IJBPAS, July, 2015, 4(7): 5368-5380

ISSN: 2277–4998

International Journal of Biology, Pharmacy and Allied Sciences (IJBPAS)
'A Bridge Between Laboratory and Reader'

www.ijbpas.com

A NEW SCHEME TO INCREASE THE RELIABILITY OF DATA TRANSMISSION IN NETWORK ON CHIP

EBRAHIM BEHROUZIAN NEJAD *

Department of Computer, Andimeshk Branch, Islamic Azad University,Andimeshk, Iran
Behrouzian.e@gmail.com

HAMID MEHDI

Department of Computer, Andimeshk Branch, Islamic Azad University,Andimeshk, Iran
Hamidmehdi@gmail.Com

BEHROOZ SAFAEIZADEH

Department of Computer, Andimeshk Branch, Islamic Azad University,Andimeshk, Iran
Behrouz_safaie@yahoo.com

***Corresponding Author: Ebrahim Behrouzian Nejad, Behrozian.e@gmail.com**

ABSTRACT

As CMOS technology scales down into the nano-technology domain and the complexity of evolving integrated circuits design increases, VLSI systems become more and more vulnerable to permanent and transient faults. Therefore error detection and correction is one of the major properties of future on-chip micro networks. In this paper we propose using Redundant Multi-Level Residue Number System and a new switch-to-switch retransmission scheme to increase the data transmission reliability in on-chip networks. Residue Number System (RNS) is an integer and non weighted number system that is useful tool for Digital Signal Processing (DSP) since it can support parallel, carry-free, high-speed and low power arithmetic. Redundant Residue Number System is an extension of RNS which also supports error detection and correction. The Multi-Level Residue Number System uses the new Residue Number System for each modulo, so in the relation of decreasing modulo the speed of operation is increased. By the combination of those systems we purpose a new numeric system which supports parallel and high speed computations, restricted carry propagation and reliable communications. This system also supports high error detection and correction capabilities. Our new method can achieve more optimizations in the terms of data security, error detection and correction, high speed data transmission and computation.

KEYWORDS: Networks on Chip, Reliability, Residue Number System, Switch-to-Switch Retransmission, Error Detection and Correction

INTRODUCTION

In the past decades, System on Board (SOB) has been the dominant methodology for designing complex digital systems. As the complexity of applications and their required algorithms have grown so rapidly, SOB has been replaced by System on Chip (SOC) methodology. Next generations of systems-on-chip (SoC) will consist of hundreds of pre-designed IPs[1] assembled together to form large chips with very complex functionality. As technology scales and chip integrity grows, on-chip communication is playing an increasingly dominant role in System-on-Chip (SoC) design[1,2].To deal with the increasingly difficult problem of on-Chip communication, it has been recently proposed to connect the IPs using a Network-on-Chip (NoC)architecture. Each core is connected to a switch by a network interface. Cores communicate with each other by sending packets via a path consisting of a series of switches and inter-switch links [1,2,3]. As devices shrink toward the nanometer scale, on-chip interconnects are becoming a critical bottleneck in meeting performance and power consumption requirements of chip designs.

Network-on-Chip (NoC) [1-3] has been proposed as a solution to provide better modularity, scalability, reliability and higher bandwidth compared to bus-based communication infrastructures. In NOC each core is connected to a switch by a network interface. Cores communicate with each other by sending packets. Fig. 1 shows an abstract view of a NOC. As shown in fig.1, A typical NoC consists of four major components: Cores, Network Interface Units (NIUs), Switches and Physical Links. Each core can be a processing Element (PE), embedded memory, DSP or etc. Other components constitute the communication fabric.

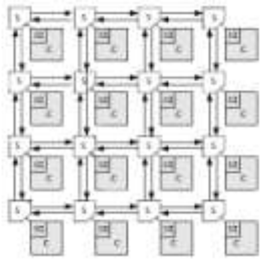


Figura1.(a)

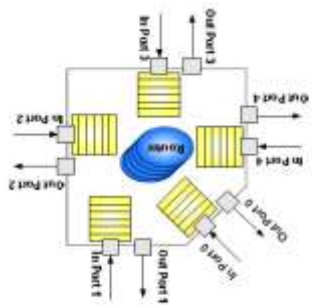


Figure 1.(b

Figure.1 (a) The typical structure of a A 4×4-2-D mesh NoC S: switch; C: core; NI: network interface (b) The typical structure of a switsh

According to ITRS's report [5], the feature size will shrink to 50nm with 4 billion transistors on a single chip, and the frequency can be up to 10GHz while the working voltage is around 1v. The more sophisticated semiconductor technology makes the more powerful and complex NOCs come true. The on-chip network is the backbone used for communication between various computing resources. The task of on-chip network is to provide pre-specified quality of service (QoS)

[1] Intellectual property

for communication, including bandwidth, latency and reliability.

