PDDVB: A Priority Division Distributed Vertical Bus for 3D Bus-NoC Hybrid Network

Gaizhen YAN , Ning WU, Lei ZHOU and Fen 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 aims at developing a TSV-cost efficient Priority Division Distributed Vertical Bus (PDDVB), which can provide both round robin service and differential service. With the proposed priority covering based arbitration scheme, 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 logic elements consumed by all the arbiters are about 69.4% of the central controlled D-TDMA for eight-node bus example. Experiment has shown that, under the 8×8×4 network configuration, 3D bus-NoC hybrid network incorporated with PDDVB outperforms 3D mesh NoC by at most 26.6% reduction in average network latency.**

I. Introduction

Most recently, with the development of 3D integrated technology, 3D NoC is becoming a promising solution to the System-on-Chip (SoC) interconnection bottleneck[1]. By vertically interconnecting wafer with Through-silicon Vias (TSVs), 3D architectures could reduce wiring length by a factor of the square root of the layer numbers [2]. Compared to the planar wire, the inter-wafer distance is extremely small, ranging from 5 to 50μm [3], and thus signals can be propagated ultrafast in the vertical direction [4]. However, the hop-by-hop vertical communication in the traditional mesh based 3D NoC do not make good use of this beneficial feature.

3D ciliated topology [1], 3D XNoTs topology [5] and 3D bus-NoC hybrid topology [6-10] have been proposed to meet the one hop character in vertical direction. However, number of TSVs is limited by current TSV technology. Too many TSVs will consume large area and negatively affect the chip yield [11]. 3D ciliated topology and 3D XNoTs topology locate one router at the bottom or middle layer for interlayer date exchange, resulting in multiple data links in vertical direction. The gigantic TSVs increased with layers are a great challenge to current TSV technology. By contrast, 3D bus-NoC hybrid topology only needs one TSV data link.

A transaction-less Dynamic Time Division Multiple Access (D-TDMA) [12] bus has been proposed for 3D bus-NoC hybrid network, in order to fully utilize the bus bandwidth. However, because of the central controlled feature, directly applying in vertical dimension would result in large TSV count, such as the work in [6], needing $(n\text{-}1)(3n+\log_2 n)$ control signals for n layer stack.

To further reduce the control signals of the vertical bus, our previous work proposed a priority-covering based Distributed D-TDMA (DD-TDMA) [13]，as shown in Fig. 1. All the active nodes are required to simultaneously send the priority codes to the shared arbitration bus and the code with the highest priority will cover all the others. Node with the code left on the arbitration bus will be the winner. By the proposed Priority Code Updating Algorithm (PCUA), Logic Continuous Coding (LCC) policy, and dynamic CMOS based transceiver, DD-TDMA can reduce the TSV cost, decrease the arbitration latency, avoid starvation and fully utilize the bus bandwidth. What is more, the priority-covering scheme can also provide the convenience for priority service.

Anyway, to guarantee starvation free, the priority code of each node is initiated different and updated by one level for every one slot to make sure they differ from each other all the time. This feature make the priority code cannot reflect the real time requirement of each traffic, or else the uniqueness of the arbitration winner cannot be guaranteed any more. Therefore, DD-TDMA shows limitation when to directly provide priority service.

Based on the previous proposed DD-TDMA, a Priority-Division Distributed Vertical Bus (PDDVB) is proposed for 3D Bus-NoC Hybrid Network in this paper. It extends [13] in the following aspects.

(1) We have introduced two priority codes: Traffic Priority Code (TPC) and Node Priority Code (NPC). And with the two priority codes, the priority assignment scheme is redesigned to provide both round robin service and differential service.

(2) Technology constraints of the TSV bus are discussed in details.

Manuscript received June 10, 2015. This work was supported in part by the National Natural Science Foundation of China under Grants 61376025 and 61106018, the Industry-academic Joint Technological Innovations Fund Project of Jiangsu under Grant BY2013003-11, and the Jiangsu Innovation Program for Graduate Education (Fundamental Research Funds for the Central Universities) under grant KYLX15_0283.

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).

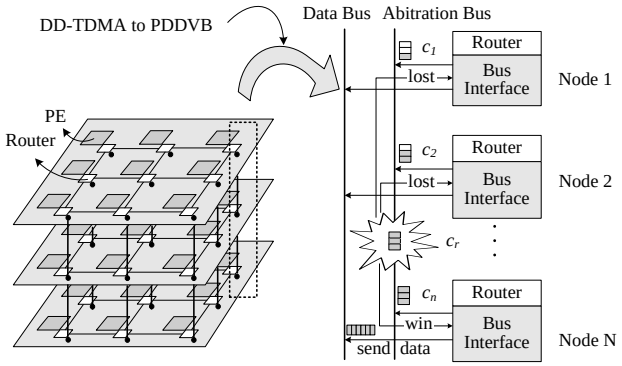


Fig. 1. Priority covering based arbitration scheme

(3) Features of most recent bus architectures are discussed and compared to PDDVB in details.

