

## PAPER

# On NoC Bandwidth Sharing for the Optimization of Area Cost and Test Application Time

Fawnizu Azmadi HUSSIN<sup>†a)</sup>, *Nonmember*, Tomokazu YONEDA<sup>†b)</sup>, *Member*, and Hideo FUJIWARA<sup>†c)</sup>, *Fellow*

**SUMMARY** Current NoC test scheduling methodologies in the literature are based on a dedicated path approach; a physical path through the NoC routers and interconnects are allocated for the transportation of test data from an external tester to a single core during the whole duration of the core test. This approach unnecessarily limits test concurrency of the embedded cores because a physical channel bandwidth is typically larger than the scan rate of any core-under-test. We are proposing a bandwidth sharing approach that divides the physical channel bandwidth into multiple smaller virtual channel bandwidths. The test scheduling is performed under the objective of co-optimizing the wrapper area cost and the resulting test application time using two complementary NoC wrappers. Experimental results showed that the area overhead can be optimized (to an extent) without compromising the test application time. Compared to other NoC scheduling approaches based on dedicated paths, our bandwidth sharing approach can reduce the test application time by up to 75.4%.

**key words:** SoC test scheduling, test wrapper, test access mechanism, NoC-reuse, bandwidth sharing

## 1. Introduction

System-on-Chip (SoC) design offers an integrated and efficient methodology for complex integrated circuits such as those used in consumer products. The rapid increase in design complexity and the short time-to-market pressure accelerates SoC adoption, due mainly to the Intellectual Property (IP) core reuse capability. An SoC consists of three basic building blocks: IP cores, communication interconnects, and external I/O interfaces. In this paper, the term SoC refers to an integrated circuit that uses a Network-on-Chip (NoC) as shared interconnects. A NoC-based SoC example is given in Sect. 2.

NoC is proposed as an advanced interconnect which, through its modularity, separates communication from computation [1] in order to facilitate its adoption in design and to improve scalability. To date, many NoC architectures have been proposed such as SPIN [2], OCTAGON [3], PROTEO [4], CLICHE [5], Æthereal [6], [7], SoCIN [8], SoCBUS [9], xPIPES [10], NOSTRUM [11], QNoC [12], and HERMES [13]; all are based on synchronous communication between nodes. Several other types of NoCs such as CHAIN [14], NEXUS [15], ANoC [16], and MANGO [17]

are based on Globally Asynchronous Locally Synchronous (GALS) communication. The copious NoC architectures highlight the growing interest in NoC as a next generation SoC interconnect.

In the literature, several NoC scheduling methodologies [18]–[20] utilizing the NoC as test data transportation paths from external testers to the CUTs have been proposed. In all these approaches, a dedicated path is established from the NoC input port to the CUT to transport the test vectors; another path is dedicated from the CUT to an output port for test response transportation. Dedicating a physical path to one core means that the path cannot be shared, thus preventing potential test concurrency—a useful tool for test schedule optimization. In addition, the assumption that the test data will be delivered in a timely manner is difficult to justify; there is no guarantee provided other than the dedicated physical path through multiple store-and-forward routers. Hence, the use of standard IEEE 1500 [21] compatible wrapper cannot guarantee uncorrupted data loaded into the scan chains in every scan cycle.

To overcome this shortcoming, the authors in [22] propose a NoC wrapper which takes advantage of the guaranteed bandwidth and latency provided by the NoC to ensure test data integrity. While using the NoC as a TAM, the test data loading time of the NoC wrapper is comparable to the IEEE 1500 wrapper, which requires a more flexible but costly dedicated TAM, as implemented in [23]–[25]. However, the NoC wrapper requires much higher guaranteed bandwidth on the NoC than the actual rate of the test data loaded into the wrapper scan chains. This is further explained in [26] in which two complementary wrapper architectures are proposed in order to overcome the limitations of the NoC wrapper in [22].

In this paper, we propose a NoC scheduling mechanism which utilizes the two types of complementary NoC wrappers for area cost and test application time (TAT) co-optimization. The proposed approach takes advantage of the NoC's ability to allocate a specific amount of sustained bandwidth for any particular packet-based connection called a *virtual channel*, making it possible to divide a physical connection for concurrent tests of multiple CUTs. The proposed bandwidth sharing achieves considerable reduction in test time, compared to the dedicated path approaches in [18]–[20].

The rest of the paper is organized as follows: The NoC and IP core models are described in Sect. 2. In Sect. 3, a brief description of the NoC wrapper architecture used in

Manuscript received August 27, 2007.

Manuscript revised January 30, 2008.

<sup>†</sup>The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630-0192 Japan.

a) E-mail: fawniz-h@is.naist.jp

b) E-mail: yoneda@is.naist.jp

c) E-mail: fujiwara@is.naist.jp

DOI: 10.1093/ietisy/e91-d.7.1999

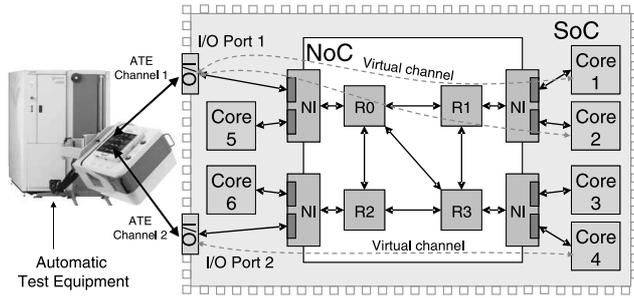


Fig. 1 SoC model based on the Æthereal NoC.

this paper is given. The test schedule and wrapper optimization methodology through bandwidth sharing is explained in Sect. 4. Some experimental results on selected benchmark circuits are given in Sect. 5. Finally, concluding remarks are offered in Sect. 6

## 2. NoC-Based SoC Model

The proposed test architecture utilizes the functional communication channel between a test source/sink and a CUT. Unlike the approaches in [18]–[20], the proposed approach does not restrict to any NoC network topology; it can be applied as long as minimum sustainable bandwidth and latency can be established and guaranteed during the test application of the target CUT. The quality-of-service guarantees ensure that the test data are available at the CUT at the right time. In this paper, the Æthereal [7] NoC, which implements data transfer through normal read/write transactions using the *shared-memory abstraction*, is used as an example in order to ease explanation.

Figure 1 shows a System-on-Chip model that implements an Æthereal NoC consisting of four routers  $R0 - R3$  and network interfaces (NI) as its communication architecture. Among others, the task of the NI is to translate the data format that is passing through. Two of the external ports are labeled *I/O port 1* and *I/O port 2*, which are used in the proposed approach to interface the external ATE ports to the NoC. Two virtual channels (VC) are shown connecting the ATE channel on port 1 to Core 1 and Core 2, respectively. Another VC connects the ATE channel on port 2 to Core 4. Each VC  $vc_k$  is guaranteed a minimum sustained bandwidth  $B_{vc_k}$ , where  $\sum_k B_{vc_k} \leq B_{max}^{i,j}$ . The term  $B_{vc_k}^{i,j}$  and  $B_{max}^{i,j}$  represent the reserved bandwidth for  $vc_k$  and the maximum link bandwidth, respectively, between each pair of routers  $R_i$  and  $R_j$  along the  $vc_k$  path. If  $B_{vc}^{i,j} < B_{max}^{i,j}$  for some link  $R_i \rightarrow R_j$ , the unreserved bandwidth can be allocated to other VCs in order to allow simultaneous test applications of multiple CUTs.

