PORTABLE CLOCK RECOVERY CIRCUITS (CRCS) FOR ON-CHIP AND OFF-CHIP SERIAL DATA COMMUNICATION

Muhammad E. S. Elrabaa †

Computer Engineering Department

King Fahd Univesity of Petroleum and Minerals (KFUPM)

Dhahran, Saudi Arabia

I. INTRODUCTION

High data rates and reliability have made serial communications the most preferred inter-chip communication method. Applications range from backplane communications, to fiber data transmission. With the emergence of large systems-on-chips, new design methodologies are being adopted to cope with design complexity and wire delays. For the 90 nm technology, different architectural explorations have shown that wire delays of up to 1.5ns are not unusual [1]. For a 5 GHz clock, this delay exceeds seven clock cycles. Also, the Semiconductors Industry Association estimates that by 2010 cross-chip wire delays are going to exceed 12 clock cycles [2]. On top of this, ASICs are being assembled from pre-designed, blochs usually heterogeneous in nature, (*i.e.* IPs) that are interconnected together. Each block has its own operating frequency and communication needs. This necessitated a paradigm shift to enable quick timing closure of the whole chip; namely, the use of networks-on-chip [3]. As on-chip global interconnect technology moves from ad-hoc wiring or bus-based to switch/router based schemes, the demand for low overhead, low-power, and portable serial communication circuits (as IP blocks for link implementation) will be ever increasing.

Two types of NRZ synchronous serial data communications are possible. In the first type, illustrated in Figure 1, NRZ data (**D**$_{Tx}$) is sent by the transmitter using a clock (**Clk**$_{Tx}$). At the receiver side, a clock-recovery circuit (**CRC**) would extract the clock (**Clk**$_{Rx}$) from the received data, re-time this clock with the data and use it to sample the data. This type of synchronous serial communication is more suited to inter-chip (*i.e.* off-chip) communications due to limitations on the number of a chip's I/O pins. However, it requires the CRC to perform both frequency and phase locking.

Key words: clock recovery, SoC, NoC, NRZ data communication, digital circuits

† Address for correspondence
P.O. Box 5065
Computer Engineering Department
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia
E-mail: elrabaa@kfupm.edu.sa

Paper Received: 31 March 2007; Revised: 21 May 2007; Accepted: 18 September 2007

In the second type of NRZ synchronous serial data communications, called source synchronous, the transmitter sends the clock along with the data. This clock arrives at the receiver side out of phase with the data, hence the receiver re-synchronizes it with the data before using it to sample the data. This would be a viable alternative to source synchronous parallel buses in deep sub-micron technologies, where global wire delays amount to several clock cycles which necessitates careful design and elaborate skew control circuitry. This is the same argument that made serial communications replace parallel communications for inter-chip communications.

A source synchronous serial link (**S3L**) is illustrated in Figure 2. The FIFO buffers (synchronous at the transmitter side and asynchronous at the receiver side) would be required if the communicating entities operate at different clock speeds [4–8]. This type is better suited for implementing on-chip communications (NoC links) with multiple-clock domains since there is no limitation on sending the clock along with the data and it only requires phase locking.

For source synchronous serial links (clock is sent with the data), as with inter-chip communications, the issue of re-timing the clock with the received data arises. Having a clock-recovery and data retiming circuit (CRC) would allow very high serial data rates and help with the timing closure of the chip. These circuits, setting in the repeaters and receivers, accurately extract the clock signals from the non return-to-zero (NRZ) serial bit streams where they are embedded. Such a circuit must maintain synchronism between the clock and the data in the presence of data phase noise (jitter), and supply and temperature fluctuations. Also, they must be agile in extracting synchronized clocks for different data packets arriving from different sources.

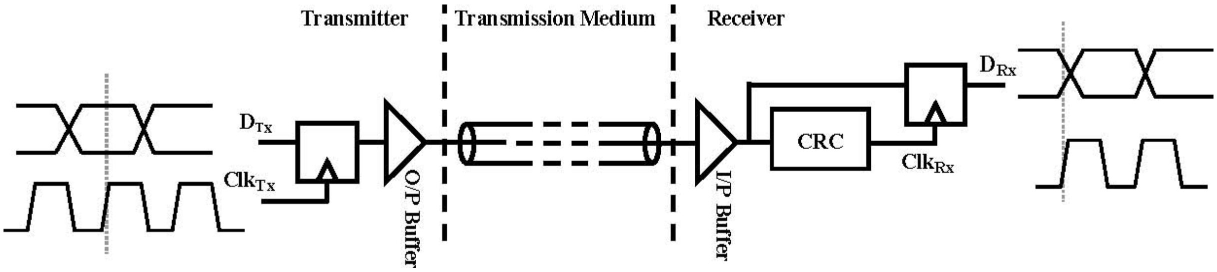


Figure 1. A typical synchronous serial link with clock-recovery circuit at the receiver for off-chip communications. The O/P and I/P buffers might implement some form of signal conditioning (such as equalization).