As for the QoS provided by NoC, the reliability of transmission is one of the most important aspects. The fault tolerant mechanism is indispensable in DSM and nanometer technology. The factors which may introduce errors into on-chip network can be categorized into two classes: one is for the crosstalk between long wires and the other for the cosmic rays, electronic-magnetic interference, variation of process etc. Those can increase the probability of transient errors and soft errors significantly which may be hardly avoided or eliminated. There will be more and more fault types affecting on-chip interconnected wires as technology is getting improved. However, it's impossible or very expensive to avoid these faults completely at design stage. Thereby error recovery and fault tolerant mechanisms are needed. These mechanisms can provide high reliability with low test and verification cost. However, traditional fault tolerant algorithms cannot be

used on chip directly due to the constraint of resource, power consumption, latency and chip area. Existing fault tolerant mechanisms fall into two major categories: one is the request retransmission mechanism based on error checking and correcting code, the other is stochastic communication.[4]

REQUEST RETRANSMISSION SCHEMES

Most of literatures consider NoC with 2D mesh topology and Wormhole switching. Wormhole switching this is employed because of its low latency and low buffer requirement. In wormhole switching, a packet is divided into flits for transmission. The header flit contains the routing information, which is used by the switches to establish the routing path. The remaining flits simply follow the path in a pipeline fashion. A flit is passed to the next switch as soon as enough buffer space is available to store it, even though there is not enough space to store the whole packet. The NOC reliability can be ensured at two level: packet level and flit level. In first method, packet level, error detection/correction codes added and attributed to a packet. In this method sending faulty flits can not be stopped. In second method, flit level, error detection/correction codes added and attributed to each flit separately. Therefore, In this method sending faulty flits can not be stopped at each time.

In these method error recovery can be down in three way : end-to-end retransmission, switch-to-switch retransmission and hybrid retransmission. There are three types of retransmission: end-to-end, switch-to-switch and hybrid [4]. *End-to-end* retransmission can use parity or CRC code to check packets. Packet checking is only performed at destination end. Intermediate switches only forward packets and have little knowledge about whether the packet is destroyed. The source end should wait for ACK or NACK packets fed back by the destination and then decide whether to retransmit the packet or not. In *switch-to-switch* mechanism, on the other hand, each intermediate switch checks all the passing packets. Therefore decoders are needed in all switches. *Hybrid* mechanism makes use of end-to-end error correcting and in some degree has lower retransmission overhead. But error correcting logic is complex and consumes large chip area.

For end-to-end retransmission mechanism, the transmission latency may be long because packets may be destroyed on the way, but intermediate switches do not have such knowledge. Switch-to-switch retransmission responds more quickly, as each switch checks the passing packets and requests retransmission when necessary. However, it brings more computing and processing on the way. In addition, end-to-end error correcting has very complex logic.[4]

RESIDU NUMBER SYSTEM (RNS)

The Residue Number System (RNS) is an unconventional and non weighted number system, which is capable of supporting parallel, carry free, high speed arithmetic. In this system, arithmetic operations act on residues - reminder of dividing original number in several definite modulo - in parallel. Consequently computations on these residues which are smaller than the original number are performed, so speed up arithmetic and decreased power consumption is achieved [6].

A Residue Number System is characterized by a moduli set $\{m_1, m_2, ..., m_n\}$, where the modulo $m_i\,(i = 1, 2, ..., n)$ are pair-wise relatively prime [5]. Any integer X in the dynamic range of M, $M = m_1 ' \ m_2 ' \ ... ' \ m_n$, is represented by a N-tuple $(x_1, x_2, x_3, ..., x_n)$, where x_i is the residue of X in modulo m_i for $i = 1, 2, ..., n$.

