DD-TDMA: A Distributed Dynamic TDMA Bus for 3D Bus-NoC Hybrid Network

Gaizhen YAN, Ning WU, Lei ZHOU, Feng GE

Abstract—**Three Dimensional (3D) bus-NoC (Network on Chip) hybrid network is efficient in exploiting the ultra-fast propagation in vertical direction. Transaction-less Dynamic Time Division Multiple Access (D-TDMA) protocol was proposed to fully utilize the vertical bus bandwidth. However, the central controlled arbiter leads to large amount of TSVs (Through Silicon Vias) for bus arbitration, which is a great challenge to current TSV technology. This paper is motivated to explore a Distributed D-TDMA (DD-TDMA) Bus to reduce TSV cost, limit maximum waiting time，decrease arbitration latency and fully utilize the bus bandwidth. Priority Code Updating Algorithm (PCUA), Logic Continuous Coding (LCC) policy, and dynamic CMOS based transceiver for arbitration bus have been proposed to meet the design goal of DD-TDMA. TSVs for arbitration are greatly reduced compared to the central controlled D-TDMA while starvation free is guaranteed. Although individual arbiter is needed in each node, the total area cost of all the arbiters is about 73% of the central controlled D-TDMA for four-node bus example. Experiment has shown that, under the 4×4×4 network configuration, 3D bus-NoC hybrid network incorporated with DD-TDMA outperforms 3D mesh NoC by at most 50% reduction in average network latency.**

Index Terms—**distributed bus, dynamic priority updating, 3D hybrid NoC, dynamic CMOS transceiver**

I. Introduction and Motivation

MOST recently, 3D NoC, an amalgamation of NoC and 3D Integration Circuit (IC), is proposed as a promising solution to System-on-Chip (SoC) interconnection bottleneck[1]. By vertically stacking multiple layers of active devices and interconnecting with Through-silicon Vias (TSVs), 3D architectures could reduce wiring length by a factor of the square root of the layer numbers [2].Compared with the planar wire, which is usually tens of millimeters, the inter-wafer distance is extremely small, ranging from 5 to 50μm [3]. Without further device scaling, the wire efficiency could be improved by 15% and the total active power could be reduced by more than 10% [4]. Other performance, such as noise immunity, logical span and transistor packing density could also be significantly improved.

3D bus-NoC hybrid Network (called 3D hybrid Network for short in this paper) takes bus as vertical interconnection and mesh [5, 6, 7] or other NoC interconnection [8, 9] as the lateral topology. Compared to other 3D topologies, the vertical one hop feature helps to utilize the ultra-low propagation latency. A central-controlled transaction-less D-TDMA [10] bus was proposed to fully utilize the bus bandwidth. D-TDMA dynamically increases and shrinks the time slots with the active nodes changing, leading nearly 100% bandwidth efficient. However, directly applying in vertical dimension would result in large TSV count, such as the work in [5] needing $(k\text{-}1)(3k+\log_2 k)$ control signals for k layer stack.

TSVs have specific geometrical characteristics. According to 2013 International Technology Roadmap for Semiconductors (ITRS) [11], current minimum global-level and intermediate-level TSV pitch are 8μm and 2μm respectively. Silicon on Insulator (SOI) technology may reduce the TSV pitch to 0.4μm [4]. Yet, the stringent surface toughness and cleanness requirement make the yield quite low, and there is still no sight in volume production in the next five years [12]. By contrast, the 4-input NAND gate area is currently 0.157μm^2 and will be reduced to 0.049μm^2 by the year 2020 [11]. The wire pitch in metal layer is also small, about 0.2μm in 65nm node technology [13]. Thus, too many TSVs would preempt the active device space and planer wire space, resulting in large area cost. What is worse, the TSV number negatively affects the total chip yield and cost. Supposing an optimistic single TSV failure rate of 10^{-4}， more than 2,000 TSVs would result in the total chip yield dropped below 80% [14].