This paper assumes that the NoC in consideration is functionally equipped with such bandwidth allocation scheme. The Æthereal NoC employs a time-slot-based time domain multiplexing (TDM) scheme, where a central arbitrator takes charge of the bandwidth allocation for the whole NoC. Figure 2 shows the conceptual view of the token-ring-

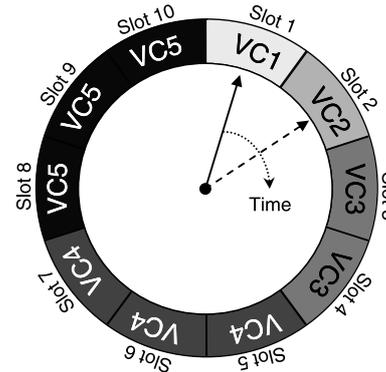


Fig. 2 Bandwidth sharing is supported by the time slot-based TDM scheme implemented by Æthereal NoC.

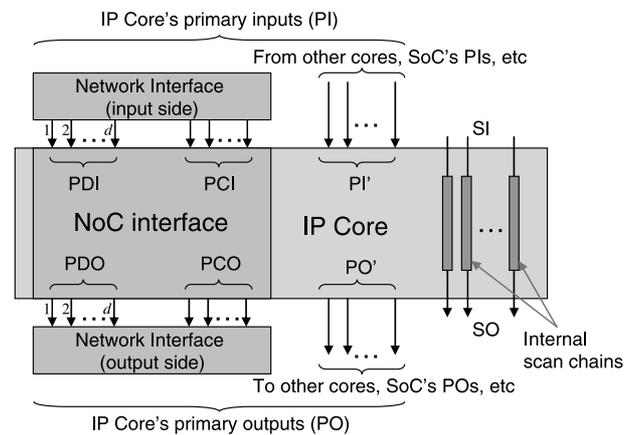
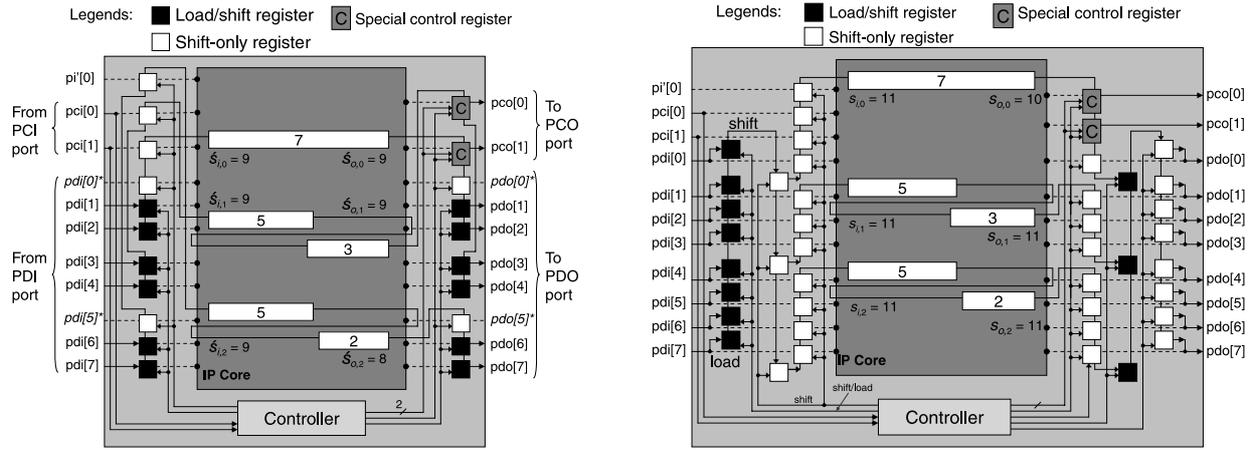


Fig. 3 IP core model interfaced to the NI port.

based TDM time slots. Each globally synchronous router port has an identical set of time slots. As virtual channels are established, sequential slots are reserved on the adjacent routers along the VC path. When connections terminate, slots are freed. The number of slots reserved represents the amount of guaranteed bandwidth reserved. Figure 2 shows five VCs,  $VC_1, VC_2, VC_3, VC_4$ , and  $VC_5$ , with 1 Gbps, 1 Gbps, 2 Gbps, 3 Gbps, and 3 Gbps bandwidths respectively, assuming that the aggregate channel bandwidth is 10 Gbps.

### 2.1 IP Core Model

IP core inputs and outputs (I/Os) shown in Fig. 3 consist of primary inputs (PI), primary outputs (PO), scan inputs (SI) and scan outputs (SO). A subset of the PIs can be categorized into *primary data inputs* (PDI) and *primary control inputs* (PCI), which are connected to the NoC input port. Correspondingly, on the output side, there are *primary data outputs* (PDO) and *primary control outputs* (PCO). The PDIs and PDOs are used to carry the test vectors from the ATE to the CUT, and the test responses from the CUT to the ATE, respectively. The remaining PI/POs (PI' and PO') are connected to other parts of the SoC, which includes other cores,



(a) Type 1 NoC wrapper architecture with scan chains made up of internal scan chains and normal (shift-only) boundary cells.

(b) Type 2 NoC wrapper with an I/O interface which performs parallel-serial shifting to match the NI bit width with the number of wrapper scan chains.

Fig. 4 NoC-reuse wrapper architectures [26].

and the SoC’s primary I/Os.

### 3. NoC Wrapper Architecture

The IEEE 1500 [21] standard wrapper is designed to be used optimally when both the following conditions are true; (i) the TAM wires connected to a core can be assigned individually, and (ii) the timing of wrapper control signals can be controlled individually by an external ATE. When reusing the NoC in the functional mode as a TAM, the number of functional TAM wires is fixed. In addition, the ATE is unable to provide to each core directly the functional control signals during the test application. These restrictions render the standard 1500 wrapper unsuitable for the SoC testing based on the NoC-reuse. In [26], we have proposed two NoC wrappers to address these limitations and showed that the two types of wrappers, with rather opposite characteristics, can be used effectively to prevent unnecessary increase in the test application time of an individual core while also optimizing the area overhead.

