J Electron Test (2010) 26:453–464
DOI 10.1007/s10836-010-5163-x

Bandwidth Analysis of Functional Interconnects Used as Test Access Mechanism

Ardy van den Berg · Pengwei Ren · Erik Jan Marinissen · Georgi Gaydadjiev · Kees Goossens

Received: 1 December 2008 / Accepted: 14 June 2010 / Published online: 10 July 2010
© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract Test data travels through a System on Chip (SOC) from the chip pins to the Core-Under-Test (CUT) and vice versa via a Test Access Mechanism (TAM). Conventionally, a TAM is implemented using dedicated communication infrastructure. However, also existing functional interconnect, such as a bus or Network on Chip (NOC), can be reused as TAM; this will reduce the overall design effort and associated silicon area. For a given core, its test set, and maximal bandwidth that the functional interconnect can offer between test equipment and core-under-test, our approach instantiates a test wrapper for the core-under-test such that the test length is minimized. Unfortunately, it is unavoidable that along with the test data also unused (idle) bits are transported. This paper presents a holistic TAM bandwidth under-utilization analysis when functional interconnect is considered for test data transportation. We classify the idle bits into four types that refer to the root-cause of bandwidth under-utilization and pinpoint design improvement opportunities. Experimental results show an average bandwidth utilization of 80%, while the remaining 20% is consumed by the idle bits.

Keywords System on Chip · Test Access Mechanism · Functional interconnect · Bus · Network on Chip · Bandwidth

1 Introduction

Rapid improvements in the semiconductor industry allow the design and manufacturing of increasingly complex chips, often referred to as Systems on Chip

Responsible Editor: C. Metra

This paper is an extended version of a paper published at the IEEE European Test Symposium (ETS), May 25–29, 2008 in Verbania, Italy [28].

A. van den Berg · P. Ren · G. Gaydadjiev · K. Goossens
Department of Computer Engineering,
Delft University of Technology,
Mekelweg 4, 2628CD Delft,
The Netherlands

G. Gaydadjiev
e-mail: g.n.gaydadjiev@ewi.tudelft.nl

E. J. Marinissen · K. Goossens
Corporate Innovation & Technology,
NXP Semiconductors,
High Tech Campus 37,
5656AE Eindhoven,
The Netherlands

Present Address:
A. van den Berg
Essent, Arnhem, The Netherlands
e-mail: ardy@mailberg.nl

Present Address:
P. Ren
ASML, Veldhoven, The Netherlands
e-mail: pengwei.ren@asml.nl

Present Address:
E. J. Marinissen (✉)
IMEC, Leuven, Belgium
e-mail: erik.jan.marinissen@imec.be

Present Address:
K. Goossens
Department of Electrical Engineering,
Eindhoven University of Technology, Potentiaal/PT g.34,
Den Dolech 2, 5612 AZ Eindhoven, The Netherlands
e-mail: k.g.w.goossens@tue.nl

(SOC). SOCs are composed of multiple, often heterogeneous cores. Each core is tested individually using on-chip isolation hardware called a wrapper. Stimuli and responses travel through the chip to and from the embedded core using a Test Access Mechanism (TAM) [29]. Conventionally, dedicated wires are used to implement the TAM.

Recently, it has been proposed to reuse existing functional interconnects, such as a bus or a NOC [8, 12], as TAM [1–3, 5, 6, 9, 14–16, 18, 19, 23]. The main advantage of this approach is the fact that it makes a dedicated TAM superfluous, leading to a reduction in design complexity and silicon area. The approach requires modifications to the conventional test wrapper, which now no longer transports test data via dedicated TAM ports, such as the WPI and WPO ports of IEEE Std. 1500 [7], but via reused functional ports instead.

The length of an SOC test dictates the required vector storage (in bits) on the automatic test equipment (ATE) and the time (in seconds) each SOC spends on the ATE. A reduction of the test length directly translates into savings in the test cost. In this paper we present an analysis of bandwidth utilization of functional interconnect serving as TAM. We identify four types of idle bits that cause under-utilization of the available bandwidth between ATE and core under test, and hence contribute to a longer-than-strictly-necessary test length. Some of these idle bits are unavoidable for a given TAM and core design, but can be eliminated or reduced by (small) design modifications, which are pinpointed by our method.

The remainder of this paper is organized as follows. Section 2 gives an overview of related prior work. Section 3 discusses various aspects and choices we made when reusing functional busses and NOCs as TAM. Section 4 describes our wrapper design approach that enables reusing functional interconnect as TAM. We define four types of idle bits that help us to explain bandwidth under-utilization in Section 5, and describe how to reduce the amount of idle bits in Section 6. Experimental results are provided and discussed in Section 7, while Section 8 concludes this paper.

2 Prior Work

Examples of previous work that propose to handle on-chip transport of test data via a reused functional bus are [3, 5, 9, 14, 15, 18, 19]. Most of these approaches are based on functional tests, of which the detection qualities are hard to assess, guarantee, and improve, and for which failure diagnosis is nearly impossible. Feige et al. [9] does apply structural scan-based tests via the ARM bus, but in a rather cumbersome way. Nahvi and Ivanov [23] were the first to propose to transport test data via a packet-switching network; they do not quantify the associated silicon area costs, but as they propose a dedicated test network, these costs must be high. Cota et al. [6] were the first to propose to reuse a functional NOC as TAM. Their approach requires knowledge of many NOC implementation details, such as the network topology, number of routers, etc. Amory et al. [2] propose a wrapper design which enables any existing functional interconnect, including bus and NOC, to be reused as TAM, provided the interconnect offers guaranteed throughput and constant latency [12, 24]. They describe how the streaming nature of scan testing (i.e., once started, it should complete