Fast Asynchronous Bit-Serial Interconnects for Network-on-Chip

Rostislav (Reuven) Dobkin, Israel Cidon, Ran Ginosar, Avinoam Kolodny, Arkadiy Morgenshtein
VLSI Systems Research Center, Electrical Engineering Department
Technion – Israel Institute of Technology
Haifa, Israel
[ran@ee.technion.ac.il]

Abstract—**The multiple wires required for on-chip bit-parallel interconnect in large systems on chip (SoC) occupy large chip area and present a significant capacitive load. The problem is exacerbated in Networks-on-Chip (NoC), which employ numerous multi-bit links with widely varying throughput demands, activity rates and standby periods. We approach this challenge with high speed on-chip serial interconnects. Conventional differential-signaling serial link circuits, typically employed for chip-to-chip I/O communications, are inappropriate for on-chip serial links, since they require complex clock and data recovery PLL-based circuits, which consume excessive power and area. Instead, we investigate low power asynchronous data transfer techniques, based on low voltage differential transition signaling. A novel bit-serial interconnect structure, comprising encoder, serializer, de-serializer and decoder circuits, is described and analyzed. Low latency synchronizers are employed at each end, enabling seamless connectivity of separate clock domains. Asynchronous NoC routers can be inserted along the link in a modular fashion, rendering this design useful for scalable NoC architectures. Dynamic and leakage power, as well as area, compare favorably with other link architectures.**

Index Terms—NoC, Serial Interconnect, Asynchronous Circuits, Differential Signaling, Low-Swing, Dual-Rail, Two-Phase, 1-of-2 Encoding, NRZ encoding**.**

I. Introduction

Large Systems on Chip (SoC), comprising a large number of modules, typically require multiple long on-chip data channels that interconnect far-away modules. Bit parallel data links provide high data rates at the cost of large chip area, routing difficulty, and high dynamic power. In addition, such links are often utilized over only a small portion of the time, but dissipate leakage power at all times. Leakage is incurred at the line drivers and also at the repeaters, which are often necessary for long interconnects.

Bit-serial interconnects address the issues of chip area, routability, and leakage power, since there are fewer wires, fewer line drivers, and fewer repeaters. However, data rate is reduced due to serialization, and hence serial links can be employed only when throughput requirements are not violated. In this work we investigate bit-serial links for their low area and low power advantages.

Large SoCs often employ multiple clocks. Transmission over long data links could cross clock domains and usually requires synchronization, whether the link is bit-parallel or serial. A common timing mechanism for serial interconnects injects a clock into the data stream at the transmitting side and recovers the clock at the receiver. Such clock-data recovery (CDR) circuits often require a power-hungry PLL, which also takes a while to converge on the proper clock frequency and phase at the beginning of each transmission. And, if the receiver and transmitter operate in different clock domains, the transaction must also be synchronized at the target, incurring additional time delay and power. Alternatively, an asynchronous data link employs handshake instead of clocks. Traditional asynchronous protocols are relatively slow due to Return-to-Zero (RTZ) requirement and the need to acknowledge transitions [1]. In this paper we consider non-RTZ (NRZ) protocols that do not require acknowledgement, improving throughput and latency of the data link.

Reduced voltage swing has been proposed for various types of data links, principally as a means for power reduction [2][3]. However, low swing may adversely affect noise immunity, and should be considered carefully.

Another commonly used technique for enhancing speed and noise immunity of interconnect involves differential (dual-rail) signal wires. This should also be carefully considered for on-chip applications, and compared with wider single-ended wires placed at larger inter-wire spacing.

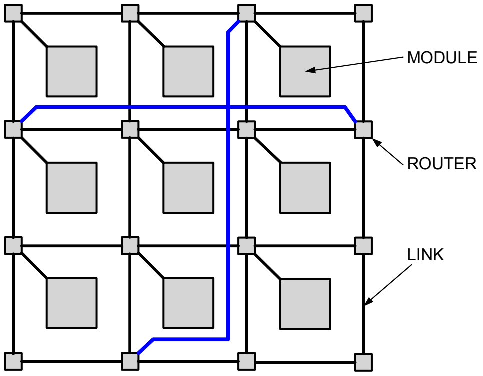

Figure 1: A NoC Mesh Architecture with Long-range Links

Networks on Chip (NoC) have been proposed as an efficient means to manage on-chip communications [4]–[6]. They typically employ routers and data links that interconnect the modules of the SoC (Figure 1). Data links