(4) Addition to the wire AND logic based priority covering operation, wire OR logic based one is also discussed.

(5) More experiments have been done to invalidate the performance of the PDDVB based 3D bus-NoC network under large network topology size.

The rest of this paper is organized as follows. Section II illustrates the newly designed priority assignment scheme. Section III lists the important implementation issues needed to be considered. Section IV gives detailed arbitration policy comparison between PDDVB and related arbitration schemes, upon which TSV cost comparison is made. To validate the key functions and performance, extensive experiments and comparison have been done in Section V. Conclusion and the future work are given in Section VI.

II. Priority Assignment Scheme of PDDVB

A. Problem Description

PDDVB arbitrates through the Traffic Priority Code and Node Priority Code. Traffic Priority Code is the index of the real time requirement of the traffic, and Node Priority Code guarantees arbitration winner uniqueness.

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

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

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

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

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

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

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

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

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

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

```
Algorithm 1: Priority Code Updating Algorithm (PCUA)
Input : TPC //Traffic Priority Code
        CNPC // Current Node Priority Code
        TS_cnt // Counter of the Waiting Time Slot
        Active Node
Output: TAC // Traffic Arbitration Code sent to bus
        NAC // Node Arbitration Code sent to bus
1:  Initialization:  set initial CNPC differently;
                     set TS_cnt to zero;
2.  if ( CNPC is with the highest priority level)
3.  |   CNPC  = lowest level priority code
4.  else
5.  |   CNPC = CNPC +1;
6.  end if
7.  if ( Active Node)
8.  |   if ( TS_cnt < TS_max)
9.  |   |   TAC = TPC
10. |   else
11. |   |   TAC = highest level priority code
12. |   end if
13. else
14. |   TAC = lowest level priority code
15. endif
16. if ( win in traffic priority arbitration )
17. |   NAC = CNPC
18. else
19. |   NAC =  lowest level priority code
20. endif
```

Fig. 2. Pseudo-code of the redesigned PCUA

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

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

Giving: priority code space *C*, arbitrating operation *F* and priority level set *P*,

Finding: a priority code assignment scheme *A*,

$$A: N \to C \quad (6)$$

Such that:

A is injective (7)

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

$$TS_i < TS_j,\ if\ p_{ti} > p_{tj} \quad (9)$$

$$|TS_i - TS_j| \le N,\ if\ p_{ti} = p_{tj} \quad (10)$$

$$TS_i \le TS_{\max} \quad (11)$$

where p_{ti} and TS_i are the traffic priority and maximum waiting time slot of node *i*. TS_{max} is the waiting time slot allowance for a specific traffic.

B. Priority Code Updating Algorithm

To meet the requirement of (7)-(11), two phase priority covering scheme is intended to be applied. The TPC of the active node will be firstly sent to the arbitration bus, in order to find the active node with the highest traffic priority level. For the node wins the traffic priority arbitration, the NPC will be sent to arbitration bus in finding the only one winner. Conversely, for the node loses the traffic arbitration, the NPC will be masked to the lowest for the absence of node priority arbitration. Therefore, the Priority Code Updating Algorithm (PCUA) is redesigned for both TPC and NPC assignment, as shown in Fig. 2.