To reduce the TSV count, we are motivated to design a distributed bus. Compared to the central controlled bus, it consumes much less wires for carrying on the arbitration policy. Our previous work [9] proposed a kind of distributed bus architecture for 3D hybrid NoC: Fake Token bus. It locates identical arbiters in each node. The request signals are shared among all arbiters, and active nodes queue up in the status register. With this scheme, for bus with k nodes, only k TSVs are needed for arbitration. And the arbitration latency cost in wires is reduced to one time of wire propagation latency. However, Fake Token bus consumes relatively more chip area. It also has limitations for priority service extension, for the request signals contain no priority information.

Priority-covering based distributed arbitration not only

Manuscript received June 10, 2015. This work was supported in part by the National Natural Science Foundation of China under Grants 61376025 and 61106018, and the Industry-academic Joint Technological Innovations Fund Project of Jiangsu under Grant BY2013003-11.

G. YAN is with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing , 210016, China, and with the College of Mathematics, Physics and Information Engineering, Anhui Science and Technology University, Fengyang, 233100,China.(email: xucs_yan@126.com)

N. WU and F. GE are with the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, 210016, China (e-mail: wunee@nuaa.edu.cn; gefen@nuaa.edu.cn).

L. Zhou is with the College of Information Engineering, Yangzhou University, Yangzhou, 225009, China. (e-mail: tomcat800607@126.com)

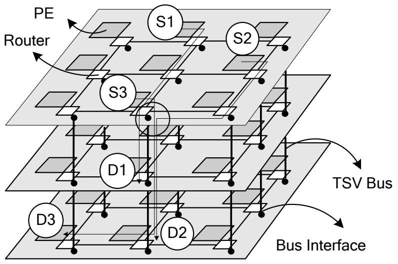


Fig. 1. Bus traffic in 3D bus-mesh hybrid NoC

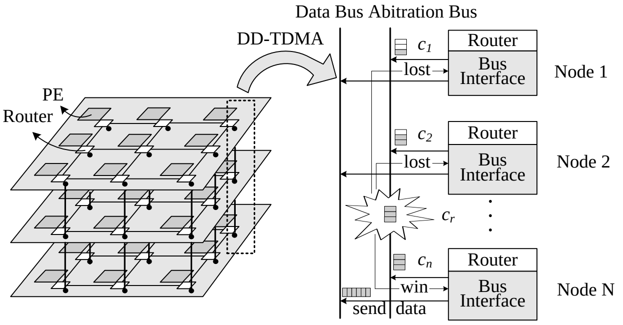


Fig. 2. Priority covering based DD-TDMA

consumes relatively less wires, but also provides the convenience for providing priority service. A node wins the arbitration by covering the priority code of others. However, when to be applied in 3D NoC, following challenge must be fully considered.

In the first, in 3D hybrid NoC, the bus traffic includes not only the one from local Process Element (PE), but also the one from the router. And router traffic might be from any other PE of the network, depending on the routing algorithm, as shown in Fig. 1. Traffic between different node pairs (S1 to D1; S2 to D2; S3 to D3) is delivered to destinations through same vertical bus interface. If differentiated service is needed, the priority should be relative to specific traffic, rather than to a node. Then, it is not feasible to bind a steady priority to a specific bus interface. If no priority service is needed, a dynamic priority assignment scheme is needed to guarantee unique winner and starvation free.

Secondly, existing priority-covering based distributed bus, such as Futurebus+ [15] and Multibus II [16], arbitrate in bit-by-bit way. Such scheme will result in large arbitration latency, which might be unacceptable in NoC communication.

Thirdly, wire AND and wire OR are the most common way to implement priority covering. And open collector or open drain drivers are usually applied for such operation. However, in an integrated circuit, they might cause large power consumption.

In this paper, a priority-covering based distributed Dynamic TDMA bus architecture, called DD-TDMA, is designed for 3D hybrid NoC. Traffic priority service is not considered yet, and will be extended in the future work. The innovative features of DD-TDMA are as follows:

A. A Priority Code Updating Algorithm (PCUA) is proposed to dynamically assign the code to each node. Under this assignment scheme, bus bandwidth is fully utilized and the maximum waiting time is kept not more than *k* time slots for *k*-node bus.

