Interconnect-Fault-Resilient Delay-Insensitive Asynchronous Communication Link Based on Current-Flow Monitoring

Naoya Onizawa, Atsushi Matsumoto and Takahiro Hanyu
Research Institute of Electrical Communication, Tohoku University
2-1-1 Katahira, Aoba-ku, Sendai, Miyagi, Japan
Email: {onizawa,matumoto,hanyu}@ngc.riec.tohoku.ac.jp

***Abstract*—Delay-insensitive asynchronous on-chip communication links are a key element to realize a highly reliable asynchronous Network-on-Chip system. However, even a single permanent fault, such as an interconnect fault, causes a deadlock state in the system. This paper presents an interconnect-fault-resilient delay-insensitive asynchronous communication link based on current-flow monitoring. Since current flow upon an interconnect is cut off by an open fault in the interconnect, the current is fed back to a transmitter, which increases a feedback current monotonically. Monitoring the feedback current makes it possible to detect the interconnect fault with delay insensitivity. The proposed link is evaluated by a 0.13μm CMOS technology with a Triple Modular Redundancy (TMR)-based asynchronous communication link which is resilient to the interconnect fault without the delay insensitivity. As a result, the energy consumption and the number of wires of the proposed link are reduced to 57% and 33%, respectively, in comparison with those of the conventional one.**

I. Introduction

As the semiconductor industry moves further into the nanometer regime, intellectual property (IP) components, such as processor cores, DSPs and memories are integrated on a single chip. Networks-on-Chip (NoC) design paradigm is scalable on-chip global-communication for multi-processor System-on-Chips (SoCs) [1]. NoCs based on Globally Asynchronous Locally Synchronous (GALS) [2] or fully-asynchronous system [3] take advantages of asynchronous circuits, such as low power consumption and communication robustness in quasi delay-insensitive (QDI) logic style [4].

On the other hand, the total length of interconnect per chip is more than a kilometer and continuously increasing [5]. Since even a single open fault in the interconnects by electromigration [6] causes fatal system errors, the interconnect fault would be more critical in the future technology nodes [7]. In the systems which require long communication links [8], such as the NoCs, it is important to consider the interconnect reliability.

Some literatures have addressed permanent faults, such as the interconnect fault in the asynchronous circuits [9]-[11]. In the conventional methods, the interconnect fault is detected by using outputs of the asynchronous circuits corresponding to a receiver in the communication link. When a received signal has an error which is caused by the interconnect fault or a delay fault upon the interconnect wires, only the interconnect fault needs to be detected. Although a timing assumption is useful to eliminate errors due to the delay fault, the QDI asynchronous circuits lose the delay insensitivity due to the timing assumption. Therefore, it is difficult to detect the interconnect fault with the delay insensitivity in the conventional methods.

This paper presents an interconnect-fault-resilient delay-insensitive asynchronous communication link for the highly reliable asynchronous systems, such as NoCs. In the proposed method, the interconnect fault is detected at a transmitter. Since the delay fault affects only the transmitted signal through the interconnect wires, the delay fault is negligible at the transmitter. To detect whether the interconnect wire is fault or not at the transmitter, the proposed communication link is designed based on a feedback architecture which is realized based on current-flow monitoring. In the proposed circuit, an input current is split into a transmitted current to the receiver and a feedback current to the transmitter itself. When the interconnect fault is occured, the feedback current is increased due to an open fault of the interconnect which carries the transmitted current. Therefore, the interconnect fault can be detected with the delay insensitivity by monitoring the feedback current at the transmitter.

The rest of the paper is organized as follows. Section II describes the delay-insensitive interconnect-fault detection based on the feedback architecture. Section III describes a fault-recovering system based on the proposed fault detection. Section IV describes hardware implementation and shows simulated results with relevant works. Section V concludes this paper.

II. Delay-Insensitive Interconnect-Fault Detection

A. Interconnect Fault in QDI Asynchronous Circuits

Fig. 1 shows a model of communication links between modules, such as IP cores and routers of the NoCs. In no fault condition shown in Fig. 1 (a), these modules can communicate by using interconnect wires which are fabricated using Cu in recent technology nodes. However, a void in the Cu interconnects causes open faults by electromigration when transistors are in operation [6] shown in Fig. 1 (b). Moreover, since technology scaling leads to increased current carrying requirements, the open-fault problem would be more critical in the future technology nodes [7].

