# Scalable and Fault-tolerant Network-on-Chip Design Using the Quartered Recursive Diagonal Torus Topology

Xianfang Tan[1], Lei Zhang[1], Shankar Neelkrishnan[1], Mei Yang[1], Yingtao Jiang[1], Yulu Yang[2]

[1]Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, NV 89154
[2]College of Information Technology and Science, Nankai University, Tianjin, China
Emails: [1]{tanx, zhangl6}@unlv.nevada.edu, sanki.er@gmail.com, {meiyang, yingtao}@egr.unlv.edu

## ABSTRACT

Network-on-a-chip (NoC) is an effective approach to connect and manage the communication between the variety of design elements and intellectual property blocks required in large and complex system-on-chips. In this paper, we propose a new NoC architecture, referred as the Quartered Recursive Diagonal Torus (QRDT), which is constructed by overlaying diagonal torus. Due to its small diameter and rich routing recourses, QRDT is determined to be well suitable to construct highly scalable NoCs. In QRDT, data packets can be routed through a proposed minimal routing algorithm based on the Johnson codes that have traditionally been used in finite state machine designs. It has been shown that this proposed routing algorithm with minor modifications is capable of handling the single link/node failure. The hardware cost of the proposed QRDT architecture and its associated routing algorithm is revealed by designing two QRDT routers which have been synthesized using TSMC 0.18μm CMOS technology.

**Categories and Subject Descriptors:** C.1.2 Multiple Data Stream Architectures (Multiprocessors), Interconnection architectures (e.g., common bus, multiport memory, crossbar switch)

**General Terms:** Algorithms, Design

**Keywords** NoC, QRDT, routing algorithm, router

## 1. INTRODUCTION

The development of VLSI technology has continuously driven the increase of on-chip capacity. The International Technology Roadmap for Semiconductors (ITRS) [7] predicts that System-on-Chips (SoCs) will grow to multi-billion transistors in next a few years. One of the major challenges in designing such highly integrated SoCs will be to find an effective way to integrate multiple pre-designed Intellectual Property (IP) cores while addressing pressing power and performance concerns [1]. With a communication centric design style, Networks-on-Chips (NoCs) have emerged as a new SoC paradigm [1] to overcome the limitations of bus-based communication infrastructure.

In general, a packet-based NoC consists of routers, the network interface between the router and the IP, and the interconnection network. The major challenges imposed upon NoC designs include scalability, energy efficiency, and reconfigurability [2]. Numerous

interconnection network topologies have been considered for NoCs, including mesh [9], torus [8], fat tree [5], honeycomb [6], and quite a few others [2]. However, these interconnection topologies either scale poorly or have no support for reconfigurability. To address these two problems, a class of topologies named Recursive Diagonal Torus (RDT) [13] has made its way to NoC designs [15].

An RDT structure is constructed by recursively overlaying 2-D diagonal torus [13], and it has the following features: recursive structure with constant node degree, smaller diameter and average distance, and embedded mesh/torus topology. Consider a torus network composed of $N \times N$ nodes, where $N = nk$ and both $n$ and $k$ are natural numbers. The rank-0 torus is formed by creating a link between node $(x, y)$ and each of its neighboring four nodes: $(mod(x\pm1, N), y)$ and $(x, mod(y\pm1, N))$. On top of rank-0 torus, rank-1 torus is formed by adding one link between node $(x, y)$ and each of the four nodes $(mod(x\pm n, N), mod(y\pm n, N))$ it needs to connect to. The direction of the rank-1 torus is at an angle of 45 degrees to the rank-0 torus. On rank-1 torus, rank-2 torus can be formed by adding 4 links to a node in the same manner. In a more general sense, a rank-$(r+1)$ torus is formed upon rank-$r$ torus.

In this paper, a special type of RDT structure, Quartered RDT (QRDT) will be presented. The QRDT architecture exhibits a number of desirable networking properties that make it particularly appreciated in building large on-chip micro-networks. By indexing the network nodes with the Johnson codes, a new minimal routing algorithm, named Johnson Coded Vector Routing (JCVR), is proposed for QRDT. We further show that the JCVR algorithm with minor modifications is capable of routing data packets when there is a single link/node failure in a QRDT-based NoC. To examine the hardware cost of the proposed QRDT architecture and the JCVR algorithm, the QRDT router is synthesized and the simulation results are reported.

The rest of the paper is organized as follows. Section 2 introduces the QRDT structure and its network properties, followed by the introduction of the Johnson code and the basic functions defined for the Johnson code in Section 3. Section 4 describes the JCVR algorithm and the fault tolerant routing variant under single link/node failure. In Section 5, the design and implementation of the QRDT router and the simulation results will be described. Finally, Section 6 concludes the paper.