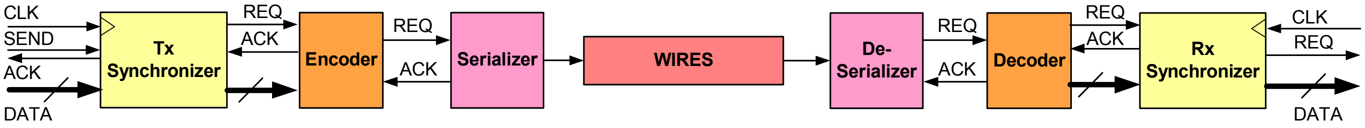

Figure 2: Fast Asynchronous Serial Link Architecture

are spatially reused (saving area), and the need for designing ad-hoc links is minimized (saving design effort and time). We seek solutions that are area- and leakage power-efficient. Bit serial links appear to be most appropriate for the majority of NoC links. Note that these links interconnect NoC routers, rather than computational modules. The NoC in Figure 1 combines normal links arranged in a 2D mesh with some longer range ones.

Using low-voltage differential pairs for on-chip interconnect was discussed in [7][8], where data was sampled at the receiver without any attention to synchronization issues. A three level voltage swing was presented in [9], requiring non-standard amplifiers.

We investigate fast differential, low-power, asynchronous bit-serial links for on-chip long distance interconnects, eliminating the clock generator and the CDR circuit. Architectural alternatives and considerations are discussed in Section II and the circuit implementation is discussed in Section III.

II. Architectural Considerations

Communications over the serial link occur at four levels (Figure 2): Synchronization, encoding, serialization, and the physical link. First, low-latency synchronization is applied: Data is synchronized at the receiver, and an acknowledgment signal is synchronized at the transmitter. Next, the encoder converts the (bit-parallel) word into the codes that will be sent over the serial lines. All this is conducted relatively slowly in a bit-parallel fashion. Last, the parallel word is serialized and driven into the wires. At the receiving end this order is reversed: A deserializer gathers received bits into parallel words, the decoder retrieves the original word, and the synchronizer inserts that word into the receiving clock domain. The synchronizer-encoder-serializer at the transmitter and the deserializer-decoder-synchronizer at the receiver perform complete Req/Ack handshake, but communication at the physical layer over the link wires is performed without any acknowledgement.

Multiple architectural and circuit alternatives are available at each level. At the physical level, data can be transferred over either single or differential wires, at either full or low voltage swing. Differential signaling suppresses common mode noise and may provide for faster communications. However, it requires more area and possibly more power (dynamic and leakage) relative to single ended signaling. Extra spacing and wire width on single wires may provide noise immunity comparable to differential lines, and they should be carefully evaluated. Low voltage swing is feasible with just two power rails when using differential signaling and sense amplifiers [10]. Other methods are described in [11][12]. Alternatively, quad power rails may be used for low swing [3]. While using low-swing, equalization may be applied in order to overcome channel saturation for certain data patterns.

A number of codes have been developed for asynchronous signaling. Bundled data is inappropriate, since it is sensitive to the relative skew of the Req and data lines, and since it requires about 1.5 transitions per bit (one on Req, one half on average on the data line). Four phase (RTZ) dual rail requires two transitions per data bit, incurring them sequentially and thus limiting the speed. We consider the two-phase NRZ Level Encoded Dual Rail (LEDR) code [13][14], which requires only one transition per bit, thus dissipating about one third less dynamic power than a bundled data code. It also achieves higher reliability than other codes [15]. The drawbacks of the LEDR code include a possible asymmetric transition count over the two rails, the need for an encoder and a decoder, and the risk of interline skew which may limit the data rate.

It is also possible to employ source-synchronous signaling similar to those used for chip-to-chip interconnects [16]−[20]. The principal disadvantage of such methods lies in the need to embed a very fast clock in the data at the transmitter and to extract it using a PLL-based CDR circuit at the receiver. These circuits dissipate high power, require large area and are therefore less attractive for on-chip interconnect.

There are also many different methods for synchronizing the data link to the transmitter and receiver clock domains. Synchronization in the context of globally-asynchronous, locally-synchronous (GALS) SoCs has been investigated in [21]−[25], and a low-latency solution is described in [26]. Since synchronization is performed on the parallel words, it is conducted at a much slower rate than the signaling rate on the serial lines.