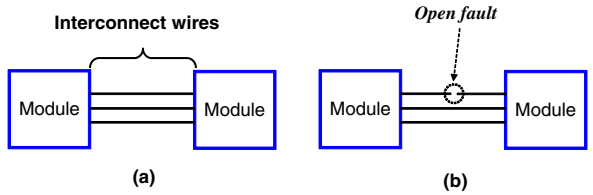

Figure 1. Model of communication links with: (a) no fault and (b) an open fault.

978-3-9810801-7-9/DATE11/©2011 EDAA

Fig. 2 shows an asynchronous communication link based on a QDI logic style, where the communication is performed by request-acknowledge based handshaking. First, the left module transmits data with the request information ($DATA + REQ$) from the input port. Second, the right module receives the transmitted data in the output port and then transmits the acknowledge information (ACK) to the left module. Since the QDI logic style avoids any timing constraints with only an assumption, where the wires at fan-out point must have roughly equal delay, asynchronous circuits are robust against delay faults due to process, voltage and temperature variations [4]. However, permanent faults, such as the interconnect fault, cause a deadlock state in QDI asynchronous circuits. Since each communication link waits for $DATA + REQ$ or ACK before it changes its output, the output never change when $DATA + REQ$ or ACK is not changed due to the interconnect fault.

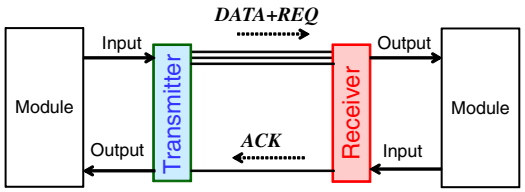

Figure 2. Delay-insensitive asynchronous communication link.

B. Related works

In an algorithm level, fault-tolerant routing algorithms under the permanent faults have been proposed for NoCs [13],[14], but these approaches are limited to specific applications, such as NoCs. In a circuit level, [9]-[11] have addressed the permanent faults, such as the interconnect fault in the asynchronous circuits. In [9], error detection and retransmission methods are implemented in asynchronous communication link for tackling intermittent and permanent faults, while the link is not implemented based on the QDI logic style. In [10], a duplication-based method has been proposed. The outputs of the duplicated circuits are compared within a specified time window to detect transient and permanent faults, but the duplicated circuit cannot be operated correctly under the permanent faults. In [11], Triple Modular Redundancy (TMR)-based method has been proposed. Since the TMR-based circuit produces the correct output when two of three circuits generate the same outputs, the TMR-based circuit can be operated under a single permanent fault. However, the majority voting scheme of TMR may cause a deadlock state or an instability in the circuit, because the other one of three circuits, which is not used in the majority voting may cause an unexpected output under an unbounded delay model in the QDI logic style. To prevent the deadlock state, a timing assumption is required in the TMR-based circuit. In this way, the conventional circuit cannot detect the interconnect fault with the delay insensitivity.

C. Concept

The problem of the conventional methods is to detect the interconnect fault at the output port of the module shown in Fig. 3 (a). In the QDI logic style, the data is received, when all incoming data is detected at the output port of the module. Even if only 1-bit data is not detected, the data cannot be received. The 1-bit missing is caused by the interconnect fault or the delay fault. However, these two faults cannot be distinguished at the output port of the module. Since the data cannot be received when the interconnect fault is occured, the communication becomes a deadlock state. To prevent the deadlock state, a timing assumption to detect the interconnect fault is useful. However, the timing assumption makes the QDI circuit lose the delay insensitivity shown at the first column in Table I. Therefore, it is impossible to detect only the interconnect fault with the delay insensitivity at the output port.

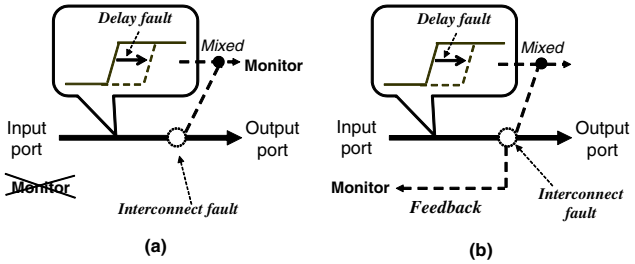

Figure 3. Link architectures with: (a) no feedback and (b) feedback.