B. Logic Continuous Coding (LCC) policy is proposed to avoid arbitrating bit by bit and reduce the arbitration critical path. With LCC, all bits of the priority code can accomplish the priority covering operation simultaneously. And TSVs for arbitration is reduced to *k*-1 for *k*-node bus.

C. Wire AND operation is applied for priority covering in this paper. A dynamic CMOS based transceiver is designed for performance and power consumption consideration. The dynamic power consumption is only relative to the bus load capacitor. The more nodes on the bus, the faster the wire AND operation will be.

The rest of this paper is organized as follows. Section II illustrates priority assignment scheme of DD-TDMA. Section III describes implementation of DD-TDMA in details. Section IV gives comparison of TSV cost and area cost among different bus architectures. DD-TDMA performance in NoC application is also given. Main contributions of this paper and future work are concluded in Section V.

II. Priority Assignment Scheme of DD-TDMA

A. Problem Description

DD-TDMA is designed based on priority covering scheme, as shown in Fig. 2. All the active nodes simultaneously send the priority codes to the shared arbitrating bus and the code with the highest priority will cover all the others. Through reading the arbitration bus, all nodes would be clear about the arbitration winner. If the code read from the bus is equal to the code the node has sent, then the node would be the winner; otherwise, it will be the loser.

Defining *C* as the priority code space under the coding rule *R*, expressed as:

$$R : C = \{c_1, c_2, c_3, ..., c_n \mid \forall i \neq j, c_i \neq c_j\} \tag{1}$$

Defining *F* as the arbitration operation and *P* as the priority level set under the arbitration operation *F*,

$$F : C \to P = \{p_1, p_2, p_3, ..., p_n \mid F : c_i = p_i, 1 \le i \le n\} \tag{2}$$

where p_i is the priority level of priority code c_i under arbitration operation *F*.

Let *C*(*t*) as the code set taking part in arbitration at time *t*,

$$C(t) = \{c_i \mid c_i \in C\}; F : C(t) \to P(t) \tag{3}$$

Defining the priority covering scheme under arbitration operation *F* as $\Phi|_F$, then ,

$$\Phi|_F \ (C(t)) = c_r \ |_{c_r \in C(t), F(c_r) = \max P(t)} \tag{4}$$

where c_r is the remaining code on the arbitrating bus. Defining the bus node set as *N*, the active bus node set as $N(t)$, and the bus node count as *k*, then

$$N = \{n_i \mid 1 \le i \le k\}, N(t) \subseteq N \tag{5}$$

With the priority covering scheme $\Phi|_F$, design goal of DD-TDMA can be expressed as:

Giving priority code space *C*, arbitration operation *R* and priority level set *P*,

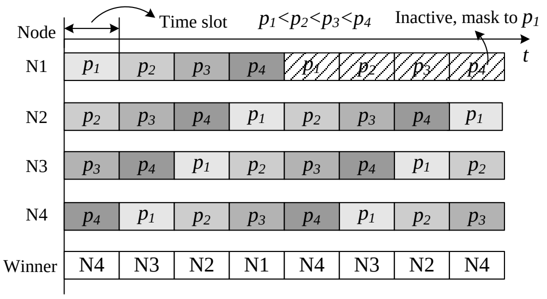
Finding a priority code assignment scheme *A*,

$$A : N \to C \tag{6}$$

```
Algorithm 1: Priority Code Updating Algorithm (PCUA)
Input :  CPC //Current  Priority Code
         Active Node
Output: Arbitration Code // code to the bus

Initiate  node priority code differently
if ( CPC with highest priority level ) then
|      Updating CPC to code with lowest  priority level
else
|      Updating CPC by increasing one  priority level
 end if
if  (Active Node)  then
|      Arbitration Code  = CPC
else
|      Arbitration Code  = code with lowest  priority level
end if
```

