Design of PRN Based Octa-Rate Clock And Data Recovery Circuit Using FPGA

Sreehitha Ghuguloth[1], Sowmya Todima[2], Sumanjali Pastham[3], Boya Pradeep Kumar[4] and Chandra Sekhar Paidimarry[5]
Dept. of ECE
Osmania University
Hyderabad, India
Email: sreehitha789@gmail.com, sowmyatodima@gmail.com, pasthamneeru05@gmail.com, pradeep.boge@gmail.com
chandrasekharpaidimarry@gmail.com

***Abstract*—Clock and data recovery (CDR) circuit plays a vital role for wired serial link communication in multi mode based system on chip (SOC). In wire linked communication systems, when data flows without any accompanying clock over a single wire, the receiver of the system is required to recover this data synchronously without losing the information. Therefore there exists a need for CDR circuits in the receiver of the system for recovering the clock or timing information from these data. The existing Octa-rate CDR circuit is not compatible to real time data, such a data is unpredictable, non periodic and has different arrival times and phase widths. Thus the proposed PRN based Octa-rate Clock and Data Recovery circuit is made compatible to real time data by introducing a Random Sequence Generator. The proposed PRN based Octa-rate Clock and Data Recovery circuit consists of PRN Sequence Generator, 16-Phase Generator, Early Late Phase Detector and Delay Line Controller. The FSM based Delay Line Controller controls the delay length and introduces the required delay in the input data. The PRN based Octa-rate CDR circuit has been realized using Xilinx ISE 13.2 and implemented on Vertex-5 FPGA target device for real time verification. The delay between the input and the generation of output is measured and analyzed using Logic Analyzer AGILENT 1962 A.**

***Keywords*—Clock and Data Recovery Circuit (CDR); system-on-chip(SOC);Finite State Machine (FSM); Pseudo Random Number (PRN);**

I. INTRODUCTION

In the conventional CDR circuits, the main parameters were introducing proper delay in the input data stream and reducing the dynamic power consumption. To overcome this, the combinational circuit in the FSM based digitally controlled delay line was replaced by the sequential circuit to provide required delay. Instead of using a single high-frequency clock signal, multiple phases of the same clock running at a frequency less than the data rate was used in the operation of the phase detector to reduce the dynamic power consumption. Therefore the 8-Phase Generator in [2] was replaced by 16-Phase Generator in [1] for reduction in dynamic power consumption.

Burugula Sai Sankalp, Boya Pradeep Kumar, Chandra Sekhar Paidimarry, Jagan mohan reddy N in [1] designed an Octa-rate CDR circuit which consists of 4 main modules namely, 16-Phase Generator, Early Late Phase Detector, Delay Line Controller and Digitally Controlled Delay Line. It is seen that though the dynamic power consumption has been reduced to an extent, the circuit was complex and not compatible to real time data. The main contributions of this paper are to make the PRN based Octa-rate CDR circuit more compatible to real time data and to reduce the dynamic power consumption seen in [1].

As the micro processors do not have enough processing power at high frequencies, the proposed CDR circuit is implemented using FPGA. The design of an application specific integrated circuit (ASIC) is not suitable for the implementation of the proposed Octa-rate CDR circuit because the quantity involved is less and thus the fabrication process would be expensive for such low volume production. Therefore FPGA is an efficient solution to the problem because it combines the speed and flexibility of microprocessors as well as the computational power of ASICs.

II. PROPOSED PRN BASED OCTA RATE CDR CIRCUIT

The proposed Octa-rate CDR circuit mainly consists of 4 blocks named as PRN Sequence Generator, 16-phase generator, Octa-rate phase detector and delay line controller. The block diagram of Octa-rate CDR circuit is as depicted in Fig. 1.

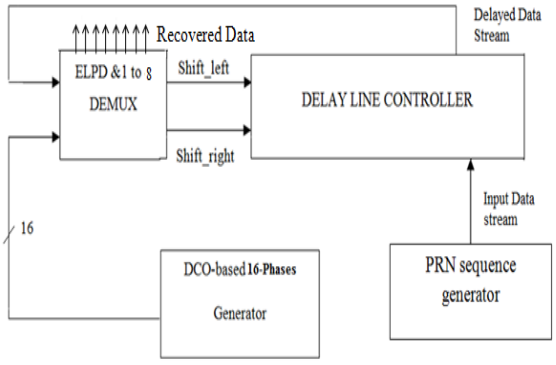

Fig. 1. Octa-rate CDR Block daigram

The design and implementation of each and every block of Octa-rate CDR circuit is discussed in detail.

