Robust Two-Phase RZ Asynchronous SoC Interconnects

Muhammad E. S. Elrabaa

***Abstract*—A novel two-phase RZ delay-insensitive asynchronous handshaking protocol for on-chip communication has been developed along with an efficient and robust dual-rail circuit implementation (Transmitter/Receiver). Performance was verified using SPICE simulations with a 0.13 μm, 1.2 V technology and compared to that of the best-in-class asynchronous transceivers in terms of forward and backward latencies, throughput, energy per bit transfer and design complexity. Results demonstrate the superior overall performance of the new transceiver.**

***Index Terms*—Asynchronous interconnects, CMOS digital integrated circuits, networks-on-chip (NoC), systems-on-chip (SoC).**

I. INTRODUCTION

Recently, systems-on-chip (SoC) designs have grown in complexity to include not only multiple clock domains but also a wide range of blocks (IPs) with various data communication needs and patterns. To satisfy the communication needs of SoCs while maintaining reasonable design and timing closure times, two new interconnect paradigms have recently emerged; networks-on-chip (NoCs) [1]–[3] and globally asynchronous locally synchronous (GALS) systems [4], [5]. Both types of interconnect system share the common problem of designing the point-to-point interconnect circuitry between routers and/or IP blocks.

Fully asynchronous interconnects are systems that can adapt to a wide range of temperature, process and voltage variations, as well as varying data rates, making them ideal for implementing the point-to-point links in NoCs and GALS. However, the main concerns with asynchronous interconnects are high latency and low throughput due to the handshaking required to implement the automatic control of data rate. Pipelining using repeaters (transceivers) is usually employed to improve throughput (and to some degree latency by providing buffering) in long interconnects.

The main objective of this work was to develop a robust asynchronous transceiver that has low-latency, high throughput and very low design-overhead (i.e., that can be simply plugged-in as a hardware macro with no required optimizations). The transceiver that has been developed combines a new single-track handshaking protocol with an efficient delay-independent custom circuit implementation that maximizes throughput and minimizes latency.

A review of asynchronous interconnects is first presented in Section II. That review is limited only to pipelined on-chip asynchronous interconnects (with no data path blocks in between pipeline stages) rather than the broader topic of asynchronous pipelines. The proposed asynchronous transceiver is introduced in Section III followed by the circuit implementation in Section IV. Simulation results that verify the basic operation of these circuits, their robustness, and relative performance compared to other repeater circuits (both asynchronous and synchronous) are provided in Section V followed by conclusions in Section VI.

II. REVIEW OF ASYNCHRONOUS INTERCONNECTS

Asynchronous interconnects have been mainly based on three types of asynchronous circuit techniques: bundled data, delay insensitive (DI) (or quasi delay insensitive QDI), and single track circuits. They all employ one of two types of handshaking and signaling protocol for data transfers: four-phase return-to-zero (RZ) signaling or two-phase non-return-to-zero (NRZ) signaling. For the first type, each and every signal starts and ends in the same state requiring four-steps for a request-acknowledgement cycle to transfer a datum. For the second type, signal transitions indicate a request or acknowledgement resulting in data transfer in two steps only. Forward latency is the time required to transfer a datum from one stage to the next. Backward latency is the time required to transfer an acknowledgement (or spacer) backward. Both latencies determine the cycle time and hence the bandwidth.

Bundled data channels utilize separate channels for data and control (handshaking). Latch control circuits latch the data at each stage when a request is received while the next stage has already acknowledged receipt of previous data [6]. A strict single-sided timing constraint must be met; the request signal must arrive after the data. Additional timing constraints may arise due to the specific circuit implementation. Lately, a new bundled data channel (MOUSETRAP) with an improved throughput and latency was proposed [7]. Latches are normally open/transparent and are closed after the data has passed through them adding more timing constraints and design complexity. Having open latches also leads to the propagation of spurious transitions and higher power consumption [8].

DI or QDI channels utilize special data encoding (dual-rail or 1-in-N) that allows receivers to detect the arrival of new data with no need for explicit request signals (and the associated timing constraints). A receiver acknowledges data reception by activating a backward acknowledgement signal. When combined with conventional four-phase RZ signaling DI or QDI schemes yield very simple and robust circuit implementations [9], [10]. Two-phase NRZ DI codes have been proposed [11], [12] to reduce latency with an increase in circuit complexity. To increase the wire efficiency (number of data bits per wire) N-of-M codes were proposed in [13] at the expense of more circuit complexity. Though DI encoding is suppose to eliminate timing constraints, specific implementations may lead to some constraints [11], [12].

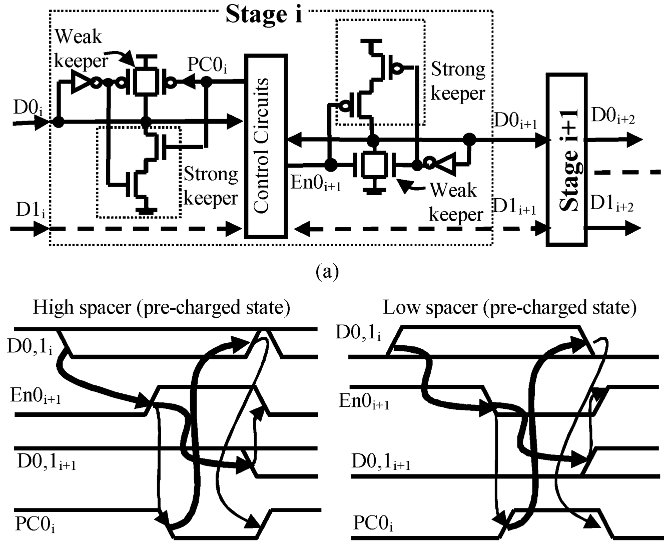
Single-track handshaking was introduced by Van Berkel *et al.* in [14] as an improvement upon bundled data channels. A single wire is used for both request and acknowledgement signals. A transmitter induces a transition on the wire indicating a request and then releases it, the receiver acknowledges data receipt by transitioning the wire back to its initial state (i.e., RZ) and then releasing it, combining two-phase handshaking with RZ signaling. This technique still suffers from the bundled data constraints in addition to new timing constraints pertaining to the transitioning of the wire and releasing it. A faster single-track circuit, GasP, was proposed in [15] where a self-resetting NAND gate (which resets after three gate transitions) controls the data latch. Thus, two new timing constraints are added; data must be read within three gate transitions and so must acknowledgement.

In [16] single-track control is extended to 1-of-N encoded channels. The proposed single-track full buffer (STFB) eliminates the acknowledge line that existed in previous QDI buffers. However, several tight timing constraints are imposed on internal and external signals' switching order. With the same number of transitions per cycle as GasP, STFB achieved significantly lower forward latency ($\sim$60%) at the expense of higher backward latency and lower throughput ($\sim$70%).

Manuscript received October 13, 2009; revised December 20, 2009. This work was supported by King Fahd University of Petroleum and Minerals (KFUPM) Grant IN070367.