Fig. 3. Pseudo-code of the proposed PCUA



Node								
N1	p_1	p_2	p_3	p_4	p_1	p_2	p_3	p_4
N2	p_2	p_3	p_4	p_1	p_2	p_3	p_4	p_1
N3	p_3	p_4	p_1	p_2	p_3	p_4	p_1	p_2
N4	p_4	p_1	p_2	p_3	p_4	p_1	p_2	p_3
Winner	N4	N3	N2	N1	N4	N3	N2	N4

Fig. 4. Graphic view of the proposed PCUA

Satisfying

A is injective (7)

$$c_r \neq null, if \quad N(t) \neq \Phi \tag{8}$$

$$TS_i \leq k \tag{9}$$

Where TS_i is the maximum waiting time slot of node i.

B. Priority Code Updating Algorithm

To meet the requirement of (7)-(9), a Priority Code Updating Algorithm (PCUA) is proposed for dynamic priority code assignment, as shown in Fig. 3. For better understanding, Graphic view of the proposed PCUA is also given in Fig. 4

At the initial time slot, all nodes are initiated with different priority code; the one with highest priority level will win the arbitration. In the next time slot, all nodes will update their codes to increase the priority level by one, no matter it is an active or inactive node. If the priority level is already the highest, it will be decreased to the lowest. This updating operation satisfies the requirement of (7), that is, guarantees the winner unique at any time. And priority code of any node would be with the highest priority level for every k time slots (k is the bus node number). The waiting time slot is not more than k, that is, requirement (9) is met.

If a node is inactive, the updated priority code will be masked to the lowest priority, as shown in Fig.4. And the winner will be selected among the remaining active nodes. From the above analysis, with the proposed Priority Code Updating Algorithm, the bus will not be idle if only there are active nodes, and the bus bandwidth can be fully utilized. Requirement of (8) is satisfied.

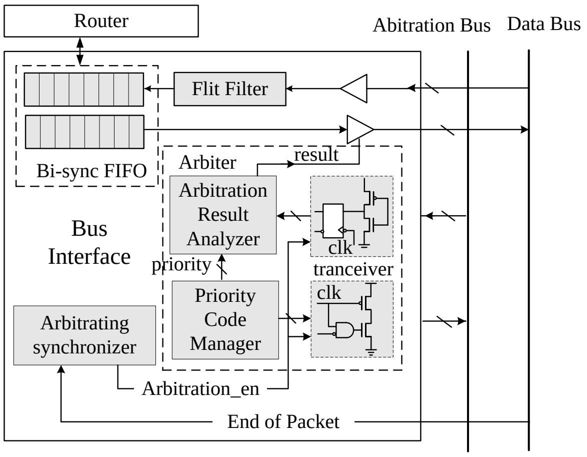


Fig. 5. Implementation architecture of DD-TDMA

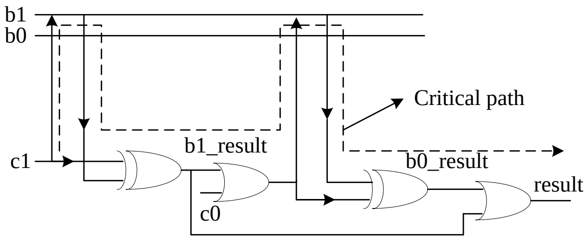


Fig. 6. Arbitration Critical path under bit-by-bit way

III. Implementation of DD-TDMA

A. Architecture of DD-TDMA

3D Hybrid topology takes bus as vertical interconnection and NoC as the lateral topology. To mix bus and NoC interconnection in one SoC (System on Chip) system, a Bus Interface is needed to bridge between the router and bus architecture. As shown in Fig. 5, the Bus Interface consists of two bi-sync FIFO buffers, a bus arbiter and an arbitration synchronizer. The bi-sync FIFO buffers bridge the router and bus clocked by two different frequencies. The arbitration synchronizer notifies all the distributed arbiters to start arbitration. The arbiter answers for the priority code management and arbitration judgment.

