

PAPER

High-Throughput Partially Parallel Inter-Chip Link Architecture for Asynchronous Multi-Chip NoCs

Naoya ONIZAWA^{†a)}, Akira MOCHIZUKI^{††b)}, *Members*, Hirokatsu SHIRAHAMA^{††c)}, *Nonmember*, Masashi IMAI^{†††d)}, Tomohiro YONEDA^{††††e)}, and Takahiro HANYU^{††††d)}, *Members*

SUMMARY This paper introduces a partially parallel inter-chip link architecture for asynchronous multi-chip Network-on-Chips (NoCs). The multi-chip NoCs that operate as a large NoC have been recently proposed for very large systems, such as automotive applications. Inter-chip links are key elements to realize high-performance multi-chip NoCs using a limited number of I/Os. The proposed asynchronous link based on level-encoded dual-rail (LEDR) encoding transmits several bits in parallel that are received by detecting the phase information of the LEDR signals at each serial link. It employs a burst-mode data transmission that eliminates a per-bit handshake for a high-speed operation, but the elimination may cause data-transmission errors due to cross-talk and power-supply noises. For triggering data retransmission, errors are detected from the embedded phase information; error-detection codes are not used. The throughput is theoretically modelled and is optimized by considering the bit-error rate (BER) of the link. Using delay parameters estimated for a 0.13 μm CMOS technology, the throughput of 8.82 Gbps is achieved by using 10 I/Os, which is 90.5% higher than that of a link using 9 I/Os without an error-detection method operating under negligible low BER ($< 10^{-20}$).

key words: *Asynchronous circuits, Network-on-Chip (NoC), burst-mode data transmission, level-encoded dual-rail (LEDR) encoding, error detection, data retransmission*

1. Introduction

The Network-on-Chip (NoC)-based design paradigm offers scalable on-chip global communication for multi-processor System-on-Chips (SoCs) [1]. NoC based on Globally Asynchronous Locally Synchronous (GALS) or fully-asynchronous systems fully utilizes the benefits of asynchronous circuits, such as low power consumption and communication robustness [2], [3]. For highly parallel computations using many processors such as neural simulators and automotive applications, asynchronous multi-

chip NoCs have been proposed [4]–[6]. To realize high-performance asynchronous multi-chip NoCs, high-speed inter-chip communication links are required even though the number of chip I/Os is limited.

Some synchronous inter-chip link proposals achieve more than 10 Gbps/link [7], [8]. The throughput is high, but requires synchronizers between the synchronous links and asynchronous computation blocks in the asynchronous multi-chip NoCs. The synchronizer is complex as the asynchronous communication speed depends on routing paths of the NoCs, which causes incurring additional delay and power dissipation.

Instead, asynchronous inter-chip links efficiently communicate with asynchronous computation blocks without synchronizers [9]–[13]. In [9], the asynchronous serial link based on a per-bit handshake achieves 315 Mbps using 5 I/Os. In [13], the asynchronous burst-mode serial link reaches to 3 Gbps using 4 I/Os by eliminating the per-bit handshake, which may cause data-transmission errors. However, the throughput is still lower than that of the on-chip links [2], [3] and it is getting lower if a data-retransmission overhead is considered.

In this paper, we introduce a high-throughput asynchronous inter-chip link architecture based on a partially parallel burst-mode data transmission scheme. In the proposed link, a chunk of bits that is called a “word” are continuously transmitted in parallel without the per-bit handshake. Instead of that, a per-word handshake is exploited in order to reduce the delay overhead of the per-bit one. There might exist wrongly transmitted bits due to dynamic delay variations, such as power-supply and crosstalk noises. To prevent data transmission errors, an error-detection scheme based on level-encoded dual-rail (LEDR) encoding [14] and a data-retransmission scheme are also introduced. The error-detection method is realized by exploiting the phase information of the LEDR signals instead of using error-detection codes that need extra I/Os. The throughput of the proposed link is theoretically modelled with considering a bit-error rate (BER) of the link. Based on the model, several parameters (e.g. the number of parallel links) are optimized for high-throughput asynchronous inter-chip links.

The rest of this paper is organized as follows. Section 2 reviews the asynchronous multi-chip NoCs and summarizes the related work. Section 3 describes the proposed link architecture. Section 4 introduces the error-detection and the data-retransmission schemes based on the LEDR encoding.

Manuscript received September 13, 2013.

Manuscript revised January 10, 2014.

[†]The author is with the Frontier Research Institute for Interdisciplinary Sciences, Tohoku University, Sendai-shi, 980–8578 Japan.

^{††}The authors are with the Research Institute of Electrical Communication, Tohoku University, Sendai-shi, 980–8577 Japan.

^{†††}The author is with the Dept. of Electronics and Information Technology, Hirosaki University, Hirosaki-shi, 036–8561 Japan.

^{††††}The author is with the National Institute of Informatics, Tokyo, 101–8430 Japan.

a) E-mail: nonizawa@m.tohoku.ac.jp

b) E-mail: pico@ngc.riec.tohoku.ac.jp

c) E-mail: shira@ngc.riec.tohoku.ac.jp

d) E-mail: miyabi@eit.hirosaki-u.ac.jp

e) E-mail: yoneda@nii.ac.jp

f) E-mail: hanyu@ngc.riec.tohoku.ac.jp

DOI: 10.1587/transinf.E97.D.1546

Section 5 models the throughput with considering the BER of the link and evaluates the throughput based on the model in a $0.13\mu\text{m}$ CMOS technology. Section 6 concludes this paper.

2. Background and Motivation

2.1 Asynchronous Multi-Chip NoCs

Figure 1 depicts an architecture of asynchronous multi-chip NoCs [4]–[6]. The asynchronous multi-chip NoCs include several asynchronous NoCs [1]–[3] that communicate using inter-chip data-transmission links. They operate as a large NoC that contains several tens of processing cores and more in a chip. Each NoC includes processing cores with asynchronous on-chip network, which consists of switching routers and on-chip data-transmission links [15], [16] designed using asynchronous circuits. Each processing core transmits/receives packets, which basically include header, body, and tail flits [2], [3]. The router decides a packet route by processing the header flit and keeps it until the tail flit is processed. During processing a packet in the router, other packets cannot use the same route basically and waits until the route is released. In the NoCs, the data-transmission throughput and the latency can be varied depending on the route and a packet congestion.

2.2 Related Work

To realize high-performance multi-chip NoCs, high-speed on-chip and inter-chip data-transmission links are required. Especially, the inter-chip links tend to be lower throughput than the on-chip links due to a limited number of I/Os of a chip. Several asynchronous inter-chip communication links have been proposed [9]–[13]. In [9]–[11], the communication links are designed based on a quasi delay-insensitive (QDI) logic style [17]. They avoid any timing constraints except for one assumption that wires at a fan-out point must have roughly equal delay. 1-bit data transmission is performed based on a handshake protocol that uses request and acknowledge information. This is called a “per-bit handshake”. In addition, spacer information has to be inserted into two consecutive data in a traditional four-phase protocol [17]. Hence, 1-bit data transmission takes four steps,

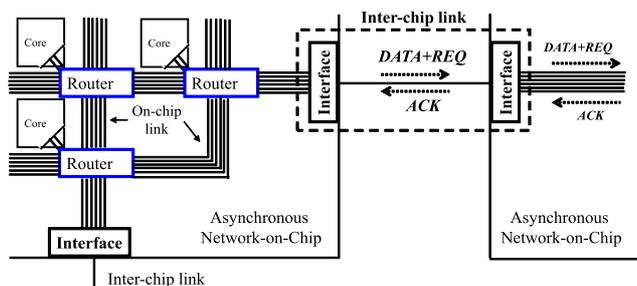


Fig. 1 Asynchronous multi-chip Network-on-Chips (NoCs) that operate as a large NoC in a chip.

which result in low throughput.