## 2. QRDT STRUCTURE AND ITS NETWORK PROPERTIES

As a special type of RDT, the QRDT network is also constructed by overlaying diagonal torus. An $N$-QRDT has $N \times N$ nodes where $N=4n$ and $n$ is a positive integer. A QRDT is composed of nodes, rank-0 links, and rank-1 links. Each node $(x, y)$, where $0 \le x, y \le N$-1, in the QRDT has 4 rank-0 links connecting to its four neighbors $(mod(x\pm1, N), y)$ and $(x, mod(y\pm1, N))$ on rank-0 torus, and 4 rank-1 links

connecting to its another four neighbors (mod($x\pm n$, $N$), mod($y\pm n$, $N$)) on rank-1 torus. In total, an $N$-QRDT has $N^2$ nodes and $4N^2$ links. Fig. 1 shows the structure of 4-QRDT.

Fig. 2(a) shows the directions of the 8 channels, and each channel is labeled by the dimension, rank number, and direction. Fig. 2(b) shows the coordinates of all eight nodes that are connected to node $(x, y)$ through respective channels shown in Fig. 2(a).
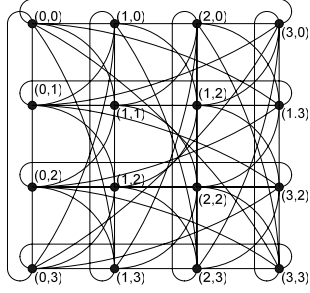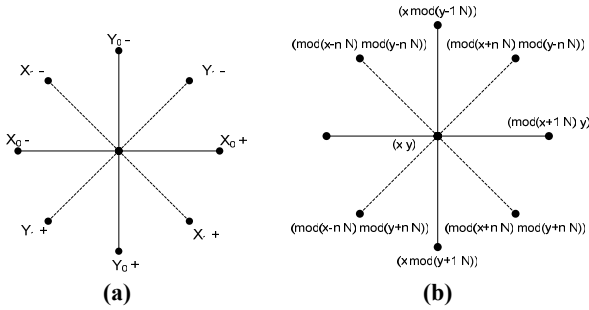


**Figure 1. Structure of 4-QRDT.**



**(a)**          **(b)**

**Figure 2. (a) Directions of 8 channels. (b) Coordinates of 8 nodes connected to (x, y) through the channels.**

**Lemma 1**: The diameter of the $N$-QRDT network is $n+1$, where $n=N/4$.

**Lemma 2**: The average distance of the $N$-QRDT is

$$\frac{\dfrac{32n^3}{3} + 20n^2 - \dfrac{32}{3}n + 2}{16n^2 - 1}, \text{ where } n=N/4.$$

The complete proofs of these two lemmas, based on the routing algorithm to be presented in Section 4, are provided in the appendix.

Table 1 lists the diameters and the average distances of QRDT, mesh, torus, and hypercube with different network sizes. One can see that QRDT has smaller diameter and average distance than the other three structures for most network sizes.

**Table 1. Comparison of four types of networks in terms of diameter (R) and average distance (AD).**

| Network Size | QRDT | | Mesh | | Torus | | Hypercube | |
|---|---|---|---|---|---|---|---|---|
| | R | AD | R | AD | R | AD | R | AD |
| 4×4 | 2 | 1.47 | 6 | 2.67 | 4 | 2 | 4 | 2 |
| 8×8 | 3 | 2.31 | 14 | 5.33 | 8 | 4 | 6 | 3 |
| 16×16 | 5 | 3.77 | 30 | 10.67 | 16 | 8 | 8 | 4 |
| 32×32 | 9 | 6.51 | 62 | 21.33 | 32 | 16 | 10 | 5 |

As QRDT is a special type of RDT structure, the Vector Routing (VR) algorithm [13] proposed for RDT can be readily applied to QRDT. The goal of the vector routing algorithm is to represent the routing vector with an expression that combines all unit vectors in rank-0 and rank-1 torus and the hop counts. However, the vector routing algorithm cannot guarantee it is minimal routing (i.e., the number of hops on the routing path may not be minimum). In this paper, we propose a minimal routing algorithm, named Johnson coded vector routing algorithm. In the following, the Johnson codes are introduced first before the JCVR algorithm is described.