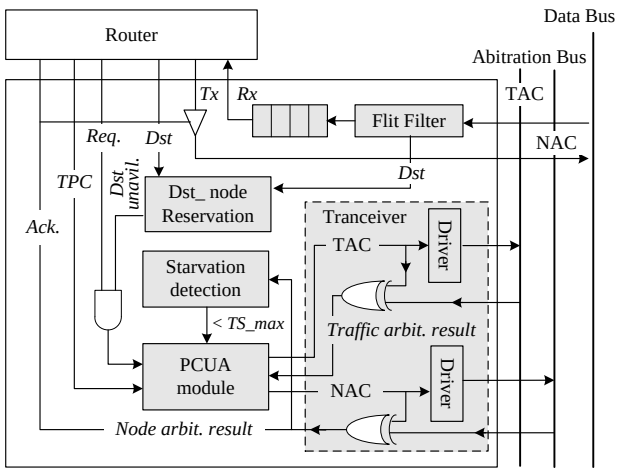


Fig. 3. BIU implementation architecture of PDDVB

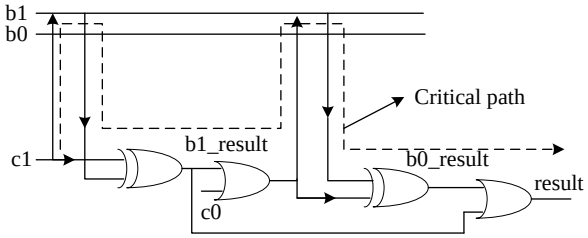


Fig. 4. Arbitration Critical path under bit-by-bit way

At the initial time slot, NPC of different node are initiated to be different. And in every time slot, all nodes will update their NPC to increase the priority level by one. For the one already with highest priority level, its NPC will be set to lowest priority level, as shown in Line 1-6 in Fig. 2. Therefore, NPC of different node will always differ from each other, and thus requirement (7) is satisfied.

During the arbitration, all the active nodes will first send the TPC to the arbitration bus, and inactive node keep the TPC with lowest priority level. Node with the highest level TPC will win the traffic arbitration, as shown in Line 7-14. This guarantees the requirement (8) and (9). But traffic priority of different node might be the same. To keep the final winner uniqueness, all the winners in traffic arbitration phase will send their NPC to arbitration bus, and NPC of losers will be masked to the lowest level, as shown Line 16-20 in Fig. 2. If TPC of all nodes are the same, then, winner will be found out through NPC. Note that, under the previous stated NPC updated scheme, NPC of any node would be with the highest priority level for every N time slots (N is the bus node number). Therefore, the waiting time slot will be not more than N, that is, requirement (10) is met. At this time, round robin service is provided.

Anyway, consistent higher priority traffic might starve the lower priority traffic. To guarantee starvation free, that is, all traffic can be transmitted within limited time slot TS_{max}, we have introduced a waiting time slot counter *TS_cnt*. If *TS_cnt* is not less than TS_{max}, the TPC will be immediately updated to the highest level for fast transmission. Therefore, requirement (11) can be met.

III. Implementation of PDDVB

A. Architecture of Bus Interface Unit

3D bus-NoC hybrid topology mixes bus and NoC interconnection in one SoC system. Bus Interface Unit (BIU) is necessary to bridge the two different interconnection architectures. As shown in Fig. 3, the designed BIU mainly consists of a bi-sync FIFO buffers, a Dst_node reservation module, a starvation detection module, a priority code updating module and a transceiver.

The bi-sync FIFO buffer is applied as an input buffer to bridge the router and bus clocked by two different frequencies. The Dst_node reservation module is introduced to support flit-wise arbitration. If a header flit is received, the corresponding destination node will be registered in the reservation table, and the node will be released when the tail flit is received. By checking the reservation table, the newly sent packets then will not be mixed with others in the destination node. Starvation detection module and PCUA modules just work as stated in Section II. Specific transceivers are designed for priority covering arbitration, among which the design of the drivers is of great importance. They are closed related to the arbitrating operation F and priority coding policy, which will be discussed next.

B. Priority Coding Policy

To accomplish the priority-covering based distributed arbitration, wire AND or wire OR can be taken as the arbitrating operation *F*. The priority coding policy and the arbitration bus transceiver will be designed under the two operations.

Our previous work [13] has found out that binary coding bit-by-bit arbitration way increases the arbitration latency, as shown in Fig. 4. The higher bit of the priority code c1 is first sent to the wire AND bus. After one wire propagation 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. Logic Continuous Coding (LCC) is proposed to reduce the arbitration critical path for wire AND operation. LCC code string consists of a consecutive logic-one string and a consecutive logic-zero string. And depending on the logic of the first bit, LCC code can be divided into two subsets: Logic One First (LOF) format and Logic Zero First (LZF) format. Either of the two sets is applicable for priority covering operation, but they cannot be mixed in one application. Although LCC is initiated for wire AND based arbitration, it also fit for wire OR based arbitration.

For better understanding, an eight-bit LCC coding example is shown in Table 1, among which {00000000} has the highest priority in wire AND logic and lowest priority in wire OR logic, while {11111111} is the lowest priority in wire AND logic and the highest in wire OR logic.

TABLE 1
8 BITS LCC ENCODING EXAMPLE

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

C. Arbitration Bus Transceiver

Open collector or open drain drivers are the typical circuit for wire AND or wire OR operation. However, in an integrated circuit, they might cause large power consumption. We have proposed a dynamic CMOS based arbitration bus transceiver to reduce the static power consumption. Drivers for wire AND arbitration and wire OR arbitration are shown in Fig. 5.