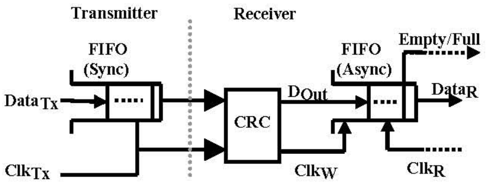


Figure 2. The general structure of a Source Synchronous Serial Link (S3L)

Many techniques for clock-recovery and data re-timing have been used for inter-chip communications [9–11]. Most of these techniques use analog PLLs (Phase-Locked Loops) or DLLs (Delay-Locked Loops). These analog CRCs suffer from several shortcomings: large area (due to analog blocks); difficulty to port to other processes or supplies; high VCO operating frequency (double the data frequency) which in turn limits the data rate, phase error accumulation in the VCO; and long lock times due to the loop damping behavior. Fully digital or semi-digital solutions were proposed to solve some of the analog PLLs problems [12–20]. Though these techniques retain many of the analog features, they suffer from poor resolution (more jitter), stability issues (two or more loops interacting), difficulty to port from one process to another (due to analog blocks), and/or large areas.

In [21], a fully digital non PLL/DLL CRC for source synchronous serial communication that demonstrated the potential of digital circuits in clock recovery and data re-timing was introduced. That CRC was based on adjusting the length of a digital delay line to align the input clock with the input NRZ data. The circuit was only suitable for source synchronous communications and did not incorporate any feedback for correction (*i.e.* open-loop). The output clock was centered with the input data through the use of a fixed delay (inverters). This required re-design of this delay for different operating frequencies/process technology. Also, though being fully digital, it utilized many non-standard circuits, making it difficult to synthesize and port from one process to another.

In this work two new improved CRCs that are also based on adjusting the length of digital delay lines are introduced. The first one performs both frequency and phase capturing of the clock hence is suited for off-chip communications, while the second one, a source synchronous serial link, is better suited for on-chip communications (with only phase locking). Unlike the CRC in [21], the two new CRCs are entirely made of standard CMOS gates. The first CRC is presented in Section 2. This includes basic concept of operation, circuit description and performance evaluation. The similar treatment for the second technique is presented in Section 3. Finally, conclusions are presented in Section 4.

2. A CRC FOR OFF-CHIP SERIAL LINK

2.1. Basic Architecture and Concept of Operation

Unlike the CRC in [21], this CRC can capture the transmitter's clock frequency from the received NRZ data stream and align its phase with the received data. The circuit also retimes the clock with each data transition. The basic idea of this CRC is to use a variable length digital delay line to measure the input data's bit duration (*i.e.* the time between a positive edge and a negative edge). This delay is then used in a feedback loop to form an oscillator with an oscillation period equal to that delay (*i.e.* oscillation frequency will be twice the maximum data frequency or equal to the data rate). The architecture of the proposed CRC is shown in Figure 3. When the incoming NRZ data (**Din**), fed to the first input of the **PED** circuit (Figure 4 (*a*)), exhibits a positive transition (0→1), the positive-edge detector (**PED**) circuit produces a pulse (**P0**) that would travel down a delay line. The inverters at the output of the **PED** circuit are meant to delay these pulses by a fixed delay to reduce the required number of stages in the delay lines and **PCCM** circuit. This simple addition reduces the power consumption of this circuit considerably compared to the one in [21].

The delay line will provide several phases of the **PED** pulse (**P_1** to **P_n**). An identical delay line will also contain a replica of this pulse, providing unloaded clock phases (**CLK_0** to **CLK_n**). When the next negative NRZ data edge arrives, the negative-edge detector (**NED**) circuit, Figure 4(*b*), will generate two complementary pulses, **T** and **T_b,** which will be used by the period capturing and clock muxing (**PCCM**) circuit, Figure 5, to latch in the appropriate pulse phase. This latched pulse phase is then used as an enable signal for a mux to select the corresponding clock phase. More than one clock phase might be selected, making the resulting feedback clock (**Clk_{fb}**) an interpolated version of the selected clock phases, enhancing the CRC's resolution significantly.

