

Design and FPGA Implementation of PLL-based Quarter-rate Clock and Data Recovery Circuit

Mohammed H. Alser¹, Maher Assaad², Fawnizu Azmadi Hussin³, and Israel Yohannes⁴

Department of Electrical and Electronic Engineering

University Technology of PETRONAS, 31750 Tronoh, Perak, Malaysia

{¹mohammed.hk_g01558, ⁴israely_g01601}@utp.edu.my, {²maher_assaad, ³fawnizu}@petronas.com.my

Abstract—This paper overviews the increased dynamic power consumption issue associated with the use of the serial links as a medium of data communication in today's multi-module based system-on-chip (SoC) and presents a novel all-digital PLL-based quarter-rate clock and data recovery circuit as a potential solution. The proposed architecture works at a frequency equal to one-fourth the received data rate and utilizes a quarter-rate early-late type phase detector, a delay line, a delay line controller, and a digitally controlled oscillator (DCO)-based 8-phases generator. The proposed architecture can be adapted easily for different FPGA families, as well as implemented as an integrated circuit. Moreover, it can be used in a deserializer as part of a SERDES in inter-module communication in SoC. The proposed architecture is designed using the Verilog language and synthesized for the Altera DE2-70 development board. Furthermore, the simulation results validate the expected functionality, such as performing quarter-rate phase detection as well as 1-to-4 demultiplexing. The synthesized design requires 117 logic elements using the above Altera board.

Keywords- system-on-chip (SoC); clock and data recovery (CDR); quarter-rate phase detector; serializer; deserializer.

I. INTRODUCTION

TECHNOLOGY SCALING has dramatically increased the number of modules that can be incorporated into a single chip to hundreds. These modules typically need to communicate with each other by on-chip communication architecture. Furthermore, as the processing speed of the chip rises, the demand for higher communication bandwidth between chips as well as modules also grows.

Typically, point-to-point parallel links are used for overcoming low communication bandwidth by increasing the data rate per channel and integrating a large number of channels into the system. On the contrary, increasing bandwidth to some extent accentuates several essential problems including timing skew issue and crosstalk between adjacent channels. The aforesaid problems cause distortion of data and place an upper limit on the length of a parallel data transmission. Moreover, improving the bandwidth by duplicating the channels in large numbers occupies a large area of silicon and it is uneconomical and impractical [1].

However, a promising solution to the parallel link problems is to replace it by a point-to-point serial link. The serial link has the ability to fulfill the strongly emerging need for a higher communication bandwidth, more efficient data transmission, and longer distance reach.