To detect only the interconnect fault, it is necessary to distinguish between the interconnect and the delay faults. Since the delay fault affects only a transmitted signal, the delay fault is negligible at the input port. The interconnect-fault detection at the input port is a key factor to realize the delay-insensitive interconnect-fault detection. However, both the interconnect and the delay faults cannot be detected at the input port in the conventional architecture shown at the 2nd column in Table I. Fig. 3 (b) shows the proposed link architecture. In the proposed link, information of the interconnect fault is obtained by feedback at the input port of the module. Since it is not necessary to distinguish between the interconnect and the delay fault, the interconnect fault is detected without a timing assumption, which realizes the delay-insensitive interconnect-fault detection shown at the 3rd column in Table I. In the next subsection, basic hardware implementation is described to realize the feedback architecture.

Table I
CHARACTERISTICS OF THE FAULT DETECTIONS.

	Output port	Input port	
		No feedback	Feedback
Monitor interconnect fault	Yes	No	Yes
Monitor delay fault	Yes	No	No
Distinguish the two faults	No	-	Yes
Detect interconnect fault	Yes	No	Yes
Delay insensitivity	No	-	Yes

D. Delay-Insensitive Interconnect-Fault Detection Based on Current-Flow Monitoring

The interconnect-fault detection by using a voltage-mode circuit in the input port is considered. Fig. 4 shows the voltage-mode circuit with no fault and an open fault in interconnects. In no interconnect-fault condition, a voltage level of the output port V_{out} becomes high when data is transmitted from the other module, while a voltage level of the input port V_{in} also becomes high. When the interconnect fault is occured, V_{out} is still low when data is transmitted

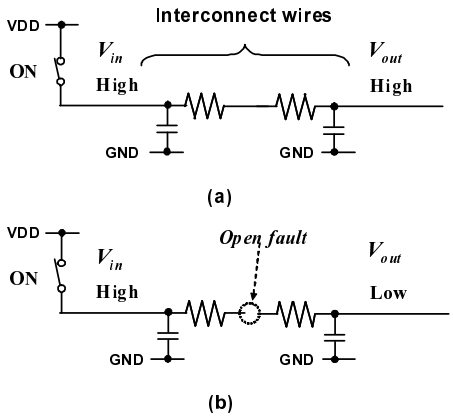

Figure 4. Voltage-mode circuits with (a) no fault, (b) an open fault.

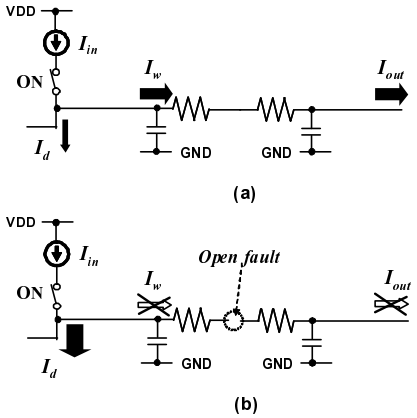

Figure 5. Proposed split-current circuits for the current-flow monitoring with (a) no fault, (b) an open fault.

from the other module. However, V_{in} becomes high due to charging capacitances in the interconnect wires. Since these voltage levels of V_{in} are the same in both no fault and open-fault conditions, the module in the input port cannot recognize the interconnect fault. Therefore, it is difficult to realize the interconnect-fault detection with the delay insensitivity in the voltage-mode circuit.