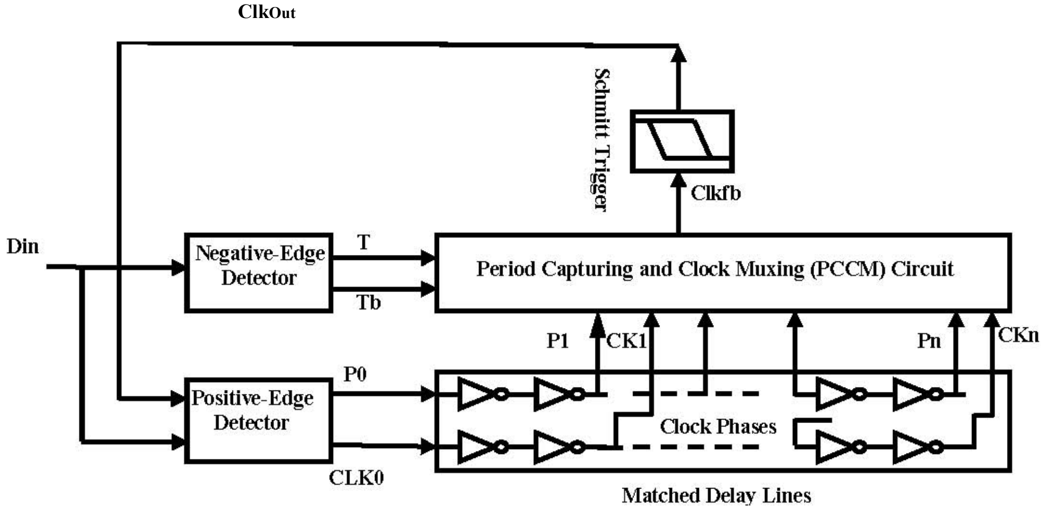
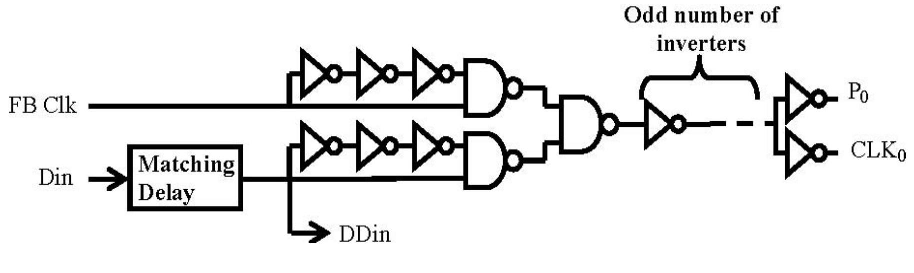
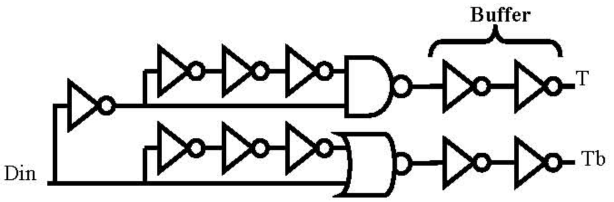


Figure 3. The architecture of the clock-recovery circuit proposed for off-chip serial communications



(*a*)



(*b*)

Figure 4. (a) The positive-edge detector circuit, (b) The negative-edge detector

The fed-back clock, reconstructed using a simple Schmitt trigger circuit, would have a delay T_{clkfb} (referenced to the positive edge of the input data):

$$T_{clkfb} = T_B + T_{NED}, \tag{1}$$

where T_B is the bit duration and T_{NED} is the delay of the **NED** circuit. This means that the captured delay in the variable delay line is T_{DL}, as a function of T_{clkfb} and the PED delay T_{PED1}:

$$T_{DL} = T_{clkfb} - T_{PED1} = T_B + T_{NED} - T_{PED1}\,. \tag{2}$$

After this feedback clock is re-constructed using the Schmitt Trigger (with a delay T_{ST}) it is fed back to the 2^{nd} input of the **PED** circuit. The output clock **Clk$_{Out}$** is produced after a delay T_{clkOut}:

$$T_{clkOut} = T_{clkfb} + T_{ST}\,. \tag{3}$$

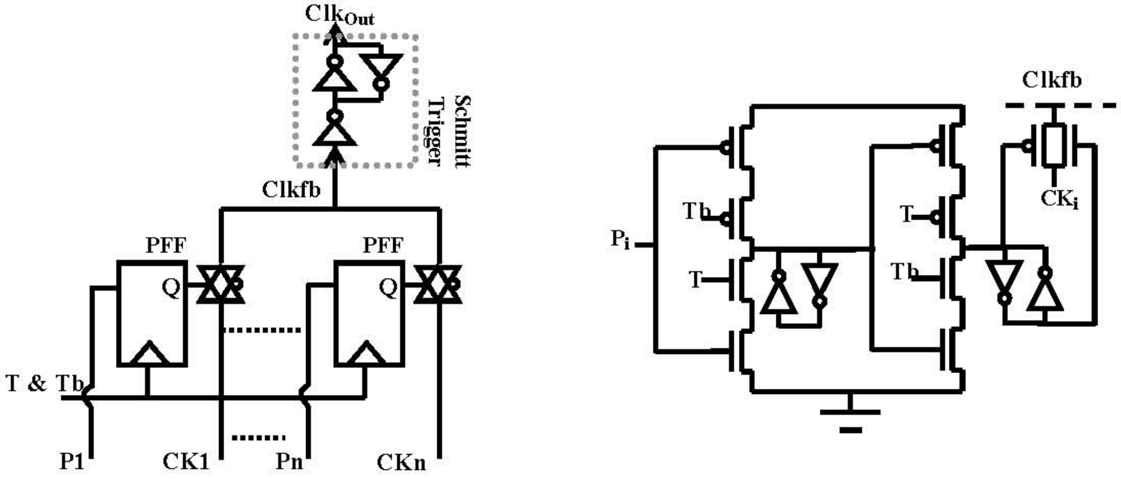
This closes the loop and from this point onward forms a ring oscillator with a total delay of

$$T_{CLK} = T_{PED2} + T_{DL} + T_{ST} \tag{4}$$

where T_{PED2} is the delay through the second input of the **PED** circuit. Substituting T_{DL} from (2) into this Equation:

