JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL.15, NO.1, FEBRUARY, 2015 http://dx.doi.org/10.5573/JSTS.2015.15.1.085

Hybrid Test Data Transportation Scheme for Advanced NoC-Based SoCs

M. Adil Ansari, Dooyoung Kim, Jihun Jung, and Sungju Park

***Abstract*—Network-on-chip (NoC) has evolved to overcome the issues of traditional bus-based on-chip interconnect. In NoC-reuse as TAM, the test schedulers are constrained with the topological position of cores and test access points, which may negatively affect the test time. This paper presents a scalable hybrid test data transportation scheme that allows to simultaneously test multiple heterogeneous cores of NoC-based SoCs, while reusing NoC as TAM. In the proposed test scheme, single test stimuli set of multiple CUTs is embedded into each flit of the test stimuli packets and those packets are multicast to the targeted CUTs. However, the test response packets of each CUT are unicast towards the tester. To reduce network load, a flit is filled with maximum possible test response sets before unicasting towards the tester. With the aid of Verilog and analytical simulations, the proposed scheme is proved effective and the results are compared with some recent techniques.**

***Index Terms*—Packet multicast, network on chip, scan test**

I. INTRODUCTION

Network-on-chip (NoC) resolves several issues of traditional bus-based on-chip interconnect, like scalability, contention, power issues, predictability and performance limitations [1]. The NoCs (especially with advanced features like virtual circuits and collective communication services) offer several benefits over bus-based on-chip interconnects, those make it an attractive solution for future SOCs, like 1600 cores [2] and more. It is likely that wire cost will dominate over the transistor costs in future [3], which encourages to share the physical links by multiple logical interfaces. NoCs support virtual channel-based communication, which not only reduces the probability of network deadlocks but also increases the performance by extensively using the links. This in turn optimizes the number of wires between the routers [4, 5].

Furthermore, for SoCs with multiple processors, collective communication services like multicast and broadcast are required [6]. Multicasting is helpful for the data-parallel programming model, cache coherent shared memory, real-time object recognition [4] and in certain applications that involve distributed decision-making algorithms [7]. In multicasting, the replication of packets occurs at optimal bifurcation points rather than full end-to-end replication, which significantly reduces network contention and resource requirement [8].

Although, a few NoC architectures offer collective communication and virtual channel-based communication [8-10], for process-intensive applications, the dense SoCs of the future may necessitate NoC architectures with such features.

Denser SoCs and deep submicron technologies pose several challenges for integration engineers and design-for-testability (DFT) engineers. In order to minimize the test cost, several researches have been published to reuse the available functional interconnects as test access mechanisms (TAMs), rather than producing a dedicated TAM. A TAM is responsible for transporting test data between the automatic-test-equipment (ATE) and core-under-test (CUT). It is comprised of a test access port

Manuscript received May. 18, 2014; accepted Jan. 5, 2015
Hanyang University
E-mail : maa1784@gmail.com

and a number of wires that physically bridge the access ports with the CUTs. NoC has also been used as TAM to test the attached cores [1, 3, 11-19]. In packet based NoC, communication is carried out via packets; therefore, in order to reuse it as a TAM, while complying with the network protocol, the test data needs to be transported in the form of packets. Moreover, it appends the constraint of topological positions of cores and access points (APs) into the test scheduling algorithms. Scheduling at high level of abstraction i.e., at packet or flit level, offers low controllability of test data paths [3]; however, the bus wires can be grouped and separately controlled. Therefore, test scheduling of NoC reuse results in relatively higher test time as compared to that of bus reuse, which is demonstrated by Fu et al. [15] by comparing the test methods of conventional bus reuse as TAM [20-24] with that of NoC reuse [12, 13, 18, 19].

Being motivated with the above review, both [15] and [3] presented test methods those embed DFT into the network routers to break the protocol constraint. With those configurations, the NoC communication hardware acts as traditional SoC TAM; however, the schedulers are constrained with the topological position of APs and CUTs.

This paper presents a hybrid test data transportation scheme for heterogeneous cores, connected through an on-chip network. It reuses the NoC as TAM in such a way that the traditional bus reuse test scheduling algorithms can be implemented. It simultaneously examines multiple heterogeneous IP cores while complying with the network protocol and does not require to modify any network component. The proposed scheme uses multicasting and unicasting for test stimuli packets and test response packets, respectively. Based on the outcome of scheduling algorithm, each flit of test stimuli packets is partitioned and each partition is assigned to a core (refer Fig. 1). In this paper, we term the set of a single stimuli (response) bit to (from) each scan chain of the corresponding CUT as a test stimuli (response) set. The tester embeds a single test stimuli set of multiple CUTs into each flit. The packets of such flits are multicast to the corresponding CUTs and the wrapper of each targeted CUT samples only its assigned portion of incoming packet flits. However, the test response packets are unicasted to the ATE-sink, having a test response set embedded in the allocated portion of the flit.

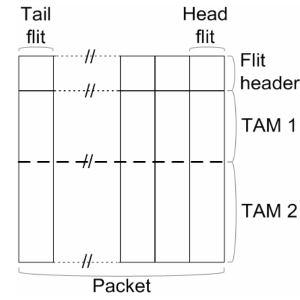

Fig. 1. Packet format and partitioned flits.