The wire AND driver, as shown in Fig. 5(a), works in two steps: pre-charge and evaluation. In logic-zero duration of the clock signal, the PMOS transistor in all nodes are turned on and the NMOS transistors are all off. 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. And then, the bus state will be evaluated in the logic-one duration of the clock signal. If one or more of the D_{in} signal is logic zero, the NMOS transistors in corresponding drivers will be turned on and low impendence path is provided for C_L discharging to logic zero. Only if all the D_{in} signals are logic one, the NMOS transistors in all drivers would be turned off and the driver high impedance state. The C_L will keep pre-charged logic one unchanged. Anyway, because of the leakage current, C_L may discharge to logic zero before the receiver reading. Therefore a bus holder should be introduced.

The wire OR driver works almost the same to wire AND driver, as shown in Fig. 5(b). In logic-one duration of the clock signal, the NMOS transistors of all drivers are turned on and PMOS transistors are all off. The load capacitor C_L will be discharged to zero. In the logic-zero duration of the clock signal, the NMOS transistors are turned off and bus state will be evaluated based on the input D_{in}. If only there is a logic-one input hitting on the PMOS transistor, the bus state will be logic one and will be kept during the whole evaluation process. Therefore, no bus holder is needed.

Based on the above analysis, neither the wire AND nor the wire OR driver can form a cut-through current in the whole arbitration process, therefore saving power consumption compared to NMOS and PMOS logic.

D. Technology Constraints

The design of PDDVB for 3D Bus-NoC hybrid network is limited by the current TSV technology, such as the bus node number and the data link bandwidth. Table II lists the

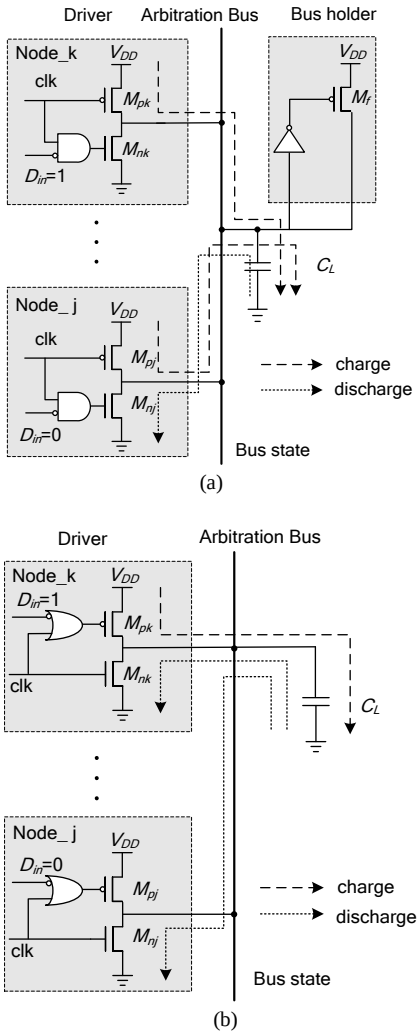


Fig. 5. Bus drivers for (a) wire AND (b) wire OR

important features of current TSV technology reported by 2013 International Technology Roadmap for Semiconductors (ITRS) [14].

The bus node number is also the stack layer number of the 3D NoC. Based on Table 2, the state of art can only stacks 5 layers at most, and 8 layers during the year 2015-2018. On the meanwhile, according to [12], giving the bus bandwidth equal to the bandwidth of the router's each connection, the bus-mesh hybrid NoC would outperform mesh NoC when the bus node is not more than nine. That means eight layers could meet both the current TSV technology and the system performance requirement, even in the near future.

TSV is large in planer diameter and short in vertical length, the vertical bus clock frequency can be much higher than the router clock. To keep bandwidth unchanged while reduce the TSV number, the bus clock frequency should be pumped up twice when the data width be halved. 1GHz clock frequency for router is reasonable because 3D chips have high constraint in heat dissipation and high clock frequency may not be preferred [8]. Supposing a router with 64 bit data width and 1GHz highest frequency, to meet the same bandwidth requirement between the uni-directional bus and bi-directional router port, the bus width and clock frequency should satisfy the value pairs listed in TABLE 3.

