° ie until the 11th bit is high. Depending on the multiphase clocks obtained from DCPS block and high signal on Rd_en, the output data is obtained across the asynchronous FIFO.

The simulation result in fig 15. shows the phase offsets from 162° upto 0° ie from the 10th bit to last bit becoming high. At this point we would obtain the output which shows the bit shift variation for each phase offset value.

3.2. Device utilization summary of the implementation

This summary gives a brief note of the available resources and used resources for the implementation

Table 7.1: Device utilization summary of the implementation

Device Utilization Summary (estimated values)				
Logic Utilization	**Used**	**Previous Design**	**Available**	**Utilization**
Number of Slice Registers	271	351	28800	0%
Number of Slice LUTs	269	493	28800	0%
Number of BUFG/BUFGCTRLs	6	11	32	18%
Number of DCM_ADVs	1	1	12	8%
Number of PLL_ADVs	1	1	6	16%

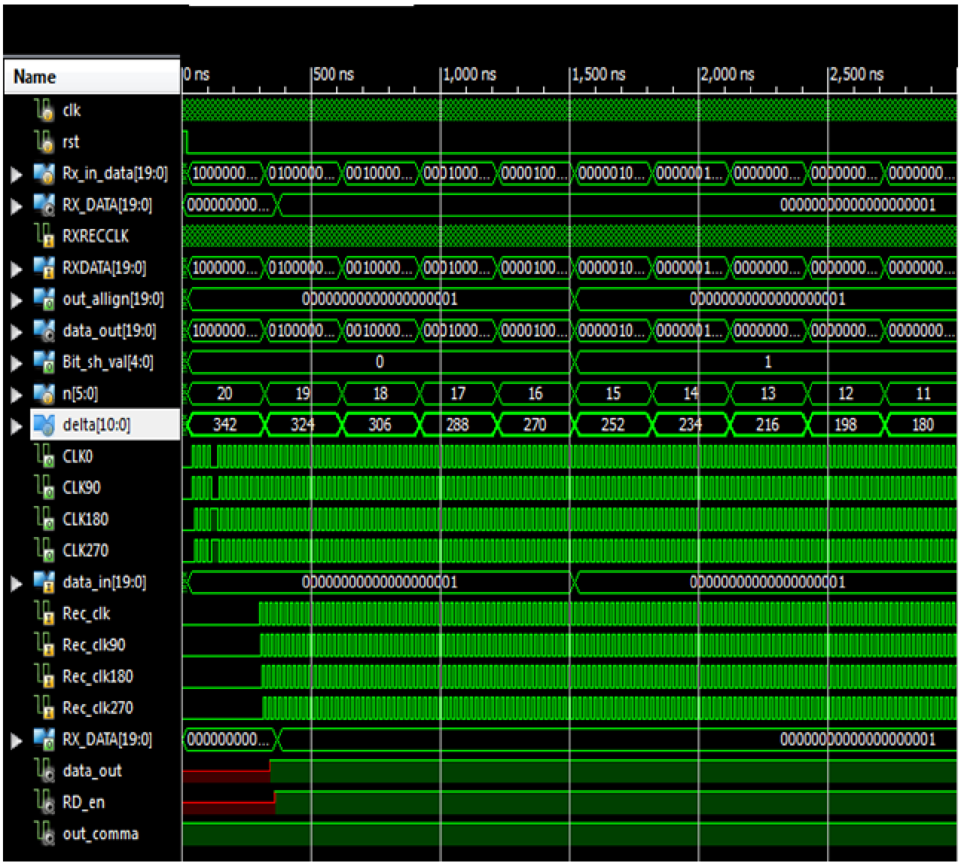

Figure 14: Simulation Results of CDT block for the phase offsets from 342° upto 180°