Further, we proposed to accumulate multiple test response sets of a single CUT into each response packet flit, if possible, before unicasting to ATE-sink. The major contribution of this work is:

- The proposed scheme for the test data transportation offers flexibility in selecting test scheduler without constraining the topological positions of APs and CUTs into the test scheduler, unlike [3, 15-17].
- The network load is reduced by minimizing the idle bits transportation by accumulating (1) a single test stimuli set of multiple CUTs into stimuli flits and (2) multiple test response sets of a single CUT into each response flit, if feasible. However, in previous work, the remaining bits of the flits are transmitted blank [12, 17].

The experimental results show that the bus-based scheduling algorithms can be implemented without modifying network components and this scheme is scalable with respect to the number of cores.

The rest of paper is organized as follows. In the following section, a review of the related work is presented. Section 0 describes the NoC-based test platform for the proposed scheme, and the proposed test data transportation scheme is detailed in Section 0. In Section 0-4, collisions between different types of packets is addressed and the remedy is discussed. Experimental work and conclusion are presented in Sections 0 and 0, respectively.

II. RELATED WORK AND LIMITATIONS

Several effective test architectures and scheduling algorithms have been proposed to effectively utilize the on-chip bus-based interconnects as TAM [25-30]. Unfortunately, they are not effective for NoC-based

chips due to constraining the protocol and the topological position of cores. Keeping in view the future demand of on-chip network interconnect, several test schemes have been presented for NoC-based SoCs, which use NoC as TAM [1, 3, 11-19]. All of these methods attempt to achieve test parallelism.

In order to minimize the test application time of SoCs, the test schedulers are desired to minimize the idle time of TAM wires or individual TAM wire. This may results with an assignment of a narrower TAM width to a core as compared to the total available TAM width. For specific core, the TAM width refers to the number of scan chains. The bus wires of a bus-based on-chip interconnect can be partitioned and separately controlled to implement the outcomes of such test schedulers [25]. However, for a packet based NoC, the network links cannot be partitioned and separately controlled unless the routers are modified for test mode like [15] and [3].

Liu et al. [12] presented a test technique for NoC-based SoCs in which the network channel width is divided (not physically) among the cores that require a narrower TAM width than the available network channel width. The cores with different TAM widths are tested using different input output-access points (IO-APs) between the device-under-test (DUT) and the ATE, as in [12, 17]. In [12], particular cores are tested by selectively using a faster clock; however, the power hungry cores are tested with a slower test clock to limit the test power while the NoC clock frequency remains constant. This composition enables test data delivery by time division multiplexing (TDM) in which, during different time-windows or time-slots (TSs), the test data is delivered to different CUTs through corresponding input-APs.

However, the test method proposed by Ahn et al. [13] also used TDM for test stimuli delivery but they made use of a single IO-AP, unlike [12]. In [13], TSs are achieved by using a faster clock for NoC than that for CUTs. During each TS, a packet carries multiple test vectors (TVs) of the targeted CUT, if the packet flit size is multiple times larger than the TAM width of the corresponding CUT. Further, a test scheduling algorithm is also presented, based on the use of multiple test clocks in the rectangle packing method [21].

Richter et al. [17] co-optimize core test scheduling and the number of pins assigned to multiple IO-APs. Furthermore, a multiple-input-signature-register (MISR) is incorporated for test response compaction, which impedes debugging. Their proposed algorithms are constrained with topological position of APs and CUTs, which may result in additional test time.

Fu et al. [15] presented a DFT for routers, which breaks the network protocol constraint during test mode. This allows to route the network channel wires in groups of wires in different directions, unlike the normal routing mechanism. For homogeneous cores, test stimuli data is multicast, but for heterogeneous cores it is delivered by sub-grouping the wires of the output ports of the routers, which allows parallel testing.

Sbiai et al. [3] presented a similar approach in which the scheduling unit is a wire, whereas in [15] it is a group of wires. Both methods, however, require a DFT part to be embedded into each router, [3] offers finer granularity than [15]. This allows to implement almost all test scheduling algorithms on NoC based systems, but they require to append a constraint of topological position of each core, which may slightly increase the test time. Moreover, although, the DFT area overhead in [3] and [15] is small and can be feasible for small-scale NoC-based chips, but it may be significant for dense SoCs.

In [16], test data is multicast to heterogeneous cores but all cores are first combined to generate a test set. D. Xiang et al. [16] assume that, in a dense SoC, multiple instances of some cores could be used. Thus, authors proposed to generate the TVs after merging a single instance of each core into a single circuit, which results in a compact test set. Hence, each TV may cover some faults of all cores; therefore, the same test packets are multicast to multiple cores. However, with hard cores this mechanism may not be viable.

On the contrary, the proposed work allows multiple heterogeneous hard cores to be tested in parallel with a single IO-AP, while exploiting the multicasting feature of the advanced NoCs. Moreover, it allows flexibility in choosing scheduling algorithm (even bus-reuse scheduling algorithms), while complying with the network protocol. The virtual channel feature may further improve the performance of the proposed scheme; however, it is not addressed in this work.