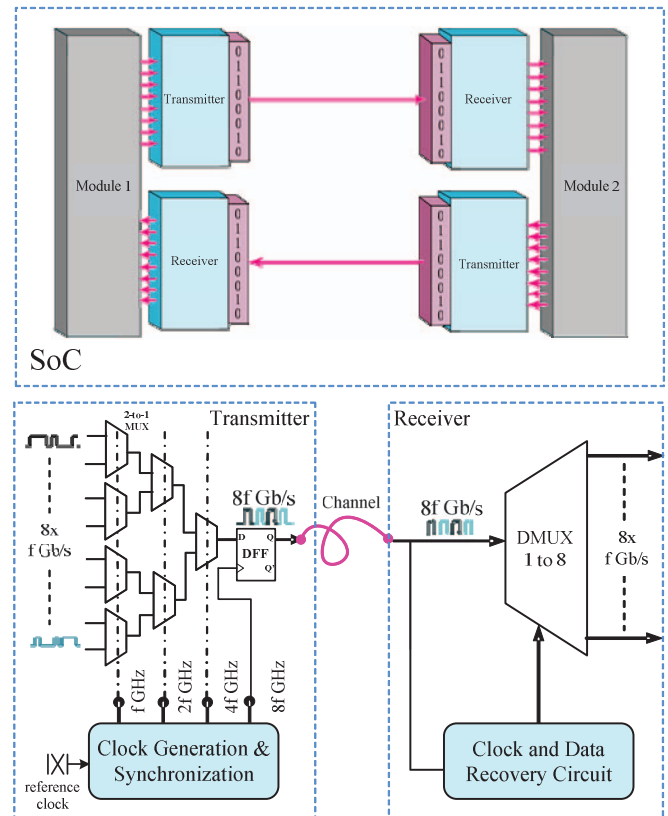


Figure 1. Inter module communication system with a simplified top-level block diagram of a point-to-point serial link.

The serial link system typically comprises, beside the channel, two main circuits [2], namely the serializer (i.e. transmitter) and the deserializer (i.e. receiver), as shown in Fig. 1. First, the serializer starts normally with converting the low-frequency parallel data streams into a single higher frequency serial data stream. An internal clock multiplication and synchronization circuit is required to clock the multiplexers used in transmitter circuit. It generates four multiples of the reference clock frequency that are synchronized to the reference clock signal. Second, the high frequency serialized data is transmitted through the channel to the receiver circuit. Meanwhile, the serializer and the receiver must be synchronized to reliably access transmitted data. Therefore finally, since the clock is embedded in the data stream, the receiver uses a clock and data recovery circuit (CDR) to extract the clock signal from the received

data and demultiplexes the data by the recovered clock, to become a low-frequency parallel data streams again.

Unfortunately, it was noticed that serializing parallel data streams into a single channel increases the overall switching activities and hence increases the dynamic power consumption [3]. As a result, the consumed power is lost in the form of heat that reduces the efficiency of the system.

However, as shown in Fig. 2, the factors affecting the dynamic power consumption issue of the serial link system can be figured out as follow:

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

$$Q = CV_{dd}$$

The current flow across the channel can be obtained as:

$$I = Q/T = CV_{dd}F$$

Where T is the time required to charge and discharge the capacitor. Then the factors affecting the dynamic power consumption can be broken down as follow:

$$P_{dyn} = IV_{dd} = CV_{dd}^2 F \quad (1)$$

Where C is the channel capacitance being switched per clock cycle, V_{dd} is the voltage swing and it is a technology dependent (i.e. for CMOS 180-nm and CMOS 130-nm, $V_{dd} = 1.8$ V, 1.3 V, respectively), and F is the switching frequency. Consequently, the only practical and realistic way to reduce the dynamic power consumption is reducing the switching frequency without affecting negatively the high communication bandwidth.

However, the power consumption in high-speed point-to-point serial links is the most crucial problem today [4]. Considering the advantages of the serial link, increasing the number of links per chip is possible only if the power consumption per link reduces. Moreover, an analytical study [5] has been conducted by S. Ardalan, shows that the receiver circuit is power-hungry element whereas it consumes approximately 60% of the transceiver total power consumption. Furthermore, it shows that CDR circuit dissipates around 30% of the total power consumption, which makes the CDR the most power-consuming block among other blocks of the serial data link.

II. CLOCK AND DATA RECOVERY OPERATION OVERVIEW

CDR circuit is a critical function in high-speed transceivers. As mentioned in section I, the received data stream is both asynchronous and noisy, requiring that a clock be extracted to allow synchronous operations. Furthermore, the data must be retimed such that the jitter accumulated during transmission is removed.

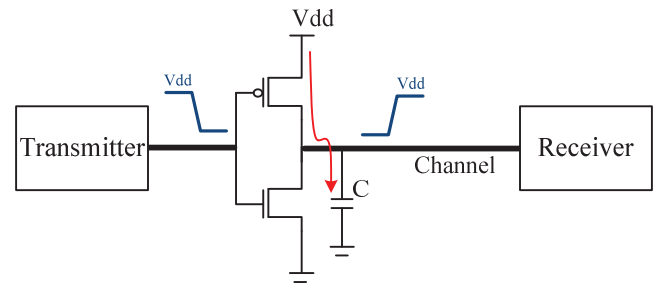


Figure 2. A simple CMOS inverter driving a capacitive external load (the channel of the point-to-point serial link system).