In [12], [13], high-speed serial links have been reported based on a burst-mode data transmission scheme [18]–[20]. In the burst-mode data-transmission method, a word that contains several tens of bits is transmitted without the per-bit handshake unlike the communication links based on the QDI logic style. Once the receiver completes to receive the word, it transmits word-level acknowledge information to the transmitter. This is called “a per-word handshake”. It greatly reduces the number of communication steps, while it is more sensitive to timing variations than the QDI-based links.

In [13], it achieves 3 Gbps based on the burst-mode data transmission method under $0.18\mu\text{m}$ CMOS with a data-retransmission mechanism. However, the throughput is still smaller than that (5 Gbps) of the asynchronous on-chip communication link based on the four-phase protocol [2] and that (17 Gbps) based on two-phase protocol that has half communication steps of the four-phase one [3]. In addition, the delay overhead of the data-retransmission scheme is not considered, so that the throughput would be even lower.

2.3 Motivation

To realize high-performance asynchronous multi-chip NoCs, a high-throughput inter-chip link using a limited number of I/Os is required. In addition, in the NoC or the multi-chip NoCs, quality of service (QoS) is also considered, such as end-to-end data transmission delays and data-transmission throughputs [21]. Our motivation is to maximize the throughput given the number of I/Os, while achieving an negligible low error probability at the link. In this paper, the error indicates a transient error due to dynamic timing variations, such as power-supply and crosstalk noises. The low error probability significantly reduces a possibility of an end-to-end data retransmission beyond chips. In order to reduce the error probability, a link-level data retransmission is efficiently realized in the proposed link architecture.

3. Partially Parallel Asynchronous Burst-Mode Data-Transmission Link

3.1 Link Architecture

Figure 2 depicts the proposed link architecture that consists of a r -bit partially parallel burst-mode inter-chip communication with a data-retransmission mechanism. Suppose a bit width of the on-chip parallel link is n . The n -bit data is divided into $r \cdot k$ -bit data, which is then transmitted at each inter-chip serial link, where $k = n/r$. Suppose n , r and k are positive integers. The on-chip and inter-chip communication links are designed based on LEDR encoding shown in Table 1, where 1-bit data is encoded using a dual-rail signal. The solid and break lines indicate dual- and single-rail signals, respectively. The number of I/Os of the link is $2r+2$.

The inter-chip data transmission is briefly described using a signal-flow chart shown in Fig. 3. The detail is de-

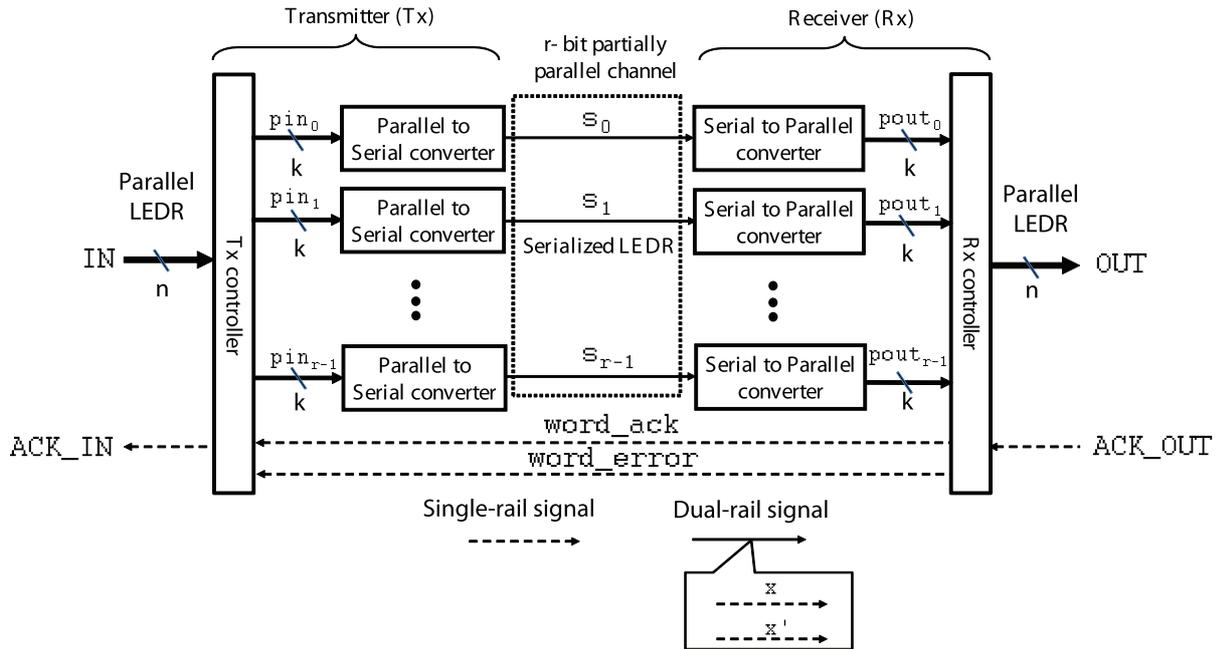


Fig. 2 The proposed r -bit partially parallel burst-mode inter-chip communication link architecture with a data-retransmission mechanism.

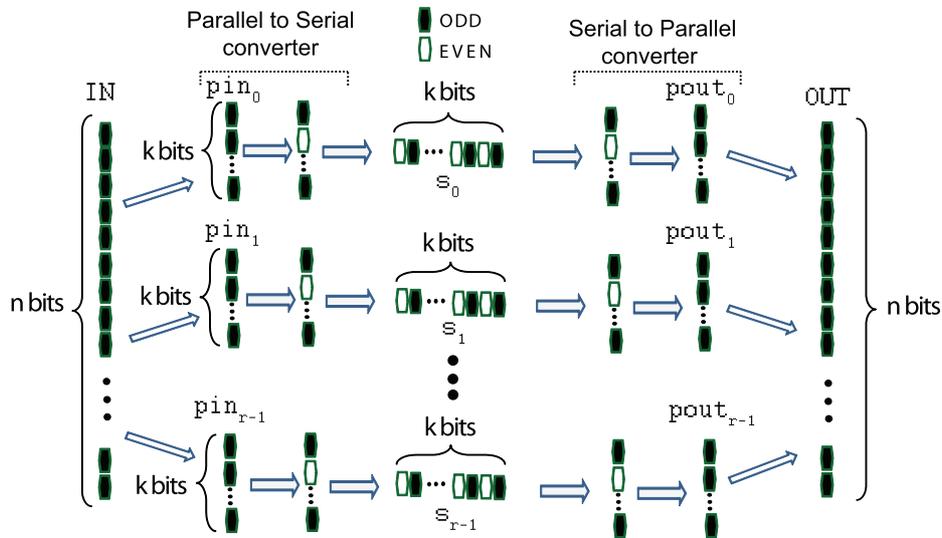


Fig. 3 A signal-flow chart in the proposed link when the phase information of IN is ODD.

