

Open-Fault Resilient Multiple-Valued Codes for Reliable Asynchronous Global Communication Links

Naoya ONIZAWA^{†a)}, Atsushi MATSUMOTO^{††b)}, and Takahiro HANYU^{†††c)}, *Members*

SUMMARY This paper introduces open-wire fault-resilient multiple-valued codes for reliable asynchronous point-to-point global communication links. In the proposed encoding, two communication modules assign complementary codewords that change between two valid states without an open-wire fault. Under an open-wire fault, at each module, the codewords don't reach to one of the two valid states and remains as "invalid" states. The detection of the invalid states makes it possible to stop sending wrong codewords caused by an open-wire fault. The detectability of the open-wire fault based on the proposed encoding is proven for m -of- n codes. The proposed code used in the multiple-valued asynchronous global communication link is capable of detecting a single open-wire fault with 3.08-times higher coding efficiency compared with a conventional multiple-valued code used in a triple-modular redundancy (TMR) link that detects an open-wire fault under the same dynamic range of logical values.

key words: fault tolerance, m -of- n codes, error detection codes, bidirectional communication

1. Introduction

The performance of digital systems rapidly grows by integrating many processing cores in a single chip thanks to reduction in feature sizes. Transistors and local wires are shrunk as technology node grows, while global wires are not [1] and thus tend to be bottleneck in the complicated systems, such as System-on-Chips (SoC). One of the promising approaches is to use global asynchronous communication based on handshaking among processing cores. The asynchronous communication releases restrictions due to a global clock signal, leading design scalability, flexibility, reliability, and low power dissipation. The systems that utilize global asynchronous communication are realized in fully asynchronous or Global Asynchronous Locally Synchronous (GALS) system [2] that combines local synchronous processing cores and global asynchronous communication.

The global asynchronous communication has been recently used in several applications: Network-on-Chips (NoC) [3], [4], Low-Density Parity-Check (LDPC) decoders [5], [6], commercialized field-programmable gate ar-

rays (FPGA) [7] Ethernet routing chips [8], and digital signal processing (DSP) chips [9]. Some asynchronous communication links have been proposed for high performance systems [10]–[12] as well as reliable systems, where asynchronous communication links are robust against various faults, such as transient and permanent faults [13]–[17].

We have proposed high-throughput wire-efficient asynchronous communication links using multiple-valued codes and current-mode circuits [18]–[21] for high-performance GALS or fully-asynchronous systems. In this paper, open-wire fault resilient multiple-valued encoding is introduced for reliable asynchronous global communication links. An open-wire fault is becoming a critical issue in current and future technology nodes where a total interconnection length in a single chip is in kilometer order and it continues to increase [1]. Even a single open-wire fault in the link by electromigration [22] causes fatal errors in a whole system [23].

To realize open-wire fault resilient asynchronous links using multiple-valued codes, we investigate a behaviour of the multiple-valued asynchronous communication link previously proposed in [20] under an open-wire fault. The multiple-valued asynchronous communication between processing cores (modules) is realized by multiplexing two codewords sent from both modules in order to reduce the number of handshaking steps and wires. In the communication link based on the previous encoding, the multiplexed codeword is wrongly detected as a "valid" state under an open-wire fault, causing wrong data communication. In the proposed encoding, under an open-wire fault, the multiplexed codeword reaches an intermediate state as "invalid" state that properly stops the wrong data communication. In addition, it is mathematically proven that the communication links using the proposed m -of- n codes certainly stop when any single open-wire fault is occurred in the link. The proposed code used in the multiple-valued asynchronous global communication link is capable of detecting a single open-wire fault with 3.08-times higher coding efficiency compared with a conventional multiple-valued code used in a triple-modular redundancy (TMR) link that detects an open-wire fault under the same dynamic range of logical values.

The rest of paper is organized as follows. Section 2 describes a model of an open-wire fault in the multiple-valued asynchronous communication links. Section 3 describes constructions of open-wire fault resilient multiple-valued codes and proves the detectability of the open-wire fault using the codes. Section 4 evaluates coding efficiency

Manuscript received December 28, 2012.

Manuscript revised April 28, 2013.

[†]The author is with the Dept. of Electrical and Computer Engineering, McGill University, Montreal, Quebec, H3A 0E9 Canada.

^{††}The author is with Gunma National College of Technology, Maebashi-shi, 371–8530 Japan.

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

a) E-mail: naoya.onizawa@mail.mcgill.ca

b) E-mail: matumoto@elc.gunma-ct.ac.jp

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

DOI: 10.1587/transinf.E96.D.1952

as a figure-of-merit (FOM) in asynchronous communication links based on the proposed encoding comparing with a duplicated and a triple-modular redundancy (TMR) links based on the conventional encoding. Section 5 concludes the paper.

2. Modeling Open-Wire Faults in Multiple-Valued Asynchronous Communication Links

2.1 Review of Conventional Multiple-Valued Asynchronous Communication Link

Figure 1 shows a block diagram of a conventional multiple-valued asynchronous communication link [20]. In the communication link, input data is encoded as a dual-rail unordered codeword of Data or Spacer at both primary and secondary modules shown in Table 1. Spacer is a separator between two consecutive Data. Both modules send codewords A and B , simultaneously. The codewords are summed up on the link and then the summed codeword C is sent back to both modules shown in Table 2.

Figure 2 shows signal states of the codeword $C(x, x')$, where each axis denotes x and x' . The four black dots represent the Data states when both modules send ‘Data’, and the white dot represents the Spacer state when both modules send ‘Spacer’. The five states are defined as valid states. The numbers of the states correspond to ones in Table 2. In the multiple-valued asynchronous communication links, the codeword is correctly sent by detecting the valid states. When ‘Data’ is sent, a summed result $x + x'$ of the codeword $C(x, x')$ is monotonically increased from logical values ‘0’

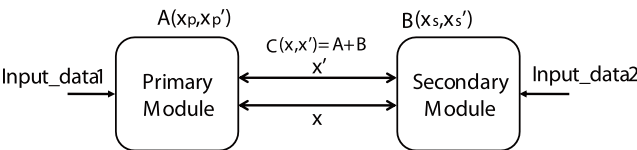


Fig. 1 Block diagram of multiple-valued asynchronous communication links.