The proposed Type 1 wrapper is conceptually illustrated in Fig. 4 (a), where dotted lines and solid lines represent the functional paths and the test data paths, respectively. It uses the same approach as in [23], [24] when forming the wrapper scan chains which minimizes the resulting test application time. In Fig. 4, PDI, PDO, PCI, PCO, and PI’ boundary cells are illustrated. The test data comes in parallel,  $n_{pdi}$  bits per clock cycle, and captured by the data-latching input boundary cells (black squares in Fig. 4). Loading these data into the  $n_{sc} (\neq n_{pdi})$  wrapper scan chains, at the scan frequency ( $f_m$ ), requires parallel-serial shifting. This bit width conversion may result in non-zero PDI bits (i.e.  $(n_{pdi} \bmod n_{sc})$  bits) that cannot be used to carry the test data, in order to avoid data corruption. These results in inefficient utilization of the NoC bandwidth, except when  $(n_{pdi} \bmod n_{sc}) = 0$ . Figure 4(a) shows two input bound-

ary cells (white squares) are not used to capture the test data from the NoC; these dummy data (not real test vectors) must be transferred from the test source to the CUT through the NoC, thereby unnecessarily wasting the NoC bandwidth.

The Type 2 NoC wrapper in Fig. 4(b) is designed to complement the Type 1 wrapper in this aspect. The load/shift registers translate the PDI bit-width,  $n_{pdi}$ , into the number of wrapper scan chains,  $n_{sc}$ , using parallel-serial shift registers similar to [27]. As a result, the required NoC bandwidth matches the scan bandwidth. The TAT for the Type 2 NoC wrapper is also the same as the IEEE 1500 wrapper. This is achieved at the cost of a larger area overhead and a more complex control scheme to realize the bit-width conversion.

### 4. Test Scheduling through Bandwidth Sharing

NoC has been proposed as an advanced SoC interconnect [7], [8], [15], [16] to provide a high bandwidth and modular infrastructure for on-chip communications. As such, in a typical SoC implementation the internal NoC bandwidth is typically larger than the external I/O bandwidth. We define the *internal NoC bandwidth* as the router-to-router and router-to-embedded cores link bandwidth or capacity (in bits-per-second) as shown in Fig. 5. *External I/O bandwidth* is defined as the link bandwidth or capacity from an I/O interface unit to the external devices. Router-to-router bidirectional links are rated at 16 Gbps (i.e. 32-bit wires at 500 MHz for each direction). The external interface through the I/O port is rated half the internal bandwidth at 8 Gbps. Each core is labeled with the corresponding scan rate. For example, core  $C_1$  has 16 wrapper scan chains. When tested at the scan frequency of 100 MHz, it requires the test data at the rate of  $16 \text{ bits} \times 100 \text{ MHz}$ , or 1.6 Gbps.

The test of core  $C_1$  utilizes only a subset of the bandwidth on the I/O port, and between routers  $R1$  and  $R2$ .

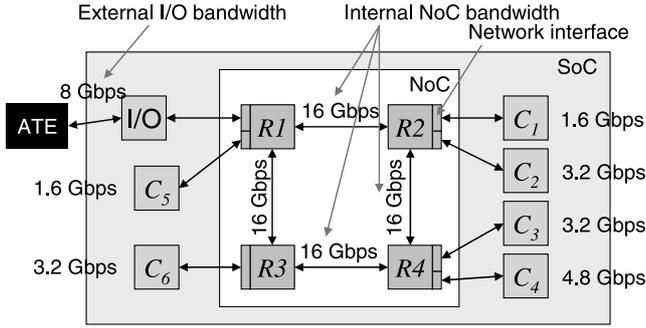


Fig. 5 Illustrative example of the NoC-based SoC model used by the proposed bandwidth sharing approach.

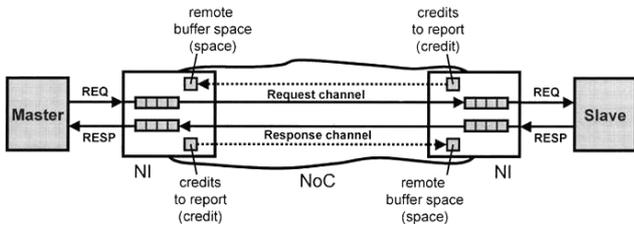


Fig. 6 Buffer-based virtual channels (request and response) between a master and a slave. Illustration by Radulescu et al., 2005 [7].

With the bandwidth sharing approach, we can allow multiple cores to be tested concurrently. For example, simultaneous testing of  $C_1$ ,  $C_3$ , and  $C_6$  requires 8 Gbps on the I/O port, 3.2 Gbps on  $R1 - R3$  link, 4.8 Gbps on  $R1 - R2$  link, and 3.2 Gbps on  $R2 - R4$  link. The shared I/O bandwidth limits further test concurrency. Nevertheless, bandwidth sharing approach allows more efficient use of NoC bandwidth compared to the dedicated path approaches.

The proposed approach is applicable to any NoC architecture that implements the bandwidth reservation scheme, such as the time-domain multiplexing (TDM) scheme implemented by  $\text{\AE}therreal$ . The NoC routers are interfaced to the cores through a buffer-based network interface (NI) architecture, as shown in Fig. 6. Test application can be implemented between the Master (Automatic Test Equipment) and the Slave (Core Under Test).

Because of the guaranteed bandwidth, the incoming data buffer at the core is always non-empty. At the core wrapper (Fig. 4 (a) and 4 (b)), new data availability is signaled by the  $\text{pci}[0]$  and  $\text{pci}[1]$  control signals; depending on the write transaction protocol, the signals could be  $\text{DATA\_STROBE}$ ,  $\text{DATA\_VALID}$ , etc. as used by the corresponding handshake protocol. These handshake signals are detected by the wrapper *Controller* (Fig. 4), which then generates the necessary sequence of control signals for parallel-serial conversion at the wrapper's inputs and outputs. After the data from the PDI port is shifted into the wrapper scan chains, an acknowledgment signal is generated to enable the network interface (NI in Fig. 6) to deliver the subsequent data.

This buffer-based architecture with credit-based flow

control transforms the bursty packet-switched data in the NoC into a steady stream of data between the Master and the Slave. The buffer architecture also separates the NoC's clock from the Core's clock; therefore, the cores' clocks can be independent from each other. For that reason, the proposed approach can also be applied to multi-clock SoCs.

In this paper, we consider the test application of such SoCs utilizing the external tester as the test source and sink. The ATE ports are connected to the SoC through these low bandwidth I/O ports, as illustrated in Fig. 1 and Fig. 5. The test data are transferred into the chip through the functional write transactions. We will assume that a virtual channel can always be established from the I/O port to the target CUT as long as  $\sum \{\text{virtual channel bandwidth}\} \leq \{\text{I/O bandwidth}\} \leq \{\text{internal NoC bandwidth}\}$ . Under this assumption, the wrapper area and test time co-optimization problem addressed in this paper can be formulated as an I/O bandwidth distribution and core test scheduling problem as follows:

$\Psi_S$ : Given an SoC  $C$  with  $M$  cores, a maximum I/O bandwidth,  $B_{max}^{i/o}$  bps, and a scan frequency for all cores,  $f_m$ , where each core consists of  $n_{ip}$  functional inputs,  $n_{op}$  functional outputs,  $n_{bi}$  bidirectionals,  $k$  internal scan chains of length  $l_1, l_2, \dots, l_k$ , for each core  $c_i \in C$  determine

- (1) the wrapper type and the allocated I/O bandwidth,  $B_{scheduled}[c_i]$ , for the test data transportation, and
- (2) the starting time,  $t_{start}[c_i]$ , and end time,  $t_{end}[c_i]$ , of the test application

such that the total test application time and the area overhead are optimized under given priority weights  $\alpha$  and  $\beta$ , respectively, where  $\{\alpha, \beta\} \in [0, 1]$  and  $\alpha + \beta = 1$ .