A. PRN Sequence Generator

The existing CDR circuits described in [1] have input data which is constant and periodic. Due to this when the data gets

synchronized with the clock that is, in locked state the circuit remains locked forever and is always in sync with the clock.

The PRN sequence generator is implemented using linear shift registers and a combinational feedback circuit. Whenever a clock pulse is applied, the outputs at each stage of the register shifts one stage to the right, the output of the final stage is lost and the input to the first stage is obtained by EXCLUSIVE-NORing a set of the stages of the shift register [4]. For an *N* stage shift register, the maximum length of the random sequence generated is, $K= 2^N-1$.

The external data stream which is generated by Pseudo Random Number Sequence Generator is given to the Delay Line Controller. The proposed CDR has a PRN sequence generator which consists of ten D flip-flops and an Ex-NOR gate. The output of the tenth and the third D flip-flops are given as inputs to the Ex-NOR gate and the output is taken at the tenth D flip-flop.

B. Sixteen-Phase Generator

The input of sixteen-phase generator is from the DCO (digitally controlled oscillator). The required 16-phase generator is generated by using finite state machine where each phase is shifted by 11.25 degrees and its complements are generated. Therefore the finite state machine has 32 states and each state has its predefined 16-bit digital word. The flow chart of Sixteen-Phase Generator is depicted in Fig. 2.

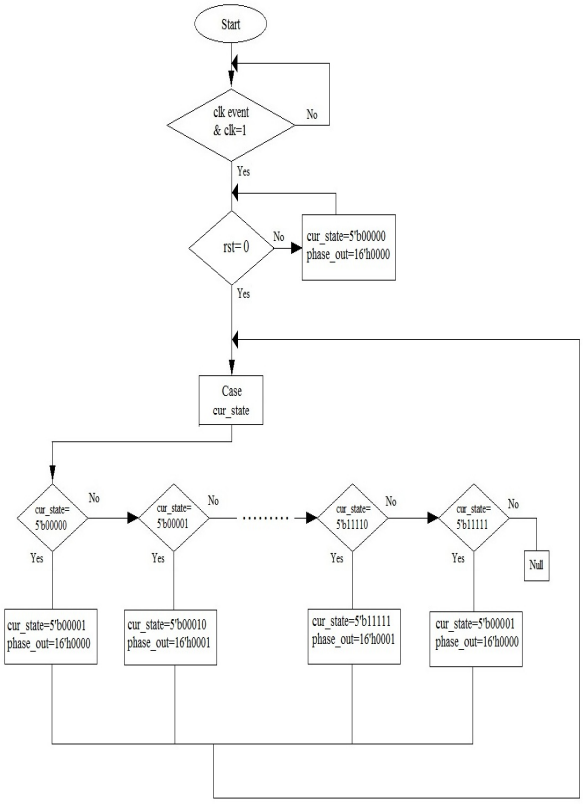

Fig. 2. Flow chart for 16-Phase generator

The finite state machine initially checks for the clock and reset condition, starts its operation when it is enabled and initializes the current state to be State 0. It gives the corresponding phase_outputs and assigns the next state value. The state transitions are governed by the positive edge of the clock. These 16 clocks of different phases which are generated are given to the phase detector as inputs.

C. Octa-Rate Phase Detector

Octa-rate phase detector detects the phase difference between the input data and the phases generated by 16-phase generator [4].

The proposed phase detector is an early-late octa-rate Alexander-based design, where the output is either early or late. The proposed phase detector samples the input data stream at *0º, 22.5º, 45º, 67.5º, 90º, 112.5º, 135º, 157.5º, 180º, 202.5º,225º, 247.5º ,270º, 292.5º , 315º* and *337.5º* of the clock phases, producing the sixteen signals *D0, D22.5, D45, D67.5, D90, D112.5, D135, D157.5, D180, D202.5, D225, D247.5, D270, D292.5, D315* and *D337.5* at the D flip-flop outputs. These sixteen signals are used to generate the *shift_right* and *shift_left*. In order to know if the data has arrived early or late, the relative clock edge positions with respect to the data edges are given by these two signals.