$$T_{CLK} = T_{PED2} + T_B + T_{NED} - T_{PED1} + T_{ST} = T_B + T_{NED} + T_{ST} + (T_{PED2} - T_{PED1})\,. \tag{5}$$

Also, if the fixed delay at the input of the **PED** ($T_{PED2} - T_{PED1}$) is set $= T_{NED} + T_{ST}$, then the oscillator's total delay T_{CLK} becomes: $T_{CLK} = T_B$ or 1 bit duration (*i.e.* bit cell). This closes the loop and forms a ring oscillator with a total delay of 1 bit duration (*i.e.* bit cell). The delay between the two input paths of the PED circuit is implemented using an even number of inverters. The first **ClkOut** pulse will reach the **PED** 2^{nd} input (referenced to the first positive data edge), using (1–5) above, at: $t_{CLK1} = T_B + (T_{PED2} - T_{PED1})$. The second pulse would be at $t_{CLK2} = 2\,T_B + (T_{PED2} - T_{PED1})$ and the *n*th pulse $t_{CLKn} = n\,T_B + (T_{PED2} - T_{PED1})$. Hence the output clock pulses are not only at twice the input frequency but are also aligned with the delayed version of the input data (DDin). This means that the CRC will continue to oscillate at twice the input's frequency even in the absence of any positive or negative input data edges. With each data transition, the circuit re-times the clock with data, hence correcting for any injected phase and/or frequency noise. The inverted version of **ClkOut**, which is in the middle of the bit cell, is used to sample and latch in the delayed data bit (DDin), as illustrated in Figure 6.



(*a*) (*b*)

Figure 5 (a). The PCCM Circuit's Schematic including the Schmitt Trigger and (b) The PFF schematic

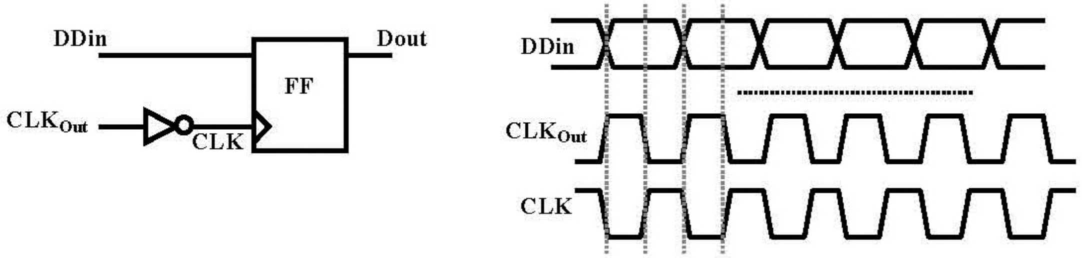


Figure 6. Timing relations between the delayed input data (DDin), CRC's output clock (CLK$_{Out}$) and sampling clock (CLK)

The total number of gates in this CRC is less than 100, most of which are inverters. This makes the circuit, very compact allowing the installation of many copies for different serial links within the same chip. Also, all blocks were designed using identical gates (*i.e.* similar gates are instances of the same cell). Hence all inverters in the circuit are identical, all FFs are identical, all NAND gates are identical, all NOR gates are identical …*etc*. This demonstrates the portability of the circuit.

2.2. Performance Evaluation

Circuit simulations using T-Spice® and a 0.13µm, 1.2V CMOS technology were used to evaluate the operation and performance of the proposed CRC. Sizes of transistors in the circuit components were optimized for 2 GBPS operation, but it can still operate with any input frequency up to 2.5 GBPS. For lower frequencies the fixed delay in the **PED** circuit is increased keeping the **PCCM** circuit the same. First, the basic operation of the circuit is demonstrated by applying NRZ data to the input. Figure 7 below shows how the CRC captures the data frequency at 2 GBPS and generates an output clock after two bit transitions. The data pattern used was 1011111111.

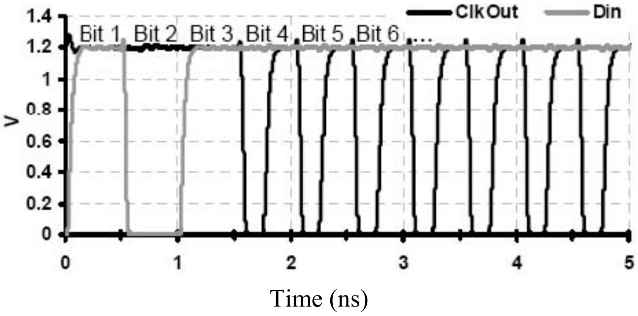


Figure 7. The input data and output clock waveforms of the CRC at 2 GBPS

Next, the response of the CRC to the injection of a large phase noise at the input data is evaluated. Figure 8 shows how the circuit fully recovers the clock within two data transitions after a large phase noise (equivalent to a half bit cell) is injected into the input data. An analog PLL would have taken hundreds or even thousands of cycles to recover.

