

A Digital Clock Re-Timing Circuit for On-Chip Source-Synchronous Serial Links

Muhammad E. S. Elrabaa

Computer Engineering Department

King Fahd University of Petroleum and Minerals (KFUPM)

Dhahran, Saudi Arabia

elrabaa@kfupm.edu.sa

Abstract— A new all-digital circuit scheme for clock and data re-timing functions for on-chip high-speed source synchronous data communications, such as in burst-mode data transmission over a network-on-chip is introduced. The new technique is non-PLL-based and is capable of re-timing the output clock with the received data within one data transition. Being fully digital makes its area much smaller than conventional circuitry. It can also be described by any hardware description language, simulated, and synthesized into any digital process. This enables it to be ported from one technology to another and support system on a chip (SOC) designs. The design concept is demonstrated with T-Spice® simulations using a 0.13μm digital CMOS technology.

Index Terms — Clock-Recovery, Networks-On-Chip, Systems-on-Chip, ASICs, Digital Circuits

I. INTRODUCTION

With the emergence of large systems-on-chips, new design methodologies were adopted. ASICs are being assembled from pre-designed blocks (i.e. IPs), usually heterogeneous in nature, 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 [1]. These networks are made of routers (switches) and links. 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 re-timing circuit would allow very high serial data rates and help with the timing closure of the chip. Such circuit must maintain synchronism between the clock and the data in the presence of data phase noise (jitter), supply and temperature fluctuations. Also, they must be agile in extracting synchronized clocks for different data packets arriving from different sources.

Many techniques for clock-recovery and data re-timing have been used for inter-chip communications [2]. Most of

these techniques use PLLs (Phase-Locked Loops) or DLLs (Delay-Locked Loops). These circuits represent an automatic feedback control systems that control the frequency and phase of the output clock. Their job is made difficult by the fact that NRZ (non-return-to-zero) data has no spectral contents at the bit rate or its even-order harmonics. This problem can be circumvented using edge detection techniques [3, 4]. Also, long streams of consecutive 1s and 0s represent a serious problem to traditional clock recovery circuits that utilize Phase-Locked-Loops (PLLs) and result in data-dependent jitter in the PLL's output. To reduce this jitter the PLL's bandwidth has to be limited which in turn limits the PLL's capture range. Also, larger loop gain reduces jitter generation but reduces stability.

Analog PLLs 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 [5-8]. 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. A fully digital non PLL/DLL technique for source synchronous serial communication demonstrated the potential of digital circuits in clock recovery and data re-timing [9]. Though fully digital, it contained some non-standard digital components. It was also meant for inter-chip communications with strict constraints on jitter transfer and hence was large and consumed relatively large power.

In this work a novel non-PLL/DLL (no loop behavior) clock and data re-timing circuit is proposed. It is fully digital containing only standard digital gates. It can retime the clock with the received data within one data transition, maintaining an approximately 90° phase shift for minimum Bit-error-rate

This research project has been funded by King Fahd University of Petroleum & Minerals under project # COE/DIGITAL/287.

(BER). The design is very simple and can be integrated within any flow for any required data rate.

The basic operation and circuit description of the proposed source synchronous serial link (S3L) with emphasis on the novel CRC are given in section 2. Performance evaluations of the CRC are presented in section 3 followed by conclusions in section 4.

II. THE PROPOSED S3L

The basic concept of the proposed S3L scheme is demonstrated in Figure 1. 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-time 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). Flow control (i.e. back pressure) is handled by higher layers of the NoC.

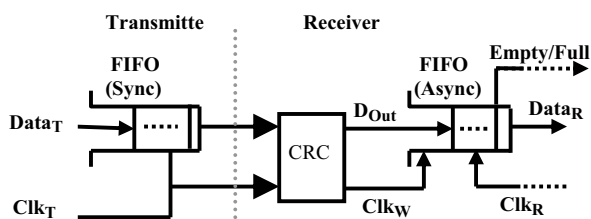


Figure 1. The general structure of the Source Synchronous Serial Link (S3L).

The architecture of the proposed CRC circuit is shown in Figure 2. The delay of a 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 ($\sim 180^\circ$ out of phase). The fixed delay was added to reduce the required stages in the variable delay line while maintaining the same resolution as in [9]. The different parts of the delay line are all made up of identical inverters.