Table 1 Level-Encoded Dual-Rail (LEDRL) code (x, x') .

Logic value	ODD	EVEN
"0"	(0,1)	(0,0)
"1"	(1,0)	(1,1)

scribed in the next subsections. A parallel LEDR data IN (n bits) is received in the transmitter (Tx) that attaches to an asynchronous NoC. The parallel data is encoded to either ODD or EVEN phase and these two phases are exploited alternatively. Suppose the phase information of the parallel data is ODD in Fig. 3. The parallel data is divided into $r \cdot k$ -bit parallel data pin_i ($0 \leq i < r$). Each k -bit parallel

data is converted to its serialized data s_i in its Parallel to Serial converter. The phase information of only even number of the serial data is changed to EVEN when ACK_IN is high shown in the example. When ACK_IN is low, the phase information of only odd number of the serial data is changed to ODD. Then, the serial data is transmitted using different phase information (ODD and EVEN), alternatively.

The serial data s_i is continuously received by detecting the change of its phase information at the receiver (Rx). The phase information of even number of the serial data is changed back to ODD and then the serial data is bundled to make k -bit parallel ODD data $pout_i$ in the Serial to Paral-

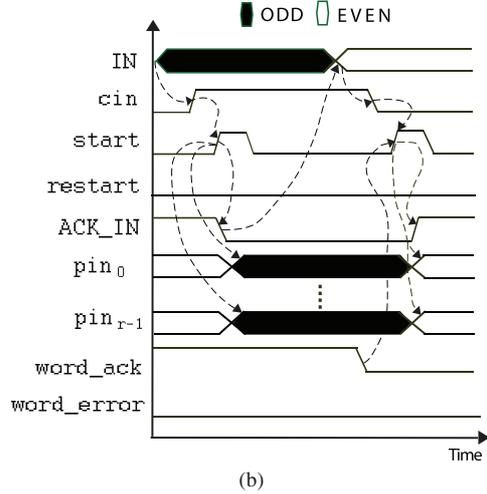
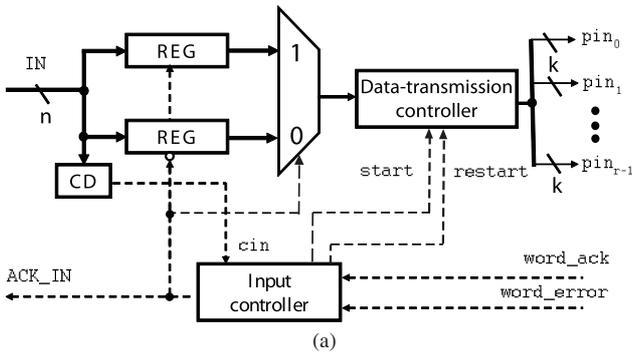


Fig. 4 Tx controller: (a) block diagram and (b) timing diagram. A parallel LEDR data (IN) is alternatively stored in one of two registers (REGs). ODD parallel data is stored in the bottom REG when ACK_IN is negated.

lel converter when ACK_OUT is high shown in the example. When ACK_OUT is low, the phase information of odd number of the serial data is changed back to EVEN. Then, the $r*k$ -bit parallel data is combined to make n -bit parallel data, which is transmitted to the NoC attached to the Rx. Once the Rx completes to receive the n -bit parallel LEDR data, it changes a word-level acknowledge signal (word_ack) to request the next data transmission to the Tx. If the received LEDR data has one or more errors, word_error is asserted to request for the data retransmission. The retransmission mechanism is described in the next section.

3.2 Transmitter

Figure 4 (a) shows a block diagram of the Tx controller. The operation of the Tx controller is described with a timing diagram shown in Fig. 4 (b). An arrival of the parallel LEDR data IN is detected using an n -bit completion detector (CD) (see Fig. 11 in [3]). The CD consists of n 2-input XOR gates, an AND-chain network, an OR-chain network and a C-element [3]. The C-element is an asynchronous storage element [22]. The output is high when both of the inputs are high and low when they are low. It holds the current state when the inputs are different. Using the C-element in

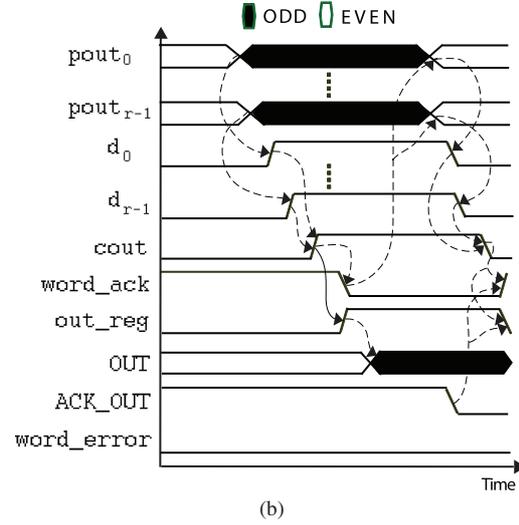
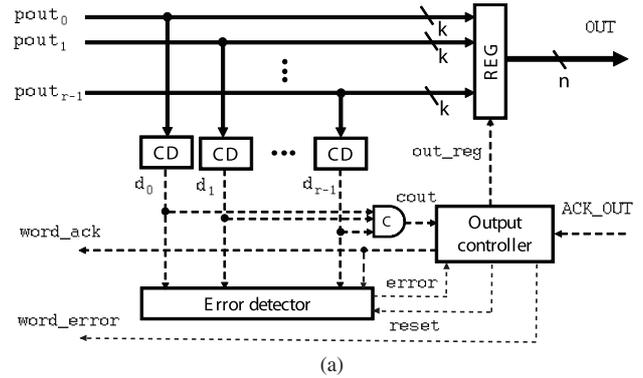


Fig. 5 Rx controller: (a) block diagram and (b) timing diagram.

the CD, the output of the CD (cin) is high when the phase information of IN is ODD and low when EVEN.

word_ack is asserted when the Rx completes to receive the n -bit data whose phase information is EVEN and is negated when ODD. When the input controller detects the high of cin and the high of word_ack, ACK_IN is negated and then IN (ODD) is stored in the bottom register (REG). IN (EVEN) is stored in the top REG when cin and word_ack are low. ACK_IN is an acknowledge signal of IN to request the next parallel LEDR data to the asynchronous NoC.

In the multiplexer, one of two parallel LEDR data is alternatively selected by ACK_IN. The selected parallel data is passed through a data-transmission controller when a pulse signal (start) is generated by the change of cin. Then, the parallel data is divided into $r*k$ -bit parallel data (pin_{*i*}) to the Parallel to Serial converters. Each Parallel to Serial converter transmits the LEDR data serially. restart and word_error signals are used for the data retransmission described in the next section.

3.3 Receiver

Figure 5 (a) depicts a block diagram of the Rx controller.

The operation of the Rx controller is described with a timing diagram shown in Fig. 5 (b). The $r \cdot k$ -bit parallel LEDR data $pout_i$ are received in the Serial to Parallel converters. An arrival of the $pout_i$ is detected using the CD. An output of the CD (d_i) is asserted when the k -bit ODD data is received and negated when the EVEN data is received. An output of the C-element ($cout$) is changed when all outputs of the CDs are asserted or negated.