The author is with the Computer Engineering Department, KFUPM, Dhahran 31261, Saudi Arabia (e-mail: elrabaa@kfupm.edu.sa).

Digital Object Identifier 10.1109/TVLSI.2010.2042240



(a)

(b)

Fig. 1. Block diagram of the proposed repeater and the handshaking protocol. (a) A Block diagram of the repeater showing the data and precharging drivers and the keeper circuits for the high-spacer encoding. (b) The proposed handshaking protocol. D0, 1$_i$ signals represent the data transfer initiated at the ith stage. Thick lines indicate signal trips across the wire segment. The protocol is shown for two types of data and spacer encoding with transitions on D0$_i$.

III. Proposed Asynchronous Transceiver

As was explained in Section II, DI or QDI techniques, depending on their implementation, can significantly reduce or eliminate timing constraints. Also, if combined with single track techniques, two-phase handshaking and RZ signaling can be realized. This would improve the bandwidth significantly and simplify the circuit implementation greatly.

Fig. 1 shows a block diagram of the newly proposed dual-rail transceiver (repeater) along with the proposed handshaking protocol. With no dedicated control lines, RZ signaling is combined with two-phase handshaking by driving each data line from the transmitter's side of a stage and returning it to its initial state (i.e., precharged or spacer state) from the receiver's side. The proper sequence of signaling is achieved by two control circuits in the repeater; an Enable control circuit that controls the driving of the output and a precharge control circuit that controls the precharging of the input. When data initiated at stage $\boldsymbol{i}$ is received at stage $\boldsymbol{i+1}$, an enable signal is asserted (En) which initiates data transfer to the $\boldsymbol{i+1}$th segment and activates a precharging signal (PC) that would then precharge the $\boldsymbol{i}$th segment, overlapping the transfer of data to the $\boldsymbol{i+1}$th segment with the precharging of the $\boldsymbol{i}$th segment and completing the data transfer cycle in two *trips*. The protocol is illustrated for two types of spacer state; a high-spacer encoded as both lines high [corresponding to the block diagram in Fig. 1(a)], and a low-spacer encoded as both lines low. Data is encoded by pulling one of the lines low or high, respectively.

As Fig. 1 shows, there are two sets of keepers on each side of a wire segment; strong keepers and weak keepers. Strong keepers at the input hold the input low (when input data arrives) until the repeater reads the data and activates the PC signal. Strong keepers at the output hold the output in the precharged state until new data arrives. When an output is driven low the weak keeper (at the output of the sender) keeps it low until the precharging circuit in the next stage precharge it (by overcoming the weak keeper) and hold it in that state by its own weak input keeper. Hence, data lines are held in the active state by strong keepers at the receiving repeaters, precharged by the receiving repeaters and held in the precharged state by strong keepers at the sending repeaters ensuring that wires are always strongly driven and preventing erroneous switching due to cross talk or SETs. Also no contention exists between any of the driving circuits and the strong keepers at all times.

IV. Circuit Implementation

Fig. 2 shows the signal transition graph (STG) of the transceiver and the circuit implementation for one of the data lines $(\mathrm{D}0_i)$. The STG/circuits associated with the other data line are identical. The enable and precharge control circuits, Fig. 2(b)–(e), behave similarly to a Muller-C element. The En signal is asserted when input data arrives $(\mathrm{D}0_{\mathbf{i}} \downarrow)$ while both transceiver outputs $(\mathrm{D}0_{\mathbf{i+1}}$ and $\mathrm{D}1_{\mathbf{i+1}})$ are high. It is then deasserted when the output $\mathrm{D}0_{\mathbf{i+1}}$ is discharged. A half keeper holds En low when input data is in the precharged (high) state. The PC signal is deasserted when $\mathrm{D}0_{\mathbf{i}}$ is low and En becomes high causing the input $\mathrm{D}0_{\mathbf{i}}$ to be charged back high. This, in turn causes the PC signal to go back high. A keeper keeps the PC signal low in case En signal goes low before the input is charged high and keeps it high if $\mathrm{D}0_{\mathbf{i}}$ goes low while En is still low (due to backpressure at the output).

The widths of the En and PC pulses are automatically set by the timing behavior of the data lines. The only timing assumption here is that En should not go low (which would require at least three gate transitions) until PC goes low (one gate transition). Even this very-easy-to-meet assumption can be eliminated by forcing En to wait for PC to go low before it goes low as indicated in Fig. 2. Circuit modifications for alternate data and spacer encoding or 1-of-N data encoding are indicated throughout Fig. 2. Designers can choose the implementation that yield the best performance for the technology they use.