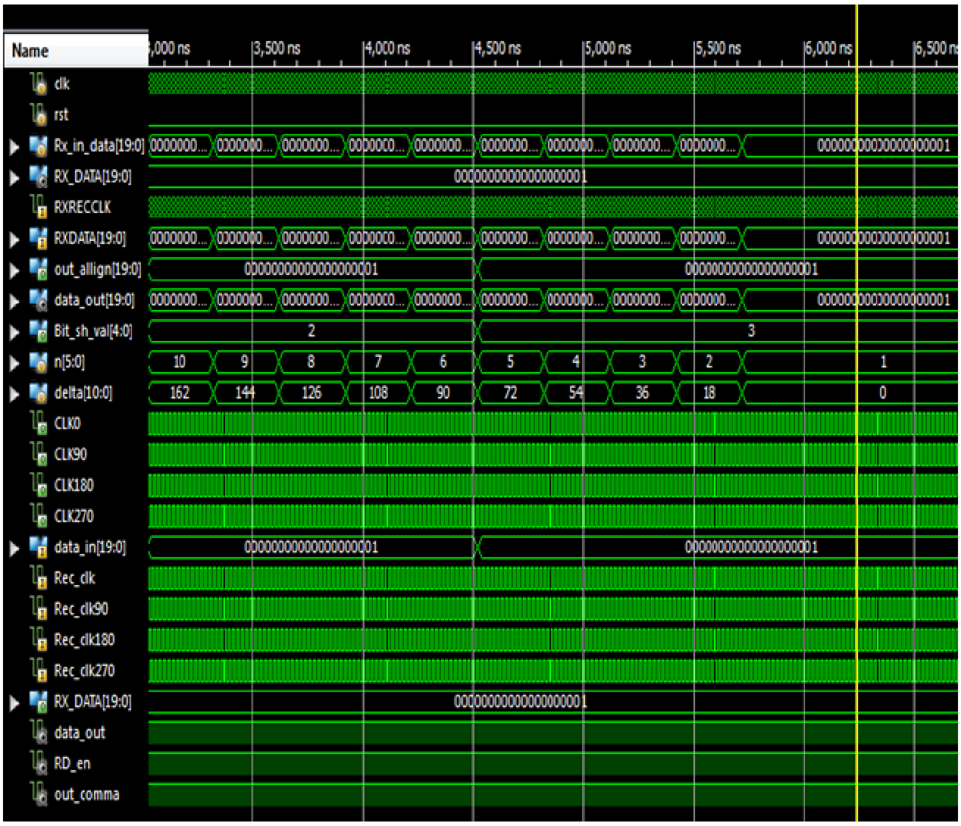

Figure 15: Simulation Results of CDT block for the phase offsets from 162° upto 0°

4. Conclusion and Future Scope

In this thesis, the clock and data slider block and its relative blocks has been successfully implemented and have been designed and implemented using Verilog HDL. Functional verification, synthesis, post synthesis simulation and static timing analysis were carried out using the Xilinx ISE design suite 14. 1, and the tests are carried out to check the result through an Emulator tool known as Chip scope Pro for an input signal of clock timing of 20ns. The performance metric is compared with the previous existing methods.

This thesis mainly focuses on the resource utilization. From this we can understand that the area, power and timing constraints are comparatively minimum from the existing architectures for the following modules.

- Comma detector and data alignment block
- Dynamic clock phase shifting block
- Changeable Delay Tuning block
- Complete integrated block

Almost all the GTX transceivers existing presently have overcome problems relating to phase synchronizations and clock timings. But there are still problems which have not found any solutions to beat the temperature variations, physical media dispersions and clock jitters etc. To design a rugged module which would overcome all these physical parameters and problems associated to latency changes is my future work.

References

[1] Xue Liu, Qing-Xu Deng, Ze-Ke Wang, "Design and FPGA Implementation of High-Speed, Fixed-Latency Serial Transceivers", IEEE Transactions on Nuclear Science, vol. 61, issue 1, pp. 561-56702/2014

[2] A. Aloisio, F. Cevenini, R. Giordano, and V. Izzo, "High-speed, fixed-latency serial links with FPGAs for synchronous transfers, " IEEE Trans. Nucl. Sci., vol. 56, no. 5, pp. 2864–2873, Oct. 2009.

[3] R. Giordano and A. Aloisio, "Fixed-latency, Multi-Gigabit serial links with Xilinx FPGAs, " IEEE Trans. Nucl. Sci., vol. 58, no. 1, pp. 194–201, Feb. 2011.

[4] A. Aloisio, F. Ameli, V. Bocci, M. Della Pietra, R. Giordano, and V. Izzo, "Design, implementation and test of the timing trigger and control receiver for the LHC, " J. Inst., vol. 8, no. 2, p. T02003, Feb. 2013.