# 3. JOHNSON CODES AND FUNCTIONS TO MANIPULATE THE CODES

## 3.1 Johnson Codes

The $m$-bit Johnson code has $M$ ($M=2m$) code words, and it can be denoted as $\{C_0, C_1, C_2, \cdots, C_i, C_{i+1}, \cdots, C_{M-2}, C_{M-1}\}$, where code word $C_i$ is given as

$$C_i = b_0^{(i)} b_1^{(i)} \cdots b_{j-1}^{(i)} b_j^{(i)} b_{j+1}^{(i)} \cdots b_{m-2}^{(i)} b_{m-1}^{(i)}, \text{ and}$$

$$b_j^{(i)} = \begin{cases} 0 & i+j < m \;\; or \;\; i+j \geq 2m \\ 1 & m \leq i+j < 2m \end{cases}.$$

For example, the eight 4-bit Johnson code words are 0000, 0001, 0011, 0111, 1111, 1110, 1100 and 1000.

On the other hand, given an $m$-bit Johnson code word $C_i$, $i$ can be determined by the $f_a$ function as

$$i = f_a(C_i) = \begin{cases} \displaystyle\sum_{j=0}^{m-1} b_j^{(i)} & if \;\; b_0^{(i)} = 0 \\ m + \displaystyle\sum_{j=0}^{m-1} \overline{b_j^{(i)}} & if \;\; b_0^{(i)} = 1 \end{cases}.$$

## 3.2 Basic Functions of Johnson Codes

Two basic functions of Johnson codes are defined: the distance function and the direction function. These functions will be used in the JCVR algorithm.

The distance function is used to determine the Hamming distance between two Johnson-codes. Given two $m$ bits Johnson-codes: $A$ and $B$, where $A= a_0 a_1 a_2 \ldots a_{m-1}$ and $B = b_0 b_1 b_2 \ldots b_{m-1}$, the distance $p$ between them is $p = \displaystyle\sum_{i=0}^{m-1} (a_i \oplus b_i)$.

The direction function is used to determine the output direction. The direction $c$ is calculated as:

Case 1:   $p=0$,     then     $c=0$

Case 2:   $a_0=0$ and $b_0=0$, then $c = \begin{cases} 1 & if \;\; \displaystyle\sum_{i=0}^{m-1} b_i > \sum_{i=0}^{m-1} a_i \\ -1 & if \;\; \displaystyle\sum_{i=0}^{m-1} b_i < \sum_{i=0}^{m-1} a_i \end{cases}$;

Case 3:   $a_0=1$ and $b_0=1$, then $c = \begin{cases} 1 & if & \sum_{i=0}^{m-1} b_i < \sum_{i=0}^{m-1} a_i \\ -1 & if & \sum_{i=0}^{m-1} b_i > \sum_{i=0}^{m-1} a_i \end{cases}$ ;

Case 4:   $a_0=0$ and $b_0=1$, then $c = \begin{cases} 1 & if & \sum_{i=0}^{m-1} \overline{b_i} \le \sum_{i=0}^{m-1} a_i \\ -1 & if & \sum_{i=0}^{m-1} \overline{b_i} > \sum_{i=0}^{m-1} a_i \end{cases}$ ;

Case 5:   $a_0=1$ and $b_0=0$, then $c = \begin{cases} 1 & if & \sum_{i=0}^{m-1} \overline{b_i} \ge \sum_{i=0}^{m-1} a_i \\ -1 & if & \sum_{i=0}^{m-1} \overline{b_i} < \sum_{i=0}^{m-1} a_i \end{cases}$ .

For simplicity, we denote the calculation of the distance and direction as one function $f_r$ as $(p,c) = f_r(A,B)$.

# 4. ROUTING ALGORITHM FOR QDRT

## 4.1 JCVR Algorithm

The routing steps from a source node to its destination node can be derived as $\vec{A} = vecX_1 \vec{X}_1 + vecY_1 \vec{Y}_1 + vecX_0 \vec{X}_0 + vecY_0 \vec{Y}_0$, where $\vec{X}_1$, $\vec{Y}_1$, $\vec{X}_0$, and $\vec{Y}_0$ are the unit vectors in rank-1 torus and rank-0 torus (with their directions shown in Fig. 2(a)) and their respective coefficients, $vecX_1$, $vecY_1$, $vecX_0$, $vecY_0$, give the number of hops on corresponding dimensions.

In the JCVR algorithm, the coefficients for the unit vectors are calculated at the source node. For $N$-QRDT, given the source and the destination addresses $S=(S_x, S_y)$, $D=(D_x, D_y)$, where $S_x$ and $S_y$ ($D_x$ and $D_y$) represent the $X$ and $Y$ coordinates of the source (destination) node in $m$-bit ($m=N/2$) Johnson code, respectively. The direction and distance values of the two coordinates, $(p_x, c_x)$ and $(p_y, c_y)$, can be determined by calculating $f_r$ as discussed in Section 3.2.