A double-edge detector (DET) circuit, shown in Figure 3, generates 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 [10]. However, the implementation shown in Figure 3 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 muxing (PCCM) circuit to capture the relative phase between the data and input clock. The PCCM circuit, shown in Figure 4, would select the appropriate phase(s) and output them to the Schmitt trigger to re-construct the clock signal. The PCCM circuit is made of pulsed Flip-Flops (PFF) and transmission gate muxes. The PFFs, shown in Figure 4, are regular FFs that are triggered by the two complementary pulses T and Tb .

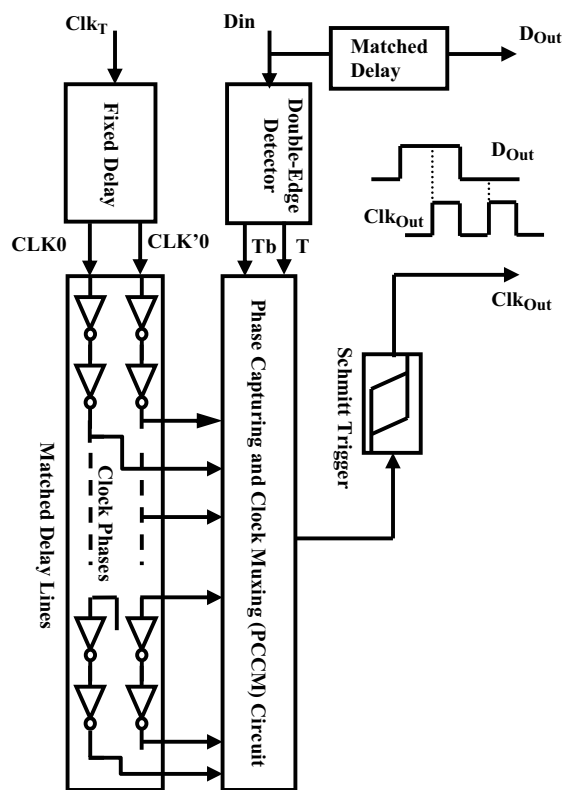


Figure 2. The architecture of the proposed clock-retiming Circuit (CRC).

The proposed CRC operates as follows;

- The fixed delay delays the incoming clock by about one half of a bit cell and produces two complementary phases which travel down the two matched delay lines,
- When an input transition occurs, the DED circuit generates the two complementary pulses, T and Tb that are applied as trigger signals to all the PFFs in the PCCM circuit,
- Each PFF captures the corresponding clock phase (CK_i) at that instance. PFFs that capture the appropriate clock phase would then enable their

muxes and pass the complement of the captured phase (CK'_i),

- The Schmitt trigger reconstructs the output clock from the mixture of phases from all the enabled muxes. With more than one phase usually selected, the muxes in the PCCM circuit and the Schmitt trigger perform phase interpolation yielding higher resolution,
- The matched delay shown in Figure 2 is made up of even number of inverters that delays the input data (D_{in}) by an equivalent delay of the DED, PFF, mux and Schmitt trigger. This makes the captured clock Clk_{Out} about 90° phase-shifted with the data D_{Out} . With each data transition the circuit re-times the clock with data, hence correct for any injected phase and/or frequency noise.

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

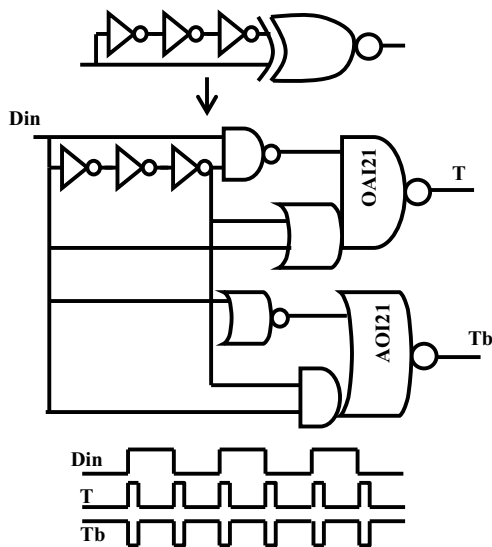


Figure 3. The Double-edge detector circuit implementation.

III. PERFORMANCE EVALUATION

Circuit simulations using T-Spice[®] and a $0.13\mu\text{m}$, 1.2V CMOS technology were used to evaluate the operation and performance of the proposed CRC circuit. 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 and the number of stages in the delay lines and PCCM circuit have to be increased. It should be noted that the data frequency (in fact the whole NoC) is frozen at design time.