Before explaining the schedule optimization algorithm (Sect. 4.3), we first clarify two required components of the algorithm in Sects. 4.1 and 4.2.

#### 4.1 Optimum Wrapper under Bandwidth Constraint

In order to achieve the objective (1) of  $\Psi_S$ , we first defined, in [26], the problems of optimizing the number of wrapper scan chains ( $n_{sc}$ ) for both the Type 1 and the Type 2 wrappers under given constraints as follows:

$\Psi_B$ : Given a core as in  $\Psi_S$ , a scan frequency,  $f_m$ , and a maximum bandwidth for the virtual channel between the core and the ATE,  $B_{max}^{vc}$  bps, find the number of wrapper scan chains,  $n_{sc}$ , such that (i) the TAT is minimum, (ii) the required bandwidth,  $B_{req} \leq B_{max}^{vc}$ , and (iii)  $n_{sc}$  is minimum subject to objectives (i) and (ii).

$\Psi_T$ : Given a core as in  $\Psi_S$ , a scan frequency,  $f_m$ , and a maximum TAT,  $T_{max}$ , find the number of wrapper scan chains,  $n_{sc}$ , such that (i) the required bandwidth,  $B_{req}$ , is minimum, (ii)  $TAT \leq T_{max}$ , and (iii)  $n_{sc}$  is minimum subject to objectives (i) and (ii).

It was shown in [23] that the TAT of a core is a monotonic decreasing function with regards to increasing number of wrapper scan chains. Therefore, the optimum solution to  $\Psi_B$  can be found in polynomial time, even when an exhaustive search is used. In [26] we implemented a binary search function to find the optimum test application time and the corresponding required bandwidth for both the Type 1 and the Type 2 wrappers. The search result is the Pareto-optimal point (the concept of Pareto-optimal was discussed in [25]) where the corresponding wrapper configurations require a sustained bandwidth,  $B_{req} \leq B_{max}^{vc}$ . A similar search algorithm was also implemented for problem  $\Psi_T$  in [26].

The area overhead of the wrappers is contributed mainly by the quantity of the boundary cells. We will assume that the area overhead due to the wrapper controller is comparable for both wrappers, therefore will not be used when deciding the wrapper type. The area overhead for Type 1 and Type 2 wrappers can be estimated by Eqs. (1) and (2), respectively. The extra  $(+n_{pdi} + n_{pdo} + 2 \cdot n_{sc})$  in Eq. (2) are due to the additional input/output buffers in the Type 2 wrapper (Fig. 4 (b)) that perform bit-width matching. Equation (3) gives the total cost of using a Type 2 instead of the Type 1 wrapper. Equation (4) gives the opposite cost.

$$H_{t1} = n_{ip} + n_{bi} + n_{op} \tag{1}$$

$$H_{t2} = n_{ip} + n_{bi} + n_{op} + n_{pdi} + n_{pdo} + 2 \cdot n_{sc} \tag{2}$$

$$Cost_{(t1 \rightarrow t2)} = \alpha \cdot \left( \frac{T_{t2} - T_{t1}}{T_{t1}} + \frac{B_{t2} - B_{t1}}{B_{t1}} \right) + \beta \cdot \frac{H_{t2} - H_{t1}}{H_{t1}} \tag{3}$$

$$Cost_{(t2 \rightarrow t1)} = \alpha \cdot \left( \frac{T_{t1} - T_{t2}}{T_{t2}} + \frac{B_{t1} - B_{t2}}{B_{t2}} \right) + \beta \cdot \frac{H_{t1} - H_{t2}}{H_{t2}} \tag{4}$$

For a given maximum bandwidth,  $B_{max}$ , the optimum configuration of a core  $c_i$  is determined by solving  $\Psi_B(c_i, B_{max})$  to obtain the respective TAT ( $T_{t1}$  and  $T_{t2}$ ) and required bandwidth ( $B_{t1}$  and  $B_{t2}$ ) for the Type 1 and the Type 2 wrappers, respectively. If  $Cost_{(t1 \rightarrow t2)} < Cost_{(t2 \rightarrow t1)}$ , then the Type 2 wrapper is selected as a better wrapper configuration for the given  $B_{max}$ . Otherwise, the Type 1 wrapper is chosen. This cost function will be the basis for wrapper selection under given cost weights  $\alpha$  and  $\beta$ , as defined in  $\Psi_S$ .

#### 4.2 Lower Bound on Test Time

The authors in [24] proposed an architecture independent tight lower bound for dedicated TAM based test application, considering both fixed and flexible length internal scan chains. In this section, a similar lower bound based on bandwidth utilization is explained for use in the optimization algorithm. The first lower bound is based on the dominant core effect. For each core  $c_i \in C$ , assuming that it is given

the maximum available bandwidth,  $B_{max}^{i/o}$ , its test time can be determined by  $T(\Psi_B(c_i, B_{max}^{i/o}))$ , which represents the TAT returned by the  $\Psi_B$  search algorithm for Core  $c_i$  when the given maximum bandwidth is  $B_{max}^{i/o}$ . Even with unlimited bandwidth, the TAT of an SoC cannot be shorter than the TAT of the longest core  $c_i \in C$ . Therefore the first lower bound can be written as

$$T_{LB}^1 = \max_{c_i \in C} \{T(\Psi_B(c_i, B_{max}^{i/o}))\} \tag{5}$$

For a bounded  $B_{max}^{i/o}$ ,  $T_{LB}^1$  does not represent a meaningful lower bound. Therefore, a tighter lower bound based on the I/O capacity to transfer test vectors into the SoC is formulated as follows. Assuming that the wrapper for a core  $c_i$  forms one scan chain, its TAT can be represented by Eq. (6) where scan-in depth,  $si = n_{ip} + n_{bi} + \sum_k l_k$ , scan-out depth,  $so = n_{op} + n_{bi} + \sum_k l_k$ , and  $v_m$  is the number of test vectors. The second lower bound can be calculated as in Eq. (7), where  $f_m$  is the scan frequency for all cores. The overall lower bound is the maximum of  $T_{LB}^1$  and  $T_{LB}^2$  (Eq. (8)).

$$T(c_i) = (\max(si, so) + 1) \times v_m + \min(si, so) \tag{6}$$

$$T_{LB}^2 = \sum_{c_i \in C} \{T(c_i)\} / (B_{max}^{i/o} / f_m) \tag{7}$$

$$T_{LB} = \max(T_{LB}^1, T_{LB}^2) \tag{8}$$

#### 4.3 Schedule Optimization through Rectangle Packing

We now introduce the concept of rectangles to represent core tests, then explain a flexible scheduling methodology based on NoC bandwidth sharing, which is inspired by the scheduling algorithm in [25]. The use of rectangles have previously been proposed in [25], [28] for dedicated TAM based scheduling approach. In this paper, the height of a rectangle represents the required NoC bandwidth to obtain the test application time represented by the horizontal length. Figure 7 illustrates two pairs of rectangles, each representing the test of Core 6 of p93791 circuit (ITC'02 benchmark [29]) when  $B_{max} = 2000$  Mbps and 800 Mbps, respectively. For this example, the NoC port's PDI/PDO bit-width is  $n_{pdi} = n_{pdo} = 64$  bits.