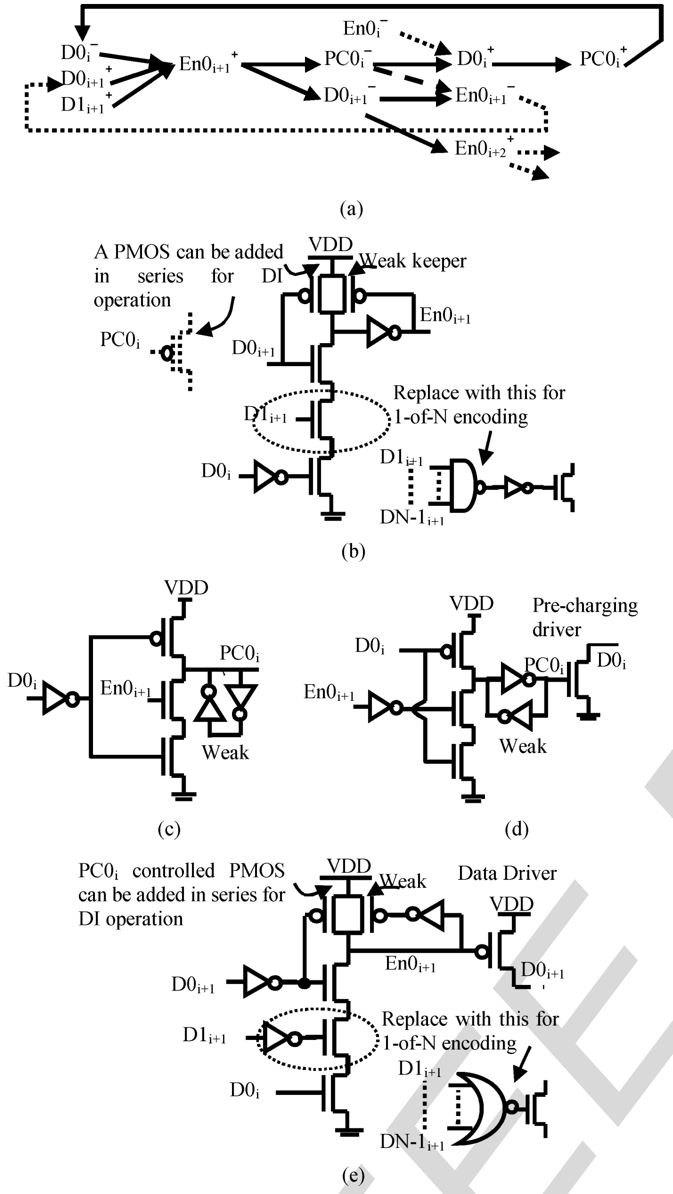
V. Circuit Performance

A. Basic Operation

Spice simulations using a 0.13-μm, 1.2 V CMOS technology were used to verify the operation of the proposed repeater. The simulated pipeline comprised a data producer followed by three stages of repeaters followed by a data consumer with 50 μm wires in between and 500 Mbs data injection rate. Transistors were simply sized to achieve 50 and 100 ps fall and rise times, respectively (equivalent to a two-input, fan-out of three, NAND gate). Fig. 3 shows the simulation waveforms of one of the repeaters illustrating how the proper sequence of events on input data, En, PC signals, and output data is achieved with a forward latency of $\sim$ 130 ps. The complete pipeline was simulated as follows; first data is injected at a constant rate of 500 Mbs but is not being consumed [see Fig. 4(a)], then the consumer starts consuming the data at the same injection rate, Fig. 4(b). The pipeline gets filled after the injection of three data bits (all stages' inputs and outputs D1-D4 are low) and no new data can be injected. As soon as the consumer starts consuming data (as evident from the precharging of D4) the data starts moving forward along the pipeline. The backward latency is $\sim$ 150 ps making the minimum cycle time $\sim$ 280 ps (i.e., maximum throughput of 3.5 Gbs).

B. Maximum Throughput

The throughput as a function of wire segment length has been evaluated using a multirepeater pipeline with specially designed data producer and consumer circuits that inject/consume data at the maximum possible rate. The producer injects new data every time the first stage's input is precharged while the consumer precharges the last stage's output every time it goes low. Fig. 5 shows the data throughput and latency per stage versus wire length. As expected, latency increases linearly with wire length while throughput decreases as $1/x$. Even at a wire length of 1000 μm the throughput is still



(a)

(b)

(c)

(d)

(e)

Fig. 2. Circuit implementation of the proposed two-phase RZ protocol. (a) Signal transition graph (STG) of the proposed repeater. The dashed transition from $\mathrm{PC0}_{\mathrm{i}}^{-}$ to $\mathrm{En0}_{\mathrm{i+1}}^{-}$ can be added to remove any timing assumption. (b) Enable control circuit for 1-of-2 and 1-of-N Data encoding. (c) Precharging control circuit. (d) PC circuit for low spacer. (e) Enable control circuit for low spacer.

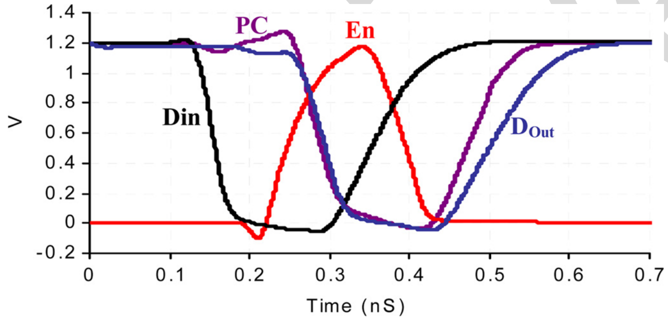
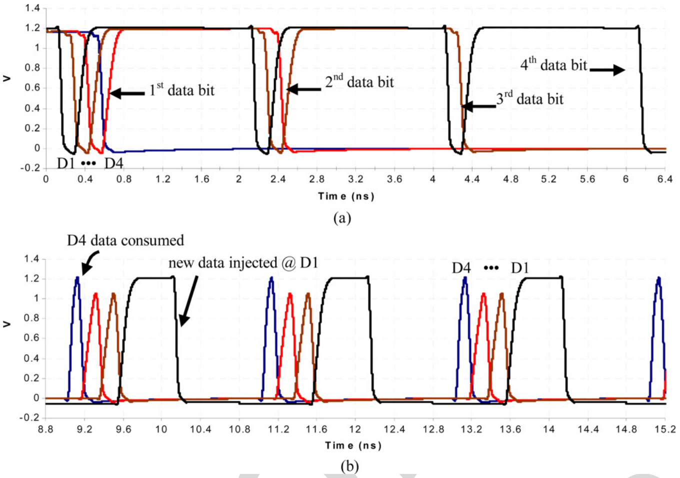


Fig. 3. Signal waveforms of one of the repeater stages.



(a)

(b)

Fig. 4. Data waveforms along the asynchronous pipeline for the following scenario: (a) data being injected but not consumed and the pipeline gets filled after three data bits; (b) consumer starts consuming the data causing it to move forward along the pipeline.

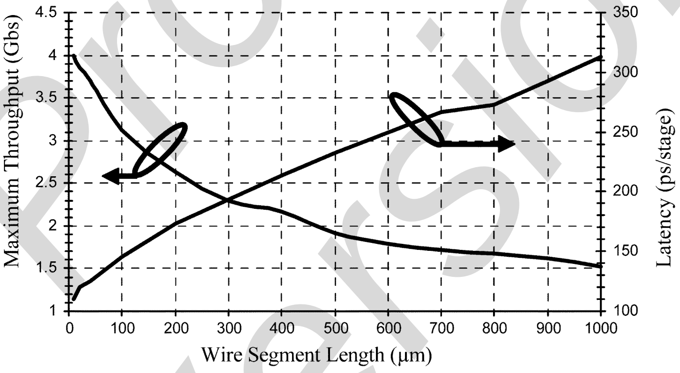


Fig. 5. Maximum throughput and latency per stage versus wire length.

relatively high (1.5 Gbs). At 0 wire length latency and throughput are 110 ps and 4 Gbs, respectively.

C. Robustness

To demonstrate the robustness of the new circuits, simulations were performed with two pipelines; an attacker and a victim. An inter-stage wire length of 1000 μm was used with a cross capacitance between the attacker and victim equivalent to half the total wire capacitance. This cross capacitance represents the worst case value under an attack by a single attacker. The attacker attacks the victim under the two most vulnerable conditions; when the victim is in the precharged state and when it is in the evaluation state (low output). Other attacks (during switching) would either slow down the victim or speed it up. As Fig. 6 shows, the repeater is immune to the cross talk noise due to the keepers' strong drive. Subsequent pipeline stages are not affected at all.

D. Comparison With Other Repeaters

The performance of the proposed repeater has been compared to that of the following asynchronous repeaters: the dual-rail DI buffer described in [12] (called LETS) with MOUSETRAP [7] pipelining style, the dual-rail (1-of-2) STFB [16] and GasP [15]. These were chosen for having the best performance published so far. The performance of two conventional repeaters; a two-phase QDI asynchronous repeater and a synchronous repeater (a CMOS FF), were also evaluated to provide reference points. The sizes of all repeaters were optimized

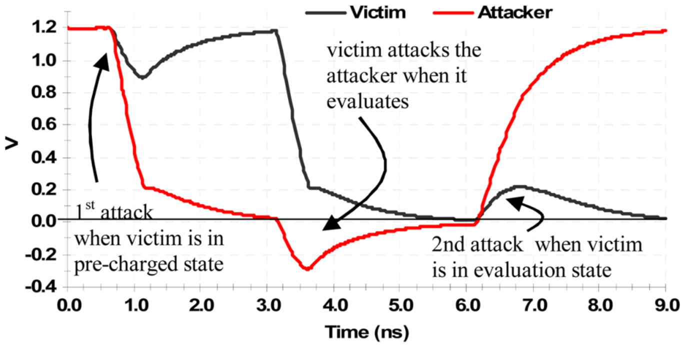


Fig. 6. Attacker and victim waveforms for two attack conditions.