In NoC, the packet size is not fixed. To fully utilize the bus bandwidth, DD-TDMA is designed to arbitrate packet-wise. Whenever a tail flit (indicating end of a packet) is detected, the arbitration synchronizer will notify the arbiter to update the priority code according to PCUA described in section II. The priority code transceivers are also enabled.

To accomplish the priority-covering based distributed arbitration, wire AND is taken as the arbitration operation F. the priority coding policy and the arbitration bus transceiver are all designed under this operation.

B. Priority Coding Policy

Most previous work accomplishes the wire AND logic in bit-by-bit way. Under such policy, operation result of higher bit will directly determine whether the lower bit is taken into the arbitration. This process prolongs the arbitration critical path, as shown in Fig. 6. The higher bit of the priority code $c1$ is first sent to the wire AND bus. After one wire propagation

TABLE I
EIGHT BIT PRIORITY ENCODING EXAMPLE OF LCC

Code		Priority level (wire AND)
LOF Format	LZF Format	
11111111	11111111	0
11111110	01111111	1
11111100	00111111	2
11111000	00011111	3
11110000	00001111	4
11100000	00000111	5
11000000	00000011	6
10000000	00000001	7
00000000	00000000	8

time, the wire AND result of *b1* will be read and compared to *c1*. If they are not equal (b1_result = 1), *c0* will be masked and not take part in the lower bit arbitration, only if all bits finish the arbitration, the final result can be achieved.

To reduce the arbitration critical path, we are motivated to adopt an alternative priority coding policy, with which, all bits can accomplish wire AND operation simultaneously. The proposed policy is called Logic Continuous Coding (LCC). LCC follows two rules:

Rule 1: LCC code string consists of consecutive logic ones and consecutive logic zeroes. Full logic ones and full logic zeroes are also acceptable

Rule 2: At most a logic one string and a logic zero string can be included in LCC string.

According to the above coding policy, two priority code subsets exist. The first is the codes with a series of ones followed by a series of zeroes, which are called in Logic One First (LOF) format; the second is the codes with a series of zeroes followed by a series of ones, which are called in Logic Zero First (LZF) format. Both of the two sets are applicable, however, they cannot be mixed in one application.

In a *k* bit binary string, there would be 2*k* legal codes and *k*+1 codes in each code subset. That is, at least *k*-1bits are needed for LCC coding to meet the requirement of the *k*-node bus. For better understanding, an eight-bit coding example is shown in Table I, among which {00000000} has the highest priority, while {11111111} is the lowest. The priority level is only relative to the number of zeroes in the code string, without consideration of LOF format or LZF format. The more zeroes in the code string, the higher the priority level is. Whenever the lower priority codes meet the higher priority one, the later will always win the arbitration and hold its code unchanged on the bus. The lowest priority is notated as Level 0 priority, and the higher priorities are numbered in ascending order.

C. Arbitration Bus Transceiver

NMOS logic or PMOS logic can be applied for wire AND logic to avoid forming cut-through current path. However, there is a dilemma in the load transistor design. To reduce the static power consumption, the load transistor should be designed with as large resistance as possible. But to speed up the signal edge transition, small resistance is preferred. Therefore, it's hard to well balance between power consumption and performance. A dynamic CMOS based arbitration bus transceiver is designed in this paper, as shown in Fig. 7. The transceiver operation follows three steps.

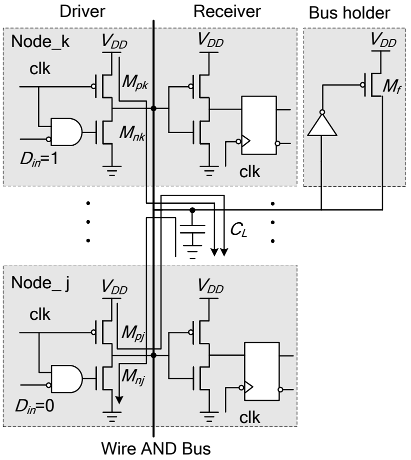


Fig. 7. Dynamic CMOS based wire AND bus transceiver

The first step: pre-charging. When the clock signal *clk* equals zero, the PMOS transistor in all nodes are turned on and the NMOS transistors are all off, no matter what logic D_{in} is. This situation is much like an inverter with logic zero input and the load capacitor C_L will be charged, where C_L is the sum of the TSV capacitance and all the receiver input capacitance. For a bus with *k* nodes, there would be *k* charge current. C_L Should be charged to logic one before the clock ends zero logic.