The top left rectangle is obtained using the wrapper optimization algorithm  $\Psi_B$  described in Sect. 4.1, when given as input the maximum allocated bandwidth,  $B_{max}^{vc} = 2,000$  Mbps. The algorithm iteratively searches for the

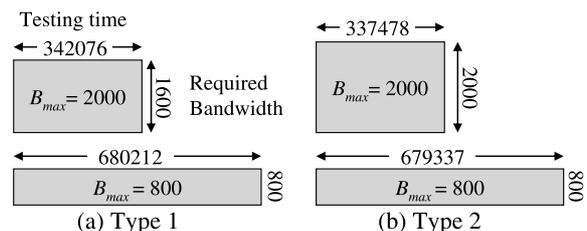
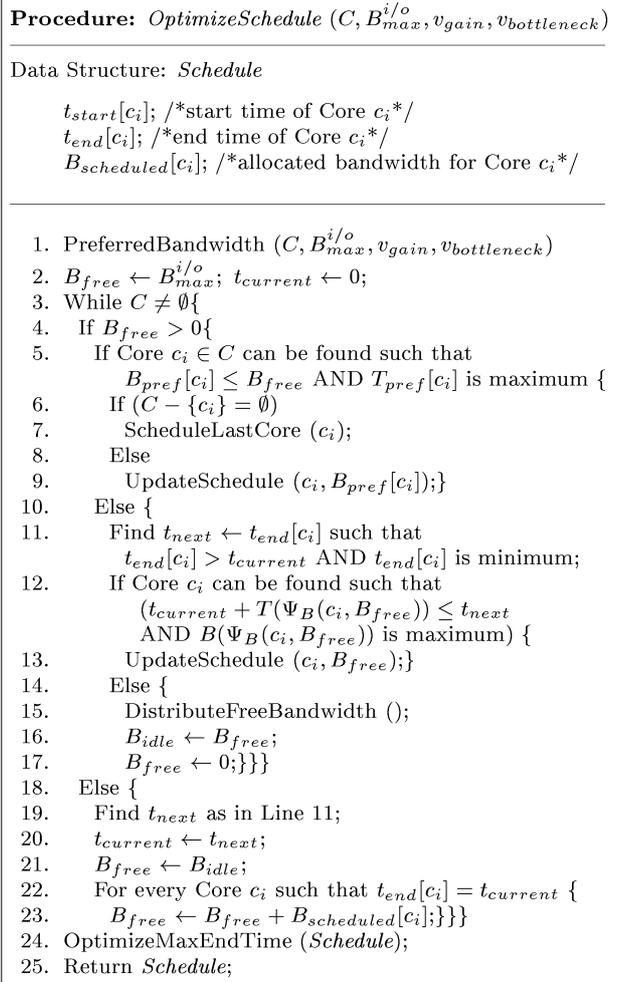


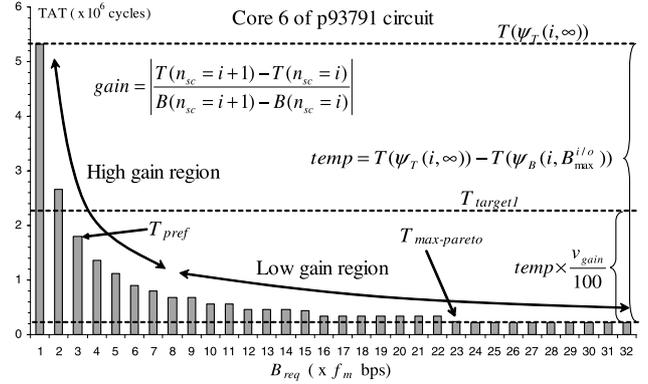
Fig. 7 Rectangles represent tests of Core 6 of p93791 [29] benchmark circuit.



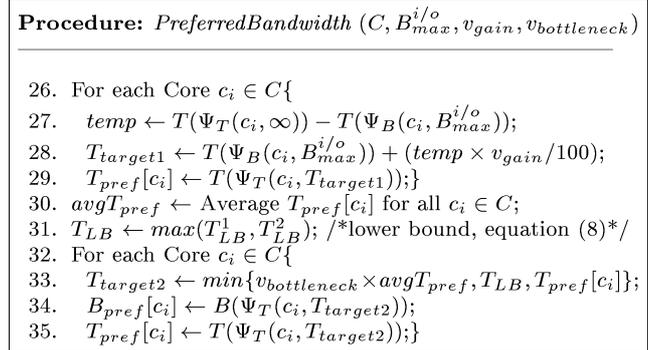
**Fig. 8** Pseudo code algorithm for solving  $\Psi_S$ .

wrapper configuration that produces the smallest test application time, which fulfills the Pareto-optimal criteria, under the bandwidth constraint. Since the Type 1 wrapper cannot effectively utilize all the allocated bandwidth, the algorithm finds the next Pareto-optimal point with a TAT of 342,076 clock cycles which requires 1,600-Mbps NoC bandwidth. The same procedure is repeated for the Type 2 wrapper. With a more efficient bandwidth matching architecture, the Pareto-optimal wrapper is found with a TAT of 337,478 clock cycles and a required bandwidth of 2,000 Mbps (top right rectangle). For  $B_{max}^{vc} = 2,000$  Mbps, these two wrapper configurations are candidates for scheduling.

The complete scheduling algorithm is given in Fig. 8. It starts by obtaining the *preferred bandwidth* for each core in the SoC  $C$  (Fig. 8). As illustrated in Fig. 9, the preferred bandwidth results after configuring the core wrapper with the number of scan chains in the “high gain” region. Gain represents the potential reduction in TAT of a core per additional unit of bandwidth allocated to that core. Therefore, rather than allocating more bandwidth to a core when it is already in the low gain region, it would be wiser to assign that bandwidth to a different core that is still in the high gain



**Fig. 9** High (preferred) and low gain regions.  
 $T_{pref} = \Psi_T(c_i, T_{target1})$ ; line 29 of Fig. 10.



**Fig. 10** Calculating the preferred bandwidth for each core in SoC  $C$ .

region.

Figure 10 describes the algorithm to determine the preferred bandwidth for all cores. In line 28, a proper value of input percent  $v_{gain}$  shifts the target TAT from  $T_{max-pareto}$  to the high gain region. Figure 9 illustrates some of the variables, and show how  $T_{target1}$  is calculated using the variable  $v_{gain}$ . Lines 26-29 are evaluated for both Type 1 and Type 2 wrapper configurations. For every core  $c_i \in C$ , Eqs. (1)-(4) are evaluated to determine the best wrapper type, for which the value of  $T_{pref}[c_i]$  is returned. The same wrapper selection procedure is performed at line 12 when evaluating  $T(\Psi_B(c_i, B_{free}))$ .

