Multi-bit Transient Fault Control for NoC Links Using 2D Fault Coding Method

Xiaowen Chen[†,‡], Zhonghai Lu[‡], Yuanwu Lei[†], Yaohua Wang[†], Shenggang Chen[†]
[†]College of Computer, National University of Defense Technology, 410073, Changsha, China
[‡]Department of Electronic Systems, KTH - Royal Institute of Technology, 16440 Kista, Stockholm, Sweden
[‡]{xiaowenc,zhonghai}@kth.se

***Abstract*—In deep nanometer scale, Network-on-Chip (NoC) links are more prone to multi-bit transient fault. Conventional ECC techniques brings heavy area, power, and timing overheads when correcting and detecting multiple transient faults. Therefore, a cost-effective ECC technique, named 2D fault coding method, is adopted to overcome the multi-bit transient fault issue of NoC links. Its key innovation is that the wires of a link are treated as its matrix appearance and light-weight Parity Check Coding (PCC) is performed on the matrix's two dimensions (horizontal matrix rows and vertical matrix columns). Horizontal PCCs and vertical PCCs work together to find the faults' position and then correct them by simply inverting them. The procedure of using the 2D fault coding method to protect a NoC link is proposed, its correction and detection capability is analyzed, and its hardware implementation is carried out. Comparative experiments show that the proposal can largely reduce the ECC hardware cost, have much higher fault detection coverage, maintain almost zero silent fault percentages, and have higher fault correction percentages normalized under the same area, demonstrating that it is cost-effective and suitable to the multi-bit transient fault control for NoC links.** [1]

I. Introduction

As the chip technology goes into the deep nanometer era, i.e., its transistor feature size is reduced to be 45nm, 40nm, 28nm, and even smaller, integrated circuits characterized by high frequency and low voltage will be increasingly susceptible to transient faults and permanent faults. The occurrence of transient faults is considered to be roughly 80%[1]. Reliability of links challenges large-scale Network-on-Chip (NoC) design. In deep nanometer scale, transient fault tolerance of NoC links faces new phenomena: **(I) The fault probability of a link wire becomes bigger.** The fault probability (ε) of a link wire can be characterized by the classic fault model[2][3] with a Gaussian distribution as

$$\varepsilon = Q\left(\frac{V_{dd}}{2\sigma_N}\right) = \int_{V_{dd}/2\sigma_N}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy \qquad (1)$$

where V_{dd} is supply voltage and σ_N is noise voltage. Fig. 1 depicts the trends of supply voltage and the ratio of noise voltage to supply voltage, according to the real technology data from TSMC® foundry[4]. As the technology shrinks, the supply voltage decreases for the main purpose of reducing the chip power consumption. However, the proportion of voltage noise in supply voltage becomes bigger. Therefore, according to Equation (1), the increase of the ratio of noise voltage to supply voltage results in the increase of the fault probability (ε) of a link wire. **(II) The fault probability of a link becomes bigger.** Because technology shrinking leads to narrower wire and smaller distance between two adjacent wires and the width of on-chip link is not subject to the limited IO resources of a chip, on-chip link can be usually designed to be 256-bit, 512-bit, and even more wider in order to improve the bandwidth performance. Equation (2) shows that, as the link width (notated as w) becomes bigger, multiple wires in a link may have transient faults concurrently, resulting in the increase of the fault probability (η) of a link[5]. Multiple faults existing on the links have become more important[6][7].

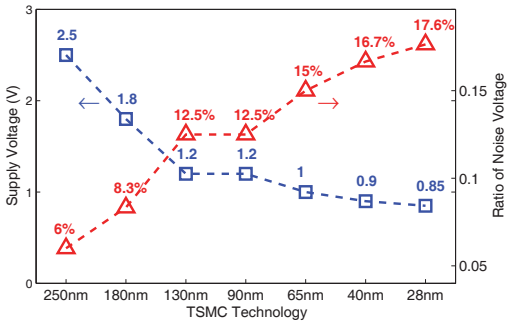

Fig. 1. Trends of supply voltage and the ratio of noise voltage from TSMC®

$$\eta = 1 - (1 - \varepsilon)^w \qquad (2)$$

NoC links are more prone to multi-bit transient fault than ever before in deep nanometer scale, and it is a need to study multi-bit transient fault control for NoC links.

Typically, fault tolerance can be achieved by redundancy. Redundancy is achieved by redundant components to cope with failing ones (*spatial redundancy*), by re-execution of a data transmission with the same component (*temporal redundancy*), and by adding information for fault detection and correction (*information redundancy*)[8]. In the paper, our scope is multi-bit transient fault control for on-chip communication links of large-scale NoCs via information redundancy.

In information redundancy, ECC (Error Correcting Codes) [9][10] is a commonly used and effective protection technique.

[1]The research is partially supported by the National Natural Science Foundation of China (No. 61502508), the Hunan Natural Science Foundation of China (No. 2015JJ3017), and the Doctoral Program of the Ministry of Education in China (No. 20134307120034).

978-1-4673-9030-9/16/$31.00 ©2016 IEEE

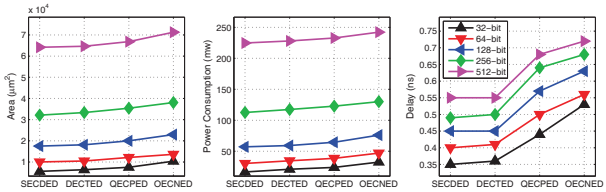

Fig. 2. Hardware cost of traditional ECC techniques such as SECDED, DECTED, QECPED, and OECNED under TSMC® 40nm technology