TABLE I
PERFORMANCE COMPARISONS BETWEEN THE NEWLY PROPOSED REPEATER AND OTHER REPEATERS

Repeater Circuit	Forward Latency (ps/stage)	Backward Latency (ps/stage)	Maximum Throughput (Gbs)	Energy per Bit transfer (fJ/bit)	No. of Transistors
New two-phase RZ Repeater	145	170	3.17	75	60 $^{+}$
LETS [12] (MOUSETRAP-style)	120	310	2.33	60	62
STFB [16]	100	230	3.03	60	34
GasP [15]	170	160	3.03	125	34
Conventional two-phase QDI	410	100	1.96	150	70
Sync. With equal Cin	310	–	3.22 *	50 $^{\#}$	24 $^{++}$
Sync. With larger size	200	–	5 *	150 $^{\#}$	24

* Not accounting for clock skew or jitter or process variations
$^{\#}$ Not including energy in clock distribution network
$^{+}$ Out of these 32 transistors are used in the keepers for robustness
$^{++}$ Without the clocking circuitry

to yield maximum throughput with equal input capacitances (Cin) to ensure a fair comparison (equal loading on circuits that drive the repeaters). The value of Cin basically determines the sizes of transistors connected to the inputs. The remaining transistors were sized for maximum throughput. With the exception of the new repeater, all asynchronous transceivers did require more design effort and time to optimize for maximum throughput. A synchronous repeater with a larger size (i.e., Cin) and minimum achievable latency per stage was also compared. For each repeater, forward and backward latencies, maximum throughput and energy per bit (EPB) were measured at a wire segment length of 100 μm and using four-stage pipelines. EPB, obtained by integrating the instantaneous power, is the average for transferring a 0 bit and a 1 bit.

Table I summarizes the simulation results for all circuits including transistor count per repeater. The new repeater achieved the best balance between forward and backward latencies resulting in highest throughput. The STFB achieved the lowest forward latency but at the expense of higher backward latency. The LETS repeater achieved lower forward latency than the new repeater but had higher backward latency. GasP had a similar performance to the new repeater in terms of latencies and throughput but its EPB is much higher (its control track has twice the data frequency). The synchronous repeater achieved the smallest EPB at a throughput close to that of the new repeater (without considering clock skew and jitter). 1-of-4 encoding could reduce the EPB of the asynchronous transceivers at the expense of more latency due to higher gates' fan-in.

Though the new repeater had more transistors than the STFB, more than half of the transistors are in the keepers which were added for robustness. Similar keepers would have to be added to the STFB to enhance its robustness. The Gasp and synchronous repeaters naturally have lower transistor counts due to their single-rail data encoding.

VI. CONCLUSION

A new two-phase QDI asynchronous handshaking protocol that utilizes dual-rail RZ data encoding has been developed. A new asynchronous repeater that implements the new protocol has also been developed and tested using SPICE simulations. The RZ data signaling has allowed the circuit implementation to be simple yet very efficient and robust. Also, since the new protocol overlaps driving an output data segment with the precharging of the preceding one, the throughput has been maximized while minimizing the latency. The new repeater achieved a throughput equal to that of a synchronous repeater with less than half the latency. Compared to the best previously published asynchronous repeaters, it achieved a superior overall performance in terms of latency, throughput, energy per bit and design complexity.

REFERENCES

[1] L. Benini and G. D. Micheli, “Networks on chips: A new SoC paradigm,” *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[2] W. J. Dally and B. Towles, “Route packets, not wires: On chip interconnection networks,” in *Proc. 38th Des. Autom. Conf.*, Jun. 2001, pp. 684–689.

[3] J. Henkel, W. Wolf, and S. Chakradhar, “On-chip networks: A scalable, communication-centric embedded system design paradigm,” in *Proc. 17th Int. Conf. VLSI Des.*, 2004, pp. 845–851.

[4] D. M. Chapiro, “**<PLEASE PROVIDE DEPARTMENT.>** Globally-asynchronous locally-synchronous systems,” Ph.D. dissertation, Stanford Univ., Stanford, CA, Oct. 1984.

[5] S. Moore, G. Taylor, R. Mullins, and P. Robinson, “Point to point GALS interconnect,” in *Proc. 8th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2002, pp. 69–75.

[6] S. B. Furber and P. Day, “Four-phase micropipeline latch control circuits,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 4, no. 3, pp. 247–253, Mar. 1996.

[7] M. Singh and S. M. Nowick, “MOUSETRAP: High-speed transition-signaling asynchronous pipelines,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 6, pp. 684–698, Jun. 2007.

[8] M. Lewis, J. Garside, and L. Brackenbury, “Reconfigurable latch controllers for low power asynchronous circuits,” in *Proc. 5th Int. Symp. Async. Circuits Syst. (ASYNC)*, 1999, pp. 27–35.

[9] W. J. Bainbridge and S. B. Furber, “Delay insensitive system-on-chip interconnect using 1-of-4 data encoding,” in *Proc. 7th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2001, pp. 118–126.

[10] A. Lines, “Asynchronous interconnect for synchronous SoC design,” in *Proc. IEEE MICRO*, Jan.–Feb. 2004, pp. 32–41.

[11] M. E. Dean, T. E. Williams, and D. L. Dill, “Efficient self-timing with level-encoded 2-phase dual-rail (LEDR),” in *Proc. Univ. California/Santa Cruz Conf. Adv. Res. VLSI*, 1991, pp. 55–70.

[12] P. B. McGee, M. Y. Agyekum, M. A. Mohamed, and S. M. Nowick, “A level-encoded transition signaling protocol for high-throughput asynchronous global communication,” in *Proc. 14th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2008, pp. 116–127.

[13] W. J. Bainbridge, W. B. Toms, D. A. Edwards, and S. B. Furber, “Delay insensitive, point-to-point interconnect using M-of-N codes,” in *Proc. 9th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2003, pp. 132–141.

[14] K. Van Berkel and A. Bink, “Single-track handshaking signaling with application to micropipelines and handshake circuits,” in *Proc. 2nd Int. Symp. Async. Circuits Syst. (ASYNC)*, 1996, pp. 122–133.

[15] I. Sutherland and S. Fairbanks, “GasP: A minimal FIFO control,” in *Proc. 7th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2001, pp. 46–53.

[16] M. Ferretti and P. A. Beerel, “Single-track asynchronous pipeline templates using 1-of-N encoding,” in *Proc. Des., Autom. Test Eur. (DATE)*, 2002, pp. 1008–1015.