In a CDR circuit, a key component called the phase detector provides phase information between the received data stream and the internally generated clock signal. Typically CDR circuit generates a clock signal internally that has a frequency equal to the data rate. However, reducing the dynamic power consumption requires reducing the frequency of the internally generated clock signal without affecting negatively the data rate. The challenging idea is the use of multiple phases of a clock running at a frequency less than the data rate in the operation of the phase detector rather than using a single high-frequency clock signal.

Consequently, we propose an all-digital quarter-rate CDR circuit using a novel quarter-rate phase detector for inter-module communication in SoC. A quarter-rate CDR circuit will operate using eight phases of a clock running at a frequency equal to one-fourth the data rate. As a result, the proposed architecture reduces the switching frequency to one-fourth, and hence the total dynamic power consumption is reduced to one-fourth, which will eventually lead to the implementation of a low-power receiver circuit and hence low-power and high-speed point-to-point serial link. The implemented architecture requires no analog components and can be easily adapted for different FPGA families as well as implemented as an integrated circuit.

III. CIRCUIT DESIGN AND IMPLEMENTATION

The proposed architecture is based on performing quarter-rate phase detection as well as 1-to-4 demultiplexing (DEMUX). As shown in Fig. 3, the basic operation of the proposed quarter-rate CDR circuit requires four important building blocks, namely the digitally controlled oscillator (DCO)-based 8-phases generator, novel quarter-rate phase detector, delay line controller, and digitally controlled delay line.

A. DCO-Based 8-Phases Generator

Since we are using a quarter-rate based CDR topology, the input data rate should be four times the frequency of any of the output phases of the 8-phases generator. For proper operation of the phase detector, eight phases and their complements are required, in which they are 22.5° out of phase. However, the proposed quarter-rate CDR circuit will be synthesized for the Altera DE2-70 development board that includes cyclone II EP2C35F672C6 FPGA on board.

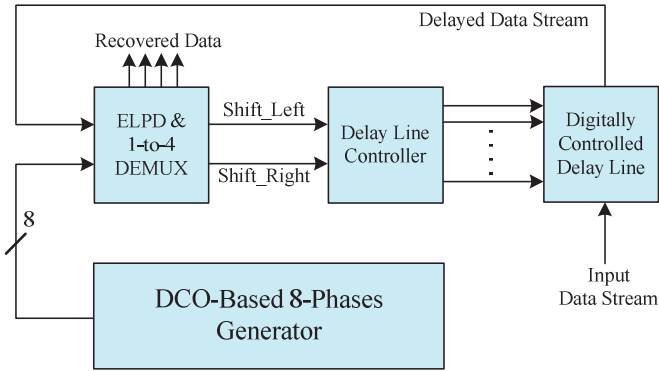


Figure 3. The proposed CDR block diagram.

Unfortunately, cyclone II PLL has a clock shift capability that enables programmable phase shifts in increments of at least 45° [6]. Moreover, it has only three output phases [7]. For the reasons mentioned above, cyclone II PLL properties do not meet the requirements of the proposed phase detector. Consequently, the eight phases required for the phase detector are realized using a finite state machine. As illustrated in Table I, the finite state machine has 16 states and each state has its own predefined 8-bits digital word. However, the finite state machine starts its operation when an enable signal is asserted and initializes the current state to be State 0. The state transitions are governed by the positive edge of the output clock (F_{DCO}) of the DCO. A digitally controlled oscillator previously designed and implemented by [2] is used in the proposed 8-phases generator design. However, the aforesaid DCO consists of two main blocks: ring oscillator and fractional divider. On the one hand, the ring oscillator consists of one NAND gate which enables/disables the oscillation and a chain of AND-OR delay elements. The ring oscillator produces a clock signal whose frequency is proportional to the number of delay elements in the ring. Consequently, reducing the number of delay elements in the ring gives higher frequency and vice versa. Moreover, changing the ring oscillator chain length via a one-hot coded word provides a coarse frequency resolution.