Half-cell (*i.e.* 250 ps)
Phase noise injected

Circuit recovers
the clock here

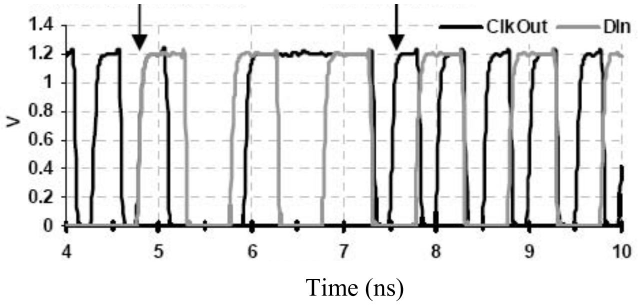
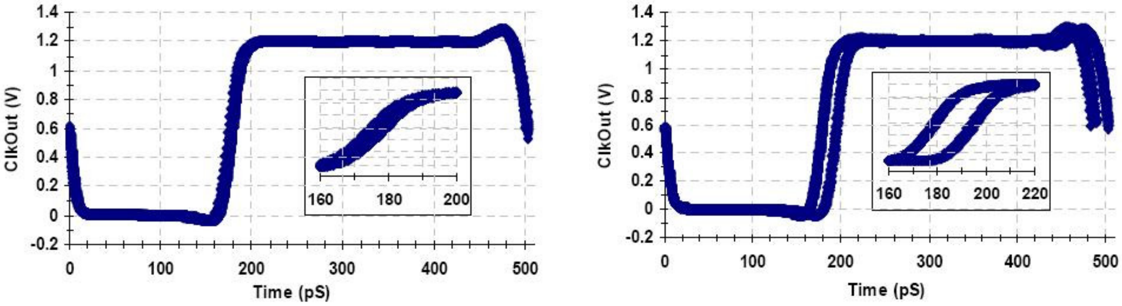


Figure 8. Output clock recovery after the injection of a large phase noise (half-cell) at 2 GBPS

The stability of the output clock's frequency was evaluated using two simulations: one with no data transitions after the initial clock frequency capturing (*i.e.* data stays constant after the two initial bit transitions) and another with a pseudo-random input stream (using 6-bit long pseudo-random patterns that are fed serially to the circuit). The output clock periods are laid out on top of one another in Figures 9 (*a*) and (*b*). The peak-to-peak period jitter in the case of no data transitions is less than 10 pS, an outstanding performance for an all-digital clock recovery circuitry. The results in Figure 9(*b*) show that, when there are data transitions, the output clock has two modes with about 10 ps difference in their periods. The peak-to-peak period jitter within each mode is still less than 10 ps. These two modes result from variations in the loop delay due to two modes of operation; when there is no positive data transition and when there is such transition. This bi-modal operation of the CRC won't cause bit errors since the faster mode only occur when there is a positive input transition and the circuit immediately goes back to the normal mode when there is no data transitions. Hence there is no accumulation of phase error that might cause an error. It should be noted that the relative difference between the two modes is smaller for lower clock frequencies.



(a) Output Clock with no data transitions after the first two

(b) Output Clock with pseudo-random data

Figure 9. Output clocks laid on top of one another. The insets show a Zoom-in of the clock edge region

Figure 10 shows the average output clock period for different data rates. The circuit components were only optimized once for the 2 GBPS data rate (0.5 nS bit cell). For other data rates, only the fixed output delay part of the **PED** circuit was changed (by changing the number of inverters). As the results in the figure indicate, the maximum frequency error is still below 0.8% for other data rates. The accumulated phase error due to the frequency error, however, is reset with every time a data transition occurs. So with proper data encoding that ensure adequate data transitions, the bit error rate can be set to a desired value.

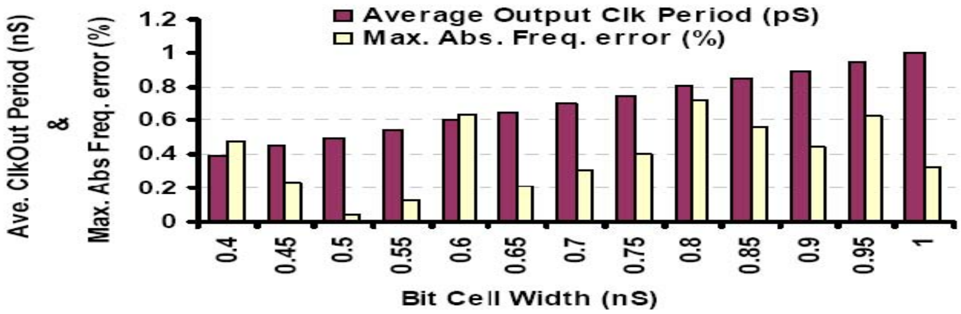


Figure 10. The average output clock Frequency and maximum absolute frequency error for different data rates

3. A CRC FOR ON-CHIP SOURCE SYNCHRONOUS SERIAL LINK (S3L)

3.1. Basic Concept of Operation

The basic concept of the **S3L** scheme was demonstrated in Figure 2. From the transmitter side, data is sent along with the transmission clock (**Clk**$_T$). The FIFO on the transmitter side is synchronous (could be a simple single Flip-Flop). Both the transmitted data and clock are received by the **CRC** which re-times the clock with the data (**D**$_{Out}$). This clock (**Clk**$_W$) is used to write to the receiver FIFO. The CRC would also have a simple FSM to detect start bits and control writing to the FIFO. The asynchronous FIFO at the receiver side is required to facilitate data transfer between two clock domains (Transmitter and Receiver). The receiver reads data using its own clock (Clk$_R$).