Table 1 Dual-rail encoding of Data and spacer in the conventional multiple-valued asynchronous communication link [20].

		Primary $A(x_p, x_{p'})$	Secondary $B(x_s, x_{s'})$
Data	“0”	(0,1)	(0,1)
	“1”	(1,0)	(1,0)
Spacer	Spacer	(0,0)	(0,0)

Table 2 Codeword C in the conventional multiple-valued asynchronous communication link [20].

Primary	State	Data	$A(x_p, x_{p'})$	Transmission lines	$C(x, x')$	$x + x'$	Secondary	Data	State
Data	“0”	(0,1)	(0,2)	1	(0,1)	(0,1)	“0”		
	“0”	(0,1)	(1,1)	2	(1,0)	(1,0)	“1”		
	“1”	(1,0)	(1,1)	3	(0,1)	(0,1)	“0”		
	“1”	(1,0)	(2,0)	4	(1,0)	(1,0)	“1”		
Spacer	-	(0,0)	(0,0)	0	(0,0)	(0,0)	-	Spacer	

to ‘2’. In contrast, when ‘Spacer’ is sent, the summed result $x + x'$ is monotonically decreased from logical values ‘2’ to ‘0’. Hence, the valid states are detected by using two logical thresholds ‘0.5’ for detecting ‘Spacer’ and ‘1.5’ for detecting ‘Data’, which correspond to the two dashed lines shown in Fig. 2. In the hardware implementation, multi-level current signals are used to represent multiple-valued codewords [18]–[20].

2.2 Fault Model of Open Wires in Multiple-Valued Asynchronous Communication Links

To discuss a reliability of the multiple-valued asynchronous communication link under an open-wire fault, a model of the open-wire fault is described. Figure 3 (a) shows a current-flow mechanism in the multiple-valued asynchronous communication links. Under no fault on a wire shown in Fig. 3 (b), the current I_A that corresponds to a logical value in a codeword is generated at the primary module and then mostly divided into two current signals $I_A/2$ at the contact

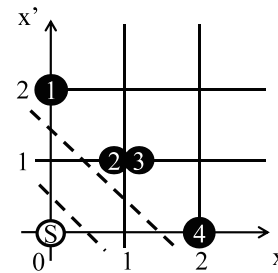


Fig. 2 Signal states of the codeword $C(x, x')$ in the conventional multiple-valued asynchronous communication link at both modules under no open-wire fault.

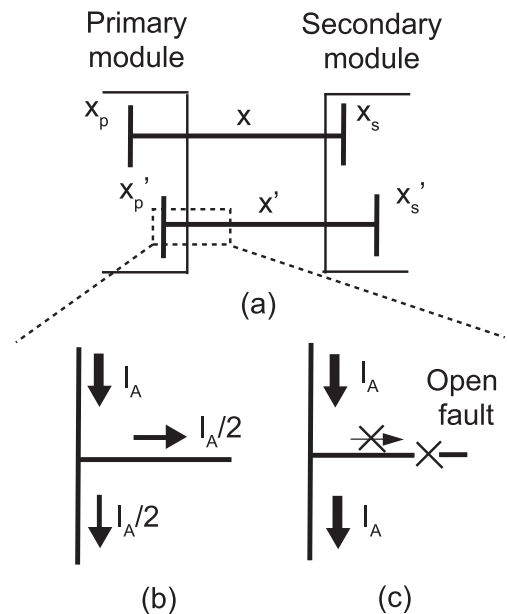


Fig. 3 Current-flow mechanism: (a) link, (b) no fault case, and (c) open-wire fault case.

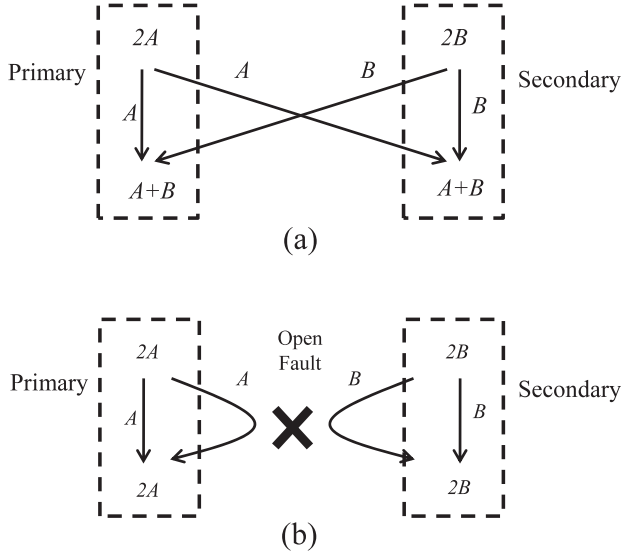


Fig. 4 Model of the multiple-valued asynchronous communication link: (a) no open-wire fault (b) open-wire fault.

point [18]. One is sent to the other module and the other one is received at the module itself. Figure 3 (c) shows a current-flow mechanism under an open-wire fault in the link. The current I_A is no longer shunted and hence the whole current is received in the module itself.

Based on the actual hardware behaviours [16], we describe the open-wire fault model in the multiple-valued asynchronous communication link. Figure 4 (a) shows a no-fault model when the link doesn't have any open-wire faults. In the hardware implementation, as the half of the input current is sent to the other module shown in Fig. 3, the both modules send twice as big as the data that the modules want to send [18]. Hence, the primary module generates "2A" to send "A" to the secondary module that generates "2B" to send "B" to the primary module. These two data are superposed on the wire and the summed data "A+B" is sent back to both modules.