The architecture of a fast on-chip serial interconnect is shown in Figure 2. The circuits are described in the next section. The design goal is to achieve the fastest throughput possible. The circuits present a lower bound on the bit time of one gate delay, and our goal is to investigate how close we can get to that target.

III. Fast On-chip Serial Interconnect Architecture

A. *Transmitter*

The transmitter consists of a pipeline comprising the synchronizer, encoder, encoder registers and the serializers. The encoder stores its output in four registers, containing S, P and their complements (Figure 3). Starting at the encoder's output, the data lines are handled differentially in dual rails.

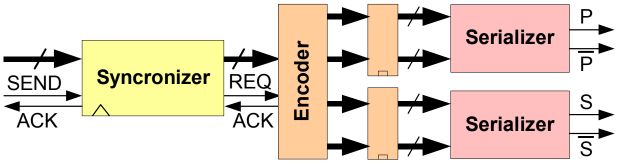

Figure 3: Transmitter Architecture

The LEDR code we employ is defined as follows. A serialized sequence B(i) of bits to be sent over the link is encoded into a sequence S(i), P(i) of State and Phase bits. S(i)=B(i) for all i. Let's assume P(0)=0. If S(i+1)=S(i) then P(i+1) is the inverse of P(i), otherwise P(i+1)=P(i). LEDR encoding is performed in parallel on an entire M-bit word, and S, P and their complements are stored in four M-bit registers (Figure 3).

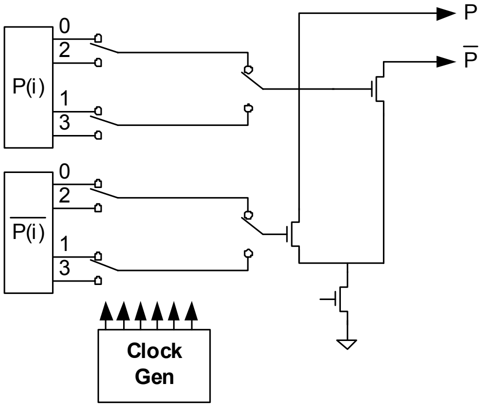

Figure 4: 4-bit Word Input-Multiplexed Serializer [27]. A similar circuit drives the S lines

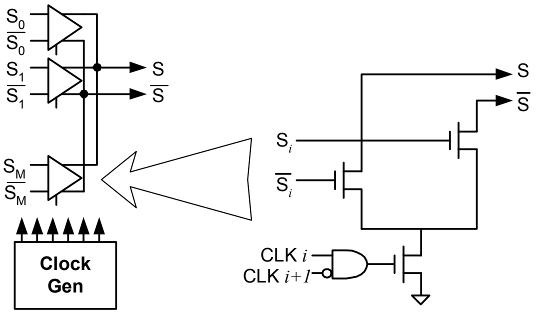

Figure 5: Output-Multiplexed Serializer [27]
A similar circuit drives the P lines

We investigate two approaches to data serialization – input and output multiplexing [27]. An input-multiplexed serializer is shown in Figure 4. The encoded P and S bits and their complements are serialized and then used to drive the two differential line drivers. This circuit uses only two differential drivers, as opposed to the output-multiplexed architecture (Figure 5). In the output-multiplexed scheme, there is a separate differential line driver for each encoded bit (a total of 2·M drivers). The outputs of all M drivers are connected directly to the lines, and only one of them is enabled at a time. The serializing multiplexers in Figure 4 and the drivers in Figure 5 are enabled sequentially with a fast multiphase clock generator. The blow-out schematics of a single driver (Figure 5) shows that the driver is enabled by the conjunction of two successive phases of the clock. The multiphase clock generator employs adjustable delay lines, and it controls the data rate over the serial lines. The relative merits of the two serializer architectures in fast technology are being investigated.

B. *Receiver*

On the receiver side, the differential lines terminate in sense amplifiers. Thanks to the two-phase dual rail NRZ LEDR encoding, only one of the differential pairs makes a transition per bit. The sense amplifiers (Figure 6) reconstruct the full-swing transition, triggering the subsequent de-serializer and decoder.