On the other hand, the fractional divider comprises an adder-accumulator. The MSB of the accumulator signed register is used to switch the input of the adder between signed integer number and its two's complement. The fractional divider is also used to switch between two adjacent ring oscillator chain lengths. Switching between two adjacent frequencies provides on average fine frequency resolution. The generated clock frequency of up to 440 MHz with a frequency step of 0.158MHz is used to clock the finite state machine.

Accordingly, Fig. 4(a) shows the input F_{DCO} signal and the eight phases (differing by 22.5°) to demonstrate the operation of the DCO-based 8-phases generator. As an overall trend, the proposed 8-phases generator suffers from a potential problem which is output glitches due to the finite transition time. However, an enhancement is added to the proposed 8-phases generator in order to eliminate the output glitches by retiming the eight output phases using a single D flip-flop for each output phase.

TABLE I
THE STATES OF THE 8-PHASES GENERATOR AND THEIR CORRESPONDING CONTENT

Current State	8 Output Clocks	Next Transition
	1 2 3 4 5 6 7 8	
State 0	0 0 0 0 0 0 0 0	State 1
State 1	1 0 0 0 0 0 0 0	State 2
State 2	1 1 0 0 0 0 0 0	State 3
State 3	1 1 1 0 0 0 0 0	State 4
State 4	1 1 1 1 0 0 0 0	State 5
State 5	1 1 1 1 1 0 0 0	State 6
State 6	1 1 1 1 1 1 0 0	State 7
State 7	1 1 1 1 1 1 1 0	State 8
State 8	1 1 1 1 1 1 1 1	State 9
State 9	0 1 1 1 1 1 1 1	State 10
State 10	0 0 1 1 1 1 1 1	State 11
State 11	0 0 0 1 1 1 1 1	State 12
State 12	0 0 0 0 1 1 1 1	State 13
State 13	0 0 0 0 0 1 1 1	State 14
State 14	0 0 0 0 0 0 1 1	State 15
State 15	0 0 0 0 0 0 0 1	State 0

This D flip-flop is clocked by the same input clock of the finite state machine, F_{DCO} signal. As shown in Fig. 4(b), all the eight output phases become glitch-free after the enhancement is added.

B. Phase Detector (PD)

The proposed phase detector is an early-late quarter-rate Alexander-based design [8]. As shown in Fig. 5, the proposed phase detector samples the input data stream at 0° , 45° , 90° , 135° , 180° , 225° , 270° , and 315° of the clock phases, producing the eight signals D_0 , D_{45} , D_{90} , D_{135} , D_{180} , D_{225} , D_{270} , and D_{315} at the D flip-flop outputs.

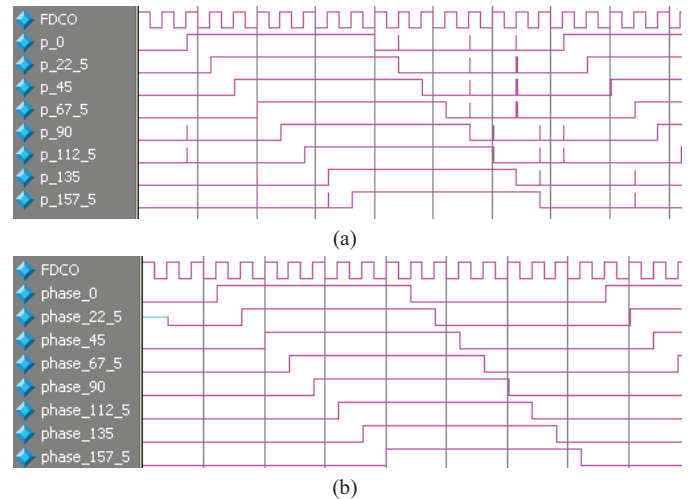


Figure 4. The gate-level simulation waveforms of the proposed (a) 8-phases generator (b) Enhanced 8-phases generator.