Hamming algorithm is often used as the base to generate SECDED (Single-Error-Correction Double-Error-Detection) code, while BCH is as the base to form the codes that can detect and correct multiple faults, e.g., DECTED (2-Error-Correction 3-Error-Detection), QECPED (4-Error-Correction 5-Error-Detection), and OECNED (8-Error-Correction 9-Error-Detection). Fig. 2 summarizes the hardware cost of SECDED, DECTED, QECPED, and OECNED under TSMC® 40nm technology. (1) For the same link width, as the number of detected and corrected faults increases, more area, power consumption, and delays are required. (2) For the same detection and correction capability, wider link consumes more hardware cost. Therefore, **scaling up conventional ECC techniques to cover multiple transient faults incur large hardware cost.**

For multi-bit transient fault tolerance, careful ECC design should be carried out by comprehensively considering its fault detection and correction capability and hardware cost, so as to obtain better cost-effectiveness. Therefore, we are motivated to adopt a 2D fault coding method to support multi-bit transient fault control for NoC links. The main contributions of the paper are summarized below:

1) The 2D fault coding method is adopted to organize the wires of a link as its matrix appearance, perform light-weight parity check coding on two dimensions (horizontal matrix rows and vertical matrix columns), and combine the horizontal and vertical fault coding information to detect and correct multiple transient faults.
2) The procedure of using the 2D fault coding method to protect a NoC link is proposed, and its correction and detection capability on NoC links is analyzed in detail. The proposal is implemented by hardware.
3) The cost-effectiveness of the proposal is proved by comparative experiments. Compared with the conventional ECC techniques, our method reduces the hardware cost largely, has higher fault detection coverage, maintains almost zero silent fault percentage, and has higher fault correction coverage normalized under the same area.

II. Related Work

Information redundancy at the data link layer widely uses coding schemes such as Hamming code, BCH code and parity bits to catch faults in NoC links. Hamming code based SEC or SECDED is the most popular, since the occurrence of 1 fault has the highest probability. As the transistor feature size shrinks, the probability of multi-bit fault grows up, some researches adopt BCH code as the base to detect and correct

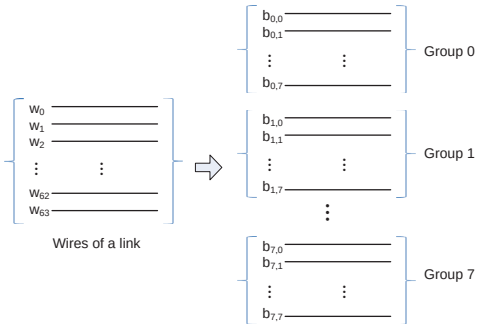

Fig. 3. An example of organizing the link wires into several groups that constitute the link's matrix appearance

two or more multi-bit faults[11]. However, BCH code has large extra hardware overhead. To reduce the hardware cost, some researchers combine a set of relatively simple SEC or SECDED codes with bit interleaving technique to correct adjacent multi-bit faults[5] [10][12]. For instance, in [5], the data is split into blocks to be interleaved, while a SEC code is applied for each block. Then an adaptive fault control method is used to select the ECC scheme dynamically. In [12], Lehtonen analyzes forward fault correction methods for nanoscale NoC. Another thought is developed by Dutta and Touba[13], who use an unequal coding scheme to protect different parts of the packet. It has similar costs as SEC codes while providing better fault detecting and correcting capability, thus the cost is relatively reduced. The ECC codes used by all these literatures are originated from Hamming code or BCH code. Due the algorithm structure itself, Hamming code with bit interleaving or BCH code can theoretically scale to detect and correct more faults, but the hardware cost will grow rapidly, so the number of detected and corrected faults by current literatures is small (< 4). We adopt a new technique (called 2D fault coding method) rather than Hamming code or BCH code to achieve fast multi-bit fault detection/correction while still maintaining high fault coverage with low cost.

The 2D fault coding method organizes the wires of a link as a matrix appearance, and performs horizontal ECC along matrix rows and vertical ECC along matrix columns. Regarding ECC on two dimensions, prior work mainly concerns the protection of memory arrays. In [14], Mohr applies product codes to memory arrays and uses horizontal byte-parity codes to enable low-latency fault detection. This scheme only addresses detecting and correcting single-bit faults within a memory array. In [15][16], Argyrides proposed a matrix code to protect SRAM-based memories against Multiple Bit Upsets (MBU). The proposed method integrates hamming code and parity code together to assure the memory reliability with low area and performance overhead. His work is only at the algorithm level without concrete hardware implementation. Similar with Argyrides's work, in [17], Kim proposed an fault coding scheme in embedded L1/L2 memories and enabled vertical fault coding across words in combination with conventional per-word horizontal fault coding, in order to achieve multi-bit fault protection. The vertical fault coding process is not

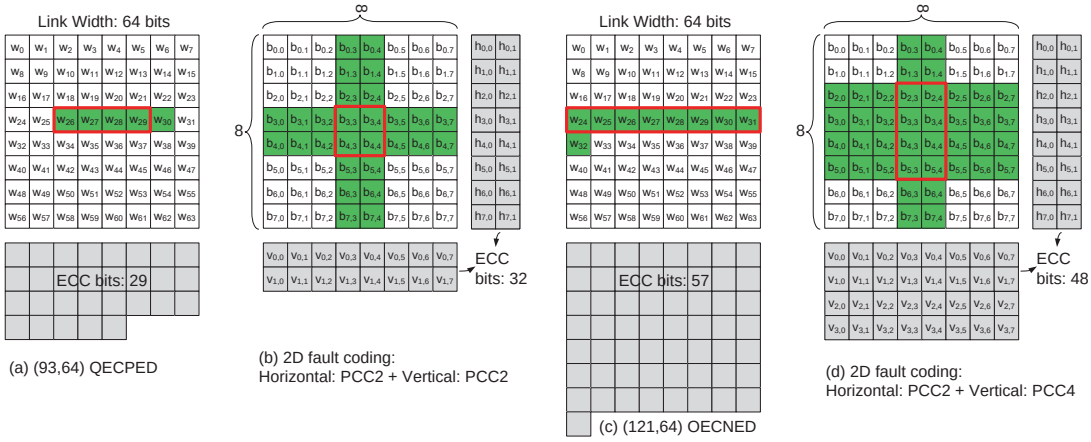

Fig. 4. Comparison of fault coverage and extra ECC bits overhead between QECPED, OECNED and the 2D fault coding method

fast due to the memory's structural characteristic that multiple words can only be loaded out sequentially. And, their jobs focused on radiation-induced faults of memory cells and did specific optimization according to memory cell structure.

