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

Fawnizu Azmadi HUSSIN\(^{(a)}\), Nonmember, Tomokazu YONEDA\(^{(b)}\), Member, and Hideo FUJIWARA\(^{(c)}\), 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 Section 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], SoCDBUS [9], xPIPEC [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, mak-
ing 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 Section 2. In Section 3, a brief description of the NoC wrapper architecture used in this paper is given. The test schedule and wrapper optimization methodology through bandwidth sharing is explained in Section 4. Some experimental results on selected benchmark circuits are given in Section 5. Finally, concluding remarks are offered in Section 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 $R_0$–$R_3$ 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 $k$ is guaranteed a minimum sustained bandwidth $B_{vc_k}$, where $\sum_k B_{vc_k} \leq B_{max}$. The term $B_{max}$ represents the maximum link bandwidth between each pair of routers $R_i$ and $R_j$ along the VC path. If $B_{vc_i}^{j} < B_{max}$ 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. Fig. 2 shows the conceptual view of the token-ring-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. Fig. 2 shows five VCs, VC1, VC2, VC3, VC4, and VC5, with 1 Gbps, 1 Gbps, 2 Gbps, 3 Gbps, and 3 Gbps bandwidth 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, 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 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. \( n_{pdi} \mod n_{sc} \) 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 \( n_{pdi} \mod n_{sc} = 0 \). Figure 4(a) shows two input boundary 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 Figure 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 bits × 100 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 \( R_1 \) and \( R_2 \). 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 \( R_1 \) link, 4.8 Gbps on \( R_1 \) link, and 3.2 Gbps on \( R_2 \) 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 Æthereal. 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 signalled by the \( \text{pci}[0] \) and \( \text{pci}[1] \) control signals; depending on the write transaction protocol, the signals could be \( \text{DATA}_{-}\text{STROBE}, \text{DATA}_{-}\text{VALID} \), 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 acknowledgement 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 \( \text{CUT} \) 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:

\[
\Psi_S: \text{Given an SoC } C \text{ with } M \text{ cores, a maximum I/O bandwidth, } B_{\text{max}} \text{ bps, and a scan frequency for all cores, } f_{\text{sc}}, \text{ where each core consists of } n_{ip} \text{ functional inputs, } n_{op} \text{ functional outputs, } n_{bi} \text{ bidirectional connections, } k \text{ internal scan chains of length } l_1, l_2, \ldots, l_k, \text{ for each core } c_i \in C \text{ determine}
\]

(1) the wrapper type and the allocated I/O bandwidth, \( B_{\text{scheduled}}[c_i] \), for the test data transportation, and

(2) the starting time, \( t_{\text{start}}[c_i] \), and end time, \( t_{\text{end}}[c_i] \), of the test application

such that the total test application time and the area overhead are co-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 (Section 4.3), we first clarify two required com-
ponents of the algorithm in sections 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, $B_{\text{max}}^{sc}$ bps, find the number of wrapper scan chains, $n_{sc}$, such that (i) the TAT is minimum, (ii) the required bandwidth, $B_{\text{req}}$, is minimum, (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, $T_{\text{max}}$, find the number of wrapper scan chains, $n_{sc}$, such that (i) the required bandwidth, $B_{\text{req}}$, is minimum, (ii) TAT $\leq T_{\text{max}}$, 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, $B_{\text{req}} \leq B_{\text{max}}^{sc}$. 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 equations (1) and (2), respectively. The extra $(+n_{pdi} + n_{pdo} + 2 \cdot n_{sc})$ in equation (2) are due to the additional input/output buffers in the Type 2 wrapper (Fig. 4(h)) 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.