The last eight signals are used to generate the *shift_right* and *shift_left* signals that indicate the relative clock edge positions with respect to the data edges. The required logic to produce the *shift_right* and *shift_left* signals is as follow:

$$\text{shift_right} = (D_0 \oplus D_{45}) + (D_{90} \oplus D_{135}) + (D_{180} \oplus D_{225}) + (D_{270} \oplus D_{315})$$

$$\text{shift_left} = (D_{45} \oplus D_{90}) + (D_{135} \oplus D_{180}) + (D_{225} \oplus D_{270}) + (D_{315} \oplus D_0)$$

However, when the phase detector is in the locked state as illustrated in Fig.6 (a), the edges of the half-quadrature phases (Φ_{45} , Φ_{135} , Φ_{225} , and Φ_{315}) are aligned with the data transitions, and hence D_0 , D_{90} , D_{180} , and D_{270} will be the recovered and retimed data.

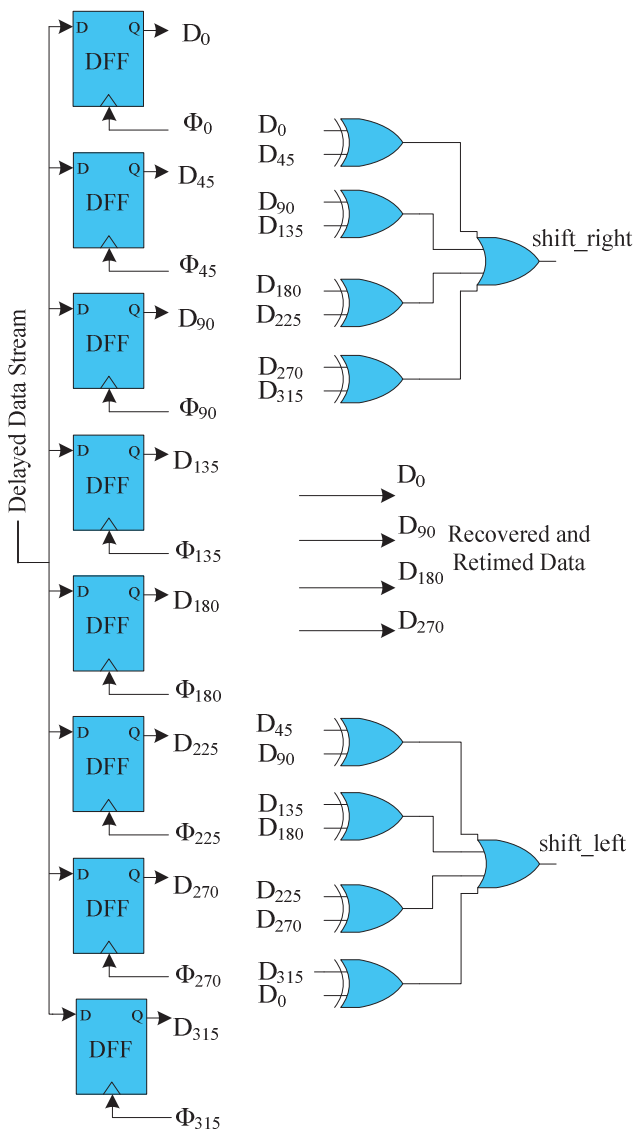


Figure 5. Block diagram of the proposed quarter-rate phase detector.