Robust Two-Phase RZ Asynchronous SoC Interconnects

Muhammad E. S. Elrabaa

***Abstract*—A novel two-phase RZ delay-insensitive asynchronous handshaking protocol for on-chip communication has been developed along with an efficient and robust dual-rail circuit implementation (Transmitter/Receiver). Performance was verified using SPICE simulations with a 0.13 μm, 1.2 V technology and compared to that of the best-in-class asynchronous transceivers in terms of forward and backward latencies, throughput, energy per bit transfer and design complexity. Results demonstrate the superior overall performance of the new transceiver.**

***Index Terms*—Asynchronous interconnects, CMOS digital integrated circuits, networks-on-chip (NoC), systems-on-chip (SoC).**

I. Introduction

Recently, systems-on-chip (SoC) designs have grown in complexity to include not only multiple clock domains but also a wide range of blocks (IPs) with various data communication needs and patterns. To satisfy the communication needs of SoCs while maintaining reasonable design and timing closure times, two new interconnect paradigms have recently emerged; networks-on-chip (NoCs) [1]–[3] and globally asynchronous locally synchronous (GALS) systems [4], [5]. Both types of interconnect system share the common problem of designing the point-to-point interconnect circuitry between routers and/or IP blocks.

Fully asynchronous interconnects are systems that can adapt to a wide range of temperature, process and voltage variations, as well as varying data rates, making them ideal for implementing the point-to-point links in NoCs and GALS. However, the main concerns with asynchronous interconnects are high latency and low throughput due to the handshaking required to implement the automatic control of data rate. Pipelining using repeaters (transceivers) is usually employed to improve throughput (and to some degree latency by providing buffering) in long interconnects.

The main objective of this work was to develop a robust asynchronous transceiver that has low-latency, high throughput and very low design-overhead (i.e., that can be simply plugged-in as a hardware macro with no required optimizations). The transceiver that has been developed combines a new single-track handshaking protocol with an efficient delay-independent custom circuit implementation that maximizes throughput and minimizes latency.

A review of asynchronous interconnects is first presented in Section II. That review is limited only to pipelined on-chip asynchronous interconnects (with no data path blocks in between pipeline stages) rather than the broader topic of asynchronous pipelines. The proposed asynchronous transceiver is introduced in Section III followed by the circuit implementation in Section IV. Simulation results that verify the basic operation of these circuits, their robustness, and relative performance compared to other repeater circuits (both asynchronous and synchronous) are provided in Section V followed by conclusions in Section VI.

II. Review of Asynchronous Interconnects

Asynchronous interconnects have been mainly based on three types of asynchronous circuit techniques: bundled data, delay insensitive (DI) (or quasi delay insensitive QDI), and single track circuits. They all employ one of two types of handshaking and signaling protocol for data transfers: four-phase return-to-zero (RZ) signaling or two-phase non-return-to-zero (NRZ) signaling. For the first type, each and every signal starts and ends in the same state requiring four-steps for a request-acknowledgement cycle to transfer a datum. For the second type, signal transitions indicate a request or acknowledgement resulting in data transfer in two steps only. Forward latency is the time required to transfer a datum from one stage to the next. Backward latency is the time required to transfer an acknowledgement (or spacer) backward. Both latencies determine the cycle time and hence the bandwidth.

Bundled data channels utilize separate channels for data and control (handshaking). Latch control circuits latch the data at each stage when a request is received while the next stage has already acknowledged receipt of previous data [6]. A strict single-sided timing constraint must be met; the request signal must arrive after the data. Additional timing constraints may arise due to the specific circuit implementation. Lately, a new bundled data channel (MOUSETRAP) with an improved throughput and latency was proposed [7]. Latches are normally open/transparent and are closed after the data has passed through them adding more timing constraints and design complexity. Having open latches also leads to the propagation of spurious transitions and higher power consumption [8].

DI or QDI channels utilize special data encoding (dual-rail or 1-in-N) that allows receivers to detect the arrival of new data with no need for explicit request signals (and the associated timing constraints). A receiver acknowledges data reception by activating a backward acknowledgement signal. When combined with conventional four-phase RZ signaling DI or QDI schemes yield very simple and robust circuit implementations [9], [10]. Two-phase NRZ DI codes have been proposed [11], [12] to reduce latency with an increase in circuit complexity. To increase the wire efficiency (number of data bits per wire) N-of-M codes were proposed in [13] at the expense of more circuit complexity. Though DI encoding is suppose to eliminate timing constraints, specific implementations may lead to some constraints [11], [12].

Single-track handshaking was introduced by Van Berkel *et al.* in [14] as an improvement upon bundled data channels. A single wire is used for both request and acknowledgement signals. A transmitter induces a transition on the wire indicating a request and then releases it, the receiver acknowledges data receipt by transitioning the wire back to its initial state (i.e., RZ) and then releasing it, combining two-phase handshaking with RZ signaling. This technique still suffers from the bundled data constraints in addition to new timing constraints pertaining to the transitioning of the wire and releasing it. A faster single-track circuit, GasP, was proposed in [15] where a self-resetting NAND gate (which resets after three gate transitions) controls the data latch. Thus, two new timing constraints are added; data must be read within three gate transitions and so must acknowledgement.

In [16] single-track control is extended to 1-of-N encoded channels. The proposed single-track full buffer (STFB) eliminates the acknowledge line that existed in previous QDI buffers. However, several tight timing constraints are imposed on internal and external signals' switching order. With the same number of transitions per cycle as GasP, STFB achieved significantly lower forward latency ($\sim$60%) at the expense of higher backward latency and lower throughput ($\sim$70%).

Manuscript received October 13, 2009; revised December 20, 2009. This work was supported by King Fahd University of Petroleum and Minerals (KFUPM) Grant IN070367.

The author is with the Computer Engineering Department, KFUPM, Dhahran 31261, Saudi Arabia (e-mail: elrabaa@kfupm.edu.sa).

Digital Object Identifier 10.1109/TVLSI.2010.2042240

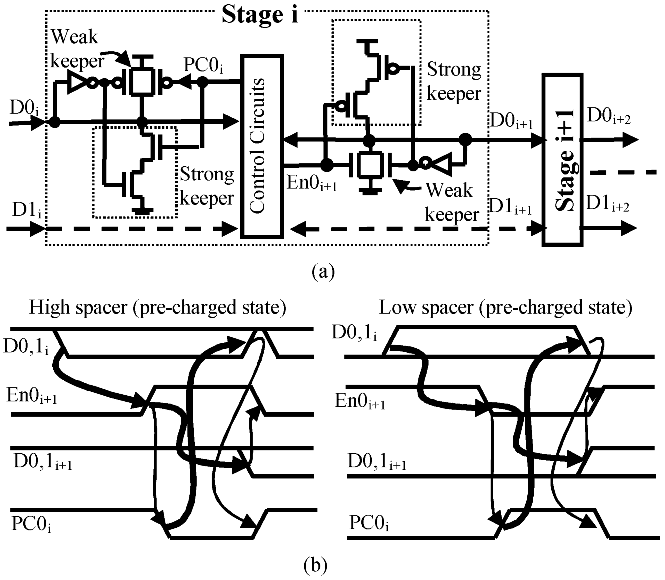


Fig. 1. Block diagram of the proposed repeater and the handshaking protocol. (a) A Block diagram of the repeater showing the data and precharging drivers and the keeper circuits for the high-spacer encoding. (b) The proposed handshaking protocol. D0, 1_i signals represent the data transfer initiated at the ith stage. Thick lines indicate signal trips across the wire segment. The protocol is shown for two types of data and spacer encoding with transitions on $D0_i$.