In this paper, we propose the current-flow monitoring to realize the delay-insensitive interconnect-fault detection. In the proposed method shown in Fig. 5, the communication link is implemented using a current-mode circuit. I_{in} in the input port is split into a transmitted current I_w and a detected current I_d. In no fault condition, I_{out} is generated through the interconnect wire when data is transmitted, while small amount of current I_d is generated. Due to electromigration, a wire resistance is increased over time, which decreases the amount of current I_w, while I_d is increased. When the interconnect fault is occurred, I_{out} is not generated when data is transmitted, while a large amount of current I_d is generated. Since I_d in the open-fault condition is larger than I_d in no fault condition, the module in the input port can recognize the interconnect fault by comparing I_d with a threshold current whose amount detemines whether the link is an open fault or not. As a result, the interconnect-fault detection in the input port can be realized based on the proposed current-flow monitoring, which results in the delay-insensitive interconnect-fault detection. In the next section, the fault recovering system for the asynchronous communication link based on the proposed method is described.

III. Fault Recovering System for Asynchronous Communication Links

Fig. 6 shows an overall structure of a fault recovering system for an asynchronous communication link. It is assumed that only a single fault in interconnects occurs. In the proposed fault-recovering system, DATA BLOCK is operated at four modes: NORMAL, FAULT, CONFIG and RECOVERY MODEs, while ACK BLOCK is operated at two modes: NORMAL and RECOVERY MODEs shown in Fig. 7. Each mode transition is started, respectively, by an interconnect fault of DATA or ACK. In the proposed link, a 4phase protocol is used [4]. Data transmission between the transmitter and the receiver is performed based on current-mode signalling, while the other blocks are implemented by a voltage-mode circuit.

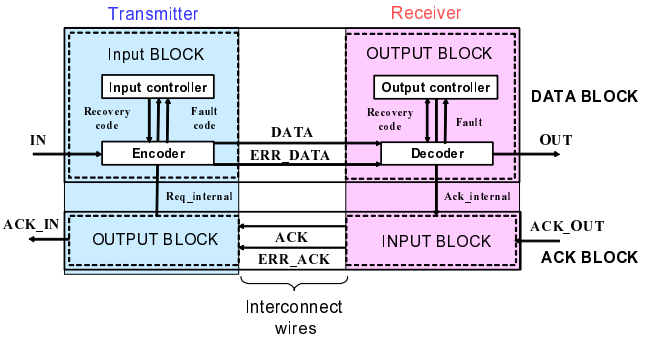

Figure 6. Overall structure of the proposed communication link.

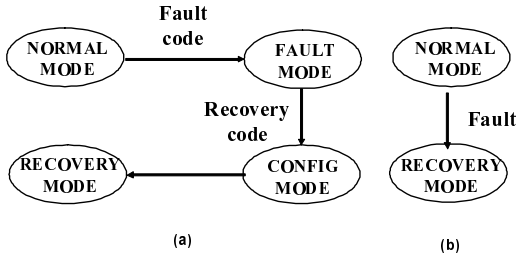

Figure 7. Mode transition in (a) DATA BLOCK and (b) ACK BLOCK.

The fault-recovering system is described using a timing diagram shown in Fig. 8. In this example, the interconnect wire corresponding to Data "3" is fault. Both DATA BLOCK and ACK BLOCK are operated at NORMAL MODE in the initial condition. Table II shows 1-of-4 encoding for DATA BLOCK at NORMAL MODE, where DATA is represented by 4-bit codeword. At NORMAL MODE, the asynchronous communication is operated as the same manner as the 1-of-4 asynchronous communication link [12]. First, (0,0,0,1) as Data "0" is transmitted from the transmitter to the receiver. Second, the receiver detects the one-hot code, and then transmits (1,0) as Ack for Data "0" to the transmitter, where the dual-rail encoding for ACK BLOCK is shown in Table III. Third, the transmitter detects Ack, and then

Table II
1-OF-4 ENCODING FOR DATA BLOCK IN NORMAL MODE.

	Logic value	DATA
Data	"0"	(0,0,0,1)
	"1"	(0,0,1,0)
	"2"	(0,1,0,0)
	"3"	(1,0,0,0)
Spacer		(0,0,0,0)

Table III
DUAL-RAIL ENCODING FOR ACK BLOCK.

	MODE	(ACK, ERR_ACK)
Ack	NORMAL	(1,0)
	RECOVERY	(0,1)
Spacer		(0,0)

transmits (0,0,0,0) as Spacer to the receiver. Fourth, the receiver detects Spacer, and then transmits (0,0) as Spacer to the transmitter. In this way, 4 steps per communication is required in the 4phase protocol.