III. NoC-Based Test Platform

This section describes the NoC-based test platform for

the proposed test scheme. Since the proposed scheme does not address NoC infrastructure testing, it is assumed that the NoC components have already been tested. The proposed test scheme is applicable for all topologies and switching schemes, with any deadlock free routing algorithm like XY-routing. XY-routing is a deterministic and deadlock free approach because the packets traverse in the X-direction and then in the Y-direction to reach their destinations.

For demonstrating the proposed scheme, a 2-D mesh topology with wormhole switching and XY-routing algorithm was assumed. 2-D mesh topology is one of the most practical and extensively used network topologies [31-35] because of its regularity, simple addressing scheme, and multiple source-destination routes [31]. In this mode, packets are divided into flits, which are the smallest unit of flow control and are equal to the network channel width, excluding flit headers, as shown in Fig. 1. In this paper, the terms network channel width and flit size are used to express the data payload bits of a flit, excluding header bits. As described in Section I, advanced NoCs offer some advanced features, and this work attempts to exploits multicasting feature for test data transportation through the network. To implement the proposed test scheme, it is assumed that the NoC architecture supports multicasting. The router architecture presented by Samman et al. [36] supports multicasting with a latency of two cycles per flit, which we considered for our experiments.

Furthermore, for NI, credit-based communication is preferred, which allows a latency of a single cycle [37]. Credit-based communication enables data forwarding only when there is a free data space in the destination buffer. If there is no data space at the destination then the data will not be transmitted, which enables other nodes to utilize communication resources. Moreover, it provides test data streaming without data loss. For our experiments, we considered latencies of one cycle for NI.

The packet based network can outperform circuit based network at high injection rate, if the packet size is less than 62 flits [38]. Therefore, we assumed the packet payload of 20 flits.

Experiments were performed on ITC'02 benchmark circuits [39]. Since no information is provided for topological positions of the cores, for experiment we mapped them in such a way that the dominant cores (in terms of test time) were at the farthest location from the IO-AP. However, with an assumption of flexibility in placing the IO-AP, we propose to select its position in such a way that the dominating cores are closer to IO-AP.

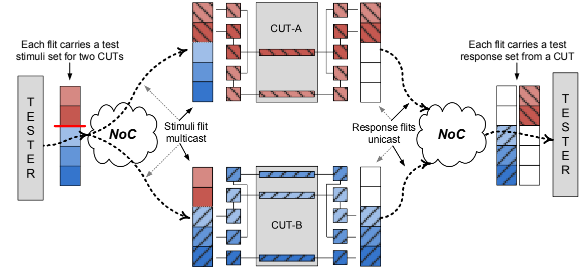

Fig. 2. An example of the proposed test data transportation scheme.

IV. Hybrid Test Data Transportation Scheme

The proposed scheme maximally utilizes the network bandwidth and provides flexibility in selecting test scheduling algorithm without constraining topological positions of APs and CUTs. Fig. 2 depicts an example of the proposed scheme. It multicasts the test stimuli packets to multiple CUTs and unicasts the test response packets to the ATE-sink, through NoC. We propose to partition each flit of test stimuli packets into multiple portions and allocate each flit portion to a particular core or a group of cores. The flit partitioning is performed based on the outcome of the preferred test scheduler.

For demonstrating the proposed scheme, we used rectangle packing test scheduling algorithm [21], which co-optimizes the wrapper and the TAM. This scheduler partitions the total TAM width into multiple TAM groups, each TAM group has a specific TAM width. All cores to be tested are distributed to each TAM group in such a way that none of the cores is assigned to more than one TAM group. For NoC as TAM, the total TAM width, the number of TAM groups and the TAM width of each group are analogous to the flit size, number of flit portions and width of each flit portion, respectively, as depicted in Fig. 1.

1. Test Stimuli Delivery

We consider the ATE channel width or TAM width of a device same as the on-chip network channel width,

which is distributed among the cores by the test scheduler. In fact, this distribution is the partitioning of flit and each flit portion is allocated to the corresponding core. ATE sends test data to the connected NI, through the IO-AP. The test stimuli packets are generated in such a way that a single test stimuli set of each CUT is embedded in the allocated bit positions of each of its flit. Subsequently, the NI multicasts them to the corresponding CUTs, as shown in Fig. 2.

2. Test Response Collection

Similarly, each CUT places their test response data on to the corresponding flit portion, which is unicasted to the ATE-sink. The allocated number of bits to a specific CUT may be less than the flit size; therefore, the remaining bits are filled with idle bits, as shown in Fig. 2. The filling of these idle bits can be performed aiming to minimize the energy consumption for test response transportation.

In order to minimize the transportation of these idle bits, we propose to embed maximum number of test response sets into each flit, instead of embedding only one test response set. Though accumulation of multiple test response sets into each flit requires additional clock cycles, it does not effect the test data streaming or test time; however, this accumulation reduces the network load, which in turn reduce energy consumption.