The reconstruction of X from its residues $(x_1, x_2, x_3, ..., x_n)$ is based on the Chinese Reminder Theory (CRT) shown by:

$$X = \left\langle \sum_{i=1}^{n} (x_i N_i)_{m_i} \times M_i \right\rangle_M$$

$$M = \prod_{i=1}^{n} M_i$$

$$M_i = \frac{M}{m_i} \quad , N_i = \left\langle M_i^{-1} \right\rangle_{m_i}, i = 1, 2, 3, \dots, n \qquad (1)$$

The notation $\langle M_i^{-1} \rangle_{m_i}$ in (1) denotes the multiplicative inverse of M_i modulo m_i.

Another advantage of this system is a security; because the conversion of RNS to weighted number system needs moduli which operate as a key.

Some applications of RNS are digital signal processing [6], digital filters [7], coding theory [8], RSA encoding algorithm [9,10], digital communication [11], Ad-Hoc networks, distributed dependable and secure data storage and retrieval [12], ability of error detection and correction [13-14] and fault tolerant systems [15].

In this system, if error occurs in the one of residues, the affect of its is not effect other residues. In the other hand, RNS inherently is fault tolerant [16-18].

The rest of this paper is organized as following; In section 2 and 3 recollect Multi-Level Residue Number System and Redundant Residue Number System, respectively. In section 4 we define a brief review of Networks on Chip concept. Multi-Level Redundant Residue Number System and its application in Networks on Chip is presented in the section 5. Some comparisons are performed between Multi-Level Redundant Residue Number System and other conventional error control mechanisms in Network on Chip. Finally, the section 6 presents some conclusion of our error control mechanism.

Multi-Level Residue Number System

Arithmetic computations on each modulo could be done on the new Residue Number System and repeated until it led very small moduli. This property of RNS causes increasing calculations speed, decreasing power consumption, increasing security and fault tolerance. In other words this process could be repeated for several levels.

The system derived from the process is called Multi-Level Residue Number System (MLRNS). However in this system the dynamic range of any sub-Residue Number System like residues in i^{th} level must be greater than the greatest modulo in the previous level like residues in $(i\text{-}1)^{th}$ level. In this paper Two-level Residue Number System is assumed in order to simplify the presentation. It should be considered that provided method could be expanded for more levels as well [19].

In Two-Level Residue Number System, two symmetrical key encryption algorithms are used together, so the system has a high security. Another advantage of Two-Level Residue Number System is the simple selection moduli set for large presentation limits. This capability is achieved while few large moduli are selected and new RNS with smaller moduli is used for the second level. When we choose few numbers of great moduli of first level: First the problem of being pair-wise relatively prime of moduli and unbalancing of them, would be solved. Second: Because of using of less moduli, the converter circuits will be simple and conversions would be done rapidly. Meanwhile since the moduli in the second level were small, internal computations of Residue Number System are performed rapidly, because of short carry propagation [20].

Notations that used for Two-Level Residue Number System in this paper are as following:

- $\{m_1, m_2, m_3, \ldots, m_n\}$: First level of Residue Number System moduli set.
- $\{m_{i1}, m_{i2}, m_{i3}, \ldots, m_{in_i}\}$: Second level of Residue Number System moduli set for modulo m_i $(i = 1, 2, 3, \ldots, n)$.
- $(r_1, r_2, r_3, \ldots, r_n)$: Residues of first level Residue Number System.
- $(r_{i1}, r_{i2}, r_{i3}, \ldots, r_{in_i})$: Residues of second level for r_i $(i = 1, 2, 3, \ldots, n)$.

Arithmetic computations a Two-Level Residue Number System are performed on the second level residues. Hence two operand arithmetic operations are defined as following:

$$\{z_{i1}, z_{i2}, z_{i3}, \ldots, z_{in_i}\} = \{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in_i}\} \circ \{y_{i1}, y_{i2}, y_{i3}, \ldots, y_{in_i}\} \quad (2)$$

Where $z_{ij} = (x_{ij} \circ y_{ij}) \bmod m_{ij}$, $i = 1,2,3,\ldots,n$, $j = 1,2,3,\ldots,n_i$, and "$\circ$" could be addition, subtraction and multiplication [21-24].