The second step: wire AND evaluation. When *clk* becomes logic one, all the PMOS transistors are turned off, and the NMOS is driven by D_{in}. If D_{in} is logic zero, the NMOS transistor is turned on and low impendence path is provided for C_L discharging to logic zero. In this situation, the driver just works like an inverter with logic one input. If D_{in} is logic one, the NMOS is turned off and the driver is much like a tri-state inverter at high impedance state. If there are multiple turned on NMOS transistors, the discharging process is just accelerated and the bus state would be finally logic zero. Only if all the NMOS transistors are turned off, there would be no discharging path for C_L and the bus state would be kept as logic one. The wire AND logic is effectively implemented. A bus holder is introduced to prevent discharging caused by leakage current.

The third step: bus reading. When the clock signal becomes logic zero again, the flip-flop in the receiver will register the bus state. On the meanwhile, the next pre-charging process also begins. If the charging process is too fast, the bus state may be covered before the flip flop reads. If such situation is met, an inverter can be inserted between the bus driver and the flip-flop receiver to meet the signal hold requirement. This requirement can also be met by shrinking the channel width of the PMOS transistors in drivers.

From the above analysis, the dynamic power consumption is only relative to the bus load capacitor. The more nodes on the bus, the faster the pre-charge and the evaluation process are. The worst case is that there is only one node taking part in the arbitration, which is similar to the data bus.

IV. COMPARISON AND EXPERIMENT

A. TSV Cost Comparison

TSVs consumed by arbitration bus greatly rely on the arbitration policy. Central controlled arbitration needs

TABLE 6
TSV COUNT COMPARISON

Arbiter	TSV count for arbitrating (*k* layer)	8 layer example
D-TDMA[11]	$(3k + \log_2 k)(k - 1)$ bits	189
Fake Token[13]	*k* bits	7
DD-TDMA	*k*-1 bits	8

TABLE III
ARBITER AREA COST COMPARISON

Arbiter	Nodes	Area(μm^2)
D-TDMA[11]	4	2012
Fake Token[13]	4	2128
DD-TDMA	4	1472

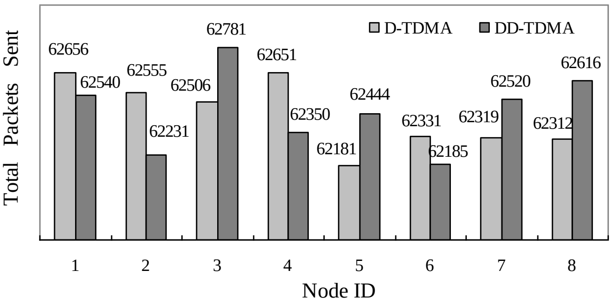
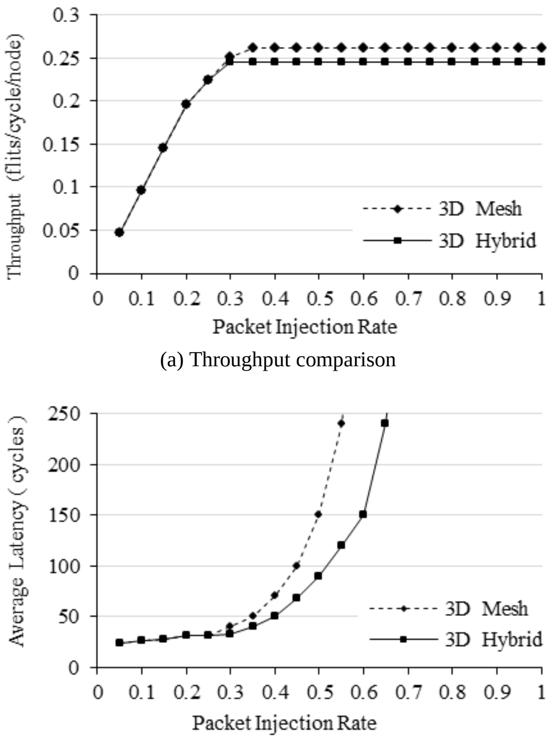