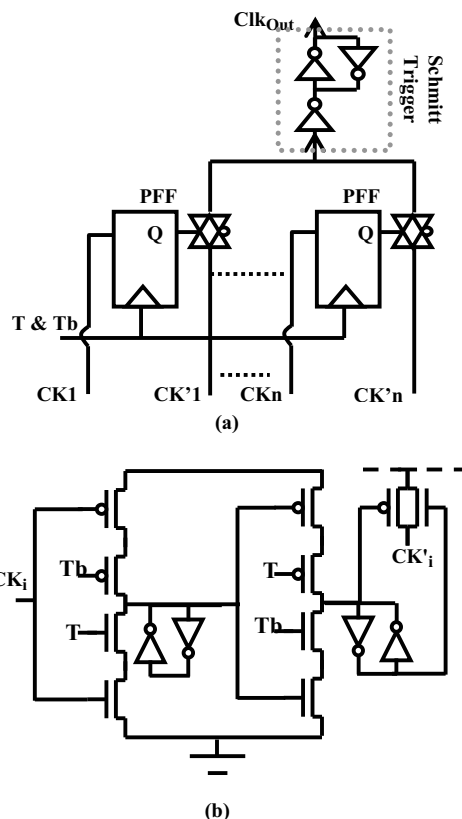


Figure 4. (a) The PCCM Circuit's Schematic including the Schmitt Trigger, and (b) The PFF schematic.

Figure 5 below shows how the CRC retimes the clock with D_{out} within 2 data transitions. For this figure, the input clock was initially about 85° phase-shifted with the data (i.e. 5° short of the required phase). The output clock is produced at the required phase. Figure 6 shows the same result but with a 0° phase shift between the input clock and D_{out} . Figure 7 shows how the circuit recovers the clock after a large phase noise (half a bit cell) is injected into the input data. An analog PLL would have taken hundreds or even thousands of cycles to recover.

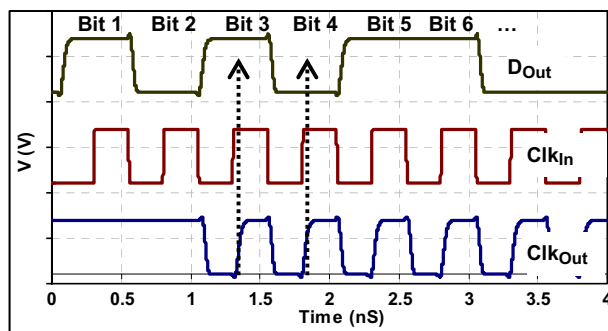


Figure 5. The input clock, output clock and data waveforms at 2 GBPS with input clock $\sim 85^\circ$ phase-shifted.

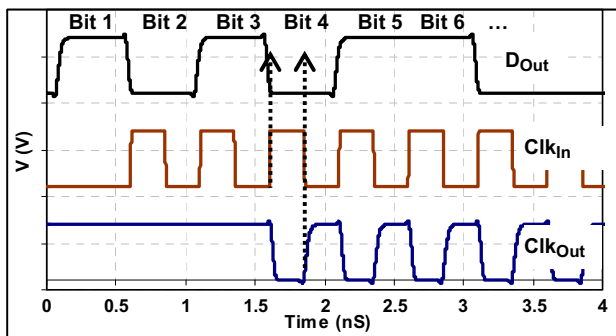
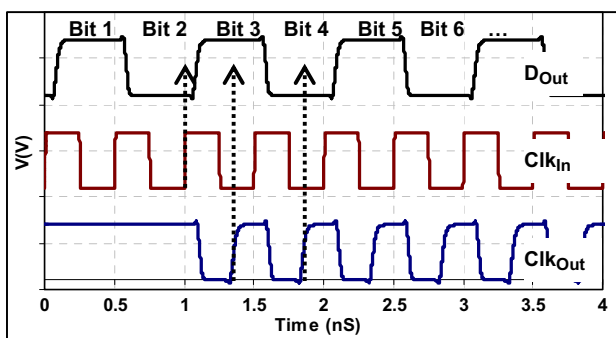
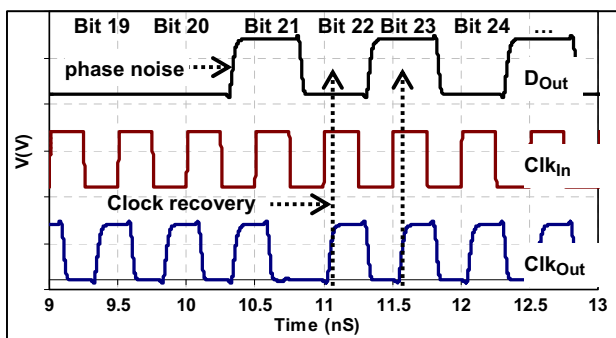


Figure 6. The input clock, output clock and data waveforms at 2 GBPS with input clock $\sim 0^\circ$ phase-shifted.