For conversion from weighted number system to Two-Level Residue Number System, first the number should be converted to the first level Residue Number System then these residues should be converted to the second level Residue Number System. This process is shown in figure 2.

Figure 2: Conversion from weighted number system to Two-Level Residue Number System.

In the reverse conversion, $n \times n_i - channel$ CRT for $i = 1,2,3,\ldots,n$ it is necessary to convert second level residues to equal residues of first level and then convert to weight number system by using a n_i-channel CRT. This process is shown in figure 3.

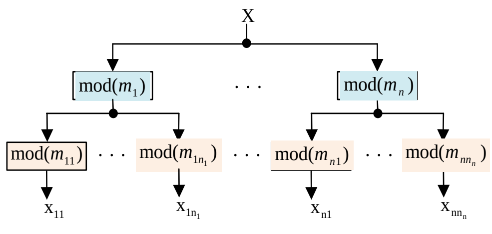


Figure 3: Conversion from Two-Level Residue Number System to weight number system

I. REDUNDANT RESIDUE NUMBER SYSTEM

Additional moduli are used in Residue Number System for error detection and correction. This system is called Redundant Residue Number System (RRNS). Redundant Residue Number System is presented by

$$\{m_1, m_2, \ldots, m_{h,m_{h+1}}, \ldots, m_{h+r}\} \text{ and } m_i > m_{i-1} \quad (3)$$

In the case all moduli were pair-wise relatively prime the presentation limit of this system is equal to:

$$\left[0, \prod_{i=1}^{h+r} m_i\right) \quad (4)$$

The interval $[0, M)$ that $M = \prod_{i=1}^{h} m_i$ constitutes the legitimate range and the interval $[M, M \times M_R)$ that $M_R = \prod_{i=h+1}^{h+r} m_i$ is associated so-called illegitimate range. Any integer belonging to the legitimate range will be labeled as legitimate and those belonging to the illegitimate range as illegitimate.

In the Redundant Residue Number System with $h+r$ modulo, a number like X which $\alpha \le X < \alpha + M$ is represented by $(x_1, x_2, \ldots, x_h, x_{h+1}, \ldots, x_{h+r})$.

The minimum distance $d_{\min}$ is a fundamental parameter associated with any error control code. The minimum distance of an RRNS code is $d_{\min}$, if the product of Redundant modulo satisfies the following relation vice versa

$$\max\{\prod_{i=1}^{d-1} m_{j_i}\} \le M_R < \max\{\prod_{i=1}^{d} m_{j_i}\}, 1 \le j_i \le h+r \quad (5)$$

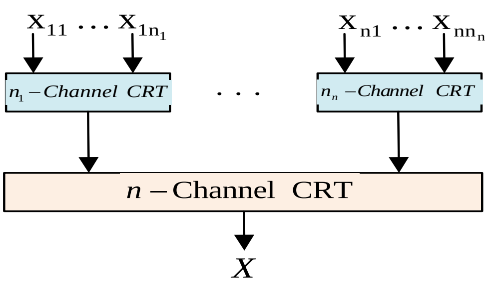
As a result, the minimum hamming distance of a Redundant Residue Number System is 3, if:

$$\max\{m_{j_1}.m_{j_2}\} \le M_R < \max\{m_{j_1}.m_{j_2}.m_{j_3}\} \quad (6)$$
$$1 \le j_1, j_2, j_3 \le h+r$$

Vice versa.

For example if a assumed Redundant Residue Number System with moduli set of $(m_1, m_2, m_3, m_4, m_5, m_6) = (3,7,11,13,16,17)$ for $d = 3$ MR should $272 \le M_R < 3536$

So minimum distance of 3 will be derived from each of these sets:



$$\{m_5, m_6 : M_R = 272\}$$
$$\{m_1, m_2, m_4 : M_R = 273\} \quad (7)$$
$$\{m_1, m_2, m_5 : M_R = 336\}$$

In addition, if we choose moduli $\{m_1, m_2, m_3, m_5 : M_R = 3696\}$ minimum distance of 4 will be derived.