The required logic to produce the shift_right and shift_left signals are shown Eq. (1) and Eq. (2).

$$shift\_right = (D0 \oplus D22.5) + (D45 \oplus D67.5) + (D90 \oplus D112.5) + (D135 \oplus D147.5) + (D180 \oplus D202.5) + (D225 \oplus D247.5) + (D270 \oplus D292.5) + (D315 \oplus D337.5) \quad (1)$$

$$Shift\_left = (D22.5 \oplus D45) + (D67.5 \oplus D90) + (D112.5 \oplus D135) + (D147.5 \oplus D180) + (D202.5 \oplus D225) + (D247.5 \oplus D270) + (D292.5 \oplus D315) + (D337.5 \oplus D0) \quad (2)$$

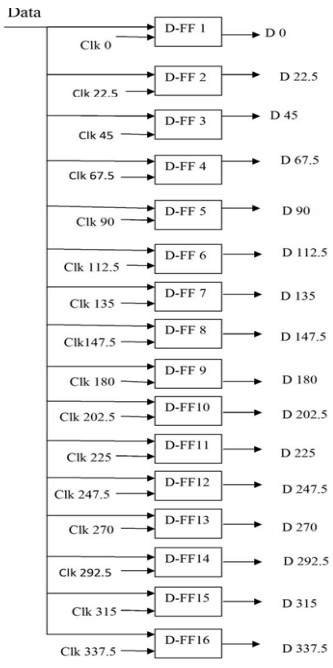

Fig. 3. Octa-rate phase detector Block diagram

D. Delay Line Controller

The delay line controller is used to control the length of the delay line. The delay line controller is generated by using finite state machine where each state is governed by the shift_right and shift_left signals received from the output of the phase detector. The operations of delay line controller are reported in the Table 1 [5].

TABLE I. DELAY LINE CONTROLLER OPERATIONS

shift_right	shift_left	Action
High	Low	Late state: the data stream is to be shifted to the right by decreasing the delay of the delay line.
Low	High	Early state: the data stream is to be shifted to the left by increasing the delay of the delay line.
High	High	Locked State: No Action.
Low	Low	No Action.

For each decision, the delay line controller updates the state which is responsible for the output delay of the delay line. The delay line controller operations are obtained by implementing the flowchart as shown in Fig. 4.

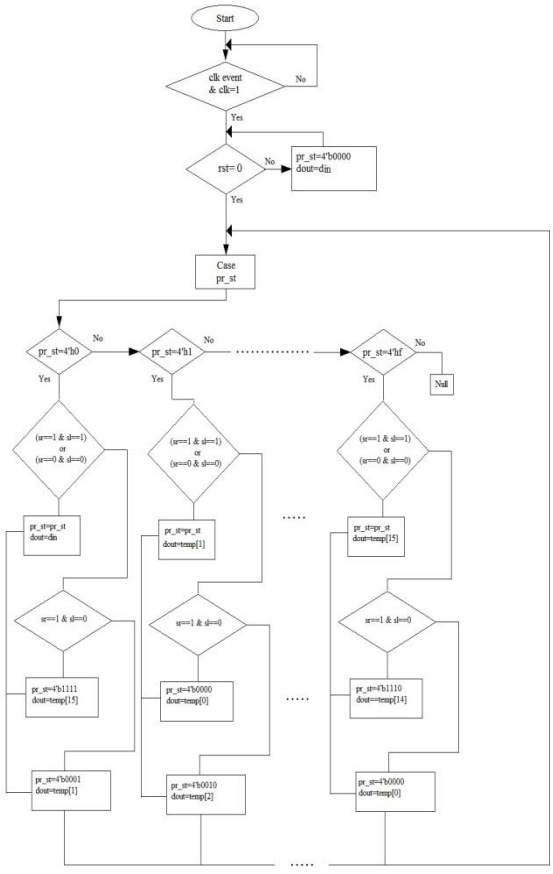

Fig. 4. Flow chart for delay line controller

The delay line controller also introduces the required delay in the input data stream. It consists of series of delay elements where each delay element consists of tap which is controlled by delay line controller. Each delay element consists of a D-flip flop [6].

