Microprocessors and Microsystems 57 (2018) 15–22

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Reliable low power NoC interconnect

M. Vinodhini*, N.S. Murty

Dept of Electronics & Communication Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India

ARTICLE INFO

Keywords:
Error Control Codes
Network on Chip
Hamming code
Error detection
Error correcting
On-Chip Interconnects

ABSTRACT

Information communicated through Network on Chip (NoC) in System on Chip (SoC) is highly prone to different sources of noise, like coupling, radiation and electromagnetic interference. The outcome is multi-bit errors, which can either be random or burst. As the demand for reliable NoC increases, optimal error correcting coding techniques become imperative for SoC and various multi-core and many-core architectures. A novel Multi-bit Error Correcting Coding with Reduced Link Bandwidth (MECCRLB) is proposed to achieve reliable data transmission through NoC. The proposed technique corrects burst error of four bits or random error of eleven bits or combined burst and random errors of total four bits for an input flit size of 32 bits. Analytical model based performance estimation for coding technique is extensively used in NoC. Reliability, link swing voltage and link power consumption are estimated using analytical model for the proposed MECCRLB coding technique. All the results obtained for MECCRLB coding technique are compared with Hamming product code with Type II HARQ. Estimated results show that at a probability of residual error of 10^{-25}, the link swing voltage and the link power are reduced by 30% and 75% respectively. Results obtained from simulation followed by synthesis indicate that there is a reduction of 65%, 44%, 27%, 28% and 49% in bit overhead, NoC router area, NoC router power, codec power and codec area respectively. Furthermore, MECCRLB coding technique achieves higher error correction capability and reduces the need for retransmission. This signifies that the proposed coding technique outperforms Hamming product code with Type II HARQ in reliability, area and power.

1. Introduction

Traditional interconnection of all Processing Elements (PEs) or IP blocks on a single chip by employing on-chip communication infrastructure, like shared buses or multi-layer buses results in issues with scalability and IP reusability. This motivates System on Chip architects to shift to Network on Chip (NoC). NoC provides scalable and high bandwidth communication infrastructure to various multi-core and many-core architectures. The critical issue in this architecture is to achieve low power consumption [1]. In these architectures, contribution of NoC to the total power consumption is significant. Typical NoC comprises of router, interconnection link and Network Interface (NI). Various components integrated in NoC router are FIFO buffer, routing computation block, switch allocator and switch fabric. Among the above, power consumed by the link plays a major role more so in DSM technologies [2].

NoC is highly susceptible to various external and internal noises that cause transient errors in the transmitted bits. The effect of transient error could be minimized by choosing an optimal error correcting coding technique. Effective and efficient error correcting coding techniques are very important in DSM technologies, where the probabilities of occurrences of multiple random and burst errors are high. The purpose of any error correcting coding technique is to locate and correct corrupted bits with the help of redundant bits and this increases the bits to be transmitted through NoC link. This in turn raises the power consumed by the NoC link. Thus, it is necessary to select a suitable error correcting coding technique to improve the reliability and to minimize the NoC power consumption.

Reliable NoC is achievable by applying proper error correcting coding technique at different layers of NoC. Error Correcting Coding (ECC) technique is a commonly used methodology to achieve reliable data transmission at data link layer. At the network layer, either ECC technique or a suitable routing algorithm could be used to get reliable data transmission through the NoC link. ECC at the network layer provides end-to-end error control, but ECC at the data link layer provides switch-to-switch error control [3].

In this paper, we propose a novel Multi-bit Error Correcting Coding technique with Reduced Link Bandwidth (MECCRLB) for reliable low power NoC. Reduced bandwidth provides low power NoC link. The technique provides correction of all combinations of burst and random errors up to four bits or four bits burst error or eleven bits random error for the input flit size of 32 bits. The main benefit of this technique is

* m_vinodhini@blr.amrita.edu
E-mail address: m_vinodhini@blr.amrita.edu (M. Vinodhini).

https://doi.org/10.1016/j.micpro.2017.12.008
Received 7 November 2016; Received in revised form 23 September 2017; Accepted 15 December 2017
Available online 16 December 2017
0141-9331/

that it enables higher error correction with minimal redundant bits, which in turn provides low power interconnect.

The organization of the remaining part of the paper is as follows. Brief discussion on related works in the field of error correcting coding techniques for NoC interconnection links are given in Section 2. Detailed explanation of the proposed MECCRLB coding technique and its implementation are given in Section 3. Analysis of the proposed MECCRLB coding technique on reliability, link swing voltage and power consumption of the link is given in Section 4. Performance evaluation of MECCRLB coding technique is given in Section 5. Conclusion on the work done and the future direction are given in Section 6.

2. Related work

Various research works performed in the area of ECC technique for NoC resulted in numerous codes. Simple Parity Check [4] code can locate and correct only single bit error. Cyclic Redundancy Check (CRC) code [5] supports only error detection but not the error correction. Hamming code [3–7] is used only to correct single bit error or to detect double bit error. Simultaneous error detection and correction are not possible with this code. In Extended Hamming code [3–7], both double bit error detection and single bit error correction are possible with added parity bit. In Multiple Continuous Error Correct Coding (MCECC) technique [8], correction is possible only when continuous bits are corrupted. Correction of burst errors is possible with Hamming code with interleaving [9] and Symbol error correcting code [10]. These codes are not effective for random errors. Hamming product code with Type II HARQ [11] can correct burst and random errors, if the corrupted bits are up to six, but with high bit overhead and retransmission. It takes 72 redundant bits for a data size of 32 bits. Bose Chaudhuri Hocquenghem (BCH) code corrects multi-bit error with increased hardware complexity. Not all the codes addressed above are effective for crosstalk avoidance and multi-bit random and burst error correction.