TABLE 2
2013 ITRS PREDICTION ABOUT TSV INTERCONNECTION

Geometric Size	Global Level		Intermediate Level	
	2013-2014	2015-2018	2013-2014	2015-2018
Minimum TSV diameter	4-10 μ m	2-3.5 μ m	1-2 μ m	0.5-2 μ m
Minimum TSV pitch	8-20 μ m	4-7 μ m	2-4 μ m	1-4 μ m
Minimum TSV depth	40-100 μ m	30-50 μ m	5-40 μ m	5-20 μ m
Maximum TSV aspect ratio	5:1-12:1	12:1-20:1	5:1-20:1	5:1-20:1
Minimum contact pitch	20 μ m	10 μ m	2-3 μ m	2-3 μ m
Number of die per stack	2-5	2-8	2-5	8-16(DRAM)

TABLE 3
BANDWIDTH REQUIREMENT OF THE VERTICAL BUS

Bus width (bits)	128	64	32	16
Clock frequency(GHz)	1	2	4	8

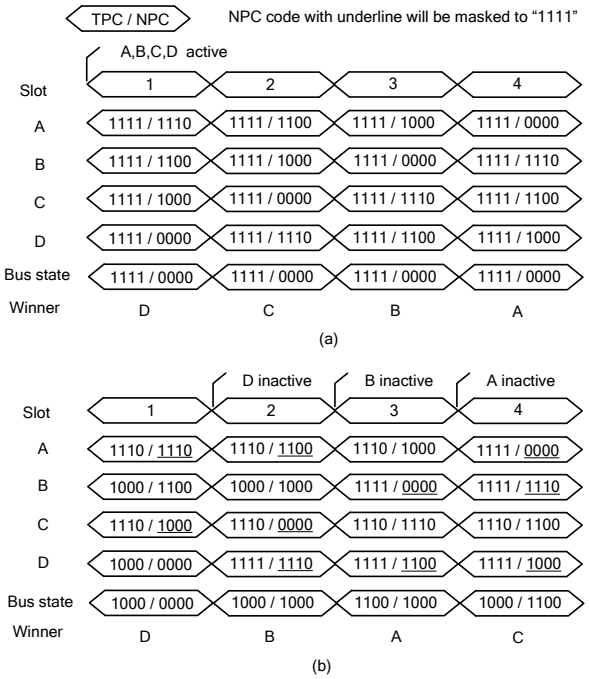


Fig. 6. Bus arbitration examples (a) round robin service demonstration (b) differential service demonstration

E. Arbitration Examples

PDDVB with four nodes under wire AND priority-covering operation is taken as an example to demonstrate the support for round robin service and differential service, as shown in Fig. 6. Fig. 6(a) is the round robin service case when traffic from all nodes is all with the same priority level. It is shown that all the active nodes take turns to win the bus arbitration. Fig. 6(b) shows the situation when the traffic is with different priority level, i.e., the differential service. TPC of node A and C are the same while node B and D have the same TPC. Because of the relatively low level TPC, Node A and C firstly lost the arbitration during traffic priority arbitration phase, and their NPCs are masked to the lowest level in the node priority arbitration phase. Winner is found out between Node B and D based on their NPC. In the following slots, the winner will be in turn Node B, A and C.

IV. ARBITRATION POLICY COMPARISON

Arbitration policy plays an important role in determining the performance of buses. Traditional arbiters like Static Priority (SP), Round Robin (RR) and Time Division Multiple Access (TDMA) cannot meet the traffic requirement well in 3D hybrid topology. SP tends to starve the lower priority node; RR has no support for differential service and TDMA shows poor bandwidth utilization. There are already several high performance bus architectures, such as Lottery Bus [15], D-TDMA [12] and Fake Token [8]. However, they might have limitations in 3D hybrid NoC application, because of failing to meet technology constraint.

Lottery Bus arbiter [15] selects the active master to service based on probabilistic assigned "tickets". Thus starvation of the low priority master is reduced. By allocating more tickets to certain master, bandwidth reserving service can also be effectively implemented. However, the high performance is guaranteed at the cost of high hardware requirement. What is more, Lottery Bus is designed transaction-based, which is already shown inefficient in hybrid NoC application [12].

D-TDMA [12] dynamically increases and shrinks the time slots with the active nodes changing, leading nearly 100% bandwidth efficient. Priority schemes and absolute bandwidth guarantees are possible by blocking timeslot allocations. However, directly applied in vertical dimension would result in large amounts of TSVs, such as the work in [6] needing $(n\text{-}1)(3n+log_2 n)$ control signals for n layer stack. If priority is supported, extra wires should be added for delivering the priority of the request.

Fake Token bus [8] locates identical arbiters in each node. The "request bus" is shared among all arbiters, and active nodes queue up in the status register. All the status registers are synchronized to shift in a round robin way and the head node will always be served. However, in such distributed bus, wires for traffic priority delivering cannot be shared, leading to even as many connections as the central controlled bus.

PDDVB is capable of providing both round robin service and differential service while avoiding starvation. The priority covering based arbitration makes it TSV cost efficient. To give an intuitional comparison, TABLE 4 lists the TSV cost of PDDVB and several previous high performance arbitration policies, where the differential service is supposed to be supported and the traffic priority level is supposed to be 8. It is found that PDDVB consumes least TSVs among the involved bus arbitration policies.