III. SIMULATION AND FPGA RESULTS

The components for Octa-rate Clock and Data Recovery Circuit were individually coded in Verilog and simulated using Xilinx Student Edition 13.2. After simulating the individual components, they were integrated into a single module to work as the complete CDR circuit and was implemented on Virtex-5 FPGA target device.

The PRN Sequence Generator was generated as illustrated in Fig. 5. This sequence is given as input to the CDR circuit. The random sequence repeats after every 1023 clock pulses.

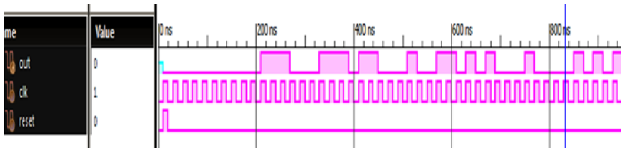

Fig. 5. Simulation result of PRN Sequence Generator

The implementation of 16-Phase Generator is described in Fig. 6. The 16 multiple phases generated whose frequency depends upon the clock which is generated by DCO.

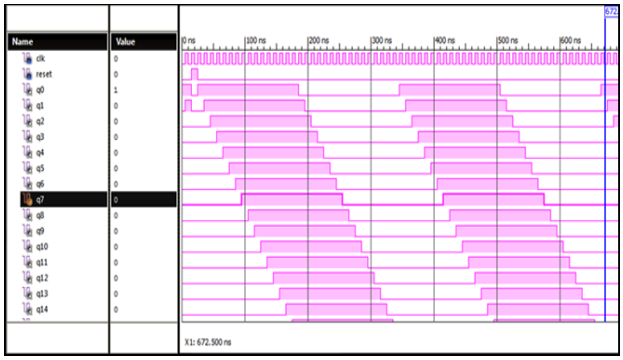

Fig. 6. Simulation Results of 16-Phase Generator

The Octa Rate Phase Detector is implemented as shown in Fig. 7. The variations in the phase detector outputs, that is the *shift_left* and *shift_right* signals are obtained. These two signals show the relative clock edge positions with respect to the data edges.

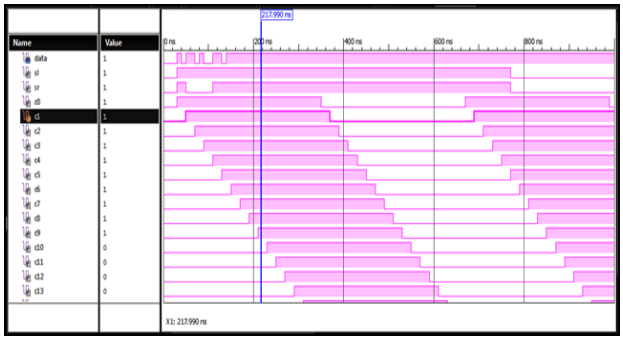

Fig. 7. Simulation results of Phase Detector

The Early state condition as depicted in Fig. 8. When the *shift_left* signal is high and *shift_right* signal is low, the data stream is to be shifted to the left and the delay must be increased in delay line so that, the number of delay elements will be increased until both *the shift_right* and *shift_left* signals are equal.

Fig. 8. Simulation output with shift_left signal is high and shift_right signal is low

The Late state condition is shown in Fig. 9. When the *shift_right* signal is high and *shift_left* signal is low, the data stream is to be shifted to the right and the delay must be decreased in delay line so that, the number of delay elements will be decreased until both the *shift_right* and *shift_left* signals are equal.

Fig. 9. Simulation output with shift_right signal is high and shift_left signal is low

Fig. 10 shows the locked state. After adjusting the delay of the data stream by passing it through a specific number of delay elements of the delay line, the CDR circuit enters the locked state. In this state both the *shift_left* and *shift_right* signals are high and the data is always in sync with the clock.