When ACK_OUT and $cout$ are asserted, the n -bit ODD data is stored in the REG by out_reg . ACK_OUT is an acknowledge signal of OUT from the NoC attached to the Rx. The n -bit EVEN data is stored in the REG when ACK_OUT and $cout$ are negated. Concurrently, $word_ack$ is changed by the output controller as the acknowledge signal of the received n -bit parallel LEDR data. $error$ and $reset$ signals are used for the data retransmission.

4. Error-Detection and Data-Retransmission Schemes

4.1 Sampling Method

Figure 6(a) shows models of crosstalk-induced jitters in synchronous and asynchronous parallel links. In the parallel data transmission, crosstalk occurs by the inductive and capacitive couplings between the transmission lines [23]–[25]. Due to the timing jitter of the transmitted parallel data shown in Fig. 6(b), a sampling point of the clock at the receiver tends to be difficult to set compared to the serial link, which limits the throughput.

Figure 6(c) shows a sampling method of an asynchronous parallel link under crosstalk environments. Unlike the synchronous link using a clock signal, the transmitted signal is preliminarily encoded at each link and it contains the data and the phase information. At the receiver, a local control signal is generated by detecting the phase infor-

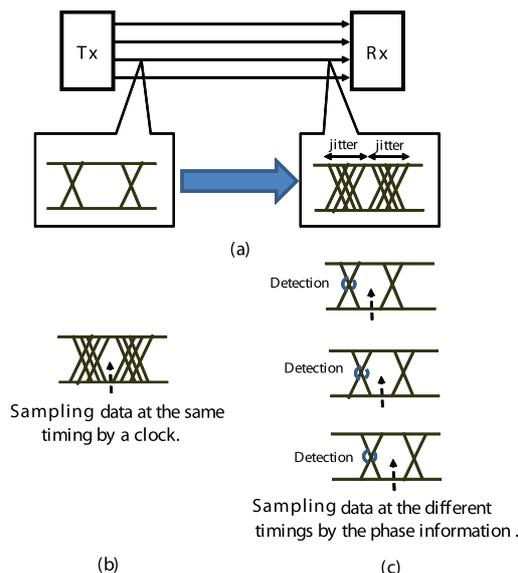


Fig. 6 Crosstalk-induced jitters: (a) model, (b) synchronous and (b) asynchronous parallel links.

mation and then the data is stored using the control signal. Hence, each serial transmitted signal can be received at different timings at the receiver.

4.2 Errors in Burst-Mode LEDR Data Transmission

The proposed asynchronous link employs the burst-mode data transmission based on the per-word handshake. Data-transmission errors might occur when two consecutive signals are too close to be detected at the receiver mainly due to dynamic timing variations, such as crosstalk and supply-voltage noises [26]. Note that a static timing variation is due to process variations. Several reliable on- and inter-chip communication links have been proposed in [11], [27], [28]. These data-transmission links exploit error-detection codes or error-correcting codes. These codes increase the number of required I/Os of the chip, which decreases the data-transmission throughput per I/O. In the proposed scheme, the embedded phase information (ODD and EVEN) of the LEDR signal is exploited to detect the errors without additional I/Os instead of using error-detection or error-correction codes.

Figure 7 depicts an example of the burst-mode data transmission based on the LEDR encoding. The LEDR encoding uses dual-rail signals to transmit 1-bit data shown in Table 1. The two different encoded signals (ODD and EVEN) are alternatively used. The transmitter changes one of the two signals to transmit 1-bit data and then the receiver detects the change of the signal to receive it. In the example shown in Fig. 7 (a), 5-bit data is correctly transmitted.

Figure 7(b) depicts an example of the data-transmission error. In the burst-mode data transmission, a subsequent signal may overwrite the precedent signal due to the dynamic timing variations. In the example, suppose the timing margin between the 3rd and the 4th signals is very small. At the receiver, the change of the phase information cannot be detected. Hence, the 3rd and the 4th signals are not stored as the receiver receives signals by detecting the phase change.

4.3 Completion-Detection Based Data-Retransmission Method

Error-detection and data-retransmission mechanisms using

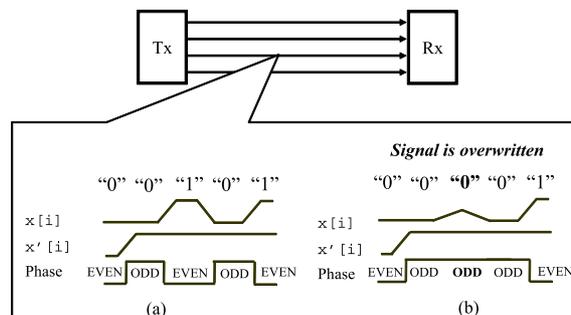


Fig. 7 Burst-mode LEDR-data transmission: (a) no errors, (b) errors due to a timing jitter.

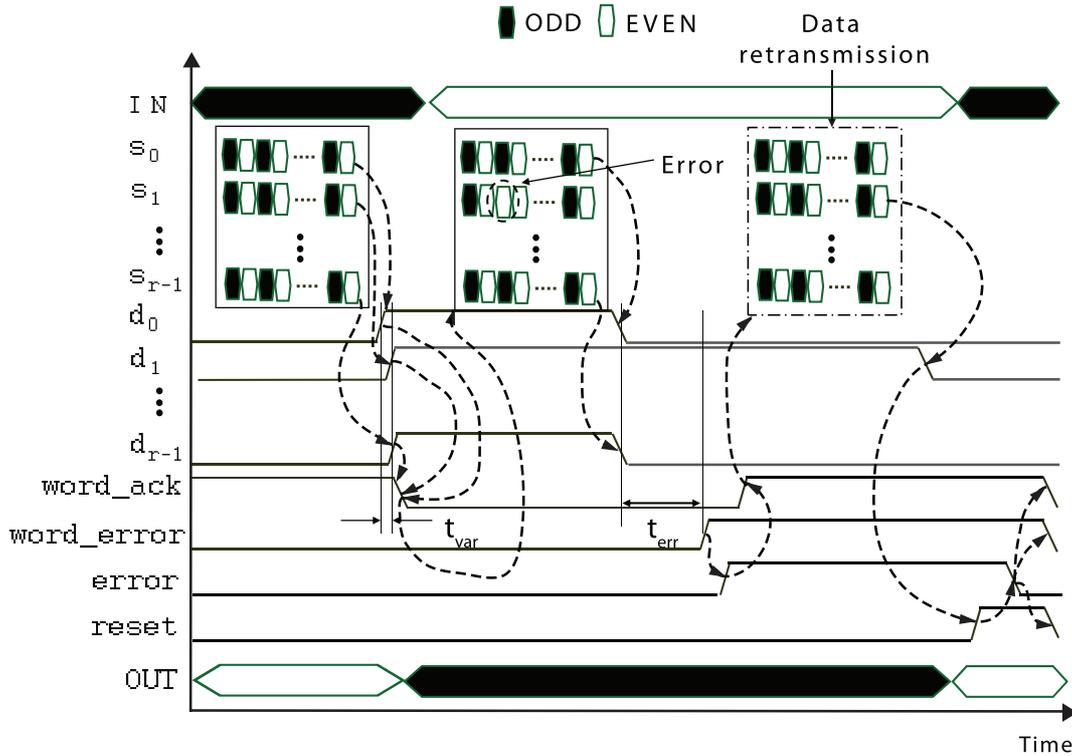


Fig. 8 Timing diagram of the proposed asynchronous inter-chip link with an error.