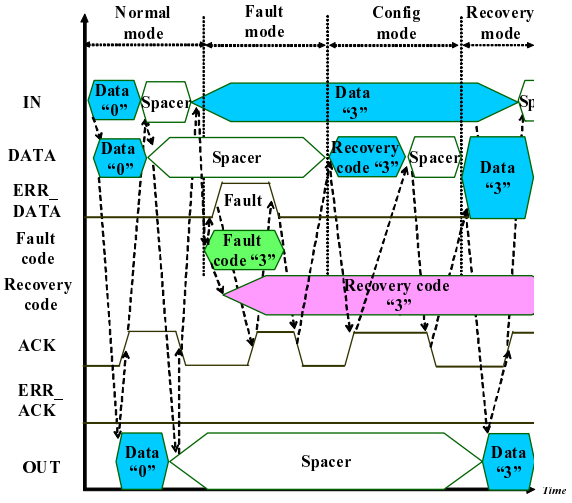

Figure 8. Timing diagram of the proposed communication link when the interconnect wire corresponding to Data "3" is fault.

When Data "3" is transmitted, the interconnect fault is detected in an encoder of the transmitter. Then, the mode transits to FAULT MODE in the transmitter. Table IV shows single-rail encoding for DATA BLOCK at FAULT MODE, where ERR_DATA is represented by 1-bit codeword. The transmitter transmits "1" as the fault information to the receiver. Concurrently, a recovery code "3" is generated based on the fault code. Once the receiver detects the fault information, the mode of the receiver transits to FAULT MODE and deals the transmitted code with the recovery code.

Then, the mode transits to CONFIG MODE in the transmitter. Table V shows 1-of-4 encoding for DATA BLOCK at CONFIG MODE. The recovery code is generated to avoid using the interconnect wire with an open fault. Once the transmitter transmits (0,0,0,1) as the recovery code "3" to the receiver, the receiver recognizes the recovery code "3" because the receiver have already received the fault information at FAULT MODE. When the receiver is operated at CONFIG MODE, the interconnect wire corresponding to Data "3" is replaced by an interconnect wire of ERR_DATA.

Table IV
SINGLE-RAIL ENCODING FOR DATA BLOCK IN FAULT MODE.

	ERR_DATA
Fault	1
Spacer	0

Table V
1-OF-4 ENCODING FOR DATA BLOCK IN CONFIG MODE.

	Logic value	DATA
Recovery code	"0"	(1,0,0,0)
	"1"	(0,1,0,0)
	"2"	(0,0,1,0)
	"3"	(0,0,0,1)
Spacer		(0,0,0,0)

Finally, the mode is transit to RECOVERY MODE in the transmitter. Table VI shows 1-of-5 encoding for DATA BLOCK at RECOVERY MODE. The transmitter transmits (0,0,0,0,1) as Data "3" to the receiver. The receiver recognizes (0,0,0,0,1) as Data "3" and then is operated at RECOVERY MODE. Then, DATA BLOCK is operated at RECOVERY MODE to continue transmitting Data under the interconnect fault corresponding to Data "3".

The mode of the fault-recovering system for ACK BLOCK is transit shown in Fig. 7 (b). First, ACK BLOCK is operated at NORMAL MODE. Once the interconnect fault of ACK is detected in INPUT BLOCK of the receiver, the dual-rail code for ACK BLOCK is changed from (1,0) to (0,1) shown in Table III. at RECOVERY MODE Since OUTPUT BLOCK of the transmitter detects the acknowledge information by ORing the dual-rail code at any modes, the acknowledge information can be received under the interconnect fault of ACK.

In the next section, a circuit diagram of the proposed communication link is described, and the performances and characteristics are compared with the conventional ones.

IV. DESIGN AND EVALUATION

A. Hardware Implementation

Fig. 9 shows a block diagram of the encoder in the transmitter in DATA BLOCK. We focus on hardware implementation of only DATA BLOCK in the proposed communication link due to page limitation. It is assumed that IN is encoded as a 1-of-4 code [12]. In NORMAL MODE, the codes of IN and DATA are the same. Fig. 10 shows a circuit diagram of an driver which includes four current drivers and a current generator. In the current driver, I_{in} is split into I_w and I_d in no interconnect-fault condition, where a voltage level V_d is low. Once the interconnect fault is occurred, I_d becomes large due to an open fault of an interconnect which carries I_w. Since I_d is copied to I_c using a current mirror, V_c becomes low due to a large amount of current I_c. Then, V_d becomes high through an inverter, which generates the fault code.