$$H_{11} = n_{ip} + n_{bi} + n_{op} \quad (1)$$

$$H_{12} = n_{ip} + n_{bi} + n_{op} + n_{pdi} + n_{pdo} + 2 \cdot n_{sc} \quad (2)$$

$$\text{Cost}_{(1 \rightarrow 2)} = \alpha \cdot \left( \frac{T_{11} - T_{12}}{T_{11}} + \frac{B_{12} - B_{11}}{B_{11}} \right) + \beta \cdot \frac{H_{12} - H_{11}}{H_{11}} \quad (3)$$

$$\text{Cost}_{(2 \rightarrow 1)} = \alpha \cdot \left( \frac{T_{21} - T_{22}}{T_{21}} + \frac{B_{21} - B_{22}}{B_{22}} \right) + \beta \cdot \frac{H_{21} - H_{22}}{H_{22}} \quad (4)$$

For a given maximum bandwidth, $B_{\text{max}}$, the optimum configuration of a core $c_i$ is determined by solving $\Psi_B(c_i, B_{\text{max}}^i)$ to obtain the respective TAT ($T_{11}$ and $T_{22}$) and required bandwidth ($B_{11}$ and $B_{22}$) for the Type 1 and the Type 2 wrappers, respectively. If $\text{Cost}_{(1 \rightarrow 2)} < \text{Cost}_{(2 \rightarrow 1)}$, then the Type 2 wrapper is selected as a better wrapper configuration for the given $B_{\text{max}}$. 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, $B_{\text{max}}^{i/o}$, its test time can be determined by $T(\Psi_B(c_i, B_{\text{max}}^{i/o}))$, which represents the TAT returned by the $\Psi_B$ search algorithm for Core $c_i$ when the given maximum bandwidth is $B_{\text{max}}^{i/o}$. Even with unlimited bandwidth, the TAT of an SoC $C$ cannot be shorter than the TAT of the longest core $c_i \in C$. Therefore the first lower bound can be written as

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

For a bounded $B_{\text{max}}^{i/o}$, $T_{1LB}$ 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 equation (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 equation (7), where $f_{m}$ is the scan frequency for all cores. The overall lower bound is the maximum of $T_{1LB}^1$ and $T_{LB}^2$ (equation (8)).

$$T(c_i) = (\max(si, so) + 1) \times v_{m} + \min(si, so) \quad (6)$$

$$T_{2LB}^2 = \sum_{c_i \in C} \{ T(c_i) \} \div (B_{\text{max}}^{i/o} / f_{m}) \quad (7)$$

$$T_{LB} = \max(T_{1LB}^1, T_{LB}^2) \quad (8)$$
in the low gain region, it would be wiser to assign that
the next Pareto-optimal point with a T\text{AT} of 342
width allocated to that core. Therefore, rather than
straint. Since the Type 1 wrapper cannot effectively
duces the smallest test application time, which fulfills
sively searches for the wrapper configuration that pro-
Fig. 9, the preferred bandwidth results after configur-
this paper, the height of a rectangle represents the re-
[28] for dedicated T\text{A}\text{T} based scheduling approach. In
We now introduce the concept of rectangles to represent
Bandwidth
\text{BW}_{\text{req}} = 2000 \text{ Mbps}
\text{BW}_{\text{req}} = 800 \text{ Mbps}
\text{BW}_{\text{req}} = 2000 \text{ Mbps}
\text{BW}_{\text{req}} = 800 \text{ Mbps}

(a) Type 1
(b) Type 2

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

4.3 Schedule Optimization through Rectangle
Packaging

We now introduce the concept of rectangles to represent
core tests, then explain a flexible scheduling method-
ology based on NoC bandwidth sharing, which is in-
spired by the scheduling algorithm in [25]. The use
of rectangles has previously been proposed in [25],
[28] for dedicated T\text{A}\text{T} based scheduling approach. In
this paper, the height of a rectangle represents the re-
quired NoC bandwidth to obtain the test application
time represented by the horizontal length. Fig. 7 illus-
trates two pairs of rectangles, each representing the test
of Core 6 of p93791 circuit (ITC’02 benchmark [29])
when \text{BW}_{\text{max}} = 2000 \text{ Mbps} and 800 \text{ Mbps}, respectively.
For this example, the NoC port’s PDI/PDO bit-width
is \text{n}_{\text{pdo}} = 64 \text{ bits}.