In some cases where the test application time is dominated by a large core such as Core 6 of p93791, selecting the high gain region for Core 6 could potentially make it a bottleneck core, thus preventing further reduction of TAT. In order to handle this kind of special cases, we need to be able to allocate as much bandwidth as possible to these potential bottleneck cores. In line 33, the variable  $v_{bottleneck}$  together with the lower bound,  $T_{LB}$  (Eq. (8)), ensures that bottleneck cores are allocated larger preferred bandwidth, even if it is in the low gain region.

The process begins with setting the current time,  $t_{current} = 0$ . During the scheduling process, a core is assigned its preferred bandwidth,  $B_{pref}$ , if the currently unused bandwidth,  $B_{free}$ , at the current time,  $t_{current}$ , is more than

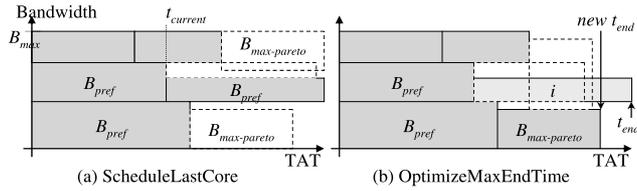
```

Procedure: UpdateSchedule ( $c_i, B_{given}$ )


---


36.  $t_{start}[c_i] \leftarrow t_{current}$ ;
37.  $t_{end}[c_i] \leftarrow t_{current} + T(\Psi_B(c_i, B_{given}))$ ;
38.  $B_{scheduled}[c_i] \leftarrow B(\Psi_B(c_i, B_{given}))$ ;
39.  $B_{free} \leftarrow B_{free} - B_{scheduled}[c_i]$ ;
40.  $C \leftarrow C - \{c_i\}$ ;
    
```

**Fig. 11** Adding core  $c_i$  to the *Schedule*.



**Fig. 12** Further optimizing the schedule. Dotted rectangles represent possible schedule/wrapper configurations.

or equal to  $B_{pref}$  (Line 9). Otherwise, the core  $c_i \in C$  that leaves minimum  $B_{free}$  after it is scheduled, is assigned bandwidth of  $B_{req} = B(\Psi_B(c_i, B_{free})) \leq B_{free}$ , that can be effectively utilized by the core  $c_i$  (Lines 12-13).  $T(\Psi_B(c_i, B_{free}))$  on the other hand, returns the corresponding TAT. Scheduling a new core  $c_i$  involves assigning several variables— $t_{start}[c_i]$ ,  $t_{end}[c_i]$ , and  $B_{scheduled}[c_i]$ —and updating  $B_{free}$  and the list of unscheduled cores,  $C$  (Fig. 11).

When no more cores can be scheduled at  $t_{current}$  while  $B_{free} > 0$  (Lines 14-17), the core  $c_i$ , whose  $t_{start}[c_i] = t_{current}$  and  $t_{end}[c_i]$  is maximum, is allocated the remaining unused bandwidth. This is repeated until either no more  $t_{end}[c_i]$  reduction of such cores is possible or  $B_{free} = 0$ . At lines 18-23, the current time and available bandwidth are updated before the *while* loop is reevaluated.

When scheduling the last core (Line 7), the core start time and assigned bandwidth is chosen such that  $t_{end}$  is minimum. This is illustrated in Fig. 12 (a) where three possible options are shown by the dotted rectangles. After all the cores are scheduled, in the final step (line 24), the current schedule of core  $c_i$  whose  $t_{end}[c_i]$  is maximum, is reconsidered for further optimization. Without modifying the schedule for other cores, core  $c_i$  is rescheduled such that the new  $t_{end}[c_i]$  is minimum (Fig. 12 (b)). This process is repeated until no more reductions can be made to  $t_{end}$ .

## 5. Experimental Results

In this section, we present experimental results for several modified ITC'02 benchmark [29] circuits (d695noc, p93791noc, p22810noc). The wrappers in Fig. 4 utilize the PDI/PDO interface between the core and the NI in its operation. From the design perspective, the cores whose  $n_{ip} + n_{bi} < n_{pdi}$  or  $n_{op} + n_{bi} < n_{pdo}$  cannot be functionally interfaced to the NoC. As a result, two, four, and five small cores are excluded from each of the benchmark circuits when  $n_{pdi} = n_{pdo} = 32$ . In addition, the optimum values (de-

termined iteratively) of  $v_{gain} \in [0..9]$  and  $v_{bottleneck} \in [1..5]$  are used, with the scan frequency,  $f_m = 100 \text{ MHz}$ . The TAT reported in this paper is in number of scan clock cycles, where each cycle is equivalent to  $1/f_m$  or  $0.01 \mu\text{s}$ . The computation time is less than 10 seconds for the largest circuit.

Table 1 tabulates the comparison of Design-For-Testability (DFT) costs between Type 1 and Type 2 wrappers and the SoC benchmark circuits. The circuit size for the ITC'02 benchmark circuits are not given, therefore we estimate the circuit size in terms of the equivalent number of NOT gates for the given number of scan flip-flops (SFF). Each scan cell and wrapper cell is estimated to be equivalent to 24 NOT gates and 31 NOT gates, respectively.

Column labeled Type 1 gives the percent overhead of the Type 1 NoC wrapper (calculated using Eq. (1)) over the SoC circuit. For the largest circuit in the ITC'02 benchmark, the overhead is 10%. For the smaller circuit (d695), the overhead is as high as 37.3%. The Type 1 wrapper cell overhead is the same as the standard IEEE 1500 wrapper overhead.

When we consider the additional area overhead of a Type 2 wrapper (on top of the overhead of Type 1 wrapper), the value ranges between 1.5% to 6.5% (for 32-bit PDI/PDO) and between 2.9% and 13% (for 64-bit PDI/PDO), for the selected circuits. The additional hardware overhead of the Type 2 wrapper is not insignificant; therefore the proposed optimization method is necessary. The calculation is based on a single wrapper scan chain (i.e.  $n_{sc} = 1$ ). The Type 2 hardware overhead would increase slightly for larger number of wrapper scan chains as indicated by Eq. (2).

In Table 2, the weights of hardware overhead cost ( $\beta$ ) and TAT cost ( $\alpha$ ) are varied according to the constraints defined in  $\Psi_S$ . In Table 2 and Table 3 hardware overhead (HOH) is represented by the total number of wrapper boundary cells required for the SoC. Other components of the wrappers such as the controller and the wiring costs are not included because they are similar for both Type 1 and Type 2 wrappers; the boundary cell structures make them unique.