The address of each intermediate node $M=(M_x, M_y)$ on the routing path from $S$ to $D$ is given as,

$$M_x = f_s(S_x, c_x, N) = \begin{cases} mod_N(S_x + n) & when & c_x \ge 0 \\ mod_N(S_x - n) & when & c_x < 0 \end{cases}$$
$$M_y = f_s(S_y, c_y, N) = \begin{cases} mod_N(S_y + n) & when & c_y \ge 0 \\ mod_N(S_y - n) & when & c_y < 0 \end{cases}$$

The JCVR algorithm for QRDT is listed below.

**Johnson Coded Vector Routing Algorithm for QRDT:**
 ***begin***
  ***initialize*** $vecX_0$, $vecY_0$, $vecX_1$, $vecY_1$ to zero
  *let* $M_x=S_x$, $M_y=S_y$
  ***while***   ($D_x \ne M_x$ **and** $D_y \ne M_y$) ***do***
   ***calculate*** $(p_x, c_x)$ and $(p_y, c_y)$ using $f_r$ function
   ***if*** ($c_x=0$ **and** $c_y=0$) ***then set*** $vecX_0$, $vecY_0$, $vecX_1$, $vecY_1$ to zero
   ***elseif*** ($p_x+p_y \le n$) ***then***
    ***if*** ($c_x \ge 0$) ***then*** $vecX_0=vecX_0 + p_x$

    ***elseif*** ($c_x<0$) ***then*** $vecX_0=vecX_0 - p_x$
    ***if*** ($c_y \ge 0$) ***then*** $vecY_0=vecY_0 + p_y$
    ***elseif*** ($c_y<0$) ***then*** $vecY_0=vecY_0 - p_y$
    ***go to end***
   ***elseif*** ($p_x+p_y>n$) ***then***
    ***if*** ($c_x \ge 0$ **and** $c_y \ge 0$) ***then*** $vecX_1=vecX_1+1$
    ***if*** ($c_x \ge 0$ **and** $c_y<0$) ***then*** $vecY_1=vecY_1-1$
    ***if*** ($c_x<0$ **and** $c_y \ge 0$) ***then*** $vecY_1=vecY_1+1$
    ***if*** ($c_x<0$ **and** $c_y<0$) ***then*** $vecX_1=vecX_1-1$
    ***calculate*** $M_x$ and $M_y$ using the $f_s$ function
   ***endif***
  ***endwhile***
 ***end***

For an $N$-QRDT, the while loop will be executed no more than $m=N/4$ times to complete the calculation of the four coefficients ($vecX_1$, $vecY_1$, $vecX_0$, $vecY_0$).

At the source node, the four coefficients are computed and encapsulated into the packet. At the source node and each intermediate node, the coefficients for the unit vectors will be checked in a default order of $vecX_1$, $vecY_1$, $vecX_0$, $vecY_0$. The packet will be routed to the direction as determined by the first unit vector with non-zero coefficient (hop count) and this vector's coefficient, depending on the routing direction (+/-), will be decremented/incremented before the data packets are actually forwarded to the next hop towards the destination.

Compared to the VR algorithm, the advantage of the JCVR algorithm is that it can achieve minimal routing (as proved in Lemma 1), which is not guaranteed in VR.

## 4.2 Fault Tolerance Routing for QRDT under Single Link/Node Failure

It has been well recognized that fault-tolerance capability is vital for a NoC system [3], since one faulty link/processor may isolate a large fraction of IP cores from being reachable by other cores. Next we show that with minor modification, the JCVR algorithm is capable of handling single link/node failure.

The following assumptions are adopted in this paper [13]. 1) Any link or node in the network can fail, and the faulty components are unusable; that is, data will not be transmitted over a faulty link or routed through a faulty node. 2) The fault model is static; that is, no new faults occur during a routing process. 3) Both source and destination nodes (on rank-0 or rank-1 torus) are fault-free. 4) The faults occur independently. 5) If a node fails, all the eight links associated with the node on rank-0/1 tori are considered unusable.

During the routing process, when the next routing link/node is found to be faulty in a QRDT network, fault-tolerance routing can be realized dynamically as described below. The faulty link encountered must be on one of the four dimensions (i.e., $\vec{X}_1$, $\vec{Y}_1$, $\vec{X}_0$, and $\vec{Y}_0$). There are two cases to consider:

**Case 1:** if there exists one or more other dimensions with non-zero coefficient, the routing steps on such dimension(s) will be completed first followed by the routing steps on the dimension with faulty link/node.

**Case 2:** otherwise, one step on the dimension on the same rank and orthogonal to the dimension with faulty link/node will be completed

first followed by the routing steps on the dimension with faulty link/node and one opposite routing step on the orthogonal dimension.