Arbitration policy also determines the arbiter area cost. We have implemented PDDVB with 4 and 8 nodes, respectively.

TABLE 4
TSV COUNT COMPARISON

Arbitration Policy	TSV count for arbitrating (n layer)	8 layer example
D-TDMA[6]	$(3n+\log_2 n+3)(n-1)$ bits	210
D-TDMA[10]	$5(n-1)$ bits	35
Fake Token[8]	$n+3n$ bits	32
PDDVB	$2(n-1)$ bits	14

TABLE 5
ARBITER COST COMPARISON IN CYCLONE III FPGA IMPLEMENTATION

Arbiter	Nodes	Logic Elements/Arbiter
SP	4	281
TDMA	4	277
RR	4	288
PDDVB	4	62
D-TDMA[12]	8	1163
Fake Token[8]	8	497
PDDVB	8	101

The synthesizing result in Cyclone III FPGA platform is shown in TABLE 5. It is shown that, although identical arbiter is needed in each node, logical elements required by PDDVB are still less than any other arbiters. That is because the node queuing up function is automatically accomplished by the TSV bus. Specifically, resources consumed by the eight nodes PDDVB arbiters are about 69.4% of D-TDMA arbiter and 20.3% of Fake Token arbiter. What is more, PDDVB has appreciable scalability. When it is extended from 4 nodes to 8 nodes, there is only about 62.9% logic element increment.

V. PERFORMANCE EVALUATION IN 3D NOC

PDDVB is designed for 3D bus-NoC hybrid network, aiming at providing round robin service and differential service. In this section, we reconstruct the NoC simulator Noxim [16], forming a 3D bus-NoC hybrid network prototype, which takes PDDVB as the vertical bus and mesh as the lateral interconnection. PDDVB can be configured to PDDVB_RR and PDDVB_D states. Under PDDVB_RR state, all the TPC are set to the lowest, i.e., round robin service is provided. Under the PDDVB_D state, the TPC are set based on equation (12).

$$TPC = \begin{cases} N_p - \dfrac{T_{\max} - T_c}{N_p}, T_c < T_{\max}; \\ N_p, T_c \geq T_{\max} \end{cases} \quad (12)$$

where T_{max} is the expected maximum traffic latency, T_c is the current traffic latency, and N_p is the priority levels of NPC code, which is equals to the vertical layers. That is, PDDVB_D imparts higher priority to the traffic with relatively larger latency, and differential service is provided.

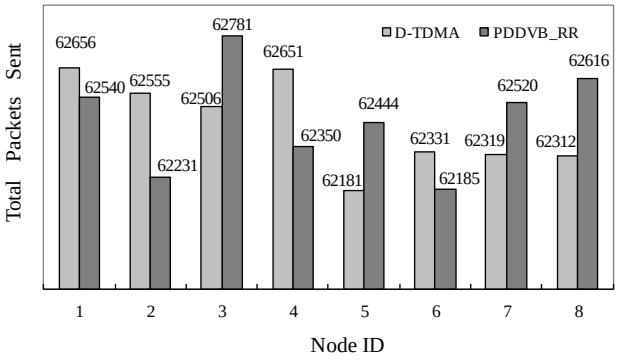


Fig. 7. Round robin service comparison between PDDVB_RR and D-TDMA

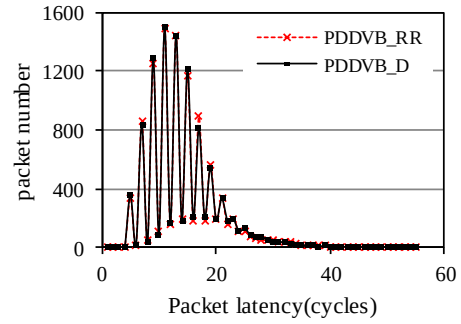
A. Evaluation of PDDVB_RR

PDDVB_RR provides no differentiated service 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 PDDVB_RR configuration. And comparison between eight-node D-TDMA and PDDVB_RR is given in Fig. 7. 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 PDDVB_RR performs as excellent as D-TDMA and almost provide equal service to all nodes. As shown in Fig. 7, the difference among the total packets sent by each node is negligible. And PDDVB_RR shows a very low Relative Standard Deviation (RSD) of 0.281%, a little smaller than D-TDMA, 0.319%.