Figure 4 (b) shows the fault model when the link has an open-wire fault. In the primary module, the transmitted data "A" isn't reach to the secondary module, but is sent to the primary module itself because of the open-wire fault. In addition, the transmitted data "B" from the secondary module isn't also reach to the primary module. Hence, the summed data in the primary module is "2A". Similarly, in the secondary module, the summed data is "2B".

Figure 5 shows signal states of the codeword $C(x, x')$ in the conventional multiple-valued asynchronous communication link that has an open-wire fault. Figure 5 (a) shows the signal states when wire x is in open-wire fault, while Fig. 5 (b) shows the signal states when wire x' is in open-wire fault. In Fig. 5 (a), the black dot numbered with 4 is the signal state where the primary module sends a codeword $A(x_p, x_{p'}) = (1, 0)$ and the secondary module sends a codeword $B(x_s, x_{s'}) = (1, 0)$. In this case, a received codeword in the primary module is $(2, 0)$, which means that the primary

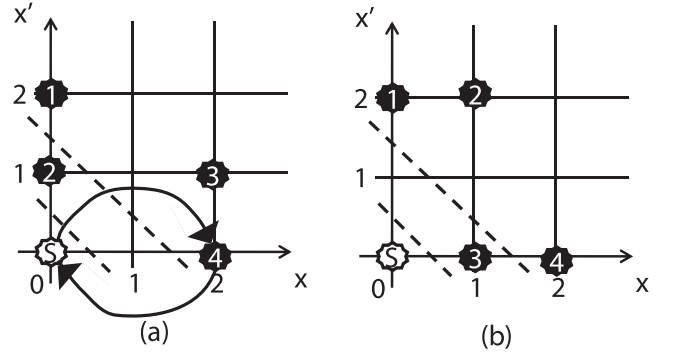


Fig. 5 Signal state of the codeword $C(x, x')$ in the conventional multiple-valued asynchronous communication links at primary module: (a) under an open-wire fault of x and (b) an open-wire fault of x' .

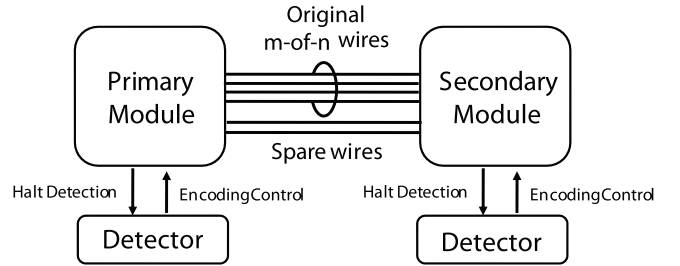


Fig. 6 Block diagram of the proposed open-fault resilient link based on multiple-valued encoding.

module receives the codeword B. Then, the primary module sends the spacer codeword. As the wire x is in open-wire fault, the received codeword in the primary module is $(0, 0)$ that is the signal state of "S" in Fig. 5 (a) even if the secondary module continues sending the Data codeword. The signal state of "S" is an indistinguishable one at the primary module. Hence, it can alternately detect the Data and Spacer eternally until the secondary module changes the codeword sent to $B(x_s, x_{s'}) = (0, 1)$. In other words, the primary module can continue to receive the codeword $B(x_s, x_{s'}) = (1, 0)$ that is not actually sent from the secondary module. It indicates that there is a possibility of the wrong acceptance of codewords by the indistinguishable state due to the open-wire fault in the conventional multiple-valued asynchronous communication link.

2.3 Open-Fault Resilient Asynchronous Communication Link Based on Multiple-Valued Encoding

To prevent the wrong data communication under an open-wire fault, we introduce open-wire fault multiple-valued codes for reliable asynchronous communication links. Figure 6 shows an open-fault resilient asynchronous communication link based on the proposed codes. It can automatically stop sending data when an open-wire fault is occurred. The open-wire fault is detected by timeout of data communication using delay elements at both modules, which then activate "Halt Detection" signals. When both detectors receive the signals, they replace the open-fault wire by spare

ones in order to maintain the functionality of the data communication link. The detailed architecture is described in Sect. 4.3.

In the next section, the proposed multiple-valued codes are described and proven to satisfy the above condition for the open-fault resilient asynchronous links.

3. Open-Fault Resilient Multiple-Valued Codes

The encoding styles of the proposed open-wire fault resilient multiple-valued codes are categorized as one-color and two-color. The one-color encoding uses Data and Spacer as shown in Table 1. As Spacer has to be inserted into two consecutive Data, the number of communication steps is 2 in the multiple-value asynchronous communication links [20]. The two-color encoding uses two different Data: ODD and EVEN [18]. By sending ODD and EVEN alternatively, the two-color encoding eliminates Spacer, which reduces the number of communication steps to 1.

In this section, the proposed open-fault resilient multiple-valued codes are described. Open-fault resilient multiple-valued codes based on one-color or two-color encoding are constructed using m -of- n codes, which assert m wires per single data transmission. It is proven that asynchronous data communication using the proposed codes are stopped when any single open-fault is occurred. Assume that the asynchronous communication links using the proposed codes are designed for point-to-point communication shown in Fig. 6.

3.1 One-Color Open-Fault Resilient Codes Based on m -of- n Encoding

3.1.1 Code Construction

A procedure of constructing the proposed open-fault resilient one-color codes based on m -of- n encoding is shown below:

1. Define original codeword sets C_p and C_s based on m -of- n encoding for the primary and secondary modules, respectively. Let C_p be a codeword set $\{c_{p1}, \dots, c_{pi}, \dots, c_{pl}\}$, where codeword $c_{pi} \in \{0, 1\}^n$. A codeword c_{pi} is represented as $(c_{pi1} \dots c_{pik} \dots c_{pin})$, where $\sum_{k=1}^n c_{pik} = m$. C_s is also represented as well as C_p .
2. Define spacer codewords s_p for the primary module and s_s for the secondary module. Let s_p be $(s_{p1} \dots s_{pk} \dots s_{pn})$, where $s_{pk} \in \{0, 1\}$. s_s is also represented as well as s_p . s_p and s_s satisfy a condition of $\sum_{k=1}^n (s_{pk} \oplus s_{sk}) = n$.
3. Define data codeword sets D_p for the primary module and D_s for the secondary module. Let a data codeword set $D_p = \{d_{p1}, \dots, d_{pi}, \dots, d_{pl}\}$, where each codeword d_{pi} is calculated by the formula $d_{pi} = (d_{pi1} \dots d_{pik} \dots d_{pin}) = (c_{pi1} + 2 * s_{p1} \dots c_{pik} + 2 * s_{pk} \dots c_{pin} + 2 * s_{pn})$. D_s is also represented as well