III. Proposed Asynchronous Transceiver

As was explained in Section II, DI or QDI techniques, depending on their implementation, can significantly reduce or eliminate timing constraints. Also, if combined with single track techniques, two-phase handshaking and RZ signaling can be realized. This would improve the bandwidth significantly and simplify the circuit implementation greatly.

Fig. 1 shows a block diagram of the newly proposed dual-rail transceiver (repeater) along with the proposed handshaking protocol. With no dedicated control lines, RZ signaling is combined with two-phase handshaking by driving each data line from the transmitter's side of a stage and returning it to its initial state (i.e., precharged or spacer state) from the receiver's side. The proper sequence of signaling is achieved by two control circuits in the repeater; an Enable control circuit that controls the driving of the output and a precharge control circuit that controls the precharging of the input. When data initiated at stage $\boldsymbol{i}$ is received at stage $\boldsymbol{i+1}$, an enable signal is asserted (En) which initiates data transfer to the $\boldsymbol{i+1}$th segment and activates a precharging signal (PC) that would then precharge the $\boldsymbol{i}$th segment, overlapping the transfer of data to the $\boldsymbol{i+1}$th segment with the precharging of the $\boldsymbol{i}$th segment and completing the data transfer cycle in two *trips*. The protocol is illustrated for two types of spacer state; a high-spacer encoded as both lines high [corresponding to the block diagram in Fig. 1(a)], and a low-spacer encoded as both lines low. Data is encoded by pulling one of the lines low or high, respectively.

As Fig. 1 shows, there are two sets of keepers on each side of a wire segment; strong keepers and weak keepers. Strong keepers at the input hold the input low (when input data arrives) until the repeater reads the data and activates the PC signal. Strong keepers at the output hold the output in the precharged state until new data arrives. When an output is driven low the weak keeper (at the output of the sender) keeps it low until the precharging circuit in the next stage precharge it (by overcoming the weak keeper) and hold it in that state by its own weak input keeper. Hence, data lines are held in the active state by strong keepers at the receiving repeaters, precharged by the receiving repeaters and held in the precharged state by strong keepers at the sending repeaters ensuring that wires are always strongly driven and preventing erroneous switching due to cross talk or SETs. Also no contention exists between any of the driving circuits and the strong keepers at all times.

IV. Circuit Implementation

Fig. 2 shows the signal transition graph (STG) of the transceiver and the circuit implementation for one of the data lines $(D0_i)$. The STG/circuits associated with the other data line are identical. The enable and precharge control circuits, Fig. 2(b)–(e), behave similarly to a Muller-C element. The En signal is asserted when input data arrives $(\mathrm{D0_i} \downarrow)$ while both transceiver outputs $(\mathrm{D0_{i+1}}$ and $\mathrm{D1_{i+1}})$ are high. It is then deasserted when the output $\mathrm{D0_{i+1}}$ is discharged. A half keeper holds En low when input data is in the precharged (high) state. The PC signal is deasserted when $\mathrm{D0_i}$ is low and En becomes high causing the input $\mathrm{D0_i}$ to be charged back high. This, in turn causes the PC signal to go back high. A keeper keeps the PC signal low in case En signal goes low before the input is charged high and keeps it high if $\mathrm{D0_i}$ goes low while En is still low (due to backpressure at the output).

The widths of the En and PC pulses are automatically set by the timing behavior of the data lines. The only timing assumption here is that En should not go low (which would require at least three gate transitions) until PC goes low (one gate transition). Even this very-easy-to-meet assumption can be eliminated by forcing En to wait for PC to go low before it goes low as indicated in Fig. 2. Circuit modifications for alternate data and spacer encoding or 1-of-N data encoding are indicated throughout Fig. 2. Designers can choose the implementation that yield the best performance for the technology they use.

V. Circuit Performance

A. Basic Operation

Spice simulations using a 0.13-μm, 1.2 V CMOS technology were used to verify the operation of the proposed repeater. The simulated pipeline comprised a data producer followed by three stages of repeaters followed by a data consumer with 50 μm wires in between and 500 Mbs data injection rate. Transistors were simply sized to achieve 50 and 100 ps fall and rise times, respectively (equivalent to a two-input, fan-out of three, NAND gate). Fig. 3 shows the simulation waveforms of one of the repeaters illustrating how the proper sequence of events on input data, En, PC signals, and output data is achieved with a forward latency of $\sim$ 130 ps. The complete pipeline was simulated as follows; first data is injected at a constant rate of 500 Mbs but is not being consumed [see Fig. 4(a)], then the consumer starts consuming the data at the same injection rate, Fig. 4(b). The pipeline gets filled after the injection of three data bits (all stages' inputs and outputs D1-D4 are low) and no new data can be injected. As soon as the consumer starts consuming data (as evident from the precharging of D4) the data starts moving forward along the pipeline. The backward latency is $\sim$ 150 ps making the minimum cycle time $\sim$ 280 ps (i.e., maximum throughput of 3.5 Gbs).

B. Maximum Throughput

The throughput as a function of wire segment length has been evaluated using a multirepeater pipeline with specially designed data producer and consumer circuits that inject/consume data at the maximum possible rate. The producer injects new data every time the first stage's input is precharged while the consumer precharges the last stage's output every time it goes low. Fig. 5 shows the data throughput and latency per stage versus wire length. As expected, latency increases linearly with wire length while throughput decreases as $1/x$. Even at a wire length of 1000 μm the throughput is still