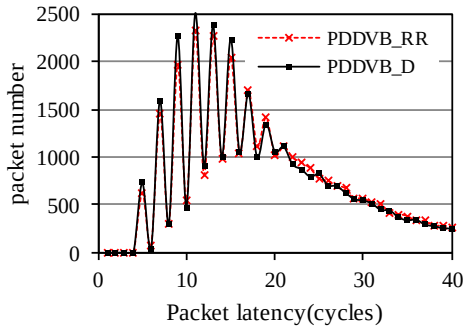
B. Evaluation of PDDVB_D

This experiment is intended to show how PDDVB_D affects the system performance. The network topology is set as 6×6×6 3D hybrid NoC, the packet size is randomly selected as 2 to 8 flits, the buffer size is set as 4 flits, and the traffic pattern is set as random. Routing algorithm is selected as last_Z. Packet Injection Rate (PIR) is set as 0.006, 0.02 and 0.028, each of which gives a low, medium and heavy traffic load. T_{max} is set as 20 cycles, 80 cycles and 150 cycles, respectively. Traffic latency distribution of the 3D hybrid NoC with PDDVB_D state and PDDVB_RR state will be compared. The results are shown in Fig. 8.

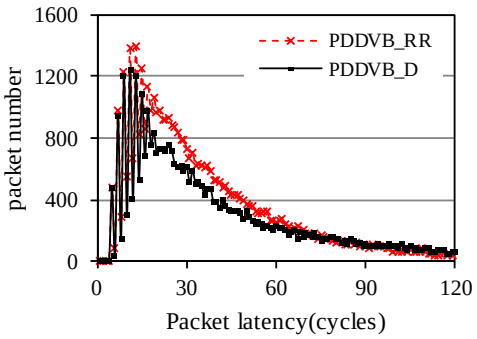
It is shown that, under the low traffic load, bus bandwidth competence is not so drastic, therefore, to increase the priority of the traffic with high latency seems to be of no effect to the system performance and the traffic latency distribution remains the same under the two situations, as shown in Fig. 8(a). Under the medium traffic load, PDDVB with differential service mode is beneficial to increase the packets with low latency, as shown in Fig. 8(b). Anyway, because there is no differential service in the lateral mesh network, performance improvement is not so prominent. Under the high traffic load, in fact the network nearly saturated, differential service seems doing harm to the latency and PDDVB decreases the packets with low latency, as shown in Fig. 8(c). That also implies that, differential service do not solve congestion problem under heavy traffic load.



(a) PIR = 0.006 packets/node/cycle, T_{max} = 20 cycles



(b) PIR = 0.020 packets/node/cycle, T_{max} = 80 cycles



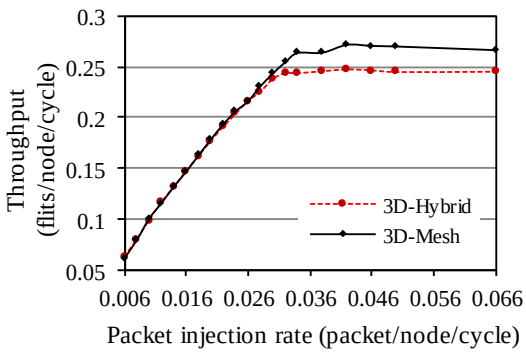
(c) PIR = 0.006 packets/node/cycle, T_{max} = 150 cycles

Fig. 8. Latency distribution comparison of 3D hybrid NoC with PDDVB_RR and PDDVB_D under (a) low traffic load (b) medium traffic load (c) heavy traffic load

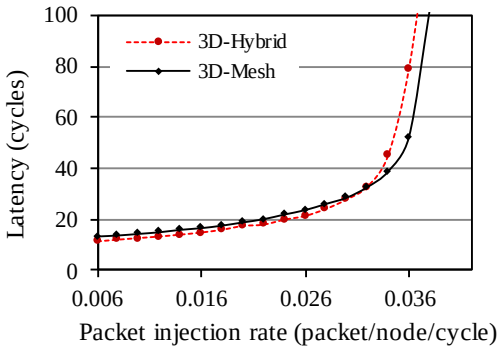
C. *Performance evaluation under different topology size*

This experiment will evaluate the scalability of PDDVB based 3D hybrid NoC. The throughput and average latency will be compared to the one of 3D mesh NoC under 5×5×5, 6×6×6 and 8×8×4 topology size. The bus clock frequency is set same to the one of NoC. All the other settings are kept the same with experiment B. The results are shown in Fig. 9.