To transport the entire test response data, the required number of flits can be achieved by the product of the length of the longest scan chain and the number of TVs. If the allocated TAM width of a CUT is multiple times narrower than the flit size then the number of accumulated test response sets into a single flit can be achieved by $\left\lfloor \frac{\text{Flit size}}{\text{TAM width of a CUT}} \right\rfloor$. With this accumulation, the quantity of test response flits reduce to:

$$\left\lceil \frac{\text{Length of the longest scan chain}}{\left\lfloor \frac{\text{Flit size}}{\text{TAM width of a CUT}} \right\rfloor} \right\rceil \times \text{No. of TVs}.$$

For example, let us consider the case shown in Fig. 2. The flit size is 5 bits, 2 bits are assigned to CUT-A, which means it has two scan-chains. Let the length of the longest scan-chain be 10 scan cells, and there is one TV. Therefore, traditionally, 10 flits transport the test response data, but after maximally loading the test response flits, as proposed, $\left\lceil \frac{10}{\left\lfloor \frac{5}{2} \right\rfloor} \right\rceil \times 1 = 5$ flits perform the same job.

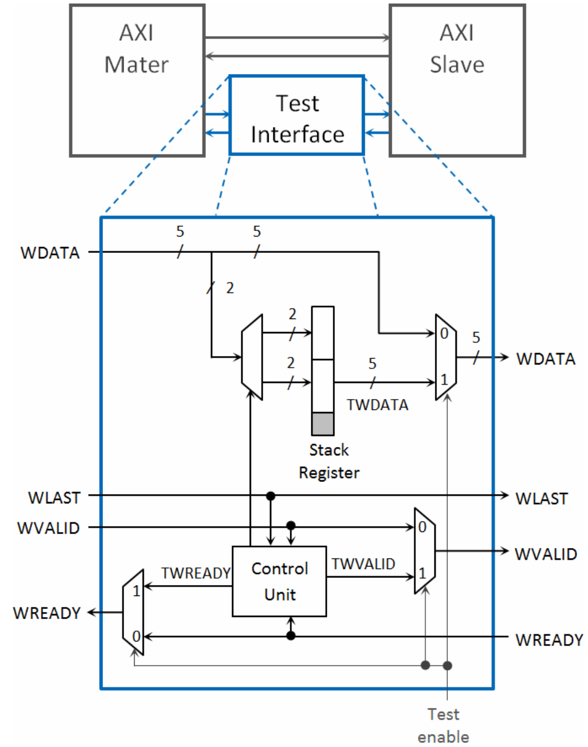

Fig. 3. Proposed interface for loading maximum test response sets before packetization.

3. Test Response Accumulator

In NoC-based S oCs, the cores communicate with NI through a protocol like Advanced eXtensible Interface (AXI) and Open Core Protocol (OCP) [40], and to load the maximum number of test response bits we worked with these interfaces. However, the packetizer of a specific NI can be modified instead. We assume that IP cores use the AXI protocol [41] to communicate with the NI, and we embed an AXI-compliant interface, as shown in Fig. 3. A similar design can be followed for other protocols as well.

This mechanism is valid if the CUT is allocated narrower (at least half) TAM width as compared to the flit size. The test response sets are accumulated into a register, before forwarding to the AXI-slave of the NI. We call it the stack-register.

The proposed test interface samples specific data bits of AXI-master (CUT side), and accumulates them into the stack-register. Subsequently, it forwards the stacked

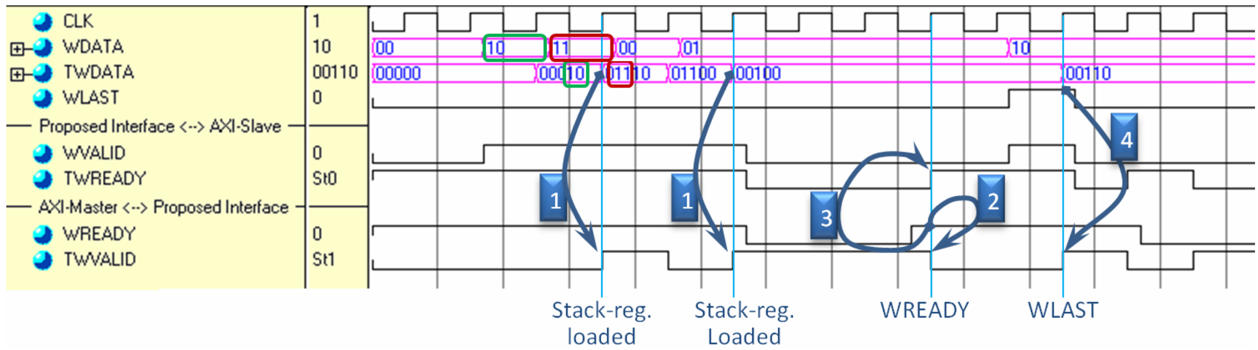

Fig. 4. Simulation results of the proposed interface to accumulate maximum test response sets.

data to the AXI-slave (NI side). Further, the NI and routers follow their usual protocols.

AXI is a burst-based protocol that offers three types of bursts – fixed-address burst, incrementing-address burst, and incrementing-address wrap burst. To transfer data to the same address, the fixed-address burst is used. The proposed interface monitors only a few AXI-signals, which are described in the following paragraphs.

WDATA is a write data bus, and WVALID signal validates the data on the WDATA bus. The WLAST signal indicates the last transfer in a write burst, and WREADY signal indicates that the slave is ready to accept the data.