Fig. 3(a) shows an example of Case 1 under single link failure. The detoured routing path is $S$-$M_0$-$M_1$-$D$, which does not introduce extra hops compared to the original routing path $S$-$M_0$-$M_2$-$D$. Similarly, as shown in Fig. 4(a), no extra hop is introduced on the detoured routing path for such a case under single node failure.

Fig. 3(b) shows an example of Case 2 with a single link failure, where the detoured routing path has two extra hops than the original routing path. As shown in Fig. 4(b), the number of extra hops on the detoured routing path is also two for such a case under single node failure.
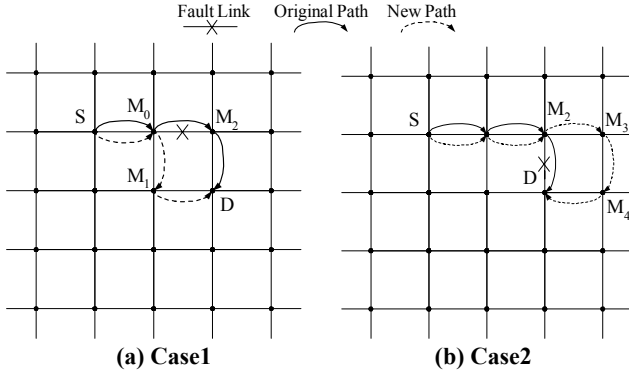


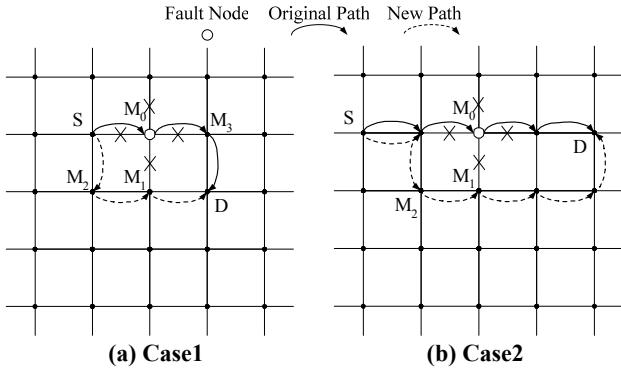**Figure 3. Two cases under single link fault.**



**Figure 4. Two cases under single node fault.**

Hence the following lemma can be derived.

**Lemma 3:** When a single faulty link/node exists in the QRDT network, the extra number of hops on the detoured route generated by the fault-tolerance JCVR routing algorithm, as compared to the number of hops on the original route for a fault-free QRDT, is no more than 2.

# 5. DESIGN AND IMPLEMENTATION OF QRDT ROUTER

To examine the hardware cost of the proposed QRDT architecture and the JCVR algorithm, the QRDT router is designed at RTL level using Verilog HDL and the design is subsequently synthesized using TSMC 0.18μm CMOS technology.

## 5.1 QRDT Router Design

The router designed for QRDT-based NoCs has up to 9 communication ports, named as N, NE, E, SE, S, SW, W, NW and L,

as there are 9 links in a node . The Local (L) is reserved to attach the router to an IP, and all the other eight ports (whose names follow the eight directions) connect the router to all its neighbors in eight directions. Each port has both input and output channels.

The main functions of the router include buffering, routing, scheduling, switching, and flow control. To implement these functions, the following components are designed: input channel module, output channel module, crossbar switch, and scheduler. Fig. 5 shows the top level block diagram of the QRDT router. The router is designed as a Verilog HDL soft-core incorporating parameterized components. In the following, the design of each component is described in brief. The details can be found in [10].
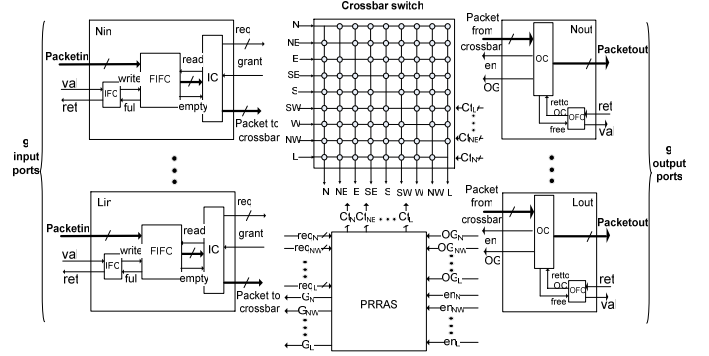


**Figure 5. Block diagram of the QRDT router.**

**Input Channel Module (ICM)**: The ICM performs the following functions. (i) It buffers the incoming packets using a FIFO buffer and it analyzes the header. (ii) It performs the routing algorithm. (iii) It generates a request ("req") to the scheduler requesting the destined output channel. Each ICM is composed of Input Flow Controller (IFC), FIFO, and Input Controller (IC). The IC at a local port implements the routing algorithm. Once a header flit reaches the IC, the routing vectors are calculated and updated in the header flit. The IC at other ports only implements the function of updating the routing vectors as described in Section 2.