III. Two Dimensional Fault Coding Method for Multi-bit Transient Link Fault Control

A. Idea

The core idea of 2D fault coding method is to treat the wires of a link as a matrix appearance, perform light-weight parity coding on each matrix row and column, and combine both of the horizontal and vertical fault coding information to detect and correct one or more transient faults. Since light-weight parity codes are adopted, hardware cost can be reduced.

Fig. 3 shows an example of organizing the wires of a link as its matrix appearance logically. Assuming that a link contains 64 wires labeled as w_0, w_1, ..., w_{63}. The wires are grouped into 8 groups. Group i (i = 0,...,7) contains the wires labeled as $w_{i\times 8+j}$ (j = 0,...,7), which are renamed as $b_{i,j}$. All the wires in group i become the elements on the i-th row of its matrix, and all of the j-th wires in all groups become the elements on the j-th column of the matrix shown in Fig. 4.

To illustrate the 2D fault coding idea, Fig. 4 compares the fault coverage and extra ECC bits overhead between two conventional ECC techniques (QECPED and OECNED) and the 2D fault coding method when they are used to protect a 64-bit link. In the figure, each box represents a bit of the link. The white ones are for data bits, while the grey ones are for ECC bits. The number of green boxes represents the maximum number of the fault that can be detected, while the number of green boxes enclosed by the red line represents the maximum number of the faults that can be corrected.

Fig. 4 (a) shows the fault coverage and the number of extra ECC bits of QECPED that can detect at most five faults and correct at most four faults. To protect a 64-bit link, 29 ECC bits are needed so that the ECC overhead is 29/64 = 45.3%. As shown in Fig. 4 (b), the 2D fault coding method adopts horizontal PCC2 for each row and vertical PCC2 for each column. PCCn represents the n-way interleaved Parity Check Coding technique, which contains n parity bits. The i-th ($i = 0, ..., n-1$) parity bit = data bit[i] $\oplus$ data bit[$i+n$] $\oplus$ data bit[$i+2\times n$] $\oplus$ $\cdots$ $\oplus$ represents an exclusive-OR operation. For instance, in Fig. 4 (b), the parity bit ($h_{0,1}$) is equal to the result of performing exclusive-OR operation on the odd bits ($b_{0,1}$,$b_{0,3}$,$b_{0,5}$,and $b_{0,7}$) at row 0. PCCn allows to detect all continuous n-bit faults, which can increase the fault coverage along rows and columns. The combination of PCCn in the horizontal and vertical directions identifies the location of erroneous bits. Once the erroneous bits are identified, they can be corrected by simply inverting them, so the hardware cost of their correction is trivial. For instance, in Fig. 4 (b), horizontal PCC2 and vertical PCC2 are adopted so that at most 28 faults (the green boxes) can be detected and at most four faults (the green boxes enclosed by the red line) can be corrected. From Fig. 4 (a) and (b), we can see that the 2D fault coding method with horizontal PCC2 and vertical PCC2 has higher fault coverage than QECPED. Although it requires 32 ECC bits that is a little more than 29 ECC bits of QECPED, PCC2 is very simple so that its total ECC hardware cost including ECC bits and ECC encoding and decoding logics are much smaller than QECPED.

Fig. 4 (c) and (d) show the fault coverage and extra ECC bits overhead of OECNED and the 2D fault coding method with horizontal PCC2 and vertical PCC4, both of which can correct at most eight faults. For detecting and correcting more faults, the ECC overhead becomes larger. For instance, the ECC bits overhead of OECNED reaches 57/64 = 89.1%. The 2D fault coding method has less ECC bits than OECNED. Because BCH algorithm is more complex than PCCn, the 2D fault coding method can save more ECC hardware overhead than OECNED. Meanwhile, the 2D fault coding method has large fault coverage so as to detect more faults.

B. Procedure of using the idea to protect a NoC link

The generical procedure of how to use the 2D fault coding method to protect a NoC link is proposed, which can guide