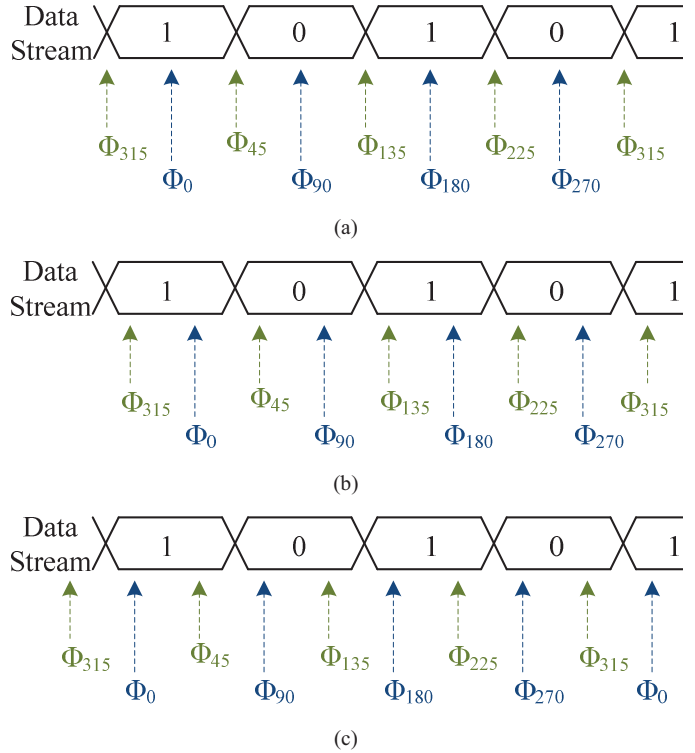


Figure 6. Detecting conditions of the proposed quarter-rate phase detector (a) Locked state (b) Late state (*shift_right* is high) (c) Early state (*shift_left* is high).

In contrast as illustrated in Fig.6 (b), when the data transition is situated between D_0 and D_{45} , the phase detector will generate a *shift_right* signal. The *shift_right* signal indicates a late state and the delay of the data stream need to be reduced. When the data transition is situated between D_{45} and D_{90} , the phase detector will generate a *shift_left* signal, which indicates an early state and the delay of the data stream need to be increased as illustrated in Fig.6 (c).

C. Delay Line Controller

Avoiding the jitter accumulation issue requires a stable controller. Consequently, a linear controller is utilized in the CDR architecture. The delay line controller is responsible for controlling the length of the delay line chain via one-hot coded word. This coded word is generated based on the received *shift_right* or *shift_left* signal from the phase detector. For each decision, the delay line controller updates the output coded word, and hence changing the output delay of the delay line.

TABLE II
DELAY LINE CONTROLLER OPERATIONS

<i>shift_right</i>	<i>shift_left</i>	Action
High	Low	Late state: shift the data stream to the right by decreasing the delay of the delay line.
Low	High	Early state: shift the data stream to the left by increasing the delay of the delay line.
High	High	Locked State: No Action.