The architecture of this CRC is very similar to the off-chip version and is shown in Figure 11. The delay of the variable length digital delay line is adjusted such that a clock phase is aligned with the edge of the input data. The complement of this phase, which would then be in the middle of the bit cell (*i.e.* 90° phase-shifted), is selected as the output clock. The delay line is made of two parts: a fixed delay (close to half a bit cell) and two matched delay lines that should have a total delay of at least one bit cell. The two matched delay lines carry two clock phases that are complement of one another (~ 180° out of phase). The fixed delay was added to reduce the required stages in the variable delay line compared to the circuit in [21] while enhancing the resolution. Again all the different parts of the delay lines are all made up of identical inverters.

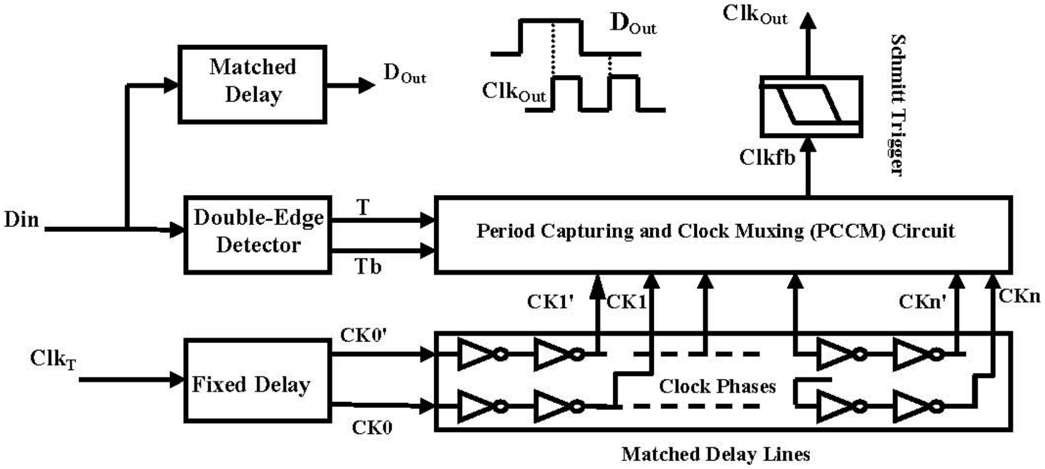


Figure 11. The architecture of the clock-recovery circuit for the on-chip S3L

An improved (compared to [21]) double-edge detector (**DET**) circuit, shown in Figure 11, is used generate two complementary pulses each time there is an input data transition. This circuit could be implemented using a differential static XOR/XNOR circuit such as in [22]. However, the implementation shown in Figure 11 was chosen because it uses standard digital gates and produces complementary pulses (**T** and **Tb**) with equal delays. These pulses are used as trigger signals by the phase capturing and clock mixing (**PCCM**) circuit to capture the relative phase between the data and input clock. The **PCCM** circuit selects the appropriate phase(s) and output them to the Schmitt trigger to re-construct the clock signal.

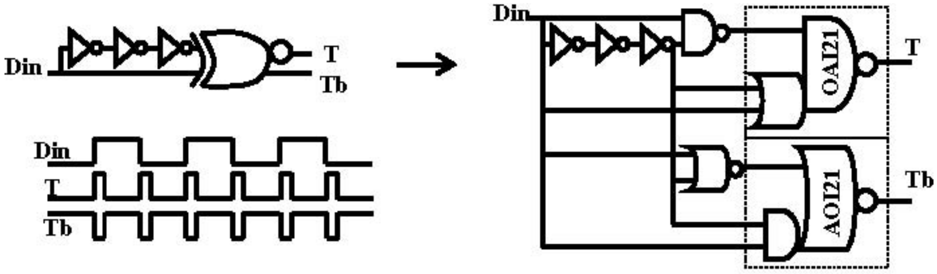


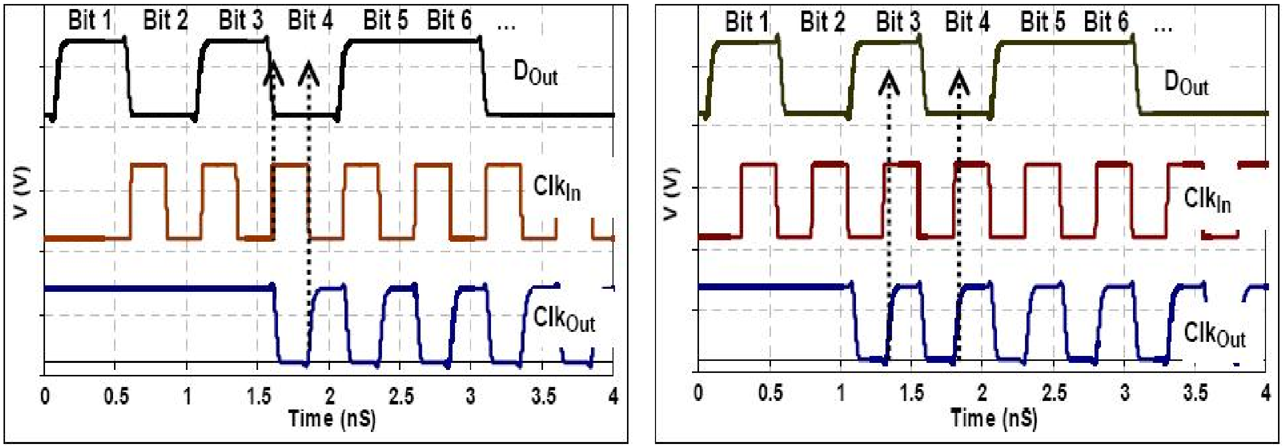
Figure 12. The Double-edge detector circuit implementation

Again, the total number of gates in this CRC is less than 100, most of which are inverters. This makes the circuit very compact, allowing the installation of many copies for different serial links within the same chip.