As the cost weight of hardware is increased (increasing  $\beta$ ), the total hardware overhead (columns labeled HOH) decreases while the test application time (columns labeled TAT) increases accordingly. This indicates that as we allow more hardware to be used, more bandwidth-efficient Type 2 wrappers can be used, allowing for a more efficient utilization of bandwidth, hence smaller “rectangles” to pack. Compared to the lower bound defined in Sect. 4.2, the TATs are on average 13% larger. The area overhead can be reduced considerably without affecting the TAT ( $\beta = 0.0$  to  $0.5$ ) for all benchmark circuits. This happens when the Type 1 wrapper is used instead of the Type 2 wrapper for those cores that do not affect the overall TAT.

Table 3 shows the resulting HOH and TAT when  $B_{max}^{i/o}$  varies from  $3.2 \text{ Gbps}$  to  $12.8 \text{ Gbps}$ , with the objective of minimizing the TAT (i.e.  $\alpha = 1, \beta = 0$ ). This illustrates that without increasing the area overhead, the TAT can be re-

**Table 1** Area overhead comparison between Type 1 and Type 2 wrappers.

Benchmark Circuit Characteristics					DFT (NoC-compatible wrappers)					
					Type 1		Type 2 (32-bit PDI)		Type 2 (64-bit PDI)	
Name	#Cores	#I/Os	#SFF	#NOT Gates	#NOT Gates	Hardware Overhead	#NOT Gates	Type 2/ Type 1	#NOT Gates	Type 2/ Type 1
d695noc	10	1,845	6,384	153,216	57,195	37.3%	67,135	6.5%	77,055	13.0%
p22810noc	28	4,283	24,723	593,352	132,773	22.4%	160,605	4.7%	188,381	9.4%
p93791noc	32	6,943	89,973	2,159,352	215,233	10.0%	247,041	1.5%	278,785	2.9%

**Table 2** TAT for several hardware cost ( $\beta$ ) and time cost ( $\alpha$ ) weights.

Cost weights	$\beta$	$\alpha$	p93791noc $B^{i/o} = 6400$ Mbps ( $T_{LB} = 435,039$ )			p22810noc $B^{i/o} = 6400$ Mbps ( $T_{LB} = 102,965$ )			d695noc $B^{i/o} = 3200$ Mbps ( $T_{LB} = 16,701$ )		
			HOH	TAT	%/ $T_{LB}$	HOH	TAT	%/ $T_{LB}$	HOH	TAT	%/ $T_{LB}$
0.00	1.00	9,303	464,252	6.7	5,768	122,091	18.6	2,396	17,827	6.7	
0.25	0.75	8,653	464,252	6.7	5,680	122,280	18.8	2,300	17,827	6.7	
0.50	0.50	7,673	471,175	8.3	5,698	122,280	18.8	2,300	17,827	6.7	
0.75	0.25	7,673	471,175	8.3	5,412	130,591	26.8	2,110	18,184	8.9	
1.00	0.00	6,557	483,411	11.1	3,810	134,466	30.6	1,676	18,494	10.7	

**Table 3** TAT for several  $B_{max}^{i/o}$ . [ $\alpha = 1, \beta = 0$ ].

$B_{max}^{i/o}$ (Mbps)	p93791noc				p22810noc			
	HOH	TAT	$T_{LB}$	%/ $T_{LB}$	HOH	TAT	$T_{LB}$	%/ $T_{LB}$
3,200	8,849	923,842	870,079	6.2	5,584	232,816	203,015	14.7
6,400	9,303	464,252	435,039	6.7	5,768	122,091	102,965	18.6
9,600	9,009	347,378	290,026	19.8	5,798	102,965	102,965	0.0
12,800	8,885	235,285	227,978	3.2	5,798	102,965	102,965	0.0

**Table 4** Test application time of dedicated path (DP) and shared bandwidth (SB) approaches. For SB,  $\alpha = 1, \beta = 0$ 

Channel Bitwidth	d695noc			p93791noc			p22810noc		
	DP	SB	%red.	DP	SB	%red.	DP	SB	%red.
16	49,135	21,768	55.7	1,861,439	907,419	51.3	655,253	229,598	65.0
32	31,317	17,827	43.1	1,211,254	464,252	61.7	510,954	125,591	75.4

duced given larger I/O bandwidth,  $B_{max}^{i/o}$ . This is typically the case because the functional I/O frequency is typically higher than the scan frequency. For the dedicated TAM based approach, TAT reduction can only be achieved by adding TAM wires.

Table 4 compares our bandwidth sharing approach with the dedicated path (DP) approaches [18]–[20]. In the DP approaches, a pair of NoC input and output ports can be used to test only one core at a time. To enable parallel testing, more I/O port pairs are required. Assuming that there is only one I/O port pair, the TAT for DP approach is the sum of each individual core test (sequential testing). Our approach enables parallelism through bandwidth sharing, which proves to be more efficient, with at least 43.1% (when  $\alpha = 1, \beta = 0$ ) smaller TAT.

## 6. Conclusion

We have presented a new approach to NoC testing through bandwidth sharing. The test schedule is optimized using a rectangle packing algorithm by optimally assigning to each core a “high gain” bandwidth—the amount of bandwidth

that gives a high reduction in TAT. The utilization of two complementary NoC wrappers allow for co-optimization of two most important properties—test application time and area overhead.

It was shown experimentally that it is not always necessary to use the expensive Type 2 wrappers in order to obtain a minimum TAT; the low-cost Type 1 wrappers can be used effectively without compromising the overall TAT. We also evaluated the efficiency of the scheduling algorithm; on average the TAT is less than 13% longer than the theoretical lower bound. Compared to the previously published NoC test scheduling based on dedicated path approach, the proposed bandwidth sharing approach reduces the TAT by an average of 58.7% for the selected case studies.

## Acknowledgments

This work was supported in part by Japan Society for the Promotion of Science (JSPS) under Grants-in-Aid for Scientific Research B(No. 15300018) and for Young Scientists (B)(No.18700046). The authors would like to thank Prof. Michiko Inoue, Dr. Satoshi Ohtake and members of Computer Design and Test Laboratory in Nara Institute of Science and Technology for their valuable comments.

## References

- [1] L. Benini and G.D. Micheli, “Networks-on-chips: A new SoC paradigm,” *Computer*, vol.35, no.1, pp.70–80, 2002.
- [2] P. Guerrier and A. Greiner, “A generic architecture for on-chip packet-switched interconnection,” *Proc. Design, Automation and Test in Europe*, pp.250–256, 2000.
- [3] F. Karim, A. Nguyen, S. Dey, and R. Rao, “On-chip communication architecture for OC-768 network processors,” *Proc. Design Automation Conference*, pp.678–683, 2001.
- [4] I. Saastamoinen, D. Siguena-Tortosa, and J. Nurmi, “Interconnect IP node for future System-on-Chip designs,” *Proc. 1st Int’l Workshop on Electronic Design, Test and Applications*, pp.116–122, 2002.
- [5] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, and A. Hemani, “A Network-on-Chip architecture and design methodology,” *Proc. IEEE Computer Society Annual Symposium on VLSI*, pp.105–112, 2002.
- [6] E. Rijpkema, “Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip,” *Proc. Design, Automation and Test in Europe*, pp.350–355, 2003.
- [7] A. Radulescu, J. Dielissen, S.G. Pestana, O.P. Gangwal, E. Rijpkema, P. Wielage, and K. Goossens, “An efficient on-chip NI offering guaranteed services, shared-memory abstraction, and flexible network configuration,” *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol.24, no.1, pp.4–17, Jan. 2005.
- [8] C.A. Zeferino and A.A. Susin, “SoCIN: A parametric and scalable