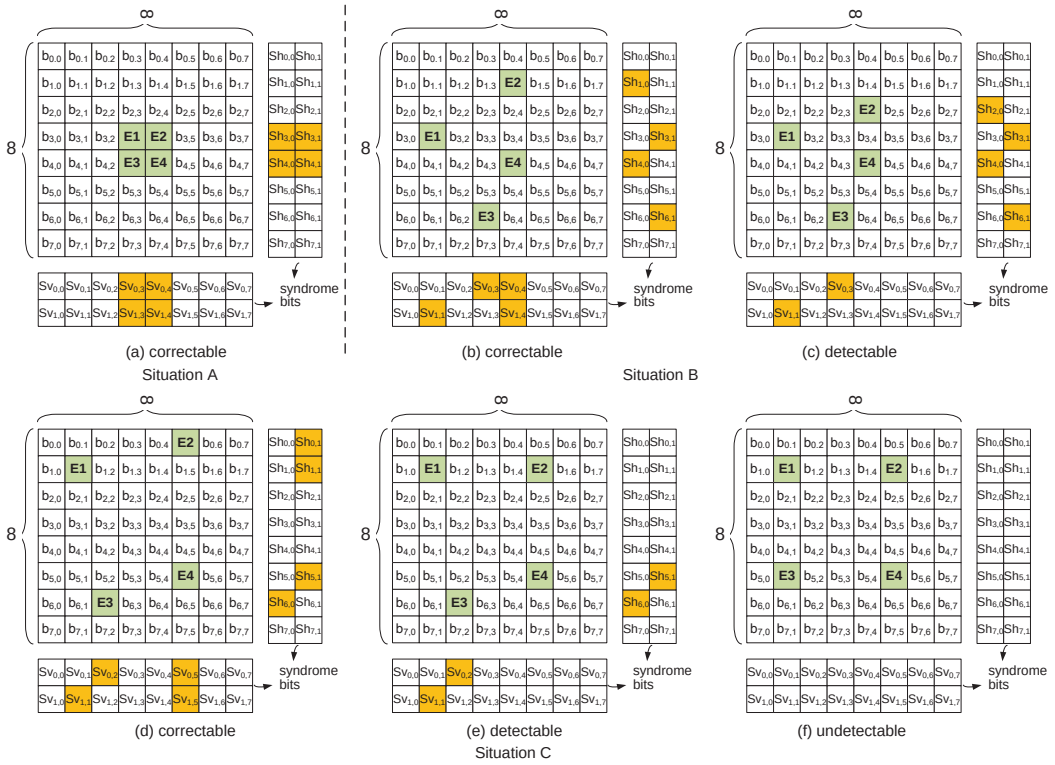

Fig. 5. Analysis of fault detection and correction capability of a 64-bit link with its 8×8 matrix appearance, horizontal PCC2 and vertical PCC2

implementing the 2D fault coding method in practice. The procedure mainly contains three steps as described below.

(I) Organize a link as a its matrix appearance.

Assuming that the link width is N and its matrix appearance contains R rows and C columns, N, R, and C must meet the equation: $N = R \times C$. Then, the wires (w_0, w_1, ..., w_N) in the link are grouped into R groups. Group i (i = 0,...,R) contains the wires labeled as $w_{i\times R+j}$ (j = 0,...,C), which are renamed as $b_{i,j}$ ($0 \leqslant i < R$, $0 \leqslant j < C$). All the wires in group i become the elements on the i-th row of its matrix, and all of the j-th wires in all groups become the elements on the j-th column of the matrix. The matrix appearance is illustrated in Fig. 4.

(II) Choose appropriate PCC techniques for horizontal row data and vertical column data, according to the objective of fault correction and detection.

If PCCm is chosen for each row data and PCCn for each column data, at most $m \times n$ faults can be corrected, and the fault coverage can span over m continuous columns and n continuous rows. $h_{i,j}$ and $v_{i,j}$ are notated as the parity bits for row data and column data respectively (see Fig. 4). They are calculated by Equation (3) and (4).

$$h_{i,j} = b_{i,j} \oplus b_{i,j+m} \oplus \cdots \oplus b_{i,j+\lfloor \frac{C-1}{m} \rfloor \times m} \quad \begin{pmatrix} 0 \leqslant i < R \\ 0 \leqslant j < m \end{pmatrix} \quad (3)$$

$$v_{i,j} = b_{i,j} \oplus b_{i,j+n} \oplus \cdots \oplus b_{i,j+\lfloor \frac{R-1}{n} \rfloor \times n} \quad \begin{pmatrix} 0 \leqslant i < n \\ 0 \leqslant j < C \end{pmatrix} \quad (4)$$

In total, M parity bits are required. M is computed by Equation (5).

$$M = m \times R + n \times C \quad (5)$$

(III) Correct erroneous bits or restart a transmission, according to the comparison of the PCC results between the original data and the received data.

In order to detect the faults, the parity bits ($h_{i,j}$ and $v_{i,j}$) of the original data ($b_{i,j}$) is computed, and the parity bits ($h'_{i,j}$ and $v'_{i,j}$) of the received data ($b'_{i,j}$) are also calculated. Then, the horizontal syndrome bits ($Sh_{i,j}$) and the vertical syndrome bits ($Sv_{i,j}$) are obtained by Equation (6) and (7) respectively.

$$Sh_{i,j} = h_{i,j} \oplus h'_{i,j} \quad (0 \leqslant i < R, 0 \leqslant j < m) \quad (6)$$

$$Sv_{i,j} = v_{i,j} \oplus v'_{i,j} \quad (0 \leqslant i < n, 0 \leqslant j < C) \quad (7)$$

A bit ($b'_{i,j}$) of the received data has its corresponding pair of ($Sh_{i,j\%m}$, $Sv_{i\%n,j}$). If both of $Sh_{i,j\%m}$ and $Sv_{i\%n,j}$ are equal to '1', the bit ($b'_{i,j}$) is suspected as an erroneous one, notated as $b^?_{i,j}$.

Condition A: If the maximum horizontal distance between any two of $b^?_{i,j}$ is less than m and the maximum vertical distance between any two of $b^?_{i,j}$ is less than n, all $b^?_{i,j}$ are confirmed to be erroneous bits. They can be corrected by simply inverting it, as described by Equation (8). $\wedge$ represents an AND operation.

$$corrected\_b_{i,j} = (Sh_{i,j\%m} \wedge Sv_{i\%n,j}) \oplus b^?_{i,j} \quad \begin{pmatrix} 0 \leqslant i < R \\ 0 \leqslant j < C \end{pmatrix} \quad (8)$$

Condition B: If Condition A is not achieved, the suspected bits can not be confirmed. And, if any one of syndrome bits becomes 1, the link fault is detected and the transmission is restarted. Condition B can be formulated by Equation (9). $\vee$ represents an OR operation. That is to say, if there is at least a pair of ($Sh_{p_1,p}$,$Sh_{p_2,p}$) with the same p and the distance ($p_2 - p_1$) $\geqslant m$, or if there is at least a pair of (Sv_{q,q_1},Sv_{q,q_2}) with the same q and the distance ($q_2 - q_1$) $\geqslant n$, the retransmission signal (notated as ξ) is asserted to be active-high.

$$\begin{aligned} &\xi = (\bigvee (Sh_{p_1,p} \wedge Sh_{p_2,p})) \vee (\bigvee (Sh_{q,q_1} \wedge Sh_{q,q_2})) \\ &s.t. \\ &\quad 0 \leqslant p_1, p_2 < R, 0 \leqslant t < m, p_2 \geqslant p_1 + m \\ &\quad 0 \leqslant q_1, q_2 < C, 0 \leqslant q < n, q_2 \geqslant q_1 + n \end{aligned} \tag{9}$$

C. Analysis of fault detection and correction capability

Fig. 4 intrinsically shows that, when protecting the NoC link, the 2D fault coding method has high fault coverage and can correct at most $m \times n$ transient faults. However, it cannot correct any $m \times n$ transient faults. This subsection analyzes the fault detection and correction capability of the 2D fault coding method under three situations categorized according to the locations of erroneous bits.