CDs are introduced. The proposed mechanisms are described using an example of a timing diagram of the proposed data-transmission link shown in Fig. 8. At the first n -bit data transmission, r -bit data is transmitted k times in parallel from the Tx without errors. Each k -bit data is received in the Serial to Parallel converter at the Rx. The $r \cdot k$ -bit data is processed using r CDs, which asserts cout depicted in Fig. 5. Then, word_ack is changed by the output controller to request the next data transmission.

The outputs of the CDs are also connected to the error detector. The error detection is realized using a time window. The error detector contains a delay element whose delay time is t_{err} that is set to be large enough to compensate timing variations among serial links due to the dynamic timing variations. The output (error) is given by:

$$\text{error} = \begin{cases} 1, & \text{if } t_{var} > t_{err} \\ 0, & \text{else if } \text{reset} = 1 \\ \text{hold}, & \text{otherwise} \end{cases} \quad (1)$$

where t_{var} is a time period that at least one d_i is different from the other ones. In this case, error is not asserted as t_{var} is smaller than t_{err} .

At the second n -bit data transmission, there exists an error in the serial link s_i . In s_i , the 3rd data is overwritten by the 4th data, so that these two data are not received at the Rx. In this case, the Serial to Parallel converter for the link s_i stores a $(k-2)$ -bit data whose phase information is EVEN. As all inputs are not set to the CD for pout_1 , d_1 is not changed while other outputs are changed. Hence, cout is stable to be high. As d_1 is never changed within

t_{err} , error is asserted. Once the output controller detects the assertion of error , word_error is also asserted and word_ack is changed.

When the assertion of word_error is detected in the input controller at the Tx controller depicted in Fig. 4, a pulse signal (restart) is generated. Then, the parallel LEDR data (EVEN) is retransmitted using the data-transmission controller. Suppose there is no error at this time. In this case, as cout is negated, the received data is stored in the REG by out_reg . Also, reset is asserted that negates error . Then, both word_error and reset are negated. Concurrently, word_ack is changed to request the next data transmission.

5. Evaluation

5.1 Throughput Model

In the proposed burst-mode data transmission link, errors occur when two consecutive signals are too close to be distinguished due to dynamic timing variations. Figure 9 shows the timing model of the two consecutive signals. Suppose the power-supply noise causes the dynamic timing variation that is approximated as a normal distribution [26], where the standard deviation is σ_{delay} . Serial LEDR data is transmitted every t_{sep} that is defined by:

$$t_{sep} = t_{dis} + t_{margin}, \quad (2)$$

where t_{dis} is the minimum time difference to distinguish

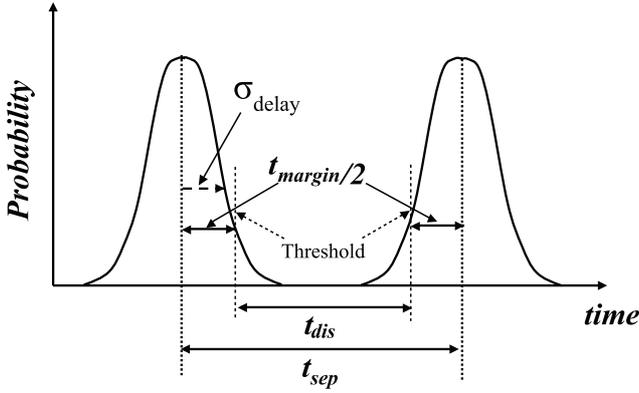


Fig. 9 Timing model between two consecutive signals under a power-supply-noise based dynamic timing variation.

these two signals and t_{margin} is the delay margin. When the probability distribution of the delay time crosses thresholds shown in Fig. 9, there will be errors. Hence, the bit-error rate (BER) is given by:

$$BER = \frac{1}{2} \operatorname{erfc}\left(\frac{t_{margin}}{2\sqrt{2}\sigma_{delay}}\right). \quad (3)$$

In the proposed link, as each link transmits serial data k (n/r) times for the n -bit data transmission, the total delay time is given by:

$$t_{total} = kt_{sep} + t_{ctr}, \quad (4)$$

where t_{ctr} is a summed delay time of controllers, such as the Tx and the Rx controllers.

During a n -bit data transmission, there are $(k-1)$ times chances of errors. Hence, the probability of the data transmission with errors is given by the following:

$$p = 1 - (1 - BER)^{r(k-1)}. \quad (5)$$

When there exist errors at the link, the error detection takes the time of t_{err} . If a n -bit data is at most m times retransmitted at the error case, the average throughput is given by:

$$\text{Throughput} = n(1 - p) \sum_{s=1}^{m+1} p^{s-1} \frac{1}{st_{total} + (s-1)t_{err}}. \quad (6)$$

5.2 Throughput Estimation

To estimate the throughput, several delay information is estimated in a $0.13\mu\text{m}$ CMOS technology. Suppose the data transmission is performed using current-mode circuits [29] and a length of the link is set to be 10 mm. σ_{delay} is estimated under a power-supply noise, where VDD is set to 1.2V. The power-supply noise is modelled as a normal distribution where the 3-sigma standard deviation ($3\sigma_{vdd}$) is set to 0.05V to 0.2V. The parameters are summarized in Table 2.

First, a single n -bit data transmission in the burst-mode link is evaluated. Figure 10 shows BERs vs. t_{sep} . t_{sep} can be adjusted to decide a desirable BER. Once t_{sep} is chosen, t_{total}

Table 2 Estimated parameters in a $0.13\mu\text{m}$ CMOS.

t_{dis}	136 ps
t_{ctr}	1724 ps
t_{err}	500 ps
σ_{delay} (when $3\sigma_{vdd}=0.1$ V)	17.6 ps

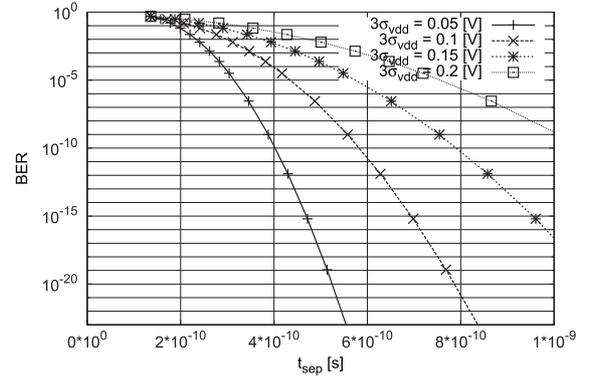
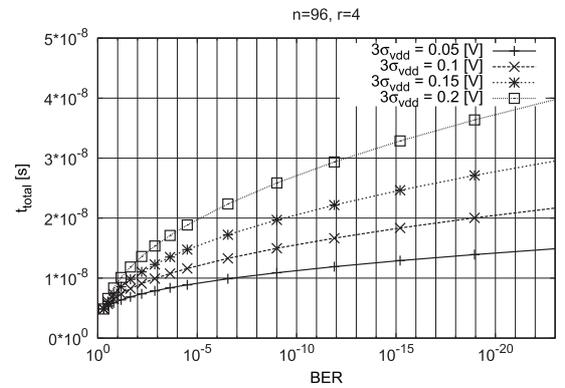
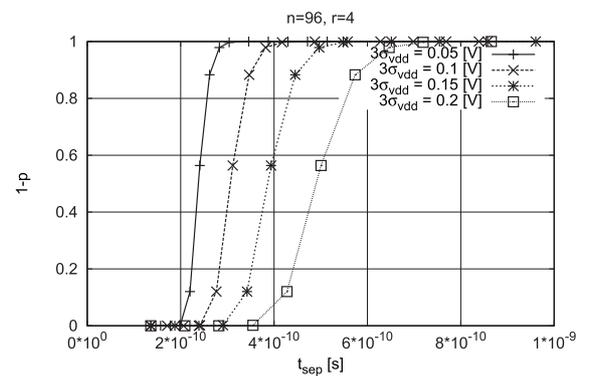


Fig. 10 BER vs. t_{sep} .