The fault code leads to transition to FAULT MODE which asserts ERR_DATA, while DATA is still (0,0,0,0). Concurrently, the recovery code is generated based on the fault code. When the mode transits to CONFIG MODE, the output of the encoder becomes the recovery code. Since the recovery code is generated to use only interconnect wires with no fault condition, the fault code returns to (0,0,0,0). At an input selector, connections between (DATA, ERR_DATA) and IN are reconfigured based on the recovery code to avoid using an interconnect wire with an open fault. When the

Table VI
1-OF-5 ENCODING FOR DATA BLOCK IN RECOVERY MODE WHEN THE INTERCONNECT CORRESPONDING TO DATA "3" IS FAULT.

	Logic value	(DATA, ERR_DATA)
Data	"0"	(0,0,0,1,0)
	"1"	(0,0,1,0,0)
	"2"	(0,1,0,0,0)
	"3"	(0,0,0,0,1)
Spacer		(0,0,0,0,0)

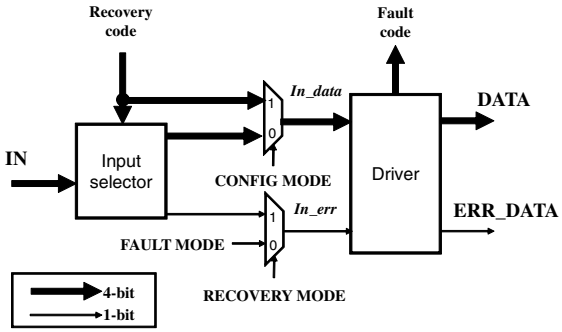

Figure 9. Block diagram of the encoder in the transmitter.

mode transits to RECOVERY MODE, data is transmitted by the 1-of-5 encoding through DATA and ERR_DATA.

Fig. 11 shows a block diagram of the decoder in the receiver of DATA BLOCK. Fig. 12 shows a circuit diagram of a detector in the receiver which includes five current detectors. In the current detector, a current I_{out} is converted to a voltage V_{out}. In NORMAL MODE, the codes of DATA and OUT are the same. Once the interconnect fault is occured, ERR_DATA becomes "1", which asserts Fault. Then, the mode is transit to FAULT MODE. In FAULT MODE, when the recovery code is transmitted from the transmitter, the recovery code is stored in the receiver, which makes the mode transit to CONFIG MODE. At an output selector, the connection between (DATA, ERR_DATA) and OUT is reconfigured based on the recovery code to avoid using an interconnect wire with an open fault. Then, data is transmitted by the 1-of-5 encoding through DATA and ERR_DATA, the mode is transit to RECOVERY MODE.

Input and output controllers are also implemented based on the QDI logic style, but the detailed structure is not

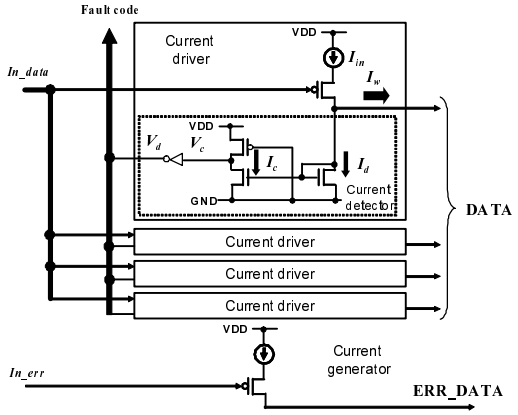

Figure 10. Circuit diagram of the driver in the transmitter.