The top left rectangle is obtained using the wrap-
er optimization algorithm \Psi_{B} described in Section 4.1,
when given as input the maximum allocated band-
width, \text{BW}_{\text{max}} = 2,000 \text{ Mbps}. The algorithm iter-
atively searches for the wrapper configuration that
produces the smallest test application time, which fulfills
the Pareto-optimal criteria, under the bandwidth con-
straint. Since the Type 1 wrapper cannot effectively
utilize all the allocated bandwidth, the algorithm finds
the next Pareto-optimal point with a T\text{AT} of 342,076
clock cycles which requires 1,600-Mbps NoC band-
width. The same procedure is repeated for the Type 2
wrapper. With a more efficient bandwidth matching ar-
chitecture, the Pareto-optimal wrapper is found with a
T\text{AT} of 337,478 clock cycles and a required bandwidth
of 2,000 Mbps (top right rectangle). For \text{BW}_{\text{max}} = 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 \text{preferred bandwidth}
for each core in the SoC \text{C} (Fig. 8). As illustrated in
Fig. 9, the preferred bandwidth results after config-
uring the core wrapper with the number of scan chains in
the “high gain” region. Gain represents the potential
reduction in T\text{AT} of a core per additional unit of band-
width 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 region.

Figure. 10 describes the algorithm to determine the
preferred bandwidth for all cores. In line 28, a proper
value of input percent \text{v}_{\text{gain}} shifts the target T\text{AT} from
\text{T}_{\text{max} - \text{pareto}} to the high gain region. Figure 9 illustrates
some of the variables, and show how \text{T}_{\text{target1}} is calcu-
lated using the variable \text{v}_{\text{gain}}. Lines 26-29 are evalu-
ated for both Type 1 and Type 2 wrapper configura-
tions. For every core \text{c}_{i} \in \text{C}, equations (1)-(4) are eval-
uted to determine the best wrapper type, for which
the value of \text{T}_{\text{pref}}[\text{c}_{i}] is returned. The same wrapper
selection procedure is performed at line 12 when eval-
uating \text{T}(\Psi_{B}(\text{c}_{i}, \text{B}_{\text{free}})).

In some cases where the test application time is
dominated by a large core such as Core 6 of p93791, se-
lecting the high gain region for Core 6 could potentially
make it a bottleneck core, thus preventing further re-
duction of T\text{AT}. In order to handle this kind of special
cases, we need to be able to allocate as much band-
width as possible to these potential bottleneck cores.
In line 33, the variable \text{v}_{\text{bottleneck}} together with the
lower bound, \text{T}_{\text{LB}} (equation (8)), ensures that bot-
tleneck 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 or equal to $B_{pref}$ (Line 9). Otherwise, the core $c_i$ is considered that leaves minimum $B_{free}$ after it is scheduled, is assigned bandwidth of $B_{req} = B_i(B(c_i, B_{free})) \leq B_{free}$, that can be effectively utilized by the core $c_i$ (Lines 12-13). $T_i$ is maximum, is reconsidered for further optimization. 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 allo-

**Fig. 9** High (preferred) and low gain regions. $T_{pref}$ is the target time and assigned bandwidth is chosen such that the current schedule of core $c_i$ 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. The scan frequency, $f_{scan} = 100 MHz$. The computation time is less than 10 seconds for the largest circuit.

![Fig. 12](image-url) Further optimizing the schedule. Dotted rectangles represent possible schedule/wrapper configurations.

| Table 1 | 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 equation (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 equation (2).

In Table 2, the weights of hardware overhead cost (\( \beta \)) and time 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 Section 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 Gbps to 12.8 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 reduced 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 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.

Acknowledgements

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

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 member of the IEEE Computer Society and a member of the IEICE (the Institute of Electronics, Information and Communication Engineers of Japan).

Hideo Fujiwara (Fellow) 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, a fellow of the IEICE, and a fellow of the IPSJ.