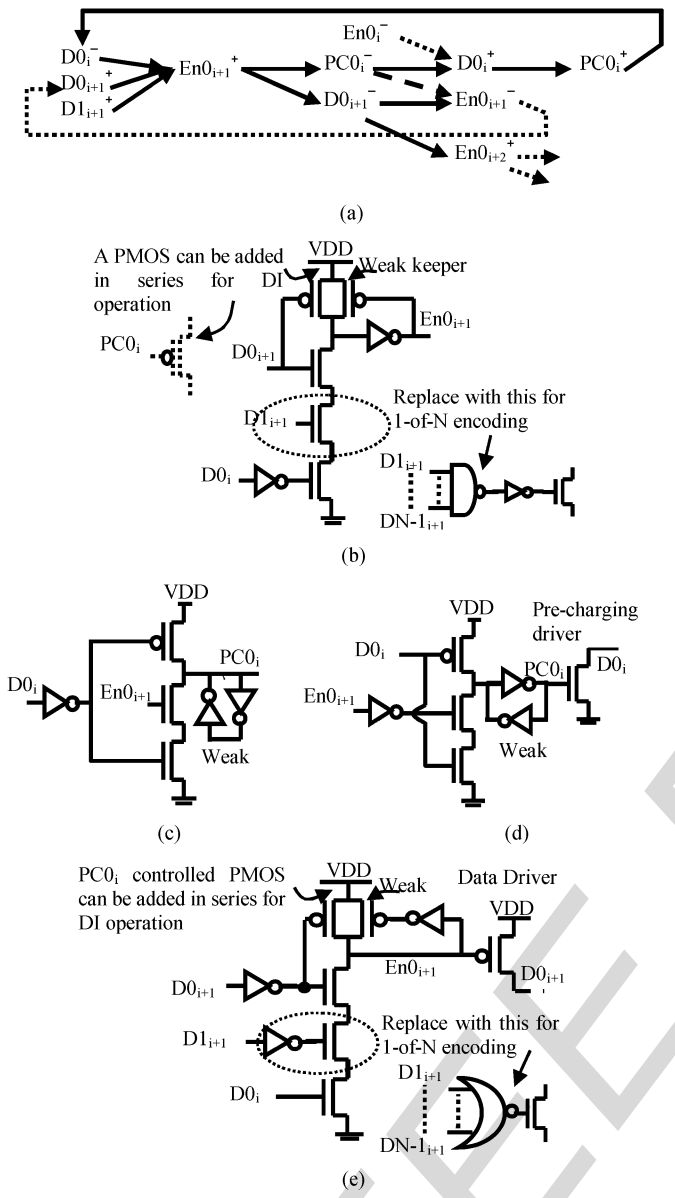


Fig. 2. Circuit implementation of the proposed two-phase RZ protocol. (a) Signal transition graph (STG) of the proposed repeater. The dashed transition from $\mathrm{PC0}_i^-$ to $\mathrm{En0}_{i+1}^-$ can be added to remove any timing assumption. (b) Enable control circuit for 1-of-2 and 1-of-N Data encoding. (c) Precharging control circuit. (d) PC circuit for low spacer. (e) Enable control circuit for low spacer.

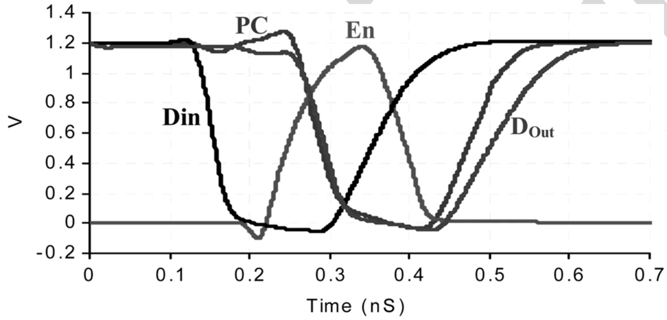


Fig. 3. Signal waveforms of one of the repeater stages.

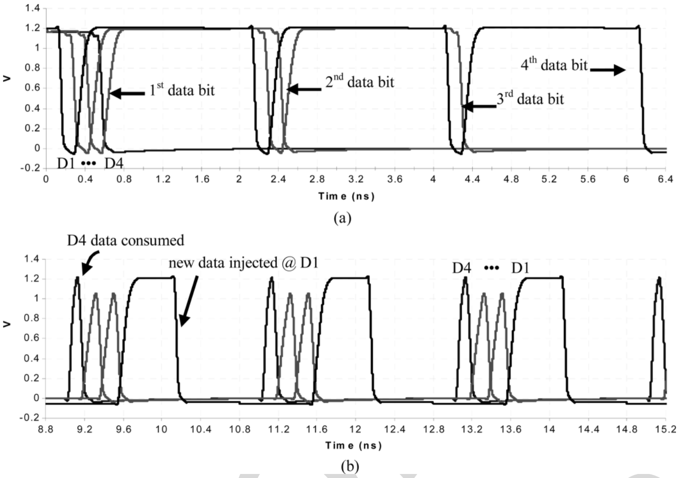


Fig. 4. Data waveforms along the asynchronous pipeline for the following scenario: (a) data being injected but not consumed and the pipeline gets filled after three data bits; (b) consumer starts consuming the data causing it to move forward along the pipeline.

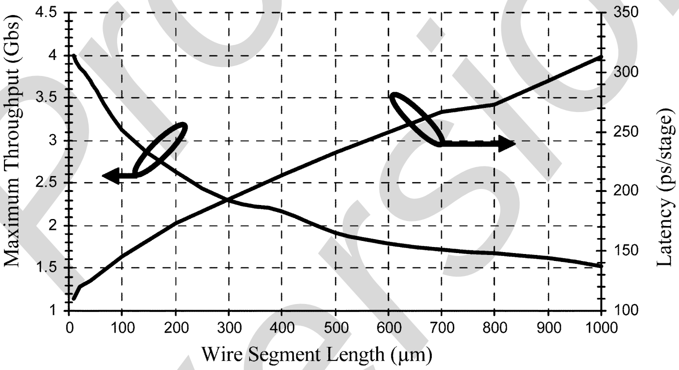


Fig. 5. Maximum throughput and latency per stage versus wire length.

relatively high (1.5 Gbs). At 0 wire length latency and throughput are 110 ps and 4 Gbs, respectively.

C. Robustness

To demonstrate the robustness of the new circuits, simulations were performed with two pipelines; an attacker and a victim. An inter-stage wire length of 1000 μm was used with a cross capacitance between the attacker and victim equivalent to half the total wire capacitance. This cross capacitance represents the worst case value under an attack by a single attacker. The attacker attacks the victim under the two most vulnerable conditions; when the victim is in the precharged state and when it is in the evaluation state (low output). Other attacks (during switching) would either slow down the victim or speed it up. As Fig. 6 shows, the repeater is immune to the cross talk noise due to the keepers' strong drive. Subsequent pipeline stages are not affected at all.

D. Comparison With Other Repeaters

The performance of the proposed repeater has been compared to that of the following asynchronous repeaters: the dual-rail DI buffer described in [12] (called LETS) with MOUSETRAP [7] pipelining style, the dual-rail (1-of-2) STFB [16] and GasP [15]. These were chosen for having the best performance published so far. The performance of two conventional repeaters; a two-phase QDI asynchronous repeater and a synchronous repeater (a CMOS FF), were also evaluated to provide reference points. The sizes of all repeaters were optimized

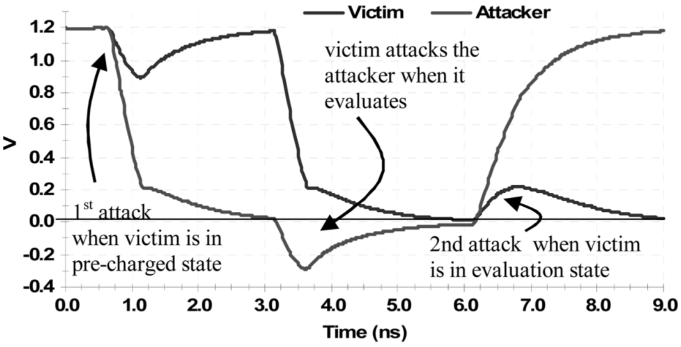


Fig. 6. Attacker and victim waveforms for two attack conditions.