As depicted before choosing redundant moduli set for minimum Hamming distance is non-unique. For choosing the optimal redundant moduli set in order to maximize the amount of M according to equation $M \times M_R = C$ (in which C is a constant) number, M_R should be minimized. The least M_R for minimum Hamming distance is equal to:

$$M_R = \max\{\prod_{i=1}^{d-1} m_{j_i}\} \quad ,1 \le j_i \le h+r \quad (8)$$

In the optimal Redundant Residue Number System $d-1$ big moduli of it should be chosen for minimum Hamming distance. In the other words:

$$d-1 = (h+r)-h$$
$$\Rightarrow d = (h+r)-h+1 = r+1 \quad (9)$$

Consequently in the optimal Redundant Residue Number System the minimum Hamming distance is equal to $r+1$ [25,26]. From here optimal Redundant Residue Number System is meant by Redundant Residue Number System.

In Redundant Residue Number System if h residues were achieved among $h+r$ residues then we could recover the number X. This property makes error detection and correction possible in Redundant Residue Number System which is the basis of error detection and correction as well.

$$\left(\prod_{i=1}^{h} m_{j_i} = M' \quad ,1 \le j_i \le n\right) \Rightarrow M < M' \quad (10)$$
$$\Rightarrow \alpha \le X < M < M' \Rightarrow \alpha \le X < M'$$

Derivation of X from h moduli is possible according to these equations by using Chinese Reminder Theorem.

Coding properties of Redundant Residue Number System are similar to Reed-Solomon well known codes. As a result by considering the minimum Hamming distance of this code is $r+1$:

- This code has the capability of detection r corrupted residues.
- This code has the capability of correcting $\left\lfloor \frac{r}{2} \right\rfloor$ corrupted residues.

- This code has the capability of simultaneous correcting up to λ corrupted residues and detection up to β corrupted residues $(\beta > \lambda)$, if and only if $\lambda + \beta \le r$.
- This code has the capability of simultaneous detection of t errors and detection of s residues which are corrupted in the path and hasn't reached if and only if $t + s \le r$.
- This code has the capability of simultaneous correction of t errors and correction of s residues which are corrupted in the path and hasn't arrived, if and only if $2t + s \le r$.

MULTI-LEVEL REDUNDANT RESIDUE NUMBER SYSTEM AND ITS APPLICATION IN NETWORKS ON CHIP

MULTI-LEVEL REDUNDANT RESIDUE NUMBER SYSTEM

In this paper Multi-Level Redundant Residue Number System is presented for increasing error detection and correction and increasing security in high speed computing without carry propagation. In Multi-Level Redundant Residue Number System, redundant moduli could be used for error detection and correction. Many errors could be detected or corrected in low levels in this method. Multi-Level Redundant Residue Number System has the capability to provide much fault tolerance for more important moduli in first level, in this method moduli which are supposed for lower levels of Redundant Residue Number System have more redundancy or have more Hamming distance in other words. There is an important issue in this system. The error detection and correction of RRNS coding are performed on moduli not on bits (Note moduli might be single bit or more). Therefore if a single-bit-error occurs, other moduli will not be effected. So, the rate of error detection and correction will be increase for single-bit-error by using small moduli.

ERROR CONTROL MECHANISM IN NETWORKS ON CHIP

Among the many NoC architectures proposed in the literature, similar to [27], we chose one that incorporates features that have been successful in many NoC designs and represents a reasonable design point. We use it as the basic architecture for incorporating the error recovery schemes used in the experiments. In the architecture we choose, the processor and memory cores communicate with each other through network components: switches, links, and network interfaces (NIs). NIs packetize data from the cores and build routing information for data communication. Each core has a sender and receiver NI for sending and receiving data from and to the core. We use an input-queued router with credit-based flow control with each packet segmented into multiple flits (Flow Control Units). We assume static routing, with paths set up at the sender NI, and wormhole flow control for data transfer. Applied error control mechanism in NoC is shown in figure 4.

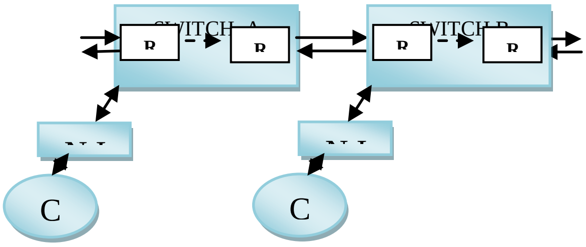


Figure 4: RNS Error control mechanism in Network on Chip

As shown in figure.3, in this scheme RNS encoder and decoder units are added to each switch for error detection/correction. RNS decoder decodes receiving flits for error detection/correction. RNS encoder unit encodes flits before sending to adjacent switch.