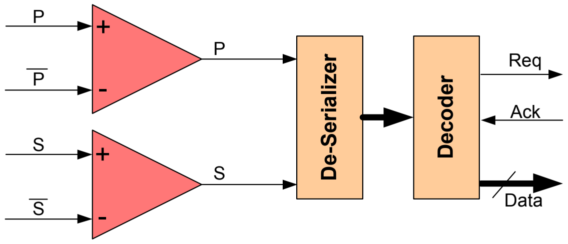

Figure 6: Receiver Architecture

The simplest de-serializer consists of a fast shift register, or an asynchronous pipeline. It is driven by an asynchronous input, and bits propagate along the pipe driven by transitions [28]. However, to enable the fastest possible pipeline, there is no time for a complete handshake including acknowledgement. The control signal is a dual rail transition, generated by a dual rail XOR gate that monitors the S, P inputs (Figure 7). A speed-optimized transition latch is shown in Figure 8. All delays along the pipe are carefully balanced. Data (S) and control transitions propagate in a wave-pipelined manner, and are sensitive to skew and delay variations. Once all bits of a packet or flit have been stored, bit-parallel decoding can take place at a slower rate. If a second (parallel) pipeline is added to Figure 7, P bits may also be recorded, enabling codes other than LEDR.

Two other alternatives are considered. The incoming bit stream may be toggled between two pipelines, and each pipeline operates at half the rate. Also, the deserializer and decoder may be combined in a tree-structured asynchronous FSM.

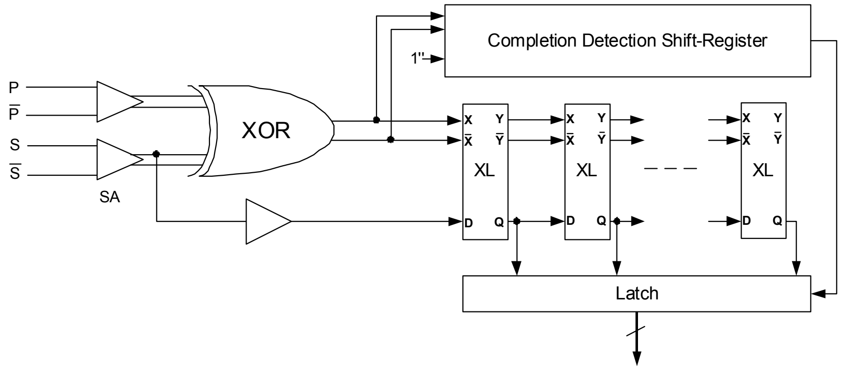

Figure 7: Fast Transition Based Pipeline

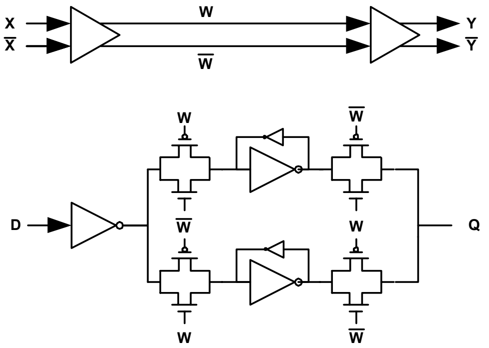

Figure 8: Transition Latch (XL)

IV. Conclusion

We presented an asynchronous approach for SoC and NoC interconnects that provide clockless, low-power and low-area solution. Different implementations were discussed and analyzed. Novel architecture and circuits, using LEDR codes and differential signaling, have been presented for asynchronous receivers that trade off a complete speed-independent handshake for speed of operation.

References

[1] J. Teifel and R. Manohar, "A High-Speed Clockless Serial Link Transceiver," Proc. ASYNC, pp. 151-161, 2003.

[2] C. Svensson, "Optimum Voltage Swing on On-Chip and Off-Chip Interconnect," J. Solid-State Circuits, 36(7), pp. 1108-1112, July 2001.

[3] R.K. Krishnamurthy, I. Lys and L.R. Carley, "Static power driven voltage scaling and delay driven buffer sizing in Mixed Swing QuadRail for sub-1 V I/O swings," Proc. ISLPED, pp. 381-386,. 1996.

[4] W. J. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection Networks," Proc. DAC, June 2001.

[5] E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny, "QoS Architectrure and Design Process for Cost Effective Networks on Chip", J. Systems Architecture, 50(2-3), pp. 105-128, Feb. 2004.