The control unit of the proposed test interface monitors WVALID and WLAST signals from the AXI-master and WREADY signal from AXI-slave, and generates control signals for the MUXes. The transitory signal/data are termed as TWVALID, TWREADY and TWDATA.

Fig. 3 shows an example of the test interface for CUT-A of Fig. 2. The flit size is 5 bits, and CUT-A is allocated 2-bits to share the test data with the ATE. The Verilog simulations of this test interface is shown in Fig. 4.

The test enable signal selects between transitory and actual signals/data by controlling the MUXes. As soon as the stack register is maximally loaded, the control unit validates the data on the WDATA bus for the AXI-slave by asserting TWVALID signal, as shown by Point 1 in Fig. 4. TWVALID remains high until AXI slave samples the data, which is ensured by the WREADY signal, highlighted by Point 2. WREADY signal remains high if the slave is ready to receive the data; otherwise, it remains low.

The proposed interface acknowledges AXI-master by asserting TWREADY signal. It remains de-asserted unless AXI-slave samples the stack-register data by acknowledging through WREADY signal, as expressed by Point 3. The WLAST signal indicates that the data on WDATA is the last data of the corresponding burst. In the proposed interface, the WLAST signal behaves like an interrupt. As soon as it is made high from the AXI-master, the data stored in the stack-register is sent to the AXI-slave by asserting the TWVALID signal, highlighted by Point 4.

4. Collisions and Their Remedy

Test data streaming is an important factor in achieving better test performance; therefore, collisions among different packets in the network must be avoided. In [14], three types of collisions are addressed those are collisions between (i) test stimuli packets, (ii) test response packets, and (iii) test stimuli and test response packets.

Multicasting with deterministic routing and full duplex communication between routers overwhelms the first and third types of collisions. To avoid second type of collision, [14] has used global combining mechanism [42] in which each router extracts the test response bits from the test response packets of different CUTs and combines them into a single packet before forwarding. This requires a combining mechanism at each router with a larger buffer size. Moreover, each router must be informed about the allocated flit portion of each CUT so that the test response bits can be extracted from the incoming packets and combined into a single packet in the corresponding bit positions. Moreover, the irregularity in the lengths of scan chains can increase complexity at ATE level for comparing test responses.

In the proposed scheme, the collision between test

response packets is prevented due to accumulating multiple test response sets into each test response flit. However, if the test response packet injection rate is higher than that of ejection rate then a faster network frequency (f_N) is required to ensure lossless dilevery of test response packet, as compared to tester frequency (f_T).

An optimum $\frac{f_N}{f_T}$ ratio can be determined by total number of test response flits (NRF) injected into the network during specific duration (number of cycles) of tester clock.

$$\frac{f_N}{f_T} = \left\lceil \frac{NRF}{DoI} \right\rceil \tag{1}$$

The duration of interest (DoI) is the number of tester clock cycles during that at least one test response packet is injected by each CUT, which can be given by (5). The ceiling function in (1) is used to result in an integer value. NRF is the total number of response flits injected into the network for test response transportation.

$$NRF = \sum_n \lceil NRP_n \times packet_size \rceil \tag{2}$$

where NRP is the number of response packets, which are accumulated and periodically injected by a CUT during DoI , that is:

$$NRP_n = \left\lceil \frac{DoI}{IPRP_n} \right\rceil \tag{3}$$

$IPRP$ is the injection period of a response packets after accumulation. It depends upon the number of payload flits, since the header and tail flits to each packet are attached by NI. $IPRP$ is given by:

$$IPRP = [A_n \times (packet_size - 2)] + 2 \tag{4}$$

where an integer "2" is used for a header and a tail flits. A_n is the number of cycles for accumulation of multiple test response sets, at the end of n CUTs, which depends upon the allocated TAM width or flit portion P_n i.e., $A_n = \left\lceil \frac{Flit_size}{P_n} \right\rceil$. Each flit portion is achieved by the outcome of test scheduler.

For DoI , let us consider that the flit is partitioned into n portions i.e. $P_1, P_2, \ldots, P_n$ and each is assigned to n different cores, where n is a positive integer. Hence,

$$DoI = max(IPRP_1,\ IPRP_2\ \ldots\ IPRP_n) \tag{5}$$

Let us consider p93791 (32 cores) benchmark SoC, which is mapped into the assumed NoC. The flit size is 16-bits and the packet size is 4-flits (1-header, 2-payload and 1-tail). The test scheduler [21] partitions 16-bits into four TAM groups those are P_n : 4, 4, 5, 3 bits, by using (1)-(5):

$$A_n : 4,\ 4,\ 3,\ 5$$
$$IPRP_n : 10,\ 10,\ 8,\ 12$$
$$DoI : \max(10,\ 10,\ 8,\ 12) = 12$$
$$NRP_n : 1.2,\ 1.2\ 1.5,\ 1$$
$$NRF = 5 + 5 + 6 + 4 = 20$$
$$\frac{f_N}{f_T} = 2$$

Eqs. (1)-(4) show that the number of payload flits affects the $\frac{f_N}{f_T}$ ratio. For preceding example, the packet with single payload flit, $f_N = 3f_T$ is sufficient for a lossless test response delivery.