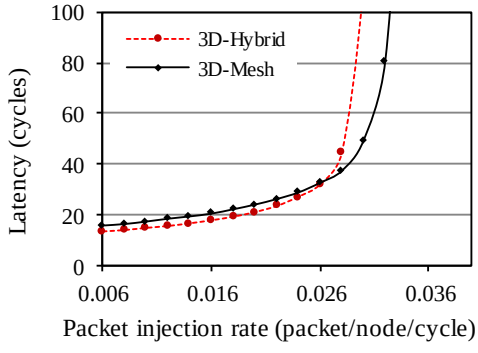
It is shown that, saturation throughput of 3D hybrid NoC is slightly lower than the one of 3D mesh NoC under 5×5×5 topology size, as shown in Fig. 9(a). In fact, under other topology size, the same trend can also be found. That is because, in the vertical direction, the link is unidirectional under 3D hybrid NoC, but bidirectional under 3D mesh NoC. The average latency of 3D hybrid NoC is lower than the one of 3D NoC when the packet injection rate is not more than 0.032 under 5×5×5 topology size, as shown in Fig. 9(b). But when the Network tends to be saturated, because of the vertical bus bandwidth competence, the latency of 3D hybrid NoC



(a)



(b)



(c)

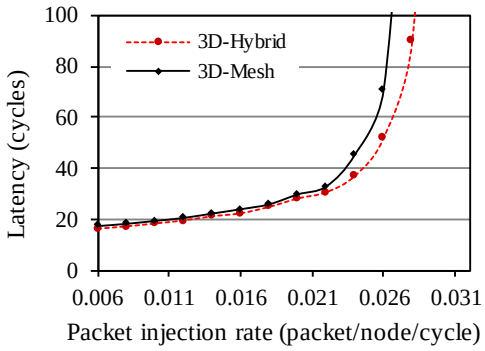


Fig. 9. Throughput and latency comparison of PDDVB based 3D hybrid NoC and 3D mesh NoC (a) throughput comparison under 5×5×5 topology (b) Average latency under 5×5×5 topology (c) Average latency under 6×6×6 (d) Average latency under 8×8×4 topology

increases more quickly than the one of 3D mesh NoC. Same trend can also be found under 6×6×6 topology size, as shown in Fig. 9(c). But when the stacked layers are not more than 4, the latency of 3D hybrid NoC outperforms 3D mesh NoC all the time, as shown in Fig. 9(d). At most 26.6% latency reduction has been found. That is because fewer nodes compete for bus utilization when layer number is less.

VI. Conclusion

In this paper, we extend the previous proposed D-TDMA vertical bus to PDDVB for differential service purpose. Both flit-wise and packet-wise arbitration are supported. Based on the experiments, it is found that if the stacked layers are not more than four, PDDVB based 3D hybrid NoC outperforms 3D mesh NoC in latency, even when the lateral topology size is as large as 8 ×8. It is also found that, under last_z routing, if higher priority levels are assigned to the packets with larger latency, PDDVB based 3D hybrid network can somewhat increase the packets with small latency under the medium traffic load. In the future work, we will further incorporate differential service in the lateral network, in order to optimize the latency distribution of the 3D hybrid networks.

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 , Sept. 2001, pp.147-151.

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

[4] T. Zhang, N. Wu, F. Zhou, et al., "A Traffic Equilibrium Mapping Method with Energy Minimization for 3D NoC-Bus Mesh Architecture", IAENG International Journal of Computer Science, vol.42, no.1, pp. 1-7, Feb. 2015.

[5] H. Matsutani, and M. Koibuchi, "Tightly-coupled multi-layer topologies for 3-D NoCs", in Proc. International Conference on Parallel Processing, Xi'an, Sept. 2007, pp. 75-85.

[6] 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, 2006, pp. 130-141.

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

[8] L. Zhou, N. Wu, X. Chen et al., "A Design Methodology for Three-Dimensional Hybrid NoC-Bus Architecture", IEICE Transactions of Electronic, vol. E96-C, no. 4, pp.492-500, Apr. 2013.

[9] 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.

[10] 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.

[11] 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, Mar.,2011, pp. 1-6.

[12] 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, Jan. 2006, pp.3-7.

[13] G. Yan, N. Wu, L. Zhou, F. Ge et al., "DDTDMA: A Distributed Dynamic TDMA Bus for 3D Bus-NoC Hybrid Network" , Proceedings of The World Congress on Engineering and Computer Science, San Francisco, USA, Oct. 2015, pp. 1-6.

[14] ITRS. "International Technology Roadmap for Semiconductors". Avaible at http://www.itrs.net, .2013

[15] K. Lahiri, A. Raghunathan, and G. Lakshminarayana, "LOTTERYBUS: a new high-performance communication architecture for system-on-chip designs", in Proceedings of the Design Automation Conference (DAC), 2001, pp. 15–20.

[16] Noxim: Network-on-Chip Simulator, Avaible at http://sourceforge.net/ projects/noxim/, 2013