Fig. 8. Starvation test and comparison



(a) Throughput comparison

(b) Average latency comparison

Fig. 9. Performance comparison between 3D hybrid and 3D Mesh NoC

individual connection between the central arbiter and the nodes, and usually consumes more TSVs than distributed controlled arbitration. TSV cost comparison among DD-TDMA and existing vertical bus in 3D bus-NoC hybrid Network is shown in TABLE II. Both Fake Token and DD-TDMA are distributed. They consume much less TSVs than the central controlled D-TDMA. In the eight node example and under the same TSV technology, the incurring area cost can be reduced more than 90%.

B. Arbiter area cost comparison

Area cost of four-node D-TDMA arbiter synthesized by Synopsys Design Compiler for TSMC 90nm CMOS technology has been given in [5]. To give an intuitional comparison, we have implemented four-node DD-TDMA and Fake Token arbiter under the same technology. And the Area cost comparison is shown in TABLE III. From the table, area consumed by DD-TDMA is about 73% of the D-TDMA. However, both central controlled D-TDMA and Distributed Fake Toke arbiter need to queue up the active nodes according to the request signals. The logic needed to be processed would be increased with proportional to the *k* power of 2 (*k* is the bus node number). DD-TDMA arbitrates with the priority code, and the priority code length linearly increases the bus node. That is, when the node number is further increased, the area cost reduction would be much larger than the four-node example.

C. Starvation Test and comparison

In this paper, it is assumed that no differentiated service is required, and all the bus nodes should be serviced within limited maximum waiting time, that is, no node is starved. This experiment is designed to validate how the nodes are serviced under DD-TDMA. And comparison between eight-node D-TDMA and DD-TDMA is given in Fig. 8. In the experiment, packets injection rate for all nodes is set as 1:8 and all nodes send packets in a random way. That is, the bus works at full load situation. The final results show that DD-TDMA performs as excellent as D-TDMA and almost provide equal service to all nodes. As shown in Fig. 8, the difference among the total packets sent by each node is negligible. And DD-TDMA shows a very low Relative Standard Deviation (RSD) of 0.281%, a little smaller than D-TDMA, 0.319%.

D. Performance Comparison

To validate the performance of DD-TDMA bus in 3D hybrid network, we have incorporated the developed bus to our FPGA based NoC test platform RcEF3Ns [17]. And 4×4×4 3D mesh and 3D hybrid architecture configuration are taken as test bench. The traffic mode is set to random uniform distribution, and flit size is set 2 to 8. The throughput and average latency are watched and compared, as shown in Fig. 9. The throughput of the two architectures is kept almost the same, and the saturated point of 3D hybrid NoC is a little lower than 3D mesh topology, as shown in Fig. 9(a). The average latency of the two architectures is also quite similar when the injection rate is less than 0.3, as shown in Fig. 9(b). With the increase of injection rate, the latency of 3D hybrid NoC increases much more slowly than 3D mesh architecture. Specifically, at the injection rate of 0.55, the latency of 3D hybrid NoC is only 50 % of the 3D mesh topology. This may benefit from the vertical one hop feature of 3D hybrid NoC.

V. CONCLUSION AND FUTURE WORK

In this paper, we have proposed and implemented a priority-covering based distributed D-TDMA bus for 3D hybrid network to reduce the TSV cost. Experiment has shown that 3D hybrid NoC with DD-TDMA bus outperforms 3D-mesh NoC in network average latency. However, in some real application, different Quality of Service (QoS) might be needed by different traffic. For DD-TDMA is naturally priority based, we will extend it for differential service supporting in the future. And a QoS supported 3D hybrid NoC would be further developed.