By joining crosstalk avoidance with error correcting coding technique, reliability of NoC is further improved. In these coding techniques, the encoded input bits are duplicated or triplicated to remove the crosstalk effects present between adjacent wires. In addition to crosstalk avoidance, these coding techniques also provide maximum number of error correction in bits by adopting enhanced error correcting coding technique. In Duplicate Add Parity (DAP) code [12], 65 bits are transmitted in place of 32 data bits for single bit error correction with crosstalk avoidance. In Crosstalk Avoidance and Double Error Correction (CADEC) code [13] and Joint crosstalk avoidance and Triple Error Correction (JTEC) code [14], 32 data bits use 45 redundant bits to correct up to two bit and three-bit errors respectively. In Crosstalk Avoidance Random Burst Error Detection and Correction (CARBEDC) code [15] and Crosstalk Avoidance Enhanced Double Error Correction (CAEDEC) code [16], 97 bits are transmitted in place of 32 data bits for multi-bit error correction up to two and three bit errors respectively. In Joint Crosstalk Avoidance-Five Bit Error Correction-Six Bit Error Detection (JCA-FBEC-SBED) code [17], 85 redundant bits are used for 32 data bits for five bit error correction. All the above referred joint crosstalk avoidance and error correcting techniques provide multi-bit random and burst error correction with increased link bandwidth.

In this work, incorporation of combined Extended Hamming code and Simple parity check code in NoC router improves the reliability of NoC link without incurring higher link bandwidth. Reduced bandwidth provides low power interconnection link. Integration of low complex Extended Hamming code with simple parity check code provides a path to develop a novel Multi-bit Error Correcting Coding technique with Reduced Link Bandwidth (MECCRLB). This technique locates and corrects all combinations of burst and random errors up to four bits or four bits burst error or eleven bits random error for the input flit size of 32 bits.

Table 1
Flit size and its corresponding group and vector size.

Flit size (K)	Number of groups (G)	Number of bits in group1 ($G1$) (K/G)	Number of bits in group2 ($G2$) (K/G)	Number of bits in group3 ($G3$) $K - 2(K/G)$	Number of vectors ($V = G3$)	Vector size (V_b)
32	3	11	11	10	10	3
64	3	22	22	20	20	3
128	3	43	43	42	42	3
256	3	86	86	84	84	3
512	3	171	171	170	170	3
1024	3	342	342	340	340	3

3. Proposed MECCRLB code

In the proposed MECCRLB code, K represents the input flit size in bits, which when partitioned, creates groups and vectors. Irrespective of the flit size, number of groups (G) created is always three in this proposed code. Number of bits in group1 ($G1$) and group2 ($G2$) are the same, which is equal to K/G and rounded off to the next higher number of bits. Number of bits in group3 ($G3$) is equal to $K - 2(K/G)$. Number of vectors (V) created is equal to number of bits in group3 ($G3$). By taking one bit from each group, vectors are created. Irrespective of the number of vectors created, Vector size (V_b) is always three as shown in Table 1. Depending upon the input flit size, the number of bits in each group and number of vectors created vary. For the input flit size of 128 bit, i.e. $K = 128$, the value of G and V will be three and by using the specified relations, $G1$ and $G2$ are calculated as 43 bits, $G3$ is calculated as 42 bits and V is calculated as 42 bits.

3.1. MECCRLB encoder

In MECCRLB encoder, parity bits are computed for input flit size of 32 bits by partitioning the input flit into 3 groups and 10 vectors as shown in Fig. 1. After partitioning, group1 and group2 have 11 bits each, group3 has 10 bits and each vector has 3 bits. Each group is encoded using Extended Hamming code and each vector is encoded using simple parity check code. MECCRLB encoder has three Extended Hamming encoders for three groups and ten simple parity check encoders for ten vectors as shown in Fig. 2. Each encoded group gives 5 Extended Hamming parity bits, which totally gives 15 Extended Hamming parity bits for three groups. Thus, group1 and group2 encoders output 16 bits each (11 data bits and 5 parity bits) and group3 encoder outputs 15 bits (10 data bits and 5 parity bits). Each vector encoder gives single parity bit. Totally, there are 25 parity bits for input flit size of 32 bits, which results into 57 bit encoded flit.

3.2. MECCRLB decoder

Fig. 3 depicts the decoding algorithm. In the decoder, the received encoded data flit is partitioned into three groups and ten vectors using the same procedure followed in the encoder. Each partitioned group and vector will hold data as well as parity bits. $G1$ and $G2$ have 16 bits each and $G3$ has 15 bits. Next, each group is decoded separately using Extended Hamming decoder and the Extended Hamming parity bits are computed. Simultaneously, each vector is decoded using simple parity check decoder and their corresponding parity bits are computed. Received parity bits are compared with the computed parity bits to calculate the syndrome bits.

After computing the parity and syndrome bits, the steps followed to locate the position of error bit are as follows:

(1) If all of the computed syndrome, extended Hamming and Hamming parity bits are zero, the received flit is error free.