- Network-on-Chip," Proc. 16th Symposium on Integrated Circuits and Systems Design, pp.169–174, 2003.
- [9] D. Wiklund and D. Liu, "SoCBUS: Switched network on chip for hard real time embedded systems," Proc. Int'l Parallel and Distributed Processing Symposium, p.78, 2003.
- [10] M. Dall'Osso, "xPIPES: A latency insensitive parameterized Network-on-Chip architecture for multi-processors SoCs," Proc. 21st Int'l Conference on Computer Design, pp.536–539, 2003.
- [11] M. Millberg, E. Nilsson, R. Thid, and A. Jantsch, "Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum Network on Chip," Proc. Design, Automation and Test in Europe, pp.890–895, 2004.
- [12] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for Network on Chip," J. Syst. Arch.: The Euromicro Journal, vol.50, no.23, pp.105–128, Feb. 2004.
- [13] F.G. Moraes, N. Laert, V. Calazans, A.V. de Mello, L.H. Mller, and L.C. Ost, "HERMES: An infrastructure for low area overhead packet-switching networks on chip," Integration, the VLSI Journal, vol.38, no.1, pp.69–93, Oct. 2004.
- [14] J. Bainbridge and S. Furber, "Chain: A delay-insensitive chip area interconnect," IEEE Micro, vol.22, no.5, pp.16–23, Sept./Oct. 2002.
- [15] A. Lines, "Asynchronous interconnect for synchronous SoC design," IEEE Micro, vol.24, no.1, pp.32–41, Jan./Feb. 2004.
- [16] E. Beigne, F. Clermidy, P. Vivet, A. Clouard, and M. Renaudin, "An asynchronous NoC architecture providing low latency service and its multi-level design framework," Proc. IEEE Int'l Symposium on Asynchronous Circuits and Systems, pp.54–63, 2005.
- [17] T. Bjerregaard and J. Sparso, "A router architecture for connection-oriented service guarantees in the MANGO clockless network-on-chip," Proc. Design, Automation and Test in Europe, pp.1226–1231, 2005.
- [18] E. Cota, L. Carro, and M. Lubaszewski, "Reusing and on-chip network for the test of core-based systems," ACM Trans. Des. Autom. Electron. Syst., vol.9, no.4, pp.471–499, Oct. 2004.
- [19] A.M. Amory, E. Cota, M. Lubaszewski, and F.G. Moraes, "Reducing test time with processor reuse in network-on-chip based systems," Proc. Integrated Circuits and Systems Design, pp.111–116, 2004.
- [20] C. Liu, Z. Link, and D.K. Pradhan, "Reuse-based test access and integrated test scheduling for network-on-chip," Proc. Design, Automation and Test in Europe, pp.303–308, 2006.
- [21] E.J. Marinissen, R. Kapur, M. Lousberg, T. McLaurin, M. Ricchetti, and Y. Zorian, "On IEEE P1500 standard for embedded core test," J. Electron. Test. Theory Appl., pp.365–383, 2002.
- [22] A. M. Amory, K. Goossens, E. J. Marinissen, M. Lubaszewski, and F. Moraes, "Wrapper design for the reuse of networks-on-chip as test access mechanism," Proc. IEEE European Test Symposium, pp.213–218, 2006.
- [23] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, "Test wrapper and test access mechanism co-optimization for system-on-chip," J. Electron. Test. Theory Appl., pp.213–230, 2002.
- [24] S.K. Goel and E.J. Marinissen, "SoC test architecture design for efficient utilization of test bandwidth," ACM Trans. Des. Autom. Electron. Syst., vol.8, no.4, pp.399–429, Oct. 2003.
- [25] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, "On using rectangle packing for SoC wrapper/TAM co-optimization," Proc. IEEE VLSI Test Symposium, pp.253–258, 2002.
- [26] F.A. Hussin, T. Yoneda, and H. Fujiwara, "Optimization of NoC wrapper design under bandwidth and test time constraints," European Test Symposium, pp.35–40, 2007.
- [27] F.A. Hussin, T. Yoneda, A. Orailoglu, and H. Fujiwara, "Power-constrained SoC test schedules through utilization of functional buses," Proc. IEEE Int'l Conference on Computer Design, pp.230–236, 2006.
- [28] R.M. Chou, K.K. Saluja, and V.D. Agrawal, "Scheduling tests for VLSI systems under power constraints," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.5, no.2, pp.175–185, June 1997.

- [29] E.J. Marinissen, V. Iyengar, and K. Chakrabarty, "A set of benchmarks for modular testing of SoCs," Proc. International Test Conference, pp.519–528, 2002.



**Fawnizu Azmadi Hussin** is a Ph.D. student in the Computer Design & Test Laboratory at Nara Institute of Science and Technology. He obtained his B.Sc. in Electrical Engineering, specializing in Computer Design from the University of Minnesota, U.S.A. and subsequently his M.Eng.Sc. in Systems and Control from the University of New South Wales, Australia. His research interests are in VLSI design and testing, especially in the area of System-on-Chip (SoC) and multiprocessor SoC. He was previously an academic staff at the Universiti Teknologi PETRONAS (Malaysia) prior to starting his Ph.D. research. He is a member of the IEEE.



**Tomokazu Yoneda** received the B.E. degree in information systems engineering from Osaka University, Osaka, Japan, in 1998, and M.E. and Ph.D. degree in information science from Nara Institute of Science and Technology, Nara, Japan, in 2001 and 2002, respectively. Presently he is an assistant professor in Graduate School of Information Science, Nara Institute of Science and Technology. His research interests are VLSI CAD, design for testability, and SoC test scheduling. He is a senior member of the IEEE.



**Hideo Fujiwara** received the B.E., M.E., and Ph.D. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1969, 1971, and 1974, respectively. He was with Osaka University from 1974 to 1985 and Meiji University from 1985 to 1993, and joined Nara Institute of Science and Technology in 1993. Presently he is a Professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. His research interests are logic design, digital systems design and test, VLSI CAD and fault tolerant computing, including high-level/logic synthesis for testability, test synthesis, design for testability, built-in self-test, test pattern generation, parallel processing, and computational complexity. He is the author of Logic Testing and Design for Testability (MIT Press, 1985). He received many awards including Okawa Prize for Publication, IEEE CS (Computer Society) Meritorious Service Awards, IEEE CS Continuing Service Award, and IEEE CS Outstanding Contribution Award. He served as an Editor and Associate Editors of several journals, including the IEEE Trans. on Computers, and Journal of Electronic Testing: Theory and Application, and several guest editors of special issues of IEICE Transactions of Information and Systems. Dr. Fujiwara is a fellow of the IEEE, a Golden Core member of the IEEE Computer Society, and a fellow of the IPSJ.