V. Experimental Results

The experiments for the proposed method were performed with ITC'02 benchmark SoCs [39] in which all cores were considered as hard cores. Table 1 compares the test results of the proposed method with other methods. First column shows the IDs of benchmark SoCs and the second column shows the ATE channel width used to test corresponding benchmark SoeCs. The ATE channel width is twice the flit size. Succeeding columns show the test results of the proposed method and previous methods, respectively.

Table 1. Comparison with previous methods

SoC IDs	ATE channel width	Test time (cycles)		
		Prop. scheme *(w/ [21])*	[3]	[17] *w/o MISR*
d695	64	21551	24261	22195
	128	11059	12844	12568
g1023	64	16889	---	17947
	128	14828	---	14808
p22810	64	224982	250310	240178
	128	133451	141101	151203
p93791	64	921935	---	912781
	128	458191	---	467441

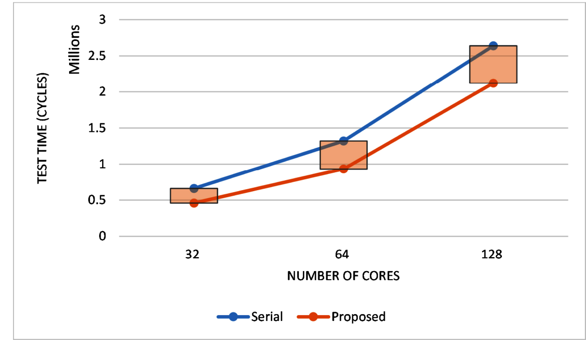

Fig. 5. Test time vs SoC size.

[15] and [3] targeted to implement the test scheduling algorithms of bus based systems into NoC based systems by embedding DFT into each router; however, [3] outperforms between them. In comparison to [3], we have achieved better results up to 13.9% because in [3], the configuration of network routers before test activity consumes several clock cycles. However, there is no such configuration required in the proposed method.

In [17], their test results have been compared with the results of test infrastructures of bus-based SoCs. They concluded that with their test scheduling algorithm, the test data delivery over the NoC does not incur any significant test time overhead. However, the scheduler was run with a flexibility in placing the number of IO-APs. A higher number of available access points increases test parallelism [17]. Distributing the tester pins among higher number of access points increases the network load due to significant idle bits transportation with the test data. The results of our experiments are similar to those of [17]; however, with the proposed test data delivery scheme, an improved scheduling framework will have positive impact on test performance.

Due to unavailability of bigger benchmark SoCs, based on NoC, we mapped multiple copies of p93791 (32 cores) on to NoC in order to demonstrate the scalability of the proposed test scheme. The flit width for this set of experiments is 64-bits. Fig. 5 shows the line chart, which illustrates the test time of 32, 64 and 128 cores. The test results of the proposed scheme competes with serial testing of each core with full flit width. It reveals that the performance of the proposed test scheme scales with the size of NoC-based SoCs.

VI. CONCLUSION

To exploit the multicasting feature of NoC architecture, a scalable hybrid test data transportation scheme is presented, which uses multicasting and unicasting for test data transportation between heterogeneous CUTs and ATE. The proposed scheme allows test engineer to develop or reuse the test scheduling framework without constraining topological position of access points and CUTs, while complying with the network protocol. The experimental results showed that the outcome of bus reuse scheduler could be effectively implemented with our proposed scheme to improve test performance.

ACKNOWLEDGEMENT

This research was supported in part by both the National Research Foundation of Korea (NRF) grant (MEST) (No. NRF-2013R1A1A2059326), and the Higher Education Commission (HEC), Govt. of Pakistan, under the scholarship program titled: Faculty Development of UESTPs/ UETs.

REFERENCES

[1] M. A. Ansari, J. Song, M. Kim and S. Park, "Parallel test method for NoC-based SoCs," *in Proc. IEEE International SoC Design Conference* (ISOCC), 2009.

[2] M. Agrawal, M. Richter and K. Chakrabarty, "A dynamic programming solution for optimizing test delivery in multicore SOCs," *in Proc. IEEE International Test Conference (ITC)*, 2012.

[3] T. SBIAI and K. NAMBA, "NoC Dynamically

Reconfigurable as TAM," in Proc. *IEEE 21st Asian Test Symposium (ATS)*, 2012.

[4] J.-Y. Kim, J. Park, S. Lee, M. Kim, J. Oh and H.-J. Yoo, "A 118.4 GBps Multi-Casting Network-on-Chip with Hierarchical star-ring Combined Topology for Real-Time Object Recognition," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 7, pp. 1399-1409, 2010.

[5] T. BJERREGAARD and S. MAHADEVAN, "A Survey of Research and Practices of Network-on-Chip," *ACM Computing Surveys*, vol. 38, no. 1, pp. 1-51, 2006.

[6] F. A. Samman, T. Hollstein and M. Glesner, "New Theory for Deadlock-Free Multicast Routing in Wormhole-Switched Virtual-Channelless Networks-on-Chip," *IEEE Transaction of Parallel and Distributed Systems*, vol. 22, no. 4, pp. 544-557, 2011.

[7] S. Rodrigo, J. Flich, J. Duato and M. Hummel, "Efficient Unicast and Multicast Support for CMPs," in 41st *IEEE/ACM International Symposium on Microarchitecture*, MICRO-41. 2008, 2008.