Situation A: The maximum horizontal distance between any two of erroneous bits is less than m and the maximum vertical distance between any two of erroneous bits is less than n.

In this situation, the combination of PCCm and PCCn can exactly obtain the positions of all erroneous bits, so that they can be corrected by simply inverting them. For instance, as shown in Fig. 5 (a), a 64-bit link appears as a 8×8 matrix, and PCC2 is performed on all rows and columns, i.e., m=n=2. If the distance between any two of the erroneous bits (E1,E2,E3,E4) is less than 2, all of their horizontal syndrome bits and vertical syndrome bits become 1 (The orange boxes represent that the syndrome bits are 1 in Fig. 5), so the exact positions of these erroneous bits can be obtained.

Situation B: The horizontal distance of at least one pair of erroneous bits is less than m, or the vertical distance of at least one pair of erroneous bits is less than n.

In this situation, the combination of PCCm and PCCn cannot exactly obtain the positions of all erroneous bits so as to correct them, but some of the erroneous bits can be detected definitely. As illustrated in Fig. 5 (b), if all of their horizontal syndrome bits and vertical syndrome bits become 1 (see the orange boxes), the erroneous bits can be positioned. However, if not all their syndrome bits are 1, some erroneous bits can not be corrected. For instance, in Fig. 5 (c), because PCC2 is adopted, a syndrome bit becomes 1 when its related odd bits are erroneous. Because the erroneous bits E2 and E4 contribute to the same vertical syndrome bit, the vertical syndrome bit of E2 and E4 is still zero. Therefore, E2 and E4 cannot be corrected, but they can be detected because E2's horizontal syndrome bit and E4's horizontal syndrome bit become 1. In Fig. 5 (b) and (c), because the distance of at least one pair of erroneous bits is less than 2, their syndrome bits become 1 definitely. Although all faults can not be corrected, we can judge that this transmission has faults and restart it.

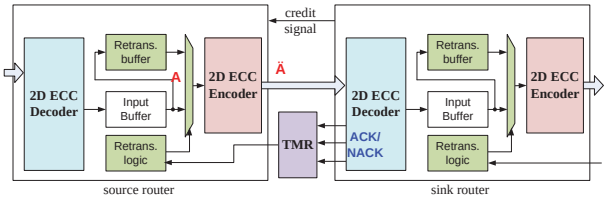

Fig. 6. Implementing the 2D fault coding method in fault-tolerant NoC

Situation C: The horizontal distance of any two of erroneous bits is more than m and the vertical distance between any two of erroneous bits is more than n.

In this situation, if all syndrome bits of all erroneous bits are 1, they can be corrected, as shown in Fig. 5 (d). If some syndrome bits are 1, they can be detected, as shown in Fig. 5 (e). However, the combination of PCCm and PCCn may not obtain the positions of any erroneous bits. For instance, in Fig. 5 (f), E1 and E2 contribute to the same horizontal syndrome bit, E3 and E4 contribute to the same horizontal syndrome bit, E1 and E3 contribute to the same vertical syndrome bit, and E2 and E4 contribute to the same vertical syndrome bit, so no syndrome bits become 1. The erroneous bits are silent in Fig. 5 (f), meaning that they cannot be detected and will cause the system fault.

Based on the analysis under the above three situations, in the procedure introduced in Subsection III-B, fault correction operation is performed under Condition A, and retransmission operation is carried out under Condition B.

IV. Implementing 2D Fault Coding Method in Fault-tolerant NoC

A. Overview

Fig. 6 illustrates how to implement the 2D fault coding method in NoC routers to protect the data on links. To highlight the implementation, only its related modules such as 2D ECC decoder, retransmission buffer, retransmission logic, and 2D ECC encoder are drawn, while the port count and the basic modules such as routing computation, VC allocation, switch allocation, etc, are omitted in the figure. Before a packet (see packet **A** in the source (left) router in Fig. 6) is sent out, it is stored in the retransmission buffer and encoded by the 2D ECC encoder. Then the encoded packet (see packet **Ä** in the sink (right) router in Fig. 6) is transferred on the link to the sink router. After the sink router receives the encoded packet **Ä**, the 2D ECC decoder is responsible for detecting and correcting one or more faults. If there are no faults or the faults are corrected, an **ACK** signal is sent to the source router and notifies the retransmission logic to remove the packet **A** from the retransmission buffer. If faults are detected but uncorrected, an **NACK** signal is returned to inform the source router of retransmitting the packet **A**. The **ACK/NACK** signal for notification is protected by Triple Modular Redundancy (TMR) to ensure its correctness.

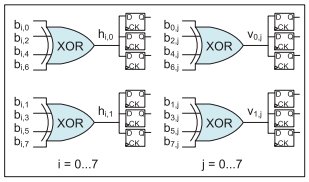

Fig. 7. 2D ECC Encoder

B. 2D ECC Encoder

The 2D ECC encoder is responsible for (I) calculating the horizontal parity bits ($h_{i,j}$) for each row data according to Equation (3) and the vertical parity bits ($v_{i,j}$) for each column data according to Equation (4), and (II) protecting the parity bits ($h_{i,j}$, $v_{i,j}$) by TMR technique. Fig. 7 shows the circuit detail of the 2D ECC encoder that uses PCC2 to protect a 64-bit (8× 8) link. The circuit is simple, because it is only comprised of 32 4-input exclusive-OR gates for parity calculation and 96 flip-flops for TMR protection.