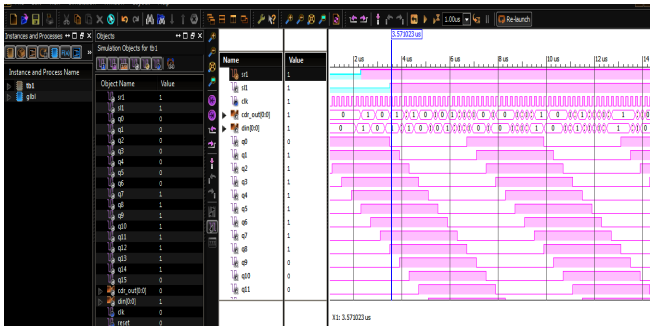

Fig. 10. Simulation output with shift_left signal is high and shift_right signal is high

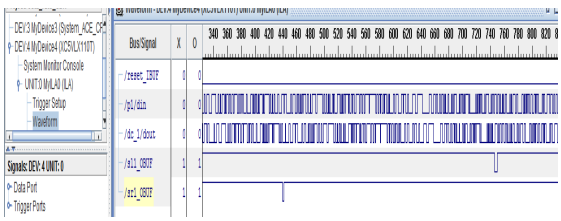

Fig. 11. Chip Scope results of Octa rate CDR circuit

The output waveform in chip-scope is obtained by dumping the Verilog code of Octa-rate clock and data recovery circuit ported on virtex-5 FPGA board. The reset is set as DIP switch in vertex 5 FPGA Target Device.

The supply power and the dynamic power consumption for the Octa rate clock and data recovery circuit is shown in the Table 3. and it is observed that the dynamic power consumed is 0.008 watts. By comparing the Octa rate clock and data recovery circuit in [1] with PRN based Octa rate clock and data recovery circuit presented in this paper, it can be observed that 42.85% of dynamic power consumption is reduced in the later as reported in Table 2.

TABLE II. DYNAMIC POWER CONSUMPTION COMPARISON

	Quarter Rate CDR in [2] (watt)	Octa Rate CDR (watt)	PRN based Octa Rate CDR(watt)
Power supply	1.056	1.056	1.051
Dynamic power	0.014	0.011	0.008

The timing analysis of the output waveforms is done using logic analyzer. The logic analyzer used for PRN based Octa rate clock and data recovery circuit is AGILENT 1692A. From the numerical results, the proposed circuit requiring less dynamic power compare to circuit.

IV. CONCLUSION

In this paper, FPGA based Octa-rate CDR implementation is discussed. Octa-rate CDR consists of 4 blocks. They are DCO-based external input data stream, 16-phase generator, early-late phase detector and delay line controller. In this proposed model first we implemented 16-phase generator. Next output of 16-phase generator is given to next module which is Octa-rate phase detector which detects phase difference between input data and clock signals generated by 16-phase generator and generates the early and late signals. These early and late signals are given as input signals to delay line controller. From the delay line controller, the actual output with the specified delay is obtained. Likewise each and every module is implemented and all these modules are integrated into single module known as Octa-rate CDR is implemented. Simulation results of Octa-rate CDR are explained. Next, the algorithms are realized in Virtex 5 FPGA Target Device. Along with that, the power analysis and timing analysis have been done and reported.

ACKNOWLEDGMENT

The authors wish to thank CV Raman Heterogeneous Computing Lab, Department of ECE, University College of Engineering (A), Osmania University, Hyderabad for providing supporting environment for carrying out this work.

REFERENCES

[1] Burugula Sai Sankalp, Boya Pradeep Kumar, Chandra Sekhar Paidimarry and Jagan mohan reddy N, "A Novel FPGA based Digital Octa-rate Clock and Data Recovery Circuit", IEEE Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), pp. 57-62, 27-29 November 2015, Hyderabad.

[2] Mohammed H. Alser, Maher Assaad, Fawnizu Azmadi Hussin and Israel Yohannes, "Design and FPGA Implementation of PLL-based Quarter-rate Clock and Data Recovery Circuit", International Conference on Intelligent and Advanced Systems (ICIAS2012), pp. 825-830,2012.

[3] Samir Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis, Pearson Education, 2005.

[4] J. D. H. Alexander, "Clock Recovery from Random Binary Data", Electronics Letters, vol. 11, pp. 541-542, October 1975.

[5] A. Hatakeyama, et al., "A 256-Mb SDRAM Using a Register-Controlled Digital DLL", IEEE J. Solid-State Circuits, vol. 32, pp. 1728-1732, Nov. 1997.

[6] 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), pp. 6 pp.-432,2006.