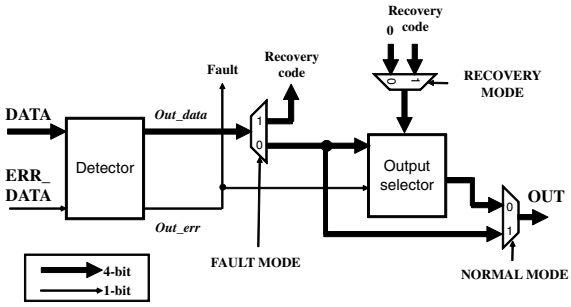

Figure 11. Block diagram of the decoder in the receiver.

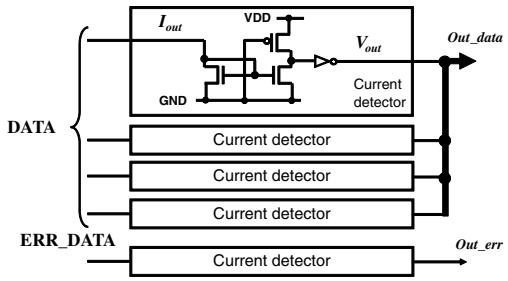

Figure 12. Circuit diagram of the detector in the receiver.

described due to page limitation.

B. Evaluation

The proposed and conventional asynchronous communication links are evaluated by HSPICE simulation under Silterra 0.13μm CMOS technology. Wire lengths between the transmitter and the receiver are set to 1mm. PI model is used for the interconnect wire, where the parameters of the wire are resistance = 91 Ω/mm and capacitance = 84 fF/mm.

Fig. 13 shows simulated waveforms of the proposed asynchronous communication link when the interconnect wire corresponding to Data "3" is fault. The unit current of I_{in} is set to 165 uA. When Data is transmitted through the interconnect wire with the open fault, I_d becomes large, which leads to transit to FAULT MODE to recover the communication link.

Table VII shows comparisons of performance in serial asynchronous communication links. The proposed communication link is compared with three 1-of-4 communication links using the single [12], the duplicated [10] and the TMR-based circuits [11]. The single 1-of-4 link is designed with delay insensitivity, but an interconnect-fault detector is not implemented. The duplication-based link can detect the interconnect fault, but the link cannot be operated correctly under the interconnect fault. The TMR-based link can be operated correctly under the interconnect fault, but the timing assumption is required, which loses the delay insensitivity. In contrast, the proposed link can be operated correctly under the interconnect fault with the delay insensitivity. The energy consumption of the proposed link is reduced to 57% with respect to the TMR-based link, while the number of wires of the proposed link are almost the same as the single 1-of-4 link. The proposed link requires 1280 transistors, where the input and output controllers use 418 transistors. As the input and output controllers can be shared in parallel

Table VII
COMPARISONS OF PERFORMANCE IN SERIAL ASYNCHRONOUS COMMUNICATION LINKS.

	1-of-4 encoding	1-of-4 encoding (duplicated)	1-of-4 encoding (TMR)	**Proposed**
Interconnect fault detection	No	**Yes**	**Yes**	**Yes**
Correct operation under the fault	No	No	**Yes**	**Yes**
Delay insensitivity	**Yes**	No	No	**Yes**
Number of wires (n bits)	**$2n$+1**	2*(2n+1)	3*(2n+1)	**$2n$+3**
Throughput [Mbps]	840	741	622	837
Power [μW]	328	616	761	584
Energy [fJ/bit]	388	831	1223	698
Number of transistors	545	1049	1659	1280 [1]

[1] Input and output controllers use 418 transistors, where parallel link shares the controllers, which can decrease area overhead of the proposed link.

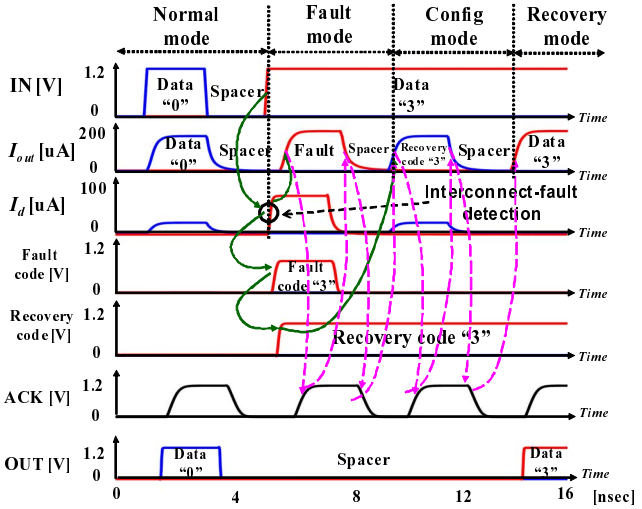

Figure 13. Simulated waveforms.

communication links, area overhead becomes small in the proposed communication link.