C. 2D ECC Decoder

The 2D ECC decoder is a little more complex than the 2D ECC encoder. It contains five functions: (I) The parity bits ($h'_{i,j}$, $v'_{i,j}$) of the received data ($b'_{i,j}$) from the link are calculated according to Equation (3) and (4). In the example shown in Fig. 8, it contains 32 4-input exclusive-OR gates. (II) The majority voters are used to get the received parity bits ($h_{i,j}$, $v_{i,j}$) from their three copies. (III) Based on (I) and (II), the syndrome bits ($Sh_{i,j}$, $Sv_{i,j}$) are calculated according to Equation (6) and (7), which uses 32 2-input exclusive-OR gates in Fig. 8. (IV) Based on (III), the corrected bits ($corrected\_b_{i,j}$) are calculated according to Equation (8), which uses 32 2-input AND gates and 32 2-input exclusive-OR gates in Fig. 8. (V) According to Equation (9), the retransmission signal (ξ) is generated. If the retransmission signal is active-high, an **NACK** signal is sent to the source router, and $corrected\_b_{i,j}$ will not be stored into the input buffers. Otherwise, $corrected\_b_{i,j}$ is stored into the input buffers, and an **ACK** signal is sent to the source router.

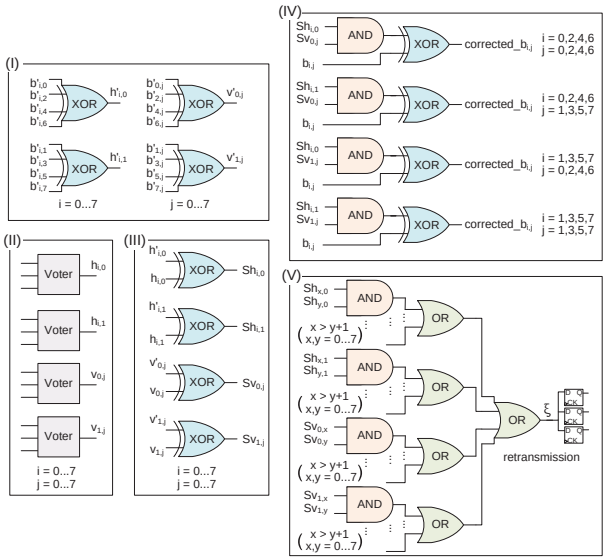

Fig. 8. 2D ECC Decoder

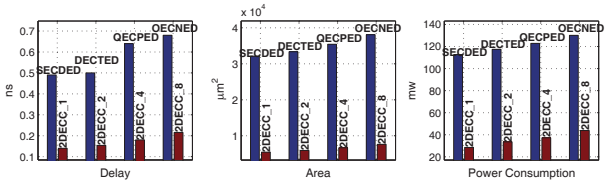

Fig. 9. Comparison of the area, power, and timing overheads between the 2D fault coding method, SECDED, DECTED, QECPED, and OECNED

D. Retransmission buffer and logic

The retransmission logic is responsible for starting the retransmission process and configuring the multiplexer at the output port. When an encoded packet **Ä** is sent out, its original packet **A** is stored into the retransmission buffer. If an **NACK** signal is returned, the retransmission logic configures the multiplexer so that the original packet **A** can be gotten from the retransmission buffer and then sent to the sink router again after going through the 2D ECC encoder. If an **ACK** signal is returned, the retransmission logic removes the original packet **A** from the retransmission buffer, and configures the multiplexer so that a new packet can be transmitted.

V. Experiments and Results

A. Experimental Setup

The focus of the experiments is to compare the effects of using different ECC techniques to protect a link, so the experimental platform is built up to contain a source router, a sink router, and a link connecting the two routers, as shown in Fig. 6. For comparison, the ECC encoder and decoder are implemented by the 2D fault coding method and the conventional SECDED, DECTED, QECPED, and OECNED techniques respectively. In the experiments, The 2D fault coding method with horizontal PCC1 and vertical PCC1, which corrects at most 1 fault, is notated as 2DECC_1. The 2D fault coding method with horizontal PCC1 and vertical PCC2, which corrects at most 2 faults, is notated as 2DECC_2. The 2D fault coding method with horizontal PCC2 and vertical PCC2, which corrects at most 4 faults, is notated as 2DECC_4. The 2D fault coding method with horizontal PCC2 and vertical PCC4, which corrects at most 8 faults, is notated as 2DECC_8. The ECC encoder and decoder are synthesized by Synopsys® Design Compiler under TSMC® 40nm technology. In the experiments, the classic fault model (see Equation (1)) is used to generate an erroneous bit for a link wire randomly. The link width increases from 32, 64, 128, 256 to 512 bits. The supply voltage is set as 0.9V, which is the nominal voltage at TSMC® 40nm technology. The voltage noise varies from 0.10V to 0.20V. 10,000 packet transmissions are carried out.

B. Hardware Cost Analysis

Fig. 9 compares the hardware cost in terms of area, power consumption, and delay between the 2D fault coding method and the conventional SECDED, DECTED, QECPED, and