3.2. Performance Evaluation

Figure 13 below shows how the CRC retimes the clock with **Dout** within 2 data transitions. Two scenarios were simulated in this figure; one with the input clock initially 85° phase-shifted with the data (*i.e.* 5° short of the required phase) and another with the input clock initially 0° phase-shifted with the data. In each case, the output clock is produced at the required phase.

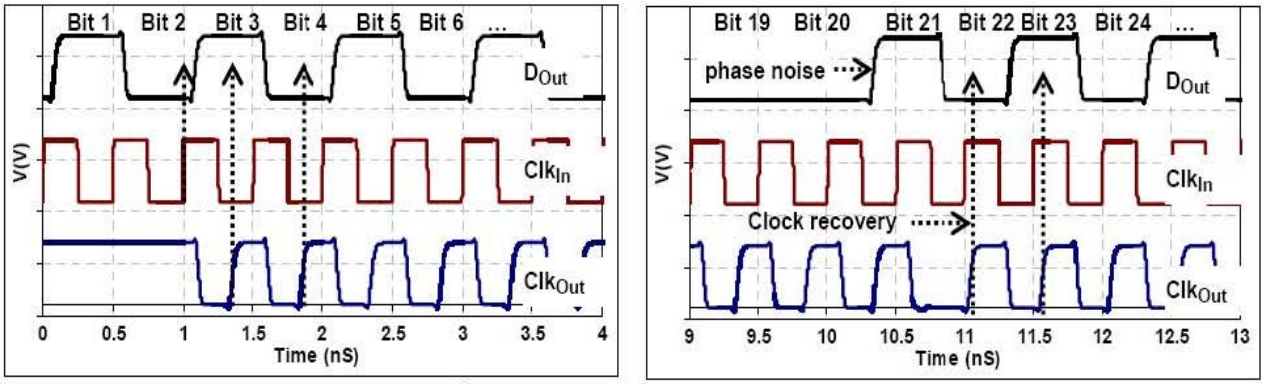
The response of the circuit to a large phase noise (half a bit cell) injection into the input data was also evaluated and the result is reported in Figure 14. The figure shows how the circuit first re-times the clock with the data with an initial **–15**° phase-shift between the input clock and data. The circuit then recovers the clock within one bit transition after a half-bit cell phase noise is injected into the data.



(*a*) (*b*)

Figure 13. The input clock, output clock and data waveforms at 2 GBPS with two different phase shifts between the input clock and data; (a) 0º phase shif; (b) 85º phase shift

Again, the stability of the output clock's frequency was evaluated with a 6-bit long pseudo-random input stream and the results are reported in Figure 15. As this figure shows, the peak-to-peak clock jitter is about 10 ps. It should be noted that this jitter would not accumulate since the CRC constantly re-times the clock with every data transition. Furthermore, the jitter is much smaller with no data transitions, again an outstanding performance for an all-digital clock re-timing circuit.



(a) Initial clock re-timing with the input clock ~ -15° phase-shifted. (b) Clock re-timing after an input phase noise is injected at Bit 21.

Figure 14. The CRC's response to a large phase noise injection (half a bit cell) in the input data

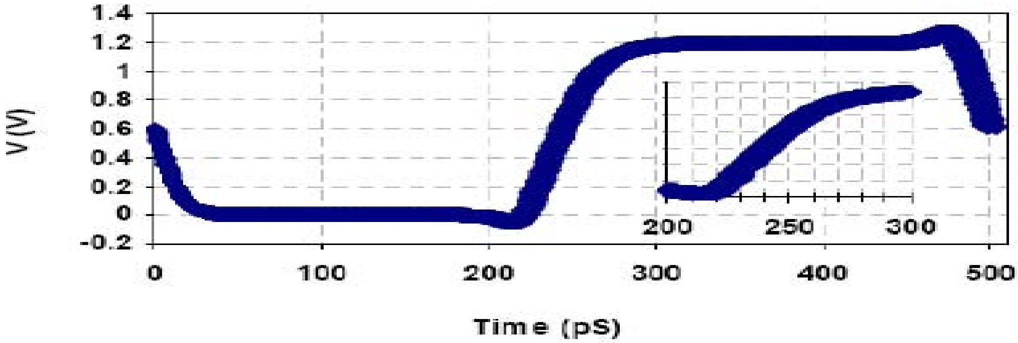


Figure 15. The output clock for a 6-bit long pseudo random input data. The inset shows the clock rising edge

4. CONCLUSIONS

Two new clock recovery circuits have been developed: one suited for off-chip communications that can perform clock frequency and phase capturing and another that is intended for source synchronous on-chip communications. Both circuits can retime the clock with the NRZ data stream within two bit transitions. Simulation results show that the output clocks of the two circuits are very stable. Output frequency error could be made very small by proper sizing of the circuit components for the intended technology. However, the circuit continuously re-times the clock phase with each bit transition. This means that with adequate data encoding the impact of frequency error on bit error rate can be virtually eliminated. Both circuits have less than 100 gates (most of which are inverters). This makes them very compact and highly portable, and able to support SoC designs with many serial links for inter-chip or intra-chip communications.