Low	Low	No Action.
-----	-----	------------

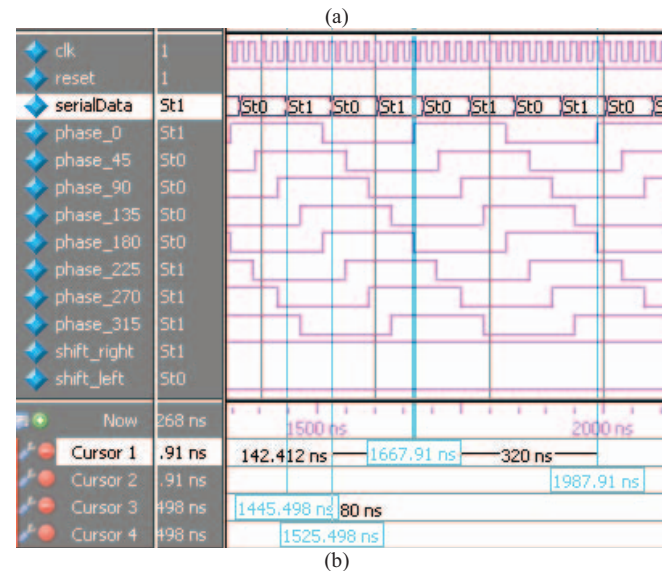
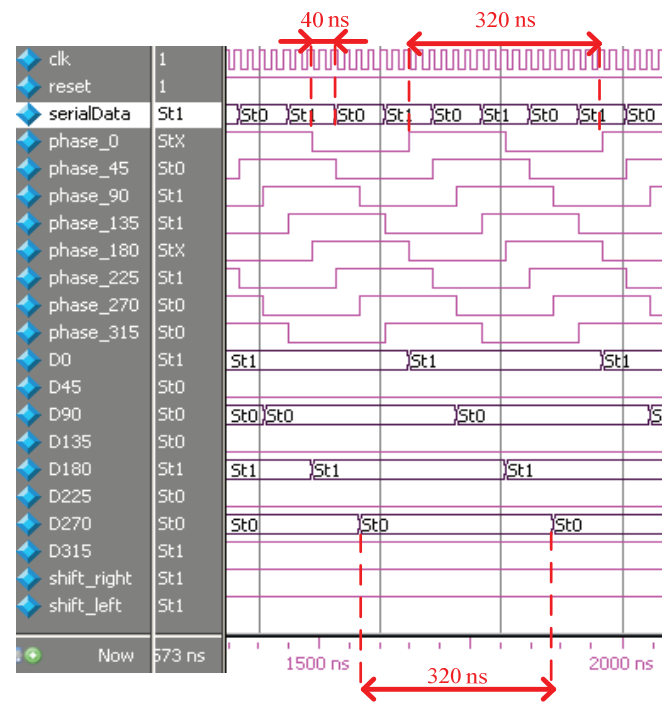
As illustrated in Table II, A shift_right signal decreases the number of delay elements, and thus decreases the delay of the input data stream. While receiving a shift_left signal increases the number of delay elements, and thus increases the delay of the input data stream.

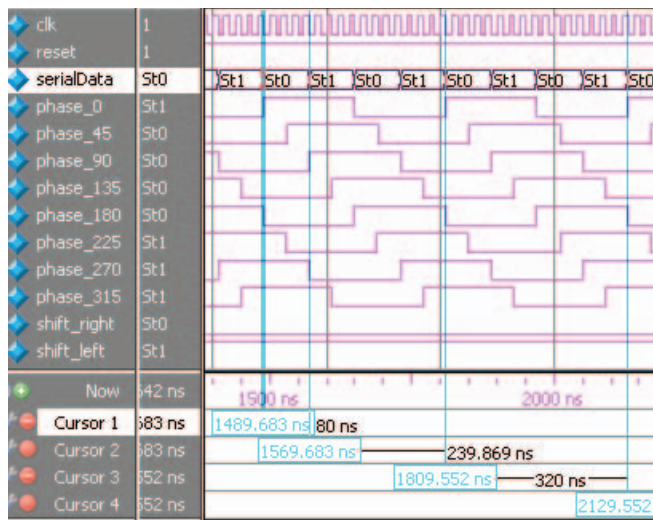
D. Digitally Controlled Delay Line

The success of the CDR operations is based on the presence of a linear relationship between the delay line controller output and the output delay of the delay line, thus a chain of linear delay elements [9] is employed in the structure of the delay line. Each delay element consists of three NAND gates. One of these three gates is used to allow the input data stream to pass through the chain via a one-hot coded word, while the other two gates are used to generate a delay. Consequently, the output signal of the delay line passes an odd number of NAND gates. Therefore, an additional NAND gate is added to the output signal of the delay line to avoid inverting the signal. However, the delayed version of the input data stream is connected to the input of the proposed phase detector.