References

[1] B.S.Feero, and P. P. Pande, “Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation”, IEEE Transactions on Computers, vol.58,no.1,pp. 32-45. Jan, 2009.

[2] J. Joyner, P. Zarkesh-Ha, and J. Meindl, “A stochastic global net-length distribution for a three-dimensional system-on-chip (3D-SoC)”,In Proc. 14th Annual IEEE International ASIC/SOC Conference, Arlington,VA , pp.147-151,Sept. 2001.

[3] K. Puttaswamy, and G. Loh, “Implementing Caches in a 3D Technology for High Performance Processors”. In Proc. The International Conference on Computer Design,pp.525-532, Oct. 2005.

[4] A. W. Topol, D. C. La Tulipe,Jr.L. Shi *et al.*, “Three-dimensional integrated circuits”. IBM Journal, pp.491-506,Jul.,2006.

[5] F. Li, C. Nicopoulos , T. Richardson et al., “Design and management of 3D chip multiprocessors using network-in-memory”, In Proc 33rd International Symposium on Computer Architecture, Boston, MA , pp. 130–141, 2006.

[6] M. Ebrahimi, M. Daneshtalab,P. Liljeberg et al.,“Cluster-based topologies for 3D Networks-on-Chip using advanced inter-layer bus architecture”. Journal of Computer and System Sciences.vol.79, no. 4, pp.475–491, Jun.,2013.

[7] A. Rahmani. K. R. Vaddina , P. Liljeberg et al., “High-Performance and Fault-Tolerant 3D NoC-Bus Hybrid Architecture Using ARB-NET-Based Adaptive Monitoring Platform”. IEEE Transactions on Computers, vol.6,no.3, pp.734-747, Mar., 2014.

[8] Y. Xu, Y. Du, B. Zhao et al., “A low-radix and low-diameter 3D interconnection network design”, In IEEE 15th International Symposium on High Performance Computer Architecture, Raleigh, NC, pp.30-42,Feb.,2009.

[9] L. Zhou, N. Wu, X. Chen et al., “A Design Methodology for Three-Dimensional Hybrid NoC-Bus Architecture”, IEICE Transactions of Electronic, pp.492-500, Apr.,2013.

[10] T. D. Richardson, C. Nicopoulos, D. Park et al., “A Hybrid SoC Interconnect with Dynamic TDMA-Based Transaction-Less Buses and On-Chip Networks”, In Proceedings of the 19th International Conference on VLSI Design, pp.3-7, Jan., 2006.

[11] ITRS. “International Technology Roadmap for Semiconductors”. Avaible at http://www.itrs.net.2013

[12] J.H. Lau. “TSV Manufacturing Yield and Hidden Costs for 3D IC Integration”, In Proc of 60th Electronic Components and Technology Conference, Las Vegas, Nevada, pp. 1031-1042, Jun. 2010,

[13] Predictive technology model. Available at http://ptm.asu.edu/

[14] C.-C.Chan, Y.-T. Yu, and I.H.-R Jiang, “3DICE: 3D IC cost evaluation based on fast tier number estimation”, in Proc. of the 12th International Society for Quality Electronic Design, Santa Clara, CA,pp. 1–6, Mar.,2011.

[15] IEEE Std 896. 2-1991, IEEE standard for futurebus+ - physical layer and profile specification. Available at http://ieeexplore.ieee.org

[16] P. L. Borrill, Microstandards special future: A Comparison of 32-bit Buses, IEEE Micro, vol., no. 6, pp. 71-79, May, 1985

[17] J. Zheng, N. Wu, L. Zhou, et al., “RcEF3Ns: A Scalable and Reconfigurable Multi-FPGA Emulation Platform for 3D NoC,” In Proceedings of the World Congress on Engineering and Computer Science, pp. 40-45, Oct. 2014.