In our proposed scheme for error control, each switch has two types of buffers. One is transmission buffer, a FIFO buffer, which stores arrival flits and another is retransmission buffer, a spare buffer which stores a copy of sent flits for probable retransmission. Unlike previous works which ACK and NACK signals were used to flit flow control between adjacent switches, in our new work only NACK signal is used. Despite overhead of retransmission buffers, the control unit of the switch in our work has simple structure and retransmission delay is very low.

In this scheme switch has been implemented in fashion so that once a flit sent form FIFO buffer, one copy of sent flit will added at the end of retransmission buffer and in each cycle shifted on unit forward.

If a NACK signal received, transmission buffer (FIFO) of the sender switch will be blocked and flit which is located in front of retransmission buffer will be sent in next cycle. One copy of this sent flit will be stored at the end of retransmission buffer again. Structure of such a switch is shown in figure.5. Note that in order to simplicity of the figure, connection lines which are needed to transmitting flits between two adjacent switches are shown as unidirectional. Also, we show only transmission circuit and other units of the switch are not shown in this figure.

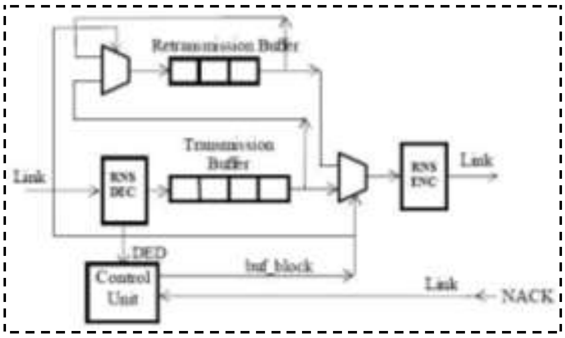


Figure 5: Proposed switch structure, include transmission buffer (FIFO) and retransmission buffer and RNS encoder/decoder

In each clock cycle, if transmission buffer of receiver switch not be full, a flit will be entered to its encoder circuit and then transmitted to next switch through connection link. We suppose each flit can be transmitted from one switch to its adjacent switch through a link during one cycle.

In this scheme single-errors will be detected and corrected automatically in RNS decoder unit of the receiver switch. Flits with two or more error (burst error) will be detected. In these cases faulty flits will be discarded and DED signal of decoder will be activated and a

NACK signal transmitted to sender switch. Sender switch send a copy of discarded flit again from its retransmission buffer.

After receiving NACK signal in sender switch, its buf_block signal will be activated and transmission buffer in sender switch will be blocked temporary. Then in next cycle flits sent from retransmission buffer of the sender switch. This operation will be continued until all discarded flit retransmitted. Then buf_block signal inactivated and in the next cycle flits will transmitted form transmission buffer (FIFO) again.

In this scheme retransmission will be down for each flit individually in with switch to switch mode. Therefore, power consumption overhead due to one flit retransmission is very low and negligible, in comparison with retransmission of a complete packet from source node to destination node. Also in most of times, probability of retransmission is very low, because in this scheme all single-errors will be detected. Because the header flit carries critical information (such as routing information), it is protected by error detect codes, which the switch checks at each hop traversal. As it was mentioned we use wormhole switching in our proposed architecture. In wormhole switching packets are split into flits (Flow Control Units) Because of inherent property of Multi-Level Residue Number System in splitting large numbers to small moduli, we use this property to split packets Small binary moduli (flits)(figure 6).