V. CONCLUSION

In this paper, an interconnect-fault delay-insensitive asynchronous communication link has been proposed for highly reliable asynchronous systems, such as asynchronous NoCs. In the proposed communication link, only an interconnect fault is detected at a transmitter based on a feedback architecture which is simply designed by a current-mode circuit, since the transmitter naturally eliminates a factor of a delay fault. Moreover, the interconnect wire with an open fault is replaced by an extra interconnect wire based on new 1-of-5 encoding, which maintains asynchronous communication. As a result the proposed link is operated correctly under a single fault in the interconnects with delay insensitivity.

ACKNOWLEDGMENT

This research was supported by JST, CREST. This simulation is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc.

REFERENCES

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

[2] M. Krstic, et al., "Globally asynchronous locally synchronous circuits: Overview and outlook, " *IEEE Des. Test Comput.*, vol. 24, no. 5, pp. 430-441, Sep./Oct. 2007.

[3] D. Lattard, E. Beigne, F. Clemidy, 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 (JSSC)*, vol. 43, no. 1 Jan. 2008.

[4] J. Sparsø, and S. Furber, "Principles of Asynchronous Circuit Design," *Kluwer Academic Publisher*, pp. 9-28, 2001.

[5] International Technology Roadmap for Semiconductors, 2007 edition, Interconnect Chapter; *http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_Interconnect.pdf*.

[6] J. Gill, T. Sullivan, S. Yankee, H. Barth and A. von Glasow, "Investigation of via-dominated multi-modal electromigration failure distributions in dual damascene Cu interconnects with a discussion of the statistical implications," *Proc. Int. Ref. Phys. Symp.*, pp. 298-304, 2002.

[7] Z. Choi, M. Tsukasa, J. M. Lee, G.-H. Choi, S. Choi and J.-T. Moon, "Effect of pre-existing void in sub-30nm Cu Interconnect Reliability," *Proc. IEEE Int. Reliability Physics Symposium (IRPS)*, pp. 903 - 905, 2010.

[8] Y. Ohtake, N. Onizawa, and T. Hanyu, "High-Performance Asynchronous Intra-Chip Communication Link Based on a Multiple-Valued Current-Mode Single-Track Scheme," *Proc. IEEE Int. Symp. Circuits and Systems,* pp.1000-1003, Taipei, Taiwan, May 2009.

[9] T. Lehtonen, P. Liljeberg and J. Plosila, "Online Reconfigurable Self-Timed Links for Fault Tolerant NoC," *VLSI Design*, vol. 2007, Article ID 94676, 2007.

[10] T. Verdel and Y. Makris, "Duplication-Based Concurrent Error Detection in Asynchronous Circuits: Shortcomings and Remedies," *Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems 2002*, pp.345-353, 2002.

[11] S. Almukharizim and O. Sinanoglu, "A Hazard-Free Majority Voter for TMR-Based Fault Tolerance in Asynchronous Circuits," *Proc. Design and Test Workshop 2007*, pp. 93-98, 2007.

[12] W. J. Bainbridge, and S. B. Furber, "Delay Insensitive System-on-Chip Interconnectusing 1-of-4 Data Encoding," *Proc. 7th IEEE Int. Symp. Asynchronous Circuits Syst.*, pp. 74-82, Mar. 2001.

[13] V. Puente, J. A. Gregorio, F. Vallejo and R. Beivide, "Immunet: A cheap and robust fault-tolerant packet routing mechanism," *ACMSIGARCH Computer Architecture News*, 32(2):198, 2004.

[14] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester and D. Blaauw, "A Highly Resilient Routing Algorithm for Fault-Tolerant NoCs," *Proc.Design, Automation and Test in Europe 2009*, pp.21-26, 2009.