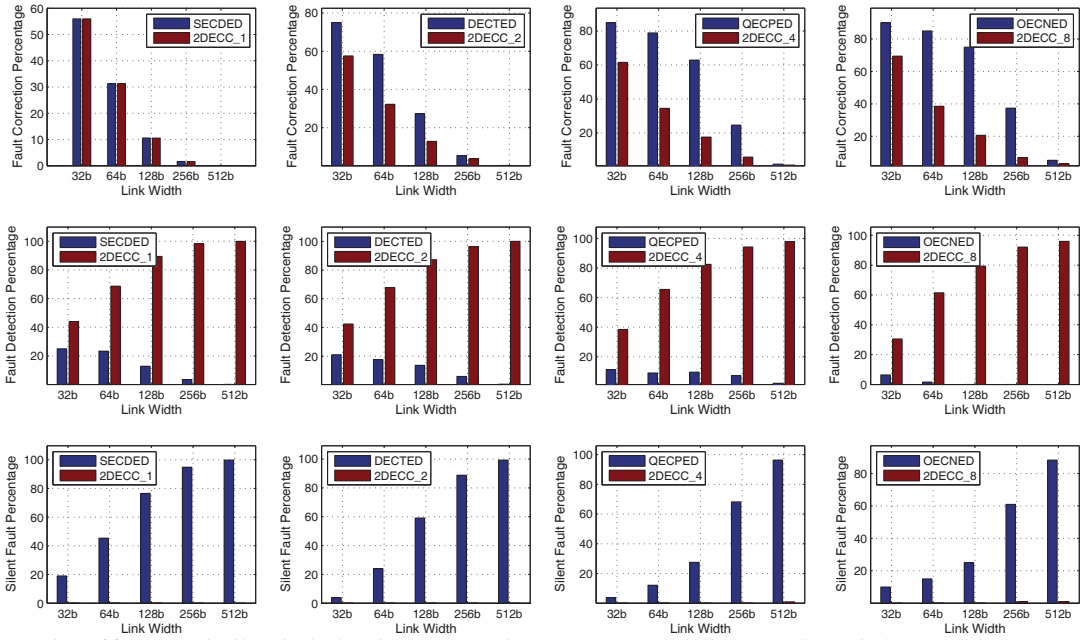

Fig. 10. Comparison of fault correction/detection/undetection percentages between our proposal, SECDED, DECTED, QECPED, and OECNED under different link widths when noise voltage is 0.2V

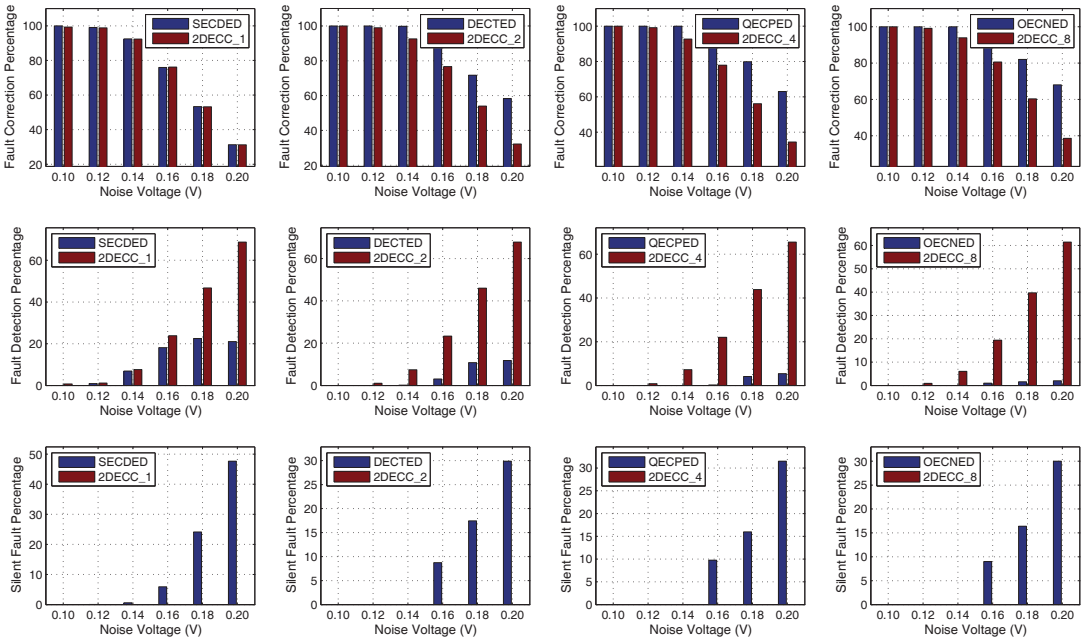

Fig. 11. Comparison of fault correction/detection/undetection percentages between our proposal, SECDED, DECTED, QECPED, and OECNED under different voltage noises when link width is 64-bit

OECNED techniques when the link width is 256-bit. We can obtain that: the 2D fault coding method has obviously smaller area, power consumption, and delay than the conventional ECC techniques. As the correction capability (i.e. the number of corrected faults) increases, the hardware cost of the conventional ECC techniques increases quickly, but the hardware cost of the 2D fault coding method increases slowly.

C. Detection and Correction Coverage

Fig. 10 and 11 compares the detection and correction capability of the 2D fault coding method and the conventional ECC techniques under different link widths and voltage noises, when they are used to correct at most 1, 2, 4, or 8 faults. From the two figures, we can see that: **(1)** The conventional ECC techniques have higher fault correction percentages due

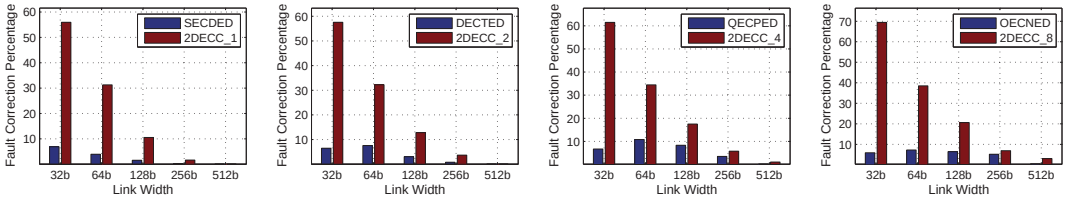

Fig. 12. Comparison of fault correction percentages normalized under the same area between our proposal, SECDED, DECTED, QECPED, and OECNED