TABLE I
PERFORMANCE COMPARISONS BETWEEN THE NEWLY PROPOSED REPEATER AND OTHER REPEATERS

Repeater Circuit	Forward Latency (ps/stage)	Backward Latency (ps/stage)	Maximum Throughput (Gbs)	Energy per Bit transfer (fJ/bit)	No. of Transistors
New two-phase RZ Repeater	145	170	3.17	75	60^{+}
LETS [12] (MOUSETRAP-style)	120	310	2.33	60	62
STFB [16]	100	230	3.03	60	34
GasP [15]	170	160	3.03	125	34
Conventional two-phase QDI	410	100	1.96	150	70
Sync. With equal Cin	310	–	3.22 *	$50^{\#}$	24^{++}
Sync. With larger size	200	–	5 *	$150^{\#}$	24

* Not accounting for clock skew or jitter or process variations
$^{\#}$ Not including energy in clock distribution network
$^{+}$ Out of these 32 transistors are used in the keepers for robustness
$^{++}$ Without the clocking circuitry

to yield maximum throughput with equal input capacitances (Cin) to ensure a fair comparison (equal loading on circuits that drive the repeaters). The value of Cin basically determines the sizes of transistors connected to the inputs. The remaining transistors were sized for maximum throughput. With the exception of the new repeater, all asynchronous transceivers did require more design effort and time to optimize for maximum throughput. A synchronous repeater with a larger size (i.e., Cin) and minimum achievable latency per stage was also compared. For each repeater, forward and backward latencies, maximum throughput and energy per bit (EPB) were measured at a wire segment length of 100 μm and using four-stage pipelines. EPB, obtained by integrating the instantaneous power, is the average for transferring a 0 bit and a 1 bit.

Table I summarizes the simulation results for all circuits including transistor count per repeater. The new repeater achieved the best balance between forward and backward latencies resulting in highest throughput. The STFB achieved the lowest forward latency but at the expense of higher backward latency. The LETS repeater achieved lower forward latency than the new repeater but had higher backward latency. GasP had a similar performance to the new repeater in terms of latencies and throughput but its EPB is much higher (its control track has twice the data frequency). The synchronous repeater achieved the smallest EPB at a throughput close to that of the new repeater (without considering clock skew and jitter). 1-of-4 encoding could reduce the EPB of the asynchronous transceivers at the expense of more latency due to higher gates' fan-in.

Though the new repeater had more transistors than the STFB, more than half of the transistors are in the keepers which were added for robustness. Similar keepers would have to be added to the STFB to enhance its robustness. The Gasp and synchronous repeaters naturally have lower transistor counts due to their single-rail data encoding.

VI. CONCLUSION

A new two-phase QDI asynchronous handshaking protocol that utilizes dual-rail RZ data encoding has been developed. A new asynchronous repeater that implements the new protocol has also been developed and tested using SPICE simulations. The RZ data signaling has allowed the circuit implementation to be simple yet very efficient and robust. Also, since the new protocol overlaps driving an output data segment with the precharging of the preceding one, the throughput has been maximized while minimizing the latency. The new repeater achieved a throughput equal to that of a synchronous repeater with less than half the latency. Compared to the best previously published asynchronous repeaters, it achieved a superior overall performance in terms of latency, throughput, energy per bit and design complexity.

REFERENCES

[1] L. Benini and G. D. Micheli, "Networks on chips: A new SoC paradigm," *IEEE Computer*, vol. 35, no. 1, pp. 70–78, Jan. 2002.

[2] W. J. Dally and B. Towles, "Route packets, not wires: On chip interconnection networks," in *Proc. 38th Des. Autom. Conf.*, Jun. 2001, pp. 684–689.

[3] J. Henkel, W. Wolf, and S. Chakradhar, "On-chip networks: A scalable, communication-centric embedded system design paradigm," in *Proc. 17th Int. Conf. VLSI Des.*, 2004, pp. 845–851.

[4] D. M. Chapiro, "**<PLEASE PROVIDE DEPARTMENT.>** Glo[illegible] asynchronous locally-synchronous systems," Ph.D. dissertation, Stanford Univ., Stanford, CA, Oct. 1984.

[5] S. Moore, G. Taylor, R. Mullins, and P. Robinson, "Point to point GALS interconnect," in *Proc. 8th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2002, pp. 69–75.

[6] S. B. Furber and P. Day, "Four-phase micropipeline latch control circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 4, no. 3, pp. 247–253, Mar. 1996.

[7] M. Singh and S. M. Nowick, "MOUSETRAP: High-speed transition-signaling asynchronous pipelines," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 6, pp. 684–698, Jun. 2007.

[8] M. Lewis, J. Garside, and L. Brackenbury, "Reconfigurable latch controllers for low power asynchronous circuits," in *Proc. 5th Int. Symp. Async. Circuits Syst. (ASYNC)*, 1999, pp. 27–35.

[9] W. J. Bainbridge and S. B. Furber, "Delay insensitive system-on-chip interconnect using 1-of-4 data encoding," in *Proc. 7th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2001, pp. 118–126.

[10] A. Lines, "Asynchronous interconnect for synchronous SoC design," in *Proc. IEEE MICRO*, Jan.–Feb. 2004, pp. 32–41.

[11] M. E. Dean, T. E. Williams, and D. L. Dill, "Efficient self-timing with level-encoded 2-phase dual-rail (LEDR)," in *Proc. Univ. California/Santa Cruz Conf. Adv. Res. VLSI*, 1991, pp. 55–70.

[12] P. B. McGee, M. Y. Agyekum, M. A. Mohamed, and S. M. Nowick, "A level-encoded transition signaling protocol for high-throughput asynchronous global communication," in *Proc. 14th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2008, pp. 116–127.

[13] W. J. Bainbridge, W. B. Toms, D. A. Edwards, and S. B. Furber, "Delay insensitive, point-to-point interconnect using M-of-N codes," in *Proc. 9th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2003, pp. 132–141.

[14] K. Van Berkel and A. Bink, "Single-track handshaking signaling with application to micropipelines and handshake circuits," in *Proc. 2nd Int. Symp. Async. Circuits Syst. (ASYNC)*, 1996, pp. 122–133.

[15] I. Sutherland and S. Fairbanks, "GasP: A minimal FIFO control," in *Proc. 7th Int. Symp. Async. Circuits Syst. (ASYNC)*, 2001, pp. 46–53.

[16] M. Ferretti and P. A. Beerel, "Single-track asynchronous pipeline templates using 1-of-N encoding," in *Proc. Des., Autom. Test Eur. (DATE)*, 2002, pp. 1008–1015.