(a)



(b)

Fig. 11 Performance of a single n -bit data transmission: (a) t_{total} vs. BER, and (b) $(1 - p)$ vs. t_{sep} .

and $(1 - p)$ are determined given n and r shown in Fig. 11, where n is 96 and r is 4.

Then, the maximum number of data retransmissions (m) is considered. Figure 12 shows the effect of m in performance, where n is 96 and r is 4 and $3\sigma_{vdd}$ is 0.1 V. The

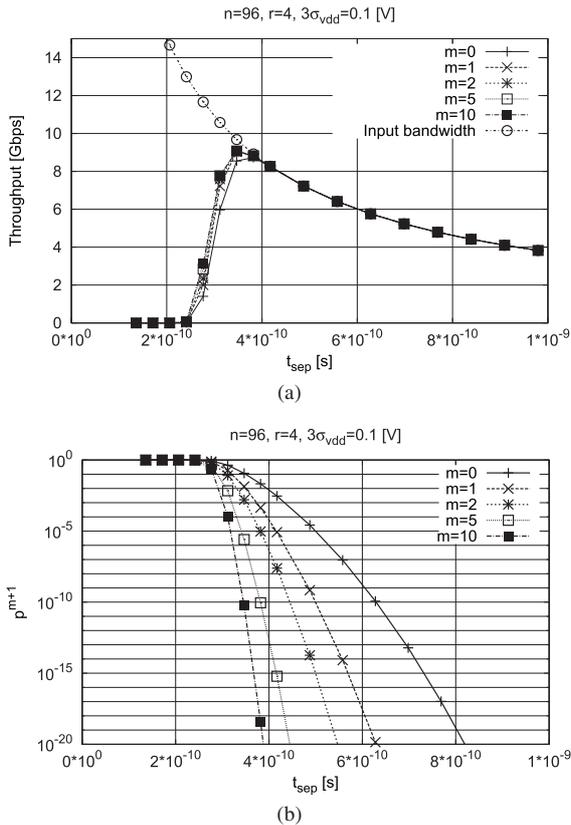


Fig. 12 Effect of the maximum number of data retransmissions (m): (a) throughput and (b) error probability of the n -bit data transmission (p^{m+1}).

input bandwidth is a data rate that the Tx provides. At the small t_{sep} , the throughput is significantly lower than the input bandwidth due to high p . A single data retransmission ($m=1$) is enough to optimize the throughput. In addition, a large m has no throughput degradation, while it decreases an error probability of the n -bit data transmission (p^{m+1}) at the same t_{sep} . In terms of throughput, the optimal t_{sep} is 347 ps and the throughput is 9.07 Gbps, while p^{m+1} is 5.55×10^{-11} when m is 10. When t_{sep} is set to 382 ps, the throughput is just 2.8% lower and p^{m+1} is significantly lower, such as 3.84×10^{-19} compared to that at the optimal point.

Figure 13 shows the throughput vs. t_{sep} depending on r and n , where $3\sigma_{vdd}$ is 0.1 V and m is 10. There is an optimal t_{sep} to maximize the throughput. Until the optimal point from smaller t_{sep} , the throughput is increased by decreasing p . From the optimal point to larger t_{sep} , the throughput is decreased by increasing t_{sep} . A large n increases the throughput as the data-transmission control delay (t_{ctr}) is relatively smaller than t_{total} . However, the throughput increase tends to be saturated around at $n=100$. If bit widths of on-chip links are small (e.g. $n=32$) in an application, buffering several parallel data at the Tx will be effective to increase the throughput.

Table 3 shows the estimated performance comparisons, where n is 96 and r is 4 and $3\sigma_{vdd}$ is 0.1 V. For performance comparisons, a data-transmission link without the error-detection method is considered. The link can be de-

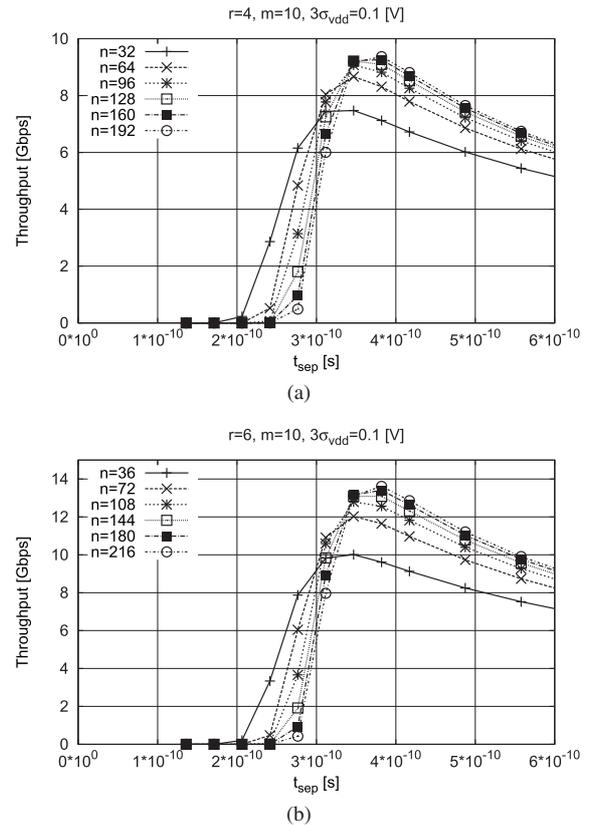


Fig. 13 Throughput vs. t_{sep} depending on n : (a) $r=4$, and (b) $r=6$.

signed based on a bundled-data logic style [17] or the encoding style. In the bundle-data logic style, data and a control signal are separately transmitted and the number of I/Os is $r+1$. However, the control signal must be received after receiving the data. Especially, in long links, deciding the delay value of the control-signal transmission is quite difficult and hence the encoding style tends to be used for the long data transmission [9], [13], [30], [31].

The data transmission without the error-detection method is designed based on the LEDR encoding used in the proposed link. The number of I/Os is 9 ($2r+1$) because the signal of `word_error` is not required. For the data transmission link without the error-detection method ($m=0$) operating at negligible low BER ($< 10^{-20}$) [8], t_{sep} is set to 797 ps to achieve the BER of 2.73×10^{-21} and the error probability of 2.51×10^{-19} (p^{m+1}).

In the proposed link, m is set to 10 to achieve the similar error probability and the estimated throughput is 8.82 Gbps. These two links use different number of I/Os and hence we define *efficiency* that is *throughput over the number of I/Os* for the performance comparison. The proposed link achieves a 71.6% higher efficiency than the data-transmission link without the error-detection method. The area overhead of the proposed link is due to the error detection and the data retransmission. The extra hardware is the error detector in the Rx controller and a few additional gates in the Tx controller to manage signals of `word_error`

Table 3 Performance comparisons ($n=96$, $r=4$, $3\sigma_{vdd}=0.1$ V).