[8] S. Yan and B. Lin, "Custom Networks-on-Chip Architectures With Multicast Routing," *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS*, vol. 17, no. 3, pp. 342-355, 2009.

[9] R. Stefan, A. Molnos, A. Ambrose and K. Goossens, "A TDM NoC Supporting QoS, Multicast and Fast Connection Set-Up," in Design, *Automation & Test in Europe Conference & Exhibition (DATE)*, 2012.

[10] F. A. Samman, T. Hollstein and M. Glesner, "Planar Adaptive Network-on-Chip Supporting Deadlock-Free and Efficient Tree-Based Multicast Routing Method," *Microprocessors and Microsystems*, vol. 36, no. 6, pp. 449-461, 2012.

[11] N. John Mark and R. Mahapatra, "A TDM Test Scheduling Method for Network-on-Chip Systems," in Sixth *IEEE International Workshop on Microprocessor Test and Verification MTV '05*, 2005.

[12] C. Liu, V. Iyengar, J. Shi and E. Cota, "Power-Aware Test Scheduling in Network-on-Chip Using Variable-Rate On-Chip Clocking," in Proc. *23rd IEEE VLSI Test Symposium*, 2005.

[13] J. H. Ahn and S. Kang, "Test Scheduling of NoC-Based SoCs Using Multiple Test Clocks," *ETRI Journal*, vol. 28, pp. 475-485, 2006.

[14] J. H. Ahn and S. Kang, "NoC-Based SoC Test Scheduling Using Ant Colony Optimization," *ETRI Journal*, vol. 30, pp. 129-140, 2008.

[15] B. Fu, Y. Han, H. Li and X. Fi, "T2-TAM: Infrastructure Resource to Provide Parallel Testing for NoC based Chip," in Proc. 8th *IEEE International Conference on ASIC*, ASICON '09, 2009.

[16] D. Xiang and Y. Zhang, "Cost-Effective Power-Aware Core Testing in NoCs Based on a New Unicast-Based Multicast Scheme," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 1, pp. 135-147, 2011.

[17] M. Richter and K. Chakrabarty, "Optimization of Test Pin-Count, Test Scheduling, and Test Access for NoC-based Multicore SoCs," *IEEE TRANSACTIONS ON COMPUTERS, SPECIAL ISSUE ON NETWORKS-ON-CHIP*, [in press].

[18] E. Cota, M. Kreutz, C. A. Zeferino, L. Carro, M. Lubaszewski and A. Susin, "The Impact of NoC Reuse on the Testing of Core-based Systems," in Proc. *IEEE 21st VLSI Test Symposium*, 2003.

[19] C. Liu, E. Cota, H. Sharif and D. K. Pradhan, "Test Scheduling for Network-on-Chip with BIST and Precedence Constraints," in Proc. *IEEE International Test Conference*, 2004.

[20] S. K. Geol and E. J. Marinissen, "Effective and Efficient Test Architecture Design for SOCs," in Proc. *IEEE International Test Conference*, 2002.

[21] V. Iyengar, K. Chakrabarty and E. J. Marinissen, "On Using Rectangle Packing for SoC Wrapper-TAM Co-Optimization," in Proc. *IEEE 20th VLSI Test Symposium (VTS 2002)*, 2002.

[22] S. Koranne and V. Iyengar, "On the Use of k - tuples for SoC Test Schedule Representation," in Proc. *IEEE International Test Conference*, 2002.

[23] A. Larsson, E. Larsson, K. Chakrabarty, P. Eles and Z. Peng, "Test-Architecture Optimization and Test Scheduling for SOCs with Core-Level Expansion of Compressed Test Patterns," in Proc. *IEEE Design, Automation and Test in Europe DATE '08*, 2008.

[24] T. Yoneda, M. Imanishi and H. Fujiwara, "An SoC Test Scheduling Algorithm using Reconfigurable Union Wrappers," in Proc. *IEEE Design, Automation*

& Test in Europe Conference & Exhibition DATE '07, 2007.

[25] V. Iyengar, K. Chankrabarty and E. J. Marinissen, "Test Wrapper and Test Access Mechanism Co-Optimization for System-on-Chip," 2001.

[26] Y. Huang, W.-T. Cheng, C.-C. Tsai and N. Mukherjee, "Resource Allocation and Test Scheduling for Concurrent Test of Core-Based SoC Design," *in Proc. IEEE 10th Asian Test Symposium*, 2001.

[27] S. K. Goel and E. J. Marinissen, "A Novel Test Time Reduction Algorithm for Test Architecture Design for Core-Based System Chips," *in Proc. IEEE The Seventh EuropeanTest Workshop*, 2002.

[28] V. Iyengar, K. Chakrabarty and E. J. Marinissen, "Efficient Test Access Mechanism Ooptimization for System-on-Chip," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 5, pp. 635-643, 2003.

[29] V. Iyengar, K. Chakrabarty and E. J. Marinissen, "Test Access Mechanism Optimization, Test Scheduling, and Tester Data Volume Reduction for System-on-Chip," *IEEE Trans. on Computers*, vol. 52, no. 12, pp. 1619-1632, 2003.