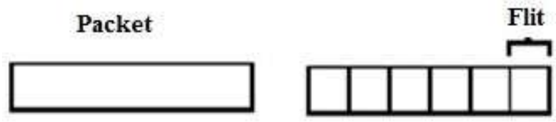


Figure 6.(a)

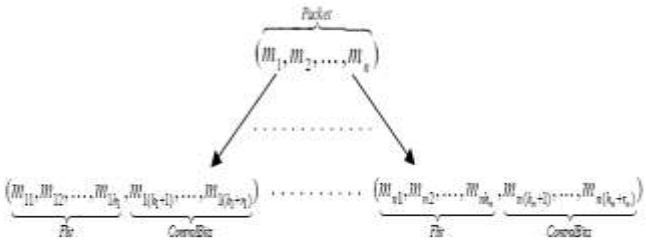


Figure 6.(b)

Figure 6. (a)splitting a packet into flits. (b) Multi-Level Residue Number System mechanism applied in wormhole switching

COMPARISON

We compare conventional coding mechanisms by two parameters: capability of error detection and correction and burst error detect. As presented in table 1, Multi-Level Redundant Residue Number System code is compared with parity, CRC and hamming mechanisms. None of other error control mechanisms are capable to split packet into small one. One of the major advantages of MLRRNS codes is the burst error detect. The Burst error detect are performed in moduli level.

Table 1. The comparison between error control mechanism

Error Control mechanism	Capability	Burst error check
Parity check Code	Error detection	No
Cyclic Redundancy Check	Error detection	Yes

Codes		
Hamming Codes	**Error detection and correction**	**No**
Multi-Level Redundant Residue Number System Codes	**Error detection and correction**	**Yes**

CONCLUSION

In this paper we propose using Redundant Multi-Level Residue Number System and a new switch-to-switch retransmission scheme to increase the data transmission reliability in on-chip networks. In our proposed scheme for error control, each switch has two types of buffers. One is transmission buffer, a FIFO buffer, which stores arrival flits and another is retransmission buffer, a spare buffer which stores a copy of sent flits for probable retransmission. Unlike previous works, in our new work only NACK signal is used. Also we introduce the Multi-Level Redundant Residue Number System (MLRRNS) as a numeric system and its capability of error detection and correction. then we proposed a new approach in applying MLRRNS error control mechanism into the Network on Chip architecture. In our scheme each switch equipped with RNS encoder/decoder units to error detection and correction. In this scheme error control and retransmission will be down for each flit individually with switch to switch mode. Therefore, delay and power consumption overhead due to one flit retransmission is very low and negligible, in comparison with retransmission of a complete packet from source node to destination node. The main property of Multi-Level Redundant Residue Number System is the splitting big numbers into small one and that future is used to split packets into small flits in wormhole switching. Another advantage of this error control mechanism is error detection/correction. Also the designers can add different redundant bits to every flits that it depends on the design. Finally, Our new method can achieve more optimizations in the terms of data security, error detection and correction, high speed data transmission and computation. Beijing, China

REFERENCES

[1] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M.Millberg, J.Oberg, et al, "A network on chip architecture and design methodology," ISVLSI, pp.117-24, USA, 2002.

[2] A.Mehran, S.Saeidi, A.Khademzadeh" Spiral: A heuristic mapping algorithm for network on chip", ,IEICE Electronics Express, Vol.4,No.15,478-484.Japan,Aug.2007.

[3] J. Henkel, W. Wolf, and S. Chakradhar, "On-chip networks: A scalable, communication-centric embedded system design paradigm," VLSI Design, pp. 845-851, India, 2004.

[4] L. Zhang, Hu. Li and X. Li , "A Routing Algorithm for Random Error Tolerance in Network-on-Chip" , Proceedings of the 12th international conference on Human-computer interaction: applications and services, pp.1210-1219, , Beijing,China, 2007.

[5] International Sematech. International Technology Roadmap for Semiconductors—2002 Update, 2002. Available at http://public.itrs.net.

[6] M. A. Soderstrand and et al. Eds, “Residue Number System arithmetic: modern applications in digital signal processing,” New York, IEEE Press, 1986.

[7] R. Conway and J. Nelson, “Improved RNS FIR Filter Architectures,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 51, No. 1, Jan. 2004.

[8] H. T. How, T. H. Liew, Ee-Lin Kuan, Lie-Liang Yang, and Lajos Hanzo, “A Redundant Residue Number System Coded Burst-by-Burst Adaptive Joint-Detection Based CDMA Speech Transceiver,” *IEEE Transactions on Vehicular Technology*, Vol. 55, No. 1, pp. 387-396, Jan. 2006.

[9] Mathieu Ciet, Michael Neve1, Eric Peeters and Jean-Jacques, “Parallel FPGA Implementation of RSA with Residue Number Systems,” *IEEE Proceedings of the Symposium on Circuits and Systems*, Vol. 2, pp. 806-810, Dec. 2003.