	Throughput [Gbps]	BER	Error probability of n -bit data transmission (p^{m+1})	Data retransmission	# of I/Os	Efficiency [Gbps/(I/Os)]
w/o error detection ($m=0$)	4.63	2.73×10^{-21}	2.51×10^{-19}	No	9 ($2r+1$)	0.514
Proposed ($m=10$)	8.82	2.33×10^{-4}	3.84×10^{-19}	Yes	10 ($2r+2$)	0.882

Table 4 Performance comparisons with related works under a $0.13 \mu\text{m}$ CMOS.

	[9]	[13]	Proposed
Handshake	Per-bit	Per-word	Per-word
Normalized throughput [Gbps]	0.436	4.15^\dagger	8.82
# of I/Os	5	4	10
Efficiency [Gbps/(I/Os)]	0.087	1.038	0.882
Error detection	No	No	Yes
Data retransmission	No	No	Yes

and restart. As the error detector is described in Fig. 13 of [30], it can be simply designed using several number of gates, which results in the small area overhead compared to the link without the error-detection method.

Table 4 shows performance comparisons with related works. Synchronous links [7], [8] need a synchronizer if they are used in the asynchronous multi-chip NoCs. The delay overhead due to the synchronizer is not easily estimated for performance comparisons as the delay time is varied depending on the asynchronous data-transmission condition in the NoCs. Hence, the performance of two asynchronous links is compared with the proposed link. As they are designed under a $0.18 \mu\text{m}$ CMOS technology, the throughput is normalized to a $0.13 \mu\text{m}$ CMOS technology in which the proposed link is designed, where the scaling rule is used in [32]. The asynchronous link in [9] is based on the per-bit handshake style and hence the throughput is very small. The asynchronous link in [13] is based on the per-word handshake style. The throughput is high, but it is evaluated under an ideal case that delay time of the acknowledgement is ignored and a wire delay between a transmitter and a receiver is not included. In addition, the error-detection and the data-retransmission functions are not included, which lowers the throughput than that under the ideal case. The proposed link achieves the high data-transmission efficiency while having the functions of the error detection and the data retransmission.

6. Conclusion

A high-throughput partially parallel inter-chip link architecture has been proposed for asynchronous multi-chip NoCs. The proposed link based on the LEDR encoding transmits chunks of bits (words) based on the per-word handshakes instead of the per-bit handshakes in order to increase the throughput. It retransmits a word once data-transmission errors are detected using the phase information of the LEDR

[†]Delay time of the acknowledgement is ignored. In addition, a wire delay between a transmitter and a receiver is not included.

signals. The BER of the link is theoretically modeled with considering the power-supply noise based dynamic timing variation. Based on the model, the optimized throughput is 8.82 Gbps using 10 I/Os in a $0.13 \mu\text{m}$ CMOS technology. This is a 90.5% higher throughput than that of a link using 9 I/Os without an error-detection method operating at negligible low BER. In future work, we plan to fabricate the proposed link by specifying design parameters based on the proposed model and measure the performance with asynchronous NoCs.

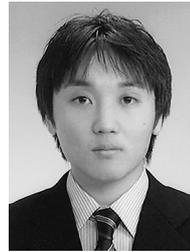
Acknowledgements

This research was supported by JST, CREST.

References

- [1] L. Benini and G.D. Micheli, "Networks on chips: A new SoC paradigm," *Computer*, vol.35, no.1, pp.70–78, 2002.
- [2] D. Lattard, E. Beigne, F. Clermidy, Y. Durand, R. Lemaire, P. Vivet, and F. Berens, "A reconfigurable baseband platform based on an asynchronous network-on-chip," *IEEE J. Solid-State Circuits*, vol.43, no.1, pp.223–235, 2008.
- [3] N. Onizawa, A. Matsumoto, T. Funazaki, and T. Hanyu, "High-throughput compact delay-insensitive asynchronous NoC router," *IEEE Trans. Comput.*, vol.63, no.3, pp.637–649, 2014.
- [4] L. Plana, J. Bainbridge, S. Furber, S. Salisbury, Y. Shi, and J. Wu, "An on-chip and inter-chip communications network for the SpiN-Naker massively-parallel neural net simulator," *Second ACM/IEEE International Symposium on Networks-on-Chip*, pp.215–216, April 2008.
- [5] T. Yoneda and M. Imai, "Dependable routing in multi-chip NoC platforms for automotive applications," *2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pp.217–224, Oct. 2012.
- [6] T. Yoneda, M. Imai, N. Onizawa, A. Matsumoto, and T. Hanyu, "Multi-chip NoCs for automotive applications," *IEEE 18th Pacific Rim International Symposium on Dependable Computing (PRDC)*, pp.105–110, Nov. 2012.
- [7] F. Spagna, L. Chen, M. Deshpande, Y. Fan, D. Gambetta, S. Gowder, S. Iyer, R. Kumar, P. Kwok, R. Krishnamurthy, C.C. Lin, R. Mohanavelu, R. Nicholson, J. Ou, M. Pasquarella, K. Prasad, H. Rustam, L. Tong, A. Tran, J. Wu, and X. Zhang, "A 78mW 11.8Gb/s serial link transceiver with adaptive RX equalization and baud-rate CDR in 32nm CMOS," *2010 IEEE International Solid-State Circuits Conference (ISSCC) Digest of Technical Papers*, pp.366–367, Feb. 2010.
- [8] A. Amirkhany, K. Kaviani, A. Abbasfar, S. Fazeel, W. Beyene, C. Hoshino, C. Madden, K. Chang, and C. Yuan, "A 4.1-pJ/b, 16-Gb/s coded differential bidirectional parallel electrical link," *IEEE J. Solid-State Circuits*, vol.47, no.12, pp.3208–3219, 2012.
- [9] A. Chandrasekaran and K. Boahen, "A 1-change-in-4 delay-insensitive interchip link," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp.3216–3219, May/June 2010.
- [10] J. Lin and K. Boahen, "A delay-insensitive address-event link," *15th IEEE Symposium on Asynchronous Circuits and Systems*, pp.55–