REFERENCES

[1] N. P. Carter and A. Hussain, "Modeling Wire Delay, Area, Power and Performance in a Simulation Infrastructure," *IBM Journal of Research and Development*, **50(2/3)**,(2006), p. 311.

[2] *International Technology Road Map For Semiconductors, ITRS, 2003 Edition*. Semiconductors Industry Association 2006, http://www.itrs.net/Links/2006Update/2006UpdateFinal.htm.

[3] L. Benini and G. D. Micheli, "Networks on Chips: A New SoC Paradigm", *IEEE Computer*, **35**(**1**)(2002), pp. 70–78.

[4] Daniel M. Chapiro, "Globally-Asynchronous Locally-Synchronous Systems," *PhD Thesis, Stanford University*, October 1984.

[5] T. Chelcea and S. M. Nowick, " A Low-Latency FIFO for Mixed-Clock Systems," *Proc. IEE Computer Society Annual Workshop on VLSI (WVLSI'00)*, April 2000.

[6] Simon Moore *et al.*, "Point to Point GALS Interconnect," *Proceedings of the Eighth International Symposium on Asynchronous Circuits and Systems (ASYNC.02),* 2002.

[7] A. Chakraborty and M. R. Greenstreet, "A Minimal Source-Synchronous Interface," *Proc. 15th Annu. IEEE Int. ASIC/SOC Conf.*, 2002, pp. 443–447.

[8] A Chakraborty and M. R. Greenstreet, "Efficient self-timed interfaces for crossing clock domains," *Proc. 9th Int. Symp. Asynchronous Circuits and Systems*, 2003, pp. 78–88.

[9] B. Razavi, (ed.), *Monolithic Phase-Locked Loops and Clock Recovery Circuits*, New York: IEEE Press, 1996.

[10] B. Razavi, "A 2.5-Gb/s 15-mW Clock Recovery Circuit," *IEEE JSSC*, **31**(**4**)(1996), pp. 472–480.

[11] S. Anand and B. Razavi, "A CMOS Clock Recovery Circuit for 2.5-Gb/s NRZ Data," *IEEE JSSC*, **36**(**3**)(2001), pp. 432–439.

[12] D-L. Chen, "A Power and Area Efficient CMOS Clock/Data Recovery Circuit for High-Speed Serial Interfaces," *IEEE JSSC*, **31**(**8**)(1996), pp. 1170–1176.

[13] T. Saeki *et al.*, "A 1.3-Cycle Lock Time, Non-PLL/DLL Clock Multiplier Based on Direct Clock Cycle Interpolation for Clock on Demand," *IEEE JSSC*, **35**(**11**)(2000), pp. 1581–1590.

[14] T-Y. Hsu *et al.*, "An All-Digital Phase-Locked Loop (ADPLL)-Based Clock Recovery Circuit," *IEEE JSSC*, **34**(**8**)(1999), pp. 1063–1073.

[15] M. Combes *et al.*, "A Portable Clock Multiplier Generator Using Digital CMOS Standard Cells," *IEEE JSSC*, **31**(**7**)(1996), pp. 958–965.

[16] M-J Lee, *et al.*, "A Second-Order Semi-Digital Clock Recovery Circuit Based on Injection Locking," *Proc. IEEE ISSCC'03*, 2003, pp. 74–75.

[17] H. Takauchi, *et al.*, "A CMOS Multichannel 10-GB/s Transceiver," *IEEE JSSC*, **38**(**12**)(2003), pp.2094–2100.

[18] R. Farjad-Rad, *et al.*, "1 33-mW 8-GB/s CMOS Clock Multiplier and CDR for Highly Integrated I/Os," *IEEE JSSC*, **39**(**9**)(2004), pp. 1553–1561.

[19] K. Krishna, *et al.*, "A 0.6 to 9.6 GB/s Binary Backplane Transceiver Core in 0.13 μm CMOS," *Proc. IEEE ISSC'05*, pp. 65–66.

[20] J. Parker, *et al.*, "A 15 mW, 3.125 GHz PLL for Serial Backplane Transceivers in 0.13μm CMOS," *Proc. IEEE ISSC'05*, pp. 412–413.

[21] M. E. S. Elrabaa, "An All-Digital Clock Recovery and Data Retiming Circuitry for High Speed NRZ Data Communications," *Institute of Electronics, Information and Communication Engineers (Japan), Transactions on Electronics*, **E85-C**,(**5**)(2002), P. 1170.

[22] M. E. S. Elrabaa, "A New Static Differential CMOS Logic with Superior Low Power Performance," *Analog Integrated Circuits and Signal Processing*, **43**(**2**)(2005), pp. 183–190.