[10] Jean-Claude Bajard, Laurent Imbert, “A Full Implementation RSA in RNS,” *IEEE Transactions on Computer*, Vol 53, No.6, pp. 769-774 Jun. 2004.

[11] J. Ramirez, et al., “Fast RNS FPL-Based Communications Receiver Design and Implementation,” *Proc. 12th Int’l Conf. Field Programmable Logic*, pp. 472-481, 2002.

[12] F. Barsi and P. Maestrini, “Error Correcting Properties of Redundant Residue Number Systems”, *IEEE Transactions on* Computers, Vol. C-22 (3), pp. 307-315. Mar. 1973.

[13] D. Mandelbaum, “Error Correction in Residue Arithmetic,” *IEEE Transactions on Computers*, Vol. C-21 (6), pp. 538-545, Jun. 1972.

[14] H. Krishna, K.-Y. Lin, and J.-D. Sun, “A coding theory approach to error control in Redundant Residue Number Systems - Part I: theory and single error correction,” *IEEE Transactions on Circuits and Systems*, Vol. 39, pp. 8-17, Jan. 1992.

[15] J.-D. Sun and H. Krishna, “A coding theory approach to error control in Redundant Residue Number Systems - Part II: multiple error detection and correction,” *IEEE Transactions on Circuits and Systems*, Vol. 39, pp. 18-34, Jan. 1992.

[16] Lie-Liang Yang and Lajos Hanzo, “Redundant Residue Number System Based Error Correction Codes,” *IEEE*

Conference on Vehicular Technology, Vol. 3, pp. 1472 – 1476, Oct. 2001.

[17] R. W. Watson and C. W. Hastings, "Self-checked computation using residue arithmetic," *Proceedings of the IEEE*, vol. 54, No.12, pp. 1920-1931, Dec. 1966.

[18] Parhami B., "RNS Representation with Redundant Residues," *Proc. of the 35th Asilomar Conference on Signals, Systems, and Computers*, pp. 1651-1655, Nov. 2001.

[19] H. M. Yassine, "Hierachical Residue Number System suitable for VLSI Arithmetic Architectures," *IEEE International Symposium on Circuits and Systems*, Vol. 2, pp. 811-814, May 1992.

[20] Alexander Skavantzos and Mohammad Abdallah, "Implementation Issues of the Two-Level Residue Number System with Pairs of Conjugate Moduli," *IEEE Transactions on Signal Processing*, Vol. 47, No. 3, pp. 826-838, Mar. 1999.

[21] A. Hariri, K. Navi, R. Rastegar, "A Simplified Modulo $(2^n - 1)$ Squaring Scheme for Residue Number System," *IEEE International Conference on Computer as a tool*, Vol. 1, pp. 615-618, Nov. 2005.

[22] S. Timarchi, K. Navi and M. Hosseinzadeh, "New Design of RNS Subtractor for modulo $(2^n + 1)$," *2th IEEE International Conference on Information & Communication Technologies: From Theory To Applications*, Apr. 24-28, 2006.

[23] M. Hosseinzadeh, K. Navi and S. Timarchi, "Design Circuit Residue Number System in Current mode," *14th Iranian Conference of Electrical Engineering*, May 16-18, 2006.

[24] M. Hosseinzadeh, K. Navi and S. Gorgin, "A New Moduli Set for RNS: $\{r^n - 2, r^n - 1, r^n\}$," *International Conference on Electrical Engineering 2007*, Apr. 11-12, 2007.

[25] E. Kinoshita and K. Lee, "A Residue Arithmetic Extension for Reliable Scientific Computation," *IEEE Transactions on Computers*, Vol. 46, No. 2, pp. 129-138, Feb. 1997.

[26] L. L. Yang, and L. Hanzo, "Redundant Residue Number System Based Error Correction Codes," *Proceedings of VTC'2001*, pp. 1472-1476, Atlantic City, USA. *Fall*, 2001.

[27] M. Hosseinzadeh, A. Khademzadeh et all, " Reliable and Secure Chip Level Communication By Redundant Residue Number System ," *Journal of Integrated Design and Process Science* " , Vol. 12, No. 1, pp. 13-22, Jan. 2008.