- 62, May 2009.
- [11] Y. Shi, S. Furber, J. Garside, and L. Plana, "Fault tolerant delay insensitive inter-chip communication," 15th IEEE Symposium on Asynchronous Circuits and Systems, pp.77–84, May 2009.
- [12] P. Roine, "A system for asynchronous high-speed chip to chip communication," Second International Symposium on Advanced Research in Asynchronous Circuits and Systems, pp.2–10, March 1996.
- [13] J. Teifel and R. Manohar, "A high-speed clockless serial link transceiver," Ninth International Symposium on Asynchronous Circuits and Systems, pp.151–161, May 2003.
- [14] M. Dean, D.L. Dill, and M. Horowitz, "Efficient self-timed level-encoded 2-phase dual-rail (LEDR)," Advanced Research in VLSI, pp.55–70, 1991.
- [15] N. Onizawa and T. Hanyu, "Highly reliable multiple-valued one-phase signaling for an asynchronous on-chip communication link," IEICE Trans. Inf. & Syst., vol.E93-D, no.8, pp.2089–2099, Aug. 2010.
- [16] N. Onizawa, A. Matsumoto, and T. Hanyu, "Long-range asynchronous on-chip link based on multiple-valued single-track signaling," IEICE Trans. Fundamentals, vol.E95-A, no.6, pp.1018–1029, June 2012.
- [17] J. Sparsø and S. Furber, Principles of asynchronous circuit design: A systems perspective, Kluwer Academic Publisher, 2001.
- [18] S.J. Lee, K. Kim, H. Kim, N. Cho, and H.J. Yoo, "Adaptive network-on-chip with wave-front train serialization scheme," 2005 Symposium on VLSI Circuits, Digest of Technical Papers, pp.104–107, June 2005.
- [19] R. Dobkin, Y. Perelman, T. Liran, R. Ginosar, and A. Kolodny, "High rate wave-pipelined asynchronous on-chip bit-serial data link," 13th IEEE International Symposium on Asynchronous Circuits and Systems, pp.3–14, March 2007.
- [20] R. Dobkin, M. Moyal, A. Kolodny, and R. Ginosar, "Asynchronous current mode serial communication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.18, no.7, pp.1107–1117, July 2010.
- [21] A. Sharifi, H. Zhao, and M. Kandemir, "Feedback control for providing QoS in NoC based multicores," Design, Automation Test in Europe Conference Exhibition (DATE), 2010, pp.1384–1389, 2010.
- [22] M. Shams and J. Ebergen, "Modeling and comparing CMOS implementations of the C-element," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.6, no.4, pp.563–567, 1998.
- [23] H.K. Jung, K. Lee, J.S. Kim, J.J. Lee, J.Y. Sim, and H.J. Park, "A 4Gbps 3-bit parallel transmitter with the crosstalk-induced jitter compensation using TX data timing control," 2008 IEEE Asian Solid-State Circuits Conference, pp.201–204, Nov. 2008.
- [24] Z. Feng, Y. Yi, Y. Zongren, P. Chiang, and H. Weiwu, "A low latency transceiver macro with robust design technique for processor interface," 2009 IEEE Asian Solid-State Circuits Conference, pp.185–188, Nov. 2009.
- [25] A. Hu and F. Yuan, "Intersignal timing skew compensation of parallel links with voltage-mode incremental signaling," IEEE Trans. Circuits and Systems I: Regular Papers, vol.56, no.4, pp.773–783, 2009.
- [26] P. Teehan, G. Lemieux, and M. Greenstreet, "Estimating reliability and throughput of source-synchronous wave-pipelined interconnect," 3rd ACM/IEEE International Symposium on Networks-on-Chip, pp.234–243, May 2009.
- [27] F.C. Cheng and S.L. Ho, "Efficient systematic error-correcting codes for semi-delay-insensitive data transmission," 2001 International Conference on Computer Design (ICCD), pp.24–29, Sept. 2001.
- [28] Q. Yu and P. Ampadu, "Adaptive error control for reliable systems-on-chip," 2008 IEEE International Symposium on Circuits and Systems (ISCAS), pp.832–835, May 2008.
- [29] A. Katoch, H. Veendrick, and E. Seevinck, "High speed current-mode signaling circuits for on-chip interconnects," IEEE International Symposium on Circuits and Systems (ISCAS), pp.4138–4141, vol.4, May 2005.
- [30] M. Imai and T. Yoneda, "Improving dependability and performance of fully asynchronous on-chip networks," 17th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), pp.65–76, 2011.
- [31] T. Bjerregaard and J. Sparso, "Implementation of guaranteed services in the MANGO clockless network-on-chip," IEE Proceedings - Computers and Digital Techniques, vol.153, no.4, pp.217–229, 2006.
- [32] B. Davari, R. Dennard, and G. Shahidi, "CMOS scaling for high performance and low power-the next ten years," Proc. IEEE, vol.83, no.4, pp.595–606, 1995.



Naoya Onizawa received the B.E., M.E. and D.E. degrees in Electrical and Communication Engineering from Tohoku University, Japan, in 2004, 2006 and 2009, respectively. He is currently an Assistant Professor in the Frontier Research Institute for Interdisciplinary Sciences at Tohoku University, Japan. He was a postdoctoral fellow at Tohoku University from 2009 to 2011 and at University of Waterloo, Canada in 2011 and at McGill University, Canada from 2011 to 2013. His main interests and activities

are in the energy-efficient VLSI design based on asynchronous circuits and multiple-valued circuits, and their applications, such as LDPC decoders, associative memories, and Network-on-Chips. He received the Best Paper Award in IEEE Computer Society Annual Symposium on VLSI in 2010. Dr. Onizawa is a Member of the IEEE.



Akira Mochizuki received the B.E., the Master of Information Science and the D.E. degrees from Tohoku University, Sendai, Japan, in 1993, 1995 and 2006, respectively. From 1995 to 2002, he joined NEC, Japan. From 2007 to 2012, he joined Renesas, Japan. In both the semiconductor companies, he was engaged in the design and development of the embedded CPU IPs. From 2002 to 2007, he was a Research Associate for researching high-performance integrated circuits in Tohoku University. He is

currently an Assistant Professor in CSIS, Tohoku University. His main interests and activities are a design of high-speed low-power VLSI and its application. He received the Ando Incentive Prize for the Study of Electronics in 2005 from the foundation of ANDO Laboratory. Dr. Mochizuki is a member of the IEEE.



Hirokatsu Shirahama received the B.E., M.E. and D.E. degrees from Tohoku University, Japan, in 2005, 2007, and 2010, respectively. Currently, he is a research coordinating staff for industry-government-academia partnerships at the Research Institute of Electrical Communication, Tohoku University. From 2010 to 2013, He joined Renesas Electronics Corporation, Japan, where he was engaged in microcontroller design. His main interests and activities

are in high-speed low-power LSI design based on current-mode circuits with power management technique and its applications.



Masashi Imai received his Ph.D. degree in Electronic Engineering from the University of Tokyo, Japan, in 2003. From 2005 to 2011, he was a project associate professor at the University of Tokyo. Currently, he is working as an associate professor at Hirosaki University. His research interests include dependable computing system design, asynchronous VLSI design, and Globally-Asynchronous Locally-Synchronous Network-on-Chip design.



Tomohiro Yoneda received Dr. Eng. degree in Computer Science from the Tokyo Institute of Technology, Tokyo, Japan in 1985. He is a Professor of National Institute of Informatics. He was a visiting researcher of Carnegie Mellon University from 1990 to 1991. His research activities currently focus on asynchronous circuit design and Networks-on-Chip.



Takahiro Hanyu received the B.E., M.E. and D.E. degrees in Electronic Engineering from Tohoku University, Sendai, Japan, in 1984, 1986 and 1989, respectively. He is currently a Professor in the Research Institute of Electrical Communication, Tohoku University. His general research interests include nonvolatile logic circuits and their applications to ultra-low-power and/or PVT-variation-free VLSI processors, and multiple-valued current-mode circuit and its application to power-aware asyn-

chronous Network-on-Chip systems. He received the Sakai Memorial Award from the Information Processing Society of Japan in 2000, the Judge's Special Award at the 9th LSI Design of the Year from the Semiconductor Industry News of Japan in 2002, the Special Feature Award at the University LSI Design Contest from ASP-DAC in 2007, the APEX Paper Award of Japan Society of Applied Physics in 2009, the Excellent Paper Award of IEICE, Japan, in 2010, Ichikawa Academic Award in 2010, the Best Paper Award of IEEE ISVLSI 2010, and the Paper Award of SSDM 2012. Dr. Hanyu is a Senior Member of the IEEE.