(a) Initial clock re-timing with the input clock $\sim -15^\circ$ phase-shifted.



(b) Clock re-timing after an input phase noise is injected at Bit 21.

Figure 7. Clock timing recovery after a 250 pS input phase noise is injected at Bit 21.

A major concern of using a digital circuit in clock re-timing might arise; that is the stability of the output clock frequency. To evaluate the stability of the output clock's frequency, simulations were carried out with a pseudo-random input stream (using 6-bit long pseudo-random patterns that are fed serially to the circuit). The output clock periods were laid out on top of one another in Figure 8.

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. This is an outstanding performance for an all-digital clock re-timing circuit.

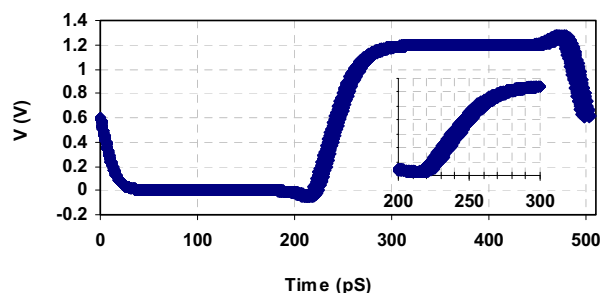


Figure 8. The output clock for a 6-bit long pseudo random input data. The inset shows the clock rising edge.

IV. CONCLUSIONS

An all-digital clock re-timing circuit for on-chip source synchronous serial links has been developed. The proposed circuit can re-time the clock with an NRZ data stream (90° phase-shifted) within two bit transitions. Simulation results show that the output clock frequency of the circuit is very stable with less than 10 pS peak-to-peak jitter. This jitter does not accumulate since the circuit continuously re-times the clock phase with each bit transition. The circuit has less than 100 gates (most of which are inverters). This makes it very compact, highly portable and can support NoC implementation with large number of serial links.

REFERENCES

- [1] L. Benini and G. D. Micheli, "Networks on chips: A new SoC paradigm", *IEEE Computer*, 35(1):70 – 78, January 2002.
- [2] B. Razavi, Ed. "Monolithic phase-locked loops and Clock Recovery Circuits," IEEE Press, 1996.
- [3] B. Razavi, "A 2.5-Gb/s 15-mW Clock Recovery Circuit," *IEEE JSSC*, Vol. 31-4, pp. 472-480, 1996.
- [4] S. Anand and B. Razavi, "A CMOS Clock Recovery Circuit for 2.5-Gb/s NRZ Data," *IEEE JSSC*, Vol. 36-3, pp. 432-439, 2001.
- [5] D-L. Chen, "A Power and Area Efficient CMOS Clock/Data Recovery Circuit for High-Speed Serial Interfaces," *IEEE Journal of Solid-State Circuits*, Vol. 31, No. 8, pp. 1170-1176, August 1996.
- [6] 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 Journal of Solid-State Circuits*, Vol. 35, No. 11, pp. 1581-1590, November 2000.
- [7] T-Y. Hsu *et al.*, "An All-Digital Phase-Locked Loop (ADPLL)-Based Clock Recovery Circuit," *IEEE JSSC*, Vol. 34-8, pp. 1063-1073, 1999.
- [8] M. Combes *et al.*, "A Portable Clock Multiplier Generator Using Digital CMOS Standard Cells," *IEEE JSSC*, Vol. 31-7, pp. 958-965, 1996.
- [9] 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*, Vol. E85-C, No. 5, P. 1170, May 2002.
- [10] M. E. S. Elrabaa, "A New Static Differential CMOS Logic with Superior Low Power Performance," *Analog Integrated Circuits and Signal Processing*, Vol. 43, No. 2, pp. 183-190, May 2005.