IV. SIMULATION RESULTS

The overall architecture for the proposed PLL-based quarter-rate clock and data recovery circuit is completely realized as digital circuits. Consequently, the four building blocks mentioned in section III are designed using Verilog-HDL code and synthesized using Altera Quartus II Web Edition v11.0 software for Altera DE2-70 development board, with a Cyclone II EP2C35F672C6 FPGA on board. The DE2-70 board is equipped with 68416 logic elements (LEs). Using Quartus II software, the proposed architecture size is 117 LEs, which is less than 1% of the total number of logic elements.





(c)

Figure 7. The gate-level simulation waveforms using ModelSim-Altera: (a) Locked state (both shift_right and shift_left are high) (b) Late state (shift_right is high) (c) Early state (shift_left is high)

However, as illustrated in Fig. 7 (b), the shift_right signal indicates that the data transition is situated between D_0 and D_{45} . As a result, the delay line controller will decrease the delay of the delay line. In Fig. 7 (c), the shift_left signal indicates that the data transition is situated between D_{45} and D_{90} , and hence the controller will increase the delay of the data stream. After adjusting the delay of the data stream through passing a specific number of delay elements of the delay line, the CDR becomes in locked state, as clearly shown in Fig. 7 (a). The edges of the half-quadrature phases (Φ_{45} , Φ_{135} , Φ_{225} , and Φ_{315}) are aligned with the data transitions, and hence D_0 , D_{90} , D_{180} , and D_{270} will be the recovered and retimed data. It is clear that the data rate of the recovered data is equal to the frequency of the output phases of the 8-phases generator as well as synchronized with the corresponding phase.

V. CONCLUSIONS

This paper presents a novel all-digital PLL-based quarter-rate clock and data recovery circuit, which includes a quarter-rate phase detector, a delay line, a delay line controller, and a digitally controlled oscillator (DCO)-based 8-phases generator. The proposed architecture can be adapted easily for different FPGA families, as well as implemented as an integrated circuit. Moreover, it can be used in a deserializer as part of a SERDES in inter-module communication in system-on-chip (SoC). Furthermore, the simulation results are included and they validate the expected functionality and properties, such as performing quarter-rate phase detection as well as 1-to-4 demultiplexing. The synthesized design requires 117 logic elements using the above Altera board.

REFERENCES

- [1] M. H. Alser and M. M. Assaad, "Design and modeling of low-power clockless serial link for data communication systems," in *National Postgraduate Conference (NPC 2011)*, pp. 1-5, 2011.
- [2] M. Assaad and M. Alser, "An FPGA-based design and implementation of an all-digital serializer for inter module communication in SoC," *IEICE Electronics Express*, vol. 8, pp. 2017-2023, 2011.
- [3] M. Ghoneima, Y. Ismail, M. Khellah, and D. Vivek, "Reducing the data switching activity on serial link buses," *7th International Symposium on Quality Electronic Design (ISQED '06)*, pp. 6 pp.-432, 2006.
- [4] A. E-Neyestanak, "Design of CMOS receivers for parallel optical interconnects", Ph.D. Dissertation, Stanford University, August 2004.
- [5] S. Ardalan, "Low Power Clock and Data Recovery Integrated Circuits", Ph.D. Dissertation, University of Waterloo, October 2007.
- [6] Altera Corporation, "Cyclone II Device Handbook", Vol. 1, pp.2-26, February 2007, <http://www.altera.com>.
- [7] Altera Corporation, "Phase-Locked Loop (ALTPLL) Megafuction User Guide", Vol. 1, pp.14, November 2009, <http://www.altera.com>.
- [8] J. D. H. Alexander, "Clock Recovery from Random Binary Data," *Electronics Letters*, vol. 11, pp. 541-542, October 1975.
- [9] F. Lin, "Research and design of low jitter, wide locking-range all-digital phase-locked and delay-locked loops", Ph.D. dissertation, March 2000.