[6] E. Bolotin, I. Cidon, R. Ginosar and A. Kolodny, "Cost considerations in Network on Chip", Integration - The VLSI Journal, 2004.

[7] I. Saastamoinen, T. Suutari, J. Isoaho and J. Nurmi, "Interconnect IP for gigascale system-on-chip," Proc. Euro. Conf. Circuit Theory and Design (ECCTD), Aug. 2001.

[8] T. Suutari, J. Isoaho and H. Tenhunen, "High-speed Serial Communication with Error Correction Using 0.25μm CMOS Technology," Proc. ISCAS, pp. IV-618-621, May 2001.

[9] C. Svensson and J. Yuan, "A 3-Level Asynchronous protocol for a Differential Two-Wire Communication Link," J. Solid State Circuits, 29(9), Sept. 1994.

[10] M. Sinha, S. Hsu, A. Alvandpour, W. Burleson, R. Krishnamurthy, and S. Borkar, "High-performance and low-voltage sense-amplifier techniques for sub-90nm SRAM," Proc. Int. SOC Conf., pp. 113-116, Sept. 2003.

[11] H. Kaul and D. Sylvester, "Transition aware global signaling (TAGS)," Proc. Int. Symp Quality Electronic Design, pp. 53-59, 2002.

[12] A. Tangel and K. Choi, "The CMOS Inverter as a comparator in ADC designs," Proc. Electrical Electronics Engineering, pp. 1-5, Nov. 2001.

[13] M.T. Dean, T. Williams *et al.* "Efficient Self-Timing with Level-Encoded 2-Phase Dual-Rail (LEDR)," Proc. ARVLSI, pp. 55-70, 1991.

[14] D.H. Linder and J.C. Harden, "Phased Logic: Supporting the Synchronous Design Paradigm with Delay-Insensitive Circuitry," IEEE Trans. Computers 45(9), pp. 1031-1044, 1996.

[15] F. Worm, P. Ienne and P. Thiran, "Soft Self-Synchronizing Codes for Self-Calibrating Communication," Proc. ICCAD, 2004.

[16] S. Sidiropoulos, "High Performance Inter-Chip Signaling," Tech. Rep. CSL-TR-98-760, Stanford Univ., 1998.

[17] C.K.K. Yang, "Design of High-Speed Serial Links in CMOS", PhD Thesis, Stanford University, 1998.

[18] W.F. Ellersick, "Data Converters for High Speed CMOS Links," PhD Thesis, Standford Univ., 2001.

[19] H.O. Johansson, J. Yuan and C. Svensson, "A 4 Gsamples/s Line-Receiver in 0.8 um CMOS," Proc. Int. Symp. VLSI Circuits, pp. 116-117, 1996.

[20] C. Svensson and J. Yuan, "High Speed CMOS Chip to Chip Commication Circuit," Proc. ISCAS, pp. 2228-2231, 1991.

[21] J. Kessels, A. Peeters, P. Wielage and S.-J. Kim, "Clock Synchronization through Handshake Signaling," Proc. ASYNC, pp. 59-68, March 2002.

[22] T. Villiger, H. Kaeslin, F.K. Gürkaynak, S. Oetiker and W. Fichtner, "Self-Timed Ring for Globally-Asynchronous Locally-Synchronous Systems," Proc. ASYNC, pp. 141-150, April 2003.

[23] J. Muttersbach, T. Villiger and W. Fichtner, "Practical Design of Globally-Asynchronous Locally-Synchronous Systems," Proc. ASYNC, pp. 52-61, April 2000.

[24] S. Moore, G. Taylor, R. Mullins and P. Robinson, "Point to Point GALS Interconnect," Proc. ASYNC, April 2002.

[25] S. W. Moore, G. S. Taylor, P. A. Cunningham, R. D. Mullins and P. Robinson, "Self-Calibrating Clocks for Globally Asynchronous Locally Synchronous Systems," Proc. Int. Conf. Computer Design (ICCD), 2000.

[26] R. Dobkin, R. Ginosar and C. P. Sotiriou, "Data Synchronization Issues in GALS SoCs," Proc. ASYNC, April 2004.

[27] M.J.E. Lee, "An Efficient I/O and Clock Recovery for TERABIT Integrated Circuits Design," PhD Thesis, Stanford Univ., 2001.

[28] I. E. Sutherland, "Micropipelines," Comm. ACM, 32(6), pp. 720-738, 1989.