**Output Channel Module (OCM)**: The OCM is responsible for flow control inside the router and outside the router. It is composed of two architectural blocks named Output Controller (OC) and Output Flow Controller (OFC).

**Crossbar Switch**: The 9×9 crossbar switch has 9 data inputs, each connected by an ICM, 9 control inputs from the scheduler, and 9 data outputs, each connected to an OCM. In Verilog soft-core implementation, a multiplexer-based crossbar switch design is actually adopted.

**Parameterized Round Robin Arbiter based Scheduler (PRRAS)**: The PRRAS design is parameterized with the router size. A 9-QRDT router composed of 9 decoders, 9 Parallel Round Robin Arbiters (PRRAs) [16], and 9 OR gates. Each PRRA is associated with one output port and responsible for arbitrating the requests to the output port from up to 9 input ports.

## 5.2 Synthesis Results

The QRDT routers using both VR and JCVR algorithms are implemented. For both designs, Verilog HDL codes [4] are generated and synthesized using Synopsys's design analyzer [12] targeting TSMC 0.18μm CMOS technology. As the VR-based QRDT router and JCVR-based QRDT router only differ on the design of the ICMs,

in the following, the performance of the ICMs and the whole router in terms of area (given as the number of 2-input NAND gates equivalent), timing, and power consumption (both dynamic power and leakage power) is reported.

The ICM contains IC, FIFO and IFC. Two types of ICs are designed: IC for the local port ($IC_L$), the port that connects the local node to the router, and IC at other ports ($IC_O$). Two types of $IC_L$ are designed based on the VR and JCVR algorithms with fault-tolerance capability under single link/node failure. Table 2 lists the area, timing delay, and power of the ICM with $IC_L$ for both routing algorithms. The results show that the ICM with VR-based $IC_L$ has less area, power consumption, and circuit delay than the ICM with JCVR-based $IC_L$.

**Table 2: Results of ICMs with $IC_L$.**

|  |  | ICM with $IC_L$ (VR) | ICM with $IC_L$ (JCVR) |
|---|---|---|---|
| **Total Area (2-input NAND gate)** | | 385.37 | 584.59 |
| **Power** | **Dynamic Power (*mW*)** | 5.9 | 6.55 |
|  | **Leakage Power (*nW*)** | 68.3 | 85.53 |
| **Timing Delay (*ns*)** | | 6.07 | 12.27 |

The results of other components can be found in [10]. Table 3 summarizes the results of the VR-based and the JCVR-based QRDT router designs. A moderate increase of area, power consumption and circuit delay for JCVR-based router is evident as compared to a VR-based router. However, as pointed out earlier, this extra cost is justifiable because JCVR can provide minimum routing, a feature that is critical in high performance NoCs but VR fails to deliver.

**Table 3: Results of QRDT routers.**

|  | VR-based Router | JCVR-based Router |
|---|---|---|
| **Area (2-input NAND gates)** | 6067.2 | 6266.4 |
| **Dynamic Power (*mW*)** | 61.2 | 62.2 |
| **Leakage Power (*nW*)** | 741.8 | 759.0 |
| **Timing Delay (*ns*)** | 6.34 | 8.1 |

## 6. CONCLUSION

In this paper, Quartered Recursive Diagonal Torus (QRDT) constructed by overlaying diagonal torus, was introduced as a novel topology for NoCs. The QRDT structure exhibits a number of desirable networking properties (including small diameter and average distance, constant node degree), which make it particularly appreciated in building large on-chip micro-networks. A minimal routing algorithm, the JCVR algorithm, was proposed for QRDT structure. It was shown that fault tolerance routing under single link/node failure can be handled by the JCVR algorithm with minor modifications. Compared with the VR algorithm, the JCVR algorithm achieves minimal routing. The tradeoff is the moderately higher implementation cost of the JCVR-based router than the VR-based router. Our future work includes the study of the layout scheme for the QRDT structure and the fault tolerant routing scheme under multiple link/node failures.

## 7. REFERENCES

[1] Benini L. and De Micheli G., 2002. Networks on chip: a new SoC paradigm, IEEE Computer, 35, 1, 70-78.

[2] Bjerregaard T., and Mahadevan S., 2006, A survey of research and practices of Network-on-chip, ACM Comput. Surv., 38, 1, 1-51.