to its strong strength that, for instance, QECPED can correct any 4 faults. The strong fault correction capability leads to its high complexity and hence large hardware cost. As the link width increases (see the 1st row in Fig. 10) and the voltage noise increases (see the 1st row in Fig. 11), the fault correction percentages of the 2D fault coding method and the conventional ECC techniques decreases, because the probability of more than 1, 2, 4, or 8 faults increases. **(2)** The 2D fault coding method has much higher fault detection percentages than the conventional ECC techniques, because, for instance, OECNED can not detect the link fault if the number of wire faults are beyond 9, but the 2D fault coding method can detect the link fault as long as one of its syndrome bits is 1. As the link width increases (see the 2nd row in Fig. 10) and the voltage noise increases (see the 2nd row in Fig. 11), the 2D fault coding method has more obvious fault detection performance. **(3)** Due to the same reason above, As the link width increases (see the 3rd row in Fig. 10) and the voltage noise increases (see the 3rd row in Fig. 11), the silent fault (i.e. undetected fault) percentages of the conventional ECC techniques increases. However, the silent fault percentages of the 2D fault coding method is almost zero, because the probability of the distribution of multiple faults that leads to all syndrome bits to be 0 is very small and can be negligible.

Although the 2D fault coding method does not have higher fault correction percentage but has higher fault detection coverage and smaller hardware cost than the conventional ECC techniques. Further, Fig. 12 shows the fault correction percentages normalized under the same area. The fault correction percentage (δ) of the conventional ECC techniques is re-calculated by Equation (10).

$$\delta = \delta \times \frac{AREA_{2D\ fault\ coding\ method}}{AREA_{conventional\ ECC}} \tag{10}$$

Under the same area, the 2D fault coding method has higher fault correction percentages, demonstrating that it is more cost-effective than the conventional ECC techniques when solving the multi-bit fault problem.

VI. Conclusion

Observing that the probability of the occurrence of multiple transient faults in deep nanometer scale rises up and the conventional ECC techniques brings heavy hardware overheads when correcting and detecting multiple transient faults, the paper adopts a cost-effective ECC technique, named 2D fault coding method, to solve the multi-bit transient fault issue. The procedure of using the 2D fault coding method to protect a NoC link is proposed, its correction and detection capability is analyzed, and its hardware implementation is also carried out. Comparative experiments demonstrate that the method is suitable to the multi-bit transient fault control for NoC links.

This paper focuses on the proposal of adopting the 2D fault coding method to safeguard NoC links. In the future, we will turn to further evaluate and optimize its performance itself by considering end-to-end and hop-to-hop fault control schemes.

References

[1] S. C. Navonil Chatterjee and K. Manna, "A spare router based reliable network-on-chip design," in *Proc. of the IEEE Int'l Symp. on Circuits and Systems*, 2014, pp. 1957–1960.

[2] D. Bertozzi, L. Benini, and G. D. Micheli, "Error control schemes for on-chip communication links: the energy-reliability tradeoff," *IEEE Trans. On Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 6, pp. 818–831, 2005.

[3] S. Sridhara and N. R. Shanbhag, "Coding for system-on-chip networks: A unified framework," *IEEE Trans. On Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 6, pp. 665–667, 2005.

[4] "Tsmc technology data," in *http://www.tsmc.com/english/dedicatedFoundry /technology/index.htm*.

[5] Q. Yu and P. Ampadu, "Adaptive error control for noc switch-to-switch links in a variable noise environment," in *Proc. of the IEEE Int'l Symp. on Defect and Fault Tolerance in VLSI Systems*, 2008, pp. 352–360.

[6] L. Xie, K. Mei, and Y. Li, "REPAIR: a reliable partial-redundancy-based router in NoC," in *Proc. of the IEEE Int'l Conf. on Networking, Architecture and Storage*, 2013, pp. 173–177.

[7] P. Poluri and A. Louri, "A soft error tolerant network-on-chip router pipeline for multi-core systems," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 107–110, 2015.

[8] M. Radetzki and C. Feng, "Methods for fault tolerance in networks-on-chip," *ACM Computing Surveys*, vol. 46, no. 1, pp. 1–38, 2013.

[9] P. Kongetira, K. Aingaran, and K. Niagara, "A 32-way multithreaded sparc processor," *IEEE Micro*, vol. 25, no. 2, pp. 21–29, 2005.

[10] H. Ando *et al.*, "Accelerated testing of a 90nm SPARC64V microprocessor for neutron SER," in *Proc. of The Workshop on System Effects on Logic Soft Errors*, 2007, pp. 1–5.

[11] T. K. Moon, "Terror correction coding: Mathematical methods and algorithms," in *New York: Wiley*, 2005, pp. 235–237.

[12] T. Lehtonen, P. Liljeberg, and J. Plosila, "Analysis of forward error correction methods for nanoscale networks-on-chip," in *Proc. of the Int'l Conf. on Nano-Networks*, 2007, pp. 1–5.

[13] A. Dutta and N. Touba, "Reliable network-on-chip using a low cost unequal error protection code," in *Proc. of the IEEE Int'l Symp. on Defect and Fault-Tolerance in VLSI Systems*, 2007, pp. 3–11.

[14] K. C. Mohr and L. T. Clark, "Delay and area efficient first-level cache soft error detection and correction," in *Proc. of the Int'l Conf. on Computer Design*, 2006, pp. 88–92.

[15] C. Argyrides and H. R. Zarandi, "Matrix codes: Multiple bit upsets tolerant method for sram memories," in *Proc. of the IEEE Int'l Symp. on Defect and Fault Tolerance in VLSI Systems*, 2007, pp. 340–348.

[16] C. Argyrides, D. K. Pradhan, and T. Kocak, "Matrix codes for reliable and cost efficient memory chips," *IEEE Trans. On Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 3, pp. 420–428, 2011.

[17] J. Kim, N. Hardavellas, K. Mai, and B. Falsafi, "Multi-bit error tolerant caches using two-dimensional error coding," in *Proc. of the ACM/IEEE Int'l Symp. on Mciroarchitecture (MICRO)*, 2007, pp. 197–209.