[5] J Adamczewski-Musch, N Kurz, S Linev and P Zumbruch "Data acquisition and online monitoring software for CBM test beams", Published under licence by IOP Publishing Ltd, Journal of Physics: Conference Series, Volume 396, Part 1

[6] P. P. M. Jansweijer and H. Z. Peek, Measure Propagation Delay Over a 1. 25 Gbps Bidirectional Data Link, Tech. Rep. ETR 2010-01, 2010 [Online]. Available: http://www.nikhef. nl/pub/services/biblio/technical reports/ETR2010-01.pdf

[7] F. Lemke, D. Slogsnat, N. Burkhardt, and U. Bruening, "A Unified DAQ Interconnection network with precise time synchronization, " IEEE Trans. Nucl. Sci., vol. 57, no. 2, pp. 412–418, Apr. 2010.

[8] Jinhong Wang, Xueye Hu, Thomas Schwarz, Junjie Zhu, J. W. Chapman, Tiesheng Dai, and Bing Zhou, "FPGA Implementation of a Fixed Latency Scheme in a Signal Packet Router for the Upgrade of ATLAS Forward Muon Trigger Electronics, " IEEE Trans. Nucl. Sci., vol. 57, no. 2, pp. 467–471, Apr. 2010.

[9] SCAN25100 24576, 1228. 8, and 614. 4 Mbps CPRI SerDes with Auto RE Sync and Precision Delay Calibration Measurement, Texas Instru- ments, 2013 [Online]. Available: http://www.ti.com. cn/product/cn/ scan25100.

[10] TLK2711 A 1. 6 TO 2. 7 GBPS TRANSCEIVER, Texas Instruments, 2012 [Online]. Available: http://www.ti.com. cn/product/cn/tlk2711a

[11] R. J. Aliaga, J. M. Monzo, M. Spaggiari, N. Ferrando, R. Gadea, and R. J. Colom, "PET system synchronization and timing resolution using high-speed data links, " IEEE Trans. Nucl. Sci., vol. 58, no. 4, pp. 1596–1605, Aug. 2011.

[12] Clifford E. Cummings and Peter Alfke, "Simulation and Synthesis Techniques for Asynchronous FIFO Design with Asynchronous Pointer Comparisons, " SNUG 2002 (Synopsys Users Group Conference, San Jose, CA, 2002) User Papers, March 2002, Section TB2, 3rd paper.

[13] Charles Rajesh Kumar. J, Vanchinathan. T, Kharthik. K, "Design of SerDes Transceiver with fixed and high throughput implementation on FPGA", Life Science Journal 2013;10(2):2849-2857] (ISSN:1097-8135). http://www.lifesciencesite.com. 394

[14] A. Aloisio, F. Cevenini, R. Giordano, and V. Izzo, "Emulating the GLink chip set with FPGA serial transceivers in the ATLAS level-1 Muon trigger, " IEEE Trans. Nucl. Sci., vol. 57, no. 2, pp. 467–471, Apr. 2010

[15] White Rabbit [Online]. Available: http://ohwr. org/projects/white- rabbit/documents

[16] A. X. Widmer and P. A. Franaszek, "Byte Oriented DC Balanced (0, 4) 8B/10B Partitioned Block Transmission Code," U. S. Patent 4 486 739, Dec. 4, 1984.

[17] DS92LV18 18-bit Bus LVDS Serializer/Deserializer - 15–66 MHz, Texas Instruments, 2006 [Online]. Available: http://www.ti.com. cn/ product/cn/ds92lv18

[18] A. Aloisio, F. Ameli, V. Bocci, M. Della Pietra, R. Giordano, and V. Izzo, "Design, implementation and test of the timing trigger and control receiver for the LHC, " J. Inst., vol. 8, no. 2, p. T02003, Feb. 2013.

[19] ML505/ML506/ML507 Evaluation Platform User Guide, UG347, v3. 1. 2, Xilinx, 2011[Online]. Available: http://www.xilinx.com/support/documentation/boards_and_kits/ug347.pdf

[20] A. Aloisio, F. Cevenini, R. Giordano, and V. Izzo, "Characterizing jitter performance of multi gigabit FPGA-embedded serial transceivers, "IEEE Trans. Nucl. Sci., vol. 57, no. 2, pp. 451–455, Apr. 2010