[3] Dumitras T., Kerner S., and Marculescu S., 2003, Towards on-chip fault-tolerant communication, Proc. ASP-DAC, 225-232.

[4] Felicijan T. and Furber S. B., 2004, An asynchronous on-chip network router with quality-of-service (QoS) support, Proc. IEEE Int'l SoC Conf., 274-277.

[5] Guerrier P. and Greiner A., 2000, A generic architecture for on chip packet-switched interconnections, Proc. Design, Automation and Test in Europe, 250-256.

[6] Hemani A., *et al.*, 2000, Network on a chip: an architecture for billion transistor era, Proc. IEEE NorChip Conf.

[7] International Technology Roadmap for Semiconductors Roadmap, DOI=http://public.itrs.net/.

[8] Marescaux T., *et al.*, 2002, Interconnection networks enable fine-grain dynamic multi-tasking on FPGAs, Proc. 12th Conf. FPGA, 795-805.

[9] Millberg M., Nilsson E., Thid R., Kumar S., and Jantsch A., 2004, The Nostrum backbone-a communication protocol stack for networks on chip, Proc. Intl. Conf. VLSI design, 693-396.

[10] Neelkrishnan S., Yang M., Jiang Y., Zhang L., Yang Y., and Lu E., and Yun X., 2008, Design and implementation of a parameterized NoC router and its application to build PRDT-based NoCs, to be presented on ITNG 2008.

[11] Ogras U.Y., Hu J., and Marculescu R., 2005, Key research problems in NoC design: a holistic perspective, Proc. CODES+ISSS, 69-74.

[12] Synopsys Synthesis Tutorial, DOI=http://www.facweb.iitkgp.ernet.in/~apal/lpcs2007/webpages/271_Syn_tut.pdf.

[13] Yang M., Li T., Jiang Y., and Yang Y., 2005, Fault-tolerant routing schemes in RDT(2,2,1)/α-based interconnection network for networks-on-chip designs, Proc. ISPAN, 52-57.

[14] Yang Y., Amano H., Shibamura H., and Sueyoshi T., 1993, Recursive diagonal torus: an interconnection network for massively parallel computers, Proc. 5th IEEE Symp. Parallel and Distrib. Processing, 591-594.

[15] Yu Y., Yang M., Yang Y., and Jiang Y., 2005, A RDT-based interconnection network for scalable NoC designs, Proc. ITCC, 723-728.

[16] Zheng S. Q. and Yang M., 2007, Algorithm-hardware codesign of fast parallel round-robin arbiters, IEEE Trans. Parallel and Distrib. Syst., 18, 1, 109-115.

## APPENDIX

### A. Proof of Lemma 1

*Proof*: To prove the lemma, we first introduce the Quad-Star Network (QSN). An *N*-QSN is constructed based on a $(2N+1)\times(2N+1)$ mesh by removing all nodes with their minimum distance (in number of hops) to the center node (indexed as $(N, N)$) greater than *N* and their related links. Fig. 6 shows the structure of a

3-QSN, with the center node ($C$) and an edge node ($E$) marked. The minimum distance between $C$ and $E$ is 3 hops.

In an $n$-QSN, the total number of nodes in the network is given by

$$U(n) = 4 \times [n + (n-1) + \cdots + 2 + 1] + 1$$
$$= 4 \times \left[ \frac{n(n+1)}{2} \right] + 1 = 2n^2 - 2n + 1 \; .$$



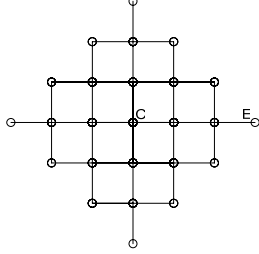**Figure 6. Structure of 3-QSN.**

And the sum of lengths (in number of hops) of all shortest paths between all nodes in the network to $C$ is

$$D(n) = 4 \times \left[ n^2 + (n-1)^2 + \cdots + 2^2 + 1^2 \right]$$
$$= \frac{4n(n+1)(2n+1)}{6}$$
$$= \frac{2n(n+1)(2n+1)}{3}$$