[30] S. K. Geol and E. J. Marinissen, "SOC test architecture design for efficient utilization of test bandwidth," *ACM Trans. on Design Automation of Electronic Systems (TODAES)*, vol. 8, no. 4, pp. 399-429, 2003.

[31] W. Zhang, L. Huo, L. Zuo, Z. Peng and W. Wu, "A Network on Chip Architecture and Performance Evaluation," *in Proc. IEEE Second International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, 2010.

[32] T. T. Ye, L. Benini and G. D. Micheli, "Packetization and routing analysis of on-chip multiprocessor networks," *ELSEVIER Journal of Systems Architecture*, vol. 50, no. 2-3, pp. 81-104, 2004.

[33] U. Y. Ogras and R. Marculescu, ""It's a small world after all": NoC performance optimization via long-range link insertion," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 7, pp. 693-706, 2006.

[34] J. H. Bahn, S. E. Lee and N. Bagherzadeh, "On Design and Analysis of a Feasible Network-on-Chip (NoC) Architecture," *in Proc. IEEE Fourth International Conference on Information Technology ITNG '07*, 2007.

[35] M. A. Al Faruque, T. Ebi and J. Henkel, "AdNoC: Runtime Adaptive Network-on-Chip Architecture," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 2, pp. 257-269, 2012.

[36] F. A. Samman, T. Hollstein and M. Glesner, "Multicast Parallel Pipeline Router Architecture for Network-on-Chip," *in Proc. IEEE Design, Automation and Test in Europe DATE '08*, 2008.

[37] B. Attia, W. Chouchene, A. Zitouni, A. Nourdin and R. Tourki, "Design and implementation of low latency network interface for network on chip," *in Proc. IEEE 5th International Design and Test Workshop (IDT)*, 2010.

[38] S. Liu, A. Jantsch and Z. Lu, "Analysis and evaluation of circuit switched NoC and packet switched NoC," *in 16th Euromicro Conference on Digital System Design*, 2013.

[39] E. J. Marinissen, V. Iyenger and K. Chakrabarty, "A set of benchmarks for modular testing of SOCs," *in Proc. IEEE International Test Conference*, 2002.

[40] K. Goossens and J. Dielissen, "Æthereal Network on Chip: Concepts, Architectures and Implementations," *IEEE Design & Test of Computers Journal*, vol. 22, no. 5, pp. 414-421, 2005.

[41] "AMBA AXI specifications," [Online]. Available: www.arm.com .

[42] J. Duato, S. Yalamanchili and L. NI, Interconnect Networks: An Engineering Apprroach, San Francisco, USA: Morgan Kaufmann Publishers, 2003.

[43] L. Benini and D. M. Giovanni, "Networks on Chips: A New SoC Paradigm," *IEEE Computer Journal*, vol. 35, no. 1, pp. 70-78, 2002.

[44] X. Yang, Z. Qing-li, F. Fang-fa, Y. Ming-yan and L. Cheng, "NISAR: An AXI Compliant On-chip NI Architecture Offering Transaction Reordering Processing," *in Proc. 7th IEEE International Conference on ASIC ASICON '07*, 2007.

M. Adil Ansari received the B.E. degree in electronic engineering from Mehran University of Engineering & Technology, Pakistan, in 2006 and the M.S. degree in computer science & engineering from Hanyang University, South Korea, in 2010. Since 2012, he is working toward Ph.D. degree in computer science and engineering at Hanyang University, South Korea. He worked as operations engineer with Pakistan Telecom. Company Ltd., from 2006 to 2008 and he remained as a teaching faculty member in COMSATS Institute and Quaid-e-Awam University, Pakistan, from 2010 to 2012. His research interests include design-for-testability of NoC-based SoCs and 3D-ICs.

Dooyoung Kim received the B.S. and M.S. in computer science and engineering from Hanyang University, South Korea in 2004 and 2006, respectively. From 2006 to 2012, he was with LG Electronics in South Korea as a research engineer in charge of ASIC Front-end. Since 2012, he is working toward the Ph.D. degree in computer science and engineering at Hanyang University. His interests include design for testability, low power test, 3D-IC testing and reliability.

Jihun Jung received the B.S. in computer science and engineering from Hanyang University, South Korea in 2010. Since 2010 he has been working toward the integrated degree of M.S. and Ph.D. in computer science and engineering at the same University. His interests include design for testability, memory test, memory ECC, 3D-IC testing, on-line test and aging monitoring.

Sungju Park received the B.S. degree in electronics from Hanyang University, South Korea, in 1983 and the M.S and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts at Amherst in 1988 and 1992, respectively. From 1983 to 1986, he was with the Gold Star Company in South Korea. From 1992 to 1995, he worked for IBM Microelectronics, Endicott, NY as a Development Staff in charge of boundary scan and LSSD scan design. Since then, he has been a Professor in the department of computer science and engineering in Hanyang University, South Korea. His research interests lie in the area of VLSI testing including scan design, built-in self test, test pattern generation, fault simulation, and synthesis of test. Additional interests include graph theory and design verification. Prof. Park is a member of the Institute of Electronics Engineers of Korea, the Korea Information Science Society, and the Institute of Electronics and Information and Communication Engineers.