	Primary module	Secondary module	
Value "0"	(0 0 1 1)	(0 0 1 1)	(a) Original 2-of-4 code
Value "1"	(0 1 0 1)	(0 1 0 1)	
Value "2"	(0 1 1 0)	(0 1 1 0)	
Value "3"	(1 0 0 1)	(1 0 0 1)	
Value "4"	(1 0 1 0)	(1 0 1 0)	
Value "5"	(1 1 0 0)	(1 1 0 0)	
Spacer	(1 0 1 0)	(0 1 0 1)	(b) Spacer codeword
Value "0"	(2 0 3 1)	(0 2 1 3)	(c) Data codeword
Value "1"	(2 1 2 1)	(0 3 0 3)	
Value "2"	(2 1 3 0)	(0 3 1 2)	
Value "3"	(3 0 2 1)	(1 2 0 3)	
Value "4"	(3 0 3 0)	(1 2 1 2)	
Value "5"	(3 1 2 0)	(1 3 0 2)	

Fig. 7 Example of the proposed open-wire fault resilient one-color code based on m -of- n encoding, where m is set to 2 and n is set to 4.

as D_p , where $D_s = \{d_{s1}, \dots, d_{sj}, \dots, d_{sl}\}$ and $d_{sj} = (d_{sj1} \dots d_{sjk} \dots d_{sjn})$.

4. Define codeword sets $\{D_p, \{s_p\}\}$ for the primary module and $\{D_s, \{s_s\}\}$ for the secondary module.

In the multiple-valued asynchronous communication, two codewords sent from the primary and the secondary modules are summed up as shown in Fig. 1 and then the summed result is compared with two thresholds to detect Data or Spacer at both modules. When both modules send Data, the summed result becomes the maximum logic value (MAXV). The MAXV is given by

$$\begin{aligned}
 \text{MAXV} &= \sum_{k=1}^n d_{pik} + \sum_{k=1}^n d_{sjk} \\
 &= \sum_{k=1}^n (c_{pik} + 2 * s_{pk}) + \sum_{k=1}^n (c_{sjk} + 2 * s_{sk}) \\
 &= 2m + 2n,
 \end{aligned} \tag{1}$$

where i and j are arbitrary. Note that $\sum_{k=1}^n (2 * s_{pk} + 2 * s_{sk})$ is equal to $2n$ because $\forall k, s_{pk} + s_{sk} = 1$ by the definition. When both modules send Spacer, the summer result becomes the minimum logical value (MINV). The MINV is given by

$$\text{MINV} = \sum_{k=1}^n (s_{pk} + s_{sk}), = n \tag{2}$$

Hence, two threshold values for Data and Spacer are set to $2m + 2n - 0.5$ and $n + 0.5$, respectively.

Figure 7 shows an example of the proposed open-wire fault resilient one-color code based on m -of- n encoding, where m is set to 2 and n is set to 4. Figure 7(a) shows the original codewords based on 2-of-4 encoding. In the example, both modules use the same assignment of the original codewords to logical values. Note that it is allowed to assign the original codewords to different values at each module. Figure 7(b) shows the spacer codewords at each module. Then, the data codewords are generated using the original and the spacer codewords as shown in Fig. 7(c). In this case, the threshold values for Data and Spacer are set to 11.5 and

4.5, respectively.

3.1.2 Proof of Open-Fault Detectability

Theorem 1. *Each module designed by using the proposed open-fault resilient one-color code stops data communication when any single open-wire fault is occurred in the communication link.*

Proof. It is assumed that the i -th wire in the link is in an open-wire fault. Generality of the proof is not lost by the assumption. It is also assumed that *module_A* indicates a module has an i -th element of the spacer codeword that is 1 and *module_B* is the counterpart module of *module_A*. As the spacer codewords at both modules are complementary, *module_B* has an i -th element of the spacer codeword that is 0.

- When both modules send the data codewords, the summation of the codewords becomes $2m + 2n$ under no open-wire fault. When the i -th wire is in an open-wire fault, the summations of the codewords: $\text{MAXV}(\text{module_A})$ and $\text{MAXV}(\text{module_B})$ monitored by each module are changed as:

$$\text{MAXV}(\text{module_A}) = \text{MAXV} + c_{Aji} - c_{Bki} + 2. \quad (3)$$

$$\text{MAXV}(\text{module_B}) = \text{MAXV} - c_{Aji} + c_{Bki} - 2. \quad (4)$$

As the i -th element of the codeword c_{Aj} and c_{Bk} is 0 or 1, $\text{MAXV}(\text{module_A})$ becomes $2m + 2n + 1$, $2m + 2n + 2$, or $2m + 2n + 3$, while $\text{MAXV}(\text{module_B})$ becomes $2m + 2n - 1$, $2m + 2n - 2$, or $2m + 2n - 3$. It means that the data codeword in the *module_B* CANNOT be detected because $\text{MAXV}(\text{module_B})$ never exceeds the threshold value of Data ($2m + 2n - 0.5$). Hence, the *module_B* stops sending new codewords.