Fig. 7 shows that the 12-QRDT is partitioned into 8 QSNs in different sizes (shown in different colored regions) based on their respective location to the original node $O$ (the node at the left-up corner). Some of the QSNs are overlapped. In general, for an $N$-QRDT, it can be partitioned to one $n$-QSN ($B$ in Fig. 7), four $n$-QSN ($F_1$, $F_2$, $F_3$, and $F_4$ in Fig. 7) and three ($n$-1)-QSN ($S_1$, $S_2$, and $S_3$ in Fig. 7), where $n = N/4$.
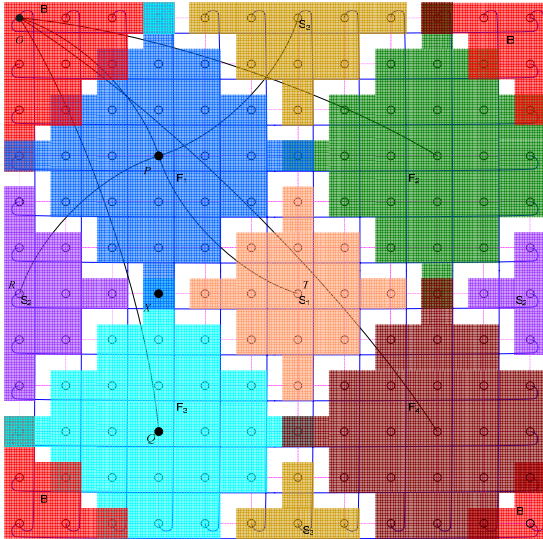


**Figure 7. Partition of QRDT to QSNs.**

Consider the $n$-QSN region $B$ in Fig. 7. Then the rank-0 path between any node in $B$ and the original node $O$ is less than or equal to $n$ hops. According to the JCVR algorithm, the routing path between any node in $B$ and the original node $O$ is less than $n$.

Then consider the $n$-QSN regions $F_1$, $F_2$, $F_3$, and $F_4$. The rank-0 path between any node in these regions and their center node is less than or equal to $n$ hops. According to the JCVR algorithm, the routing path between any node in $F_1$, $F_2$, $F_3$ and $F_4$ to the original node $O$ is less than $n+1$.

$S_1$, $S_2$, and $S_3$ are ($n$-1)-QSNs. All rank-0 paths between nodes in these regions to their original nodes are less than or equal to $n$-1 hops. According to JCVR algorithm, the routing path between any node in $S_1$, $S_2$, and $S_3$ to the original node $O$ is less than $n+1$.

Note that the combination of all these regions is equivalent to the whole QRDT network. Because the QRDT is symmetric to any of its nodes, such partition can be applied to all nodes. Then we can have the first conclusion:

The diameter of the $N$-QRDT network is less than or equal to $n+1$, where $n = \frac{N}{4}$.

As discussed in Section 4.1, the routing steps from a source node to a destination node can be derived as $\vec{A} = vecX_1 \vec{X}_1 + vecY_1 \vec{Y}_1 + vecX_0 \vec{X}_0 + vecY_0 \vec{Y}_0$. The length of a routing path is determined by the summation of the four coefficients and will not change with the sequence of routing steps. From Fig. 7, it can be seen that any routing path which involves more than 2 rank-1 links is not minimal. That is, any shortest routing path can have at most 2 rank-1 links.

Consider the shortest routing path between the original node $O$ and node $X$ as marked in Fig. 7. One of the shortest routing paths from $O$ to $X$ with 2 rank-1 links is $O \rightarrow P \rightarrow R$ (or $O \rightarrow P \rightarrow T$) plus $n$ rank-0 links from $R$ to $X$ (or $T$ to $X$), whose length is $n+2$. And one of the shortest routing paths from $O$ to $X$ with 1 rank-1 link is $O \rightarrow P$ plus $n$ rank-0 links from $P$ to $X$, whose length is $n+1$. And there are many shortest routing paths from $O$ to $X$ with only rank-0 links with lengths $3n$. Then we can have the second conclusion:

The diameter of the $N$-QRDT network is greater than or equal to $n+1$, where $n = \frac{N}{4}$.

Combing the above two conclusions, Lemma 1 holds. ∎

## B. Proof of Lemma 2

*Proof*: The calculation of the average distance in QRDT is performed by partitioning the network into a set of the QSNs.

According to the proof of Lemma 1, by the JCVR algorithm, the total lengths of all shortest paths between one selected node and all other nodes are

$$Sum = D(n) + 4 \times \{ D(n) - n[4 + (n-1)] + U(n) - [4 + (n-1)] \} + 3[D(n-1) + 2U(n-1)]$$
$$= \frac{32n^3}{3} + 20n^2 - \frac{32}{3}n + 2$$

And the number of all such paths is $N^2-1 = 16n^2-1$.

Then the average distance between one selected node and all other nodes is

$$d = \frac{D(n) + 4 \times \{ D(n) - n[4 + (n-1)] + U(n) - [4 + (n-1)] \} + 3[D(n-1) + 2U(n-1)]}{16n^2 - 1}$$
$$= \frac{\dfrac{32n^3}{3} + 20n^2 - \dfrac{32}{3}n + 2}{16n^2 - 1}$$

As the $N$-QRDT is a symmetric network, the average distance of $N$-QRDT is also $d$. ∎