- When both modules send the spacer codewords, the summation of the codewords becomes n under no open-wire fault. When the i -th wire is in an open-wire fault, the summations of the codewords: $\text{MINV}(\text{module_A})$ and $\text{MINV}(\text{module_B})$ monitored by each module are changed as:

$$\text{MINV}(\text{module_A}) = \text{MINV} + 1. \quad (5)$$

$$\text{MINV}(\text{module_B}) = \text{MINV} - 1. \quad (6)$$

$\text{MINV}(\text{module_A})$ becomes $n + 1$ while $\text{MINV}(\text{module_B})$ becomes $n - 1$. It means that the spacer codeword in the *module_A* CANNOT be detected because $\text{MINV}(\text{module_A})$ is always larger than the threshold value of Spacer ($n + 0.5$). Hence, the *module_A* stops sending new codewords.

- From the discussion, during single data commutation, both *module_A* and *module_B* stop sending new codewords when any single open-wire fault is occurred.

□

3.2 Two-Color Open-Fault Resilient Codes Based on m -of- n Encoding

To realize high-throughput asynchronous data-communication links, it is desirable to eliminate the spacer codewords [18], [19]. In this subsection, two-color open-fault resilient codes are defined and its fault detectability is proven.

3.2.1 Code Construction

A procedure of constructing the proposed open-fault resilient two-color codes based on m -of- n encoding is shown below:

1. Define original codeword sets C_p and C_s based on m -of- n encoding for the primary and secondary modules, respectively. Let C_p be a codeword set $\{c_{p1}, \dots, c_{pi}, \dots, c_{pn}\}$, where codeword $c_{pi} \in \{0, 1\}^n$. A codeword c_{pi} is represented as $(c_{pi1}, \dots, c_{pik}, \dots, c_{pin})$, where $\sum_{k=1}^n c_{pik} = m$. C_s is also represented as well as C_p .
2. Define intermediate codewords im_p for the primary module and im_s for the secondary module. Let im_p be $(im_{p1}, \dots, im_{pk}, \dots, im_{pn})$, where $im_{pk} \in \{0, 1\}$. im_s is also represented as well as im_p . im_p and im_s satisfy a condition of $\sum_{k=1}^n (im_{pk} \oplus im_{sk}) = n$.
3. Define ODD codeword sets O_p for the primary module and O_s for the secondary module. Let an ODD codeword set $O_p = \{od_{p1}, \dots, od_{pi}, \dots, od_{pn}\}$, where each codeword od_{pi} is calculated by the formula $od_{pi} = (od_{pi1}, \dots, od_{pik}, \dots, od_{pin}) = (c_{pi1} + 2 * im_{p1}, \dots, c_{pik} + 2 * im_{pk}, \dots, c_{pin} + 2 * im_{pn})$. O_s is also represented as well as O_p , where $O_s = \{od_{s1}, \dots, od_{sj}, \dots, od_{sn}\}$ and $od_{sj} = (od_{sj1}, \dots, od_{sjk}, \dots, od_{sjn})$.
4. Define EVEN codeword sets E_p for the primary module and E_s for the secondary module. Let an EVEN codeword set $E_p = \{ev_{p1}, \dots, ev_{pi}, \dots, ev_{pn}\}$, where each codeword ev_{pi} is calculated by the formula $ev_{pi} = (ev_{pi1}, \dots, ev_{pik}, \dots, ev_{pin}) = (c_{pi1} + 4 * im_{p1}, \dots, c_{pik} + 4 * im_{pk}, \dots, c_{pin} + 4 * im_{pn})$. E_s is also represented as well as E_p , where $E_s = \{ev_{s1}, \dots, ev_{sj}, \dots, ev_{sn}\}$ and $ev_{sj} = (ev_{sj1}, \dots, ev_{sjk}, \dots, ev_{sjn})$.
5. Define codeword sets $\{O_p, E_p\}$ for the primary module and $\{O_s, E_s\}$ for the secondary module.

The multiple-valued asynchronous communication using the two-color codes is also realized by sending codewords from both modules as well as one using the one-color codes. When both modules send EVEN, the summed result becomes the MAXV. The MAXV for the two-color codes is also given by

$$\begin{aligned} \text{MAXV} &= \sum_{k=1}^n ev_{pik} + \sum_{k=1}^n ev_{sjk} \\ &= \sum_{k=1}^n (c_{pik} + 4 * im_{pk}) \end{aligned}$$

	Primary module	Secondary module	
Value "0"	(0 0 0 1)	(0 0 0 1)	(a) Original 1-of-4 code
Value "1"	(0 0 1 0)	(0 0 1 0)	
Value "2"	(0 1 0 0)	(0 1 0 0)	
Value "3"	(1 0 0 0)	(1 0 0 0)	
Intermediate codeword	(1 0 1 0)	(0 1 0 1)	(b) Intermediate codeword
ODD "0"	(2 0 2 1)	(0 2 0 3)	(c) ODD And EVEN codeword (= proposed code)
ODD "1"	(2 0 3 0)	(0 2 1 2)	
ODD "2"	(2 1 2 0)	(0 3 0 2)	
ODD "3"	(3 0 2 0)	(1 2 0 2)	
EVEN "0"	(4 0 4 1)	(0 4 0 5)	
EVEN "1"	(4 0 5 0)	(0 4 1 4)	
EVEN "2"	(4 1 4 0)	(0 5 0 4)	
EVEN "3"	(5 0 4 0)	(1 4 0 4)	

Fig. 8 Example of the proposed open-fault resilient two-color codes, where $m = 1$ and $n = 4$.

$$\begin{aligned}
 & + \sum_{k=1}^n (c_{sjk} + 4 * im_{sk}) \\
 & = 2m + 4n,
 \end{aligned} \tag{7}$$

where i and j are arbitrary. Note that $\sum_{k=1}^n (4 * im_{pk} + 4 * im_{sk})$ is equal to $4n$ because $\forall k, im_{pk} + im_{sk} = 1$. When both modules send ODD, the summed result becomes the MINV. The MINV for the two-color codes is given by

$$\begin{aligned}
 \text{MINV} &= \sum_{k=1}^n od_{pik} + \sum_{k=1}^n od_{sjk} \\
 &= \sum_{k=1}^n (c_{pik} + 2 * im_{pk}) \\
 &+ \sum_{k=1}^n (c_{sjk} + 2 * im_{sk}) \\
 &= 2m + 2n.
 \end{aligned} \tag{8}$$

where i and j are arbitrary. Hence, two threshold values for EVEN and ODD are set to $2m + 4n - 0.5$ and $2m + 2n + 0.5$, respectively.

Figure 8 shows an example of the proposed open-fault resilient two-color codes, where m is set to 1 and n is set to 4. Figure 8 (a) shows the original two-color 1-of-4 codes that are used for both modules. Not that it is allowed to use different original codes at each module. Figure 8 (b) is the intermediate codewords. The ODD and EVEN codewords are calculated using the original 1-of-4 codewords and the intermediate codeword as shown in Fig. 8 (c). In this case, the threshold values of ODD and EVEN are set to 10.5 and 17.5, respectively.

3.2.2 Proof of Open-Fault-Detectability

Theorem 2. *Each module designed by using the proposed open-fault resilient two-color code stops data communication when any single open-wire fault is occurred in the link.*

Proof. It is assumed that the i -th wire in the link is in an open-wire fault. Generality of the proof is not lost by the assumption. It is also assumed that *module_A* indicates a module has an i -th element of the intermediate codeword that is 1 and *module_B* is the counterpart module of *module_A*.

- When both modules send the ODD codewords, the summation of the codewords becomes $2m + 2n$ under no open-wire fault. When the i -th wire is in an open-wire fault, the summations of the codewords: $\text{MINV}(\text{module_A})$ and $\text{MINV}(\text{module_B})$ monitored by each module are changed as:

$$\text{MINV}(\text{module_A}) = \text{MINV} + c_{Aji} - c_{Bki} + 2. \tag{9}$$

$$\text{MINV}(\text{module_B}) = \text{MINV} - c_{Aji} + c_{Bki} - 2. \tag{10}$$

As the i -th element of the codeword c_{Aj} and c_{Bk} is 0 or 1, $\text{MINV}(\text{module_A})$ becomes $2m + 2n + 1$, $2m + 2n + 2$, or $2m + 2n + 3$ while $\text{MINV}(\text{module_B})$ becomes $2m + 2n - 1$, $2m + 2n - 2$, or $2m + 2n - 3$. It means that the ODD codeword in the *module_A* CANNOT be detected because $\text{MINV}(\text{module_A})$ is always larger than the threshold value of ODD state ($2m + 2n + 0.5$). Hence, the *module_A* stops sending new codewords.

- When both modules send the EVEN codewords, the summation of the codewords becomes $2m + 4n$ under no open-wire fault. When the i -th wire is in an open-wire fault, the summations of the codewords: $\text{MAXV}(\text{module_A})$ and $\text{MAXV}(\text{module_B})$ monitored by each module are changed as:

$$\text{MAXV}(\text{module_A}) = \text{MAXV} + c_{Aji} - c_{Bki} + 4. \tag{11}$$

$$\text{MAXV}(\text{module_B}) = \text{MAXV} - c_{Aji} + c_{Bki} - 4. \tag{12}$$

As the i -th element of the codeword c_{Aj} and c_{Bk} is 0 or 1, $\text{MAXV}(\text{module_A})$ becomes $2m + 4n + 3$, $2m + 4n + 4$, or $2m + 4n + 5$ while $\text{MAXV}(\text{module_B})$ becomes $2m + 4n - 3$, $2m + 4n - 4$, or $2m + 4n - 5$. It means that the EVEN codeword in the *module_B* CANNOT be detected because $\text{MAXV}(\text{module_B})$ never exceeds the threshold value of Even state ($2m + 4n - 0.5$). Hence, the *module_B* stops sending new codewords.

- From the discussion, during single data commutation, both *module_A* and *module_B* stop sending new codewords when any single open-wire fault is occurred in the link.

□

4. Evaluation

4.1 Average Logical Values

Table 3 shows comparisons of multiple-valued codes for asynchronous communication links. The *conventional 1* encoding [20] is shown in Tables 1 and 2. The *conventional 1*, the *previous* and *proposed 1* codes are constructed based on one-color encoding and take 2 steps for data communication: Data and Spacer. *Conventional 2* and *proposed 2*

Table 3 Comparisons of multiple-valued codes for n -wire asynchronous communication links.

	# of colors	# of steps	Encoding type	MINV	MAXV	Average logical values per wire (AL)	Open-fault resilient
Conventional 1 [20]	1	2	Dual-rail	0	n	0.5	No
Conventional 2 [18]	2	1	Dual-rail	n	$3n$	2	No
Previous [16]	1	2	Dual-rail	n	$3n$	2	Yes
Proposed 1	1	2	m -of- n	n	$2(n+m)$	$1.5+m/n$	Yes
Proposed 2	2	1	m -of- n	$2(n+m)$	$4n+2m$	$3+2m/n$	Yes

Table 4 Coding efficiency of open-wire fault resilient multiple-valued codes.

# of wires	Method	Data_value	Wire_efficiency	AL	Coding_efficiency	Improvement
2	Previous	2	0.5	2	0.125	-
	Proposed 1 (1-of-2)	2	0.5	2	0.125	0%
	Proposed 2 (1-of-2)	2	0.5	4	0.125	0%
4	Previous	4	0.5	2	0.125	-
	Proposed 1 (1-of-4)	4	0.5	1.75	0.143	14.3%
	Proposed 2 (1-of-4)	4	0.5	3.5	0.143	14.3%
	Proposed 1 (2-of-4)	6	0.646	2	0.162	29.2%
	Proposed 2 (2-of-4)	6	0.646	4	0.162	29.2%
6	Previous	8	0.5	2	0.125	-
	Proposed 1 (2-of-6)	15	0.651	1.83	0.178	42.0%
	Proposed 2 (2-of-6)	15	0.651	3.67	0.178	42.0%
	Proposed 1 (3-of-6)	20	0.720	2	0.180	44.0%
	Proposed 2 (3-of-6)	20	0.720	4	0.180	44.0%
8	Previous	16	0.5	2	0.125	-
	Proposed 1 (2-of-8)	28	0.601	1.75	0.172	37.3%
	Proposed 2 (2-of-8)	28	0.601	3.5	0.172	37.3%
	Proposed 1 (3-of-8)	56	0.726	1.875	0.194	54.9%
	Proposed 2 (3-of-8)	56	0.726	3.75	0.194	54.9%
	Proposed 1 (4-of-8)	70	0.766	2	0.192	53.2%
	Proposed 2 (4-of-8)	70	0.766	4	0.192	53.2%

codes are constructed based on two-color encoding (ODD and EVEN) and take 1 step for data communication by eliminating Spacer.

These codes use different amount of logical values. Large logical values require large currents in the hardware implementation when a unit current is fixed [18]–[21]. In order to include the hardware overhead for the evaluation of the efficiency of the multiple-valued codes, average logical values per wire (AL) is defined. The AL is given by:

$$AL = \frac{MINV + MAXV}{2n}. \quad (13)$$

For example, in the *previous* encoding, the number of wire (n) is 2, the number of asserted wire in single transmission

(m) is 1, the MAXV is 6, and the MINV is 2 in [16]. Hence, the AL is 2.

The proposed codewords are constructed based on m -of- n codes. n is the number of wires and m is the number of “1” in the codeword. The ALs in the proposed encoding are described in Table 3. For example, when n is set to 4 and m is set to 1, the original codewords are (0,0,0,1), (0,0,1,0), (0,1,0,0), and (1,0,0,0). The AL in the *proposed 1* encoding is 1.75 and that in the *proposed 2* encoding is 3.5.

4.2 Coding Efficiency

Table 4 shows comparisons of Coding efficiency of the

Table 5 Comparisons of open-wire fault resilient multiple-valued asynchronous communication links ($n=8$), where a unit current is set to $100\mu\text{A}$.

	Conventional 2	Conventional 2 (Duplicated)	Conventional 2 (TMR)	Proposed 1 (3-of-8)	Proposed 2 (3-of-8)
Fault detection	No	Yes	Yes	Yes	Yes
Fault recovery	No	No	Yes	Yes	Yes
# of wires	8 (n)	16 ($2n$)	24 ($3n$)	8 (n)	8 (n)
# of spare wires	-	-	0	4 ($\lceil n/2 \rceil$)	4 ($\lceil n/2 \rceil$)
MINV	8	16	24	8	22
MAXV	24	48	72	22	38
AL	2	2	2	1.25	2.5
Wire_efficiency	0.5	0.25	0.125	0.484	0.484
Coding_efficiency	0.250	0.125	0.063	0.194	0.194
Maximum current on wire [μA]	400	400	400	400	600

them at different timing, which would receive a wrong data caused by the replacement. We assume that the wrong data transmission can be recovered at higher layer than the physical one, such as the transport layer. It is because the open-wire fault detection is occurred at just one time and hence it is very rare. Other than the case, the communication links based on the proposed codes send data correctly. After the replacement of the wires at both modules, data transmission restarts using spare wires shown in Fig. 9 (d).

4.4 Performance Comparison

Table 5 shows comparisons of open-wire fault resilient multiple-valued asynchronous communication links ($n=8$). The maximum current on a wire is calculated using the maximum logic values on a codeword when a unit current is fixed to $100\mu\text{A}$. A duplication method [24] is applied to an asynchronous communication link based on the *conventional 2* encoding for detection of an open-wire fault. The duplicated link can detect an open-wire fault, but cannot detect which link has the open-wire fault. Hence, it cannot be recovered from the condition of the open-wire fault. A triple-modular redundancy (TMR) method [25] is also applied to the link based on the *conventional 2* encoding for detection of an open-wire fault. The TMR link can mask an open-wire fault without using spacer wires and continues data transmission when an open-wire fault is occurred.

The links based on the proposed 3-of-8 encoding can find one of two groups that has an open-wire fault and hence the number of spare wires is $\lceil n/2 \rceil$ as described in the previous subsection. The MAXV and the MINV of the proposed encoding are calculated using Eqs. (1) and (2) for the one-color code and Eqs. (7) and (8) for the two-color code. The *proposed 1* and the *proposed 2* encoding require smaller MAXV and MINV than those of the TMR-based link based on the *conventional 2* encoding. The link based on the *proposed 1* encoding achieves $3.08\times$ higher coding_efficiency compared to those of the TMR-based link, while maintaining the same maximum currents on a wire. The link based on the *proposed 2* encoding requires higher currents on a wire than the others, but it operates at one step that is the half of *proposed 1*.

5. Conclusion

In this paper, we have proposed open-wire fault resilient multiple-valued codes for reliable asynchronous global communication links. Under an open-wire fault, the conventional links might receive data that is not actually sent by detecting indistinguishable states that are the same as the valid states caused by an open-wire fault. The proposed encoding eliminates the indistinguishable states and the proposed codewords reach to invalid states that is distinguishable to valid states under an open-wire fault. By detecting the invalid states, the link based on the proposed encoding makes it possible to stop the wrong data communication. A detectability of an open-wire fault based on the proposed encoding is proven for one-color and two-color m -of- n codes. Compared with a triple-module redundancy (TMR) link based on the conventional encoding, the proposed link achieves $3.08\times$ higher coding_efficiency. In addition, the proposed codes achieve up to 54.9% higher coding efficiency than the previously reported open-wire fault codes.

Acknowledgements

This research was supported by JST, CREST.

References

- [1] International Technology Roadmap for Semiconductors, 2007 edition, Interconnect Chapter;
http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_Interconnect.pdf.
- [2] D.M. Chapiro, "Globally-asynchronous locally-synchronous systems," Ph.D. dissertation, Stanford University, Stanford, CA, Oct. 1984.
- [3] 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, pp.223–235, 2008.
- [4] N. Onizawa, A. Matsumoto, T. Funazaki, and T. Hanyu, "High-throughput compact delay-insensitive asynchronous NoC router," IEEE Trans. on Computers, 2013 (to appear).
- [5] N. Onizawa, T. Hanyu, and V.C. Gaudet, "High-throughput bit-serial

- LDPC decoder LSI based on multiple-valued asynchronous interleaving," *IEICE Trans. Electron.*, vol.E92-C, no.6, pp.867–874, June 2009.
- [6] N. Onizawa, V.C. Gaudet, and T. Hanyu, "Low-energy asynchronous interleaver for clockless fully-parallel LDPC decoding," *IEEE Trans. Circuits Syst. I Reg. Paper*, vol.58, no.8, pp.1933–1942, Aug. 2011.
 - [7] J. Teifel and R. Manohar, "Highly pipelined asynchronous FPGA's," *Proc. ACM/SIGDA FPGA Symp.*, pp.133–142, Feb. 2004.
 - [8] A. Lines, "Asynchronous interconnect for synchronous SoC design," *IEEE Micro.*, vol.24, no.1, pp.32–41, Jan.-Feb. 2004.
 - [9] M. Laurence, "Introduction to octasic asynchronous processor technology," *Proc. 18th IEEE Int. Symp. on Asynchronous Circuits and Systems (ASYNC)*, pp.113–117, May 2012.
 - [10] A. Chandrasekaran and K. Boahen, "A 1-change-in-4 delay-insensitive interchip link," *Proc. Inter. Symp. on Circuits and Systems (ISCAS)*, pp.3216–3219, May 2010.
 - [11] 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," *Proc. IEEE Int. Symp. on Asynchronous Circuits and Systems (ASYNC)*, pp.116–127, April 2008.
 - [12] 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.
 - [13] S. Ogg, B. Al-Hashimi, and A. Yakovlev, "Asynchronous transient resilient links for NoC," *Proc. CODES*, pp.209–214, Oct. 2008.
 - [14] M.Y. Agyekum and S.M. Nowick, "Error-correcting unordered codes and hardware support for robust asynchronous global communication," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol.31, no.1, pp.75–88, Jan. 2012.
 - [15] N. Onizawa, A. Matsumoto, and T. Hanyu, "Interconnect-fault-resilient delay-insensitive asynchronous communication link based on current-flow monitoring," *Proc. Design Automation Test in Europe (DATE)*, pp.776–781, March 2011.
 - [16] A. Matsumoto, N. Onizawa, and T. Hanyu, "Complementary multiple-valued encoding scheme for interconnect-fault-resilient bidirectional asynchronous links," *Proc. 41st IEEE Int. Symp. on Multiple-Valued Logic (ISMVL)*, pp.236–241, May 2011.
 - [17] T. Lehtonen, P. Liljeberg, and J. Plosila, "Online reconfigurable self-timed links for fault tolerant NoC," *VLSI Design*, vol.2007, Article ID 94676, 2007.
 - [18] T. Takahashi, "Asynchronous duplex communication based on wire-shared current-mode signal multiplexing," *Supplemental Volume of the 2004 IEEE Int. Conf. on Dependable Systems and Networks*, pp.190–192, 2004.
 - [19] N. Onizawa, T. Hanyu, "Highly reliable multiple-valued one-phase signalling for an asynchronous on-chip communication link," *IEICE Trans. Inf. & Syst.*, vol.E93-D, no.8, pp.2089–2099, Aug. 2010.
 - [20] A. Matsumoto, N. Onizawa, and T. Hanyu, "One-color two-phase asynchronous communication links based on multiple-valued simultaneous control," *Proc. 40th IEEE Int. Symp. on Multiple-Valued Logic (ISMVL)*, pp.211–216, 2010.
 - [21] 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.
 - [22] 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. Reliability Physics Symp. (IRPS)*, pp.298–304, 2002.
 - [23] Z. Choi, M. Tsukasa, J.M. Lee, G.-H. Choi, S. Choi, and J.-T. Moon, "Effect of pre-existing void in sub-30 nm Cu Interconnect Reliability," *Proc. IEEE Int. Reliability Physics Symp. (IRPS)*, pp.903–905, 2010.
 - [24] 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*, pp.345–353, 2002.
 - [25] S. Almkharizim and O. Sinanoglu, "A hazard-free majority voter for TMR-based fault tolerance in asynchronous circuits," *Proc. Design and Test Workshop*, pp.93–98, 2007.



Naoya Onizawa received the B.E., M.E. and D.E. degrees from Tohoku University, Japan, in 2004, 2006 and 2009, respectively. He is currently a postdoctoral fellow in Electrical and Computer Engineering, McGill University, Canada. He was a postdoctoral fellow at Tohoku University from 2009 to 2011 and at University of Waterloo in 2011. 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.



Atsushi Matsumoto received the B.E., M.E. and D.E. degrees in the Dept. of Information Engineering, Tokyo Institute of Technology, in 2002, 2004 and 2007, respectively. He is currently an Assistant Professor in the Department of Electronic Media Technology, Gunma National College of Technology. His main interests and activities are in the asynchronous circuits, multiple-valued circuits, and their applications, such as Network-on-Chips. He received Young Researcher Award in 9th Forum on Information

Technology (FIT 2009), Japan and the Best Paper Award in IEEE Computer Society Annual Symposium on VLSI in 2010. Dr. Matsumoto is a member of the IEEE.



Takahiro Hanyu received the B.E., M.E. and D.E. degrees in Electronic Engineering from Tohoku University, 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. He received the Outstanding Paper Award at IEEE International Symposium on Multiple-Valued

Logic in 1986, the Distinctive Contribution Award at IEEE International Symposium on Multiple-Valued Logic in 1988, the Niwa Memorial Award in 1988, 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, APEX Paper Award of Japanese Society of Applied Physics in 2009, the Excellent Paper Award of IEICE, Japan, in 2010, Ichikawa Academic Award, and the Best Paper Award in IEEE Computer Society Annual Symposium on VLSI in 2010. Dr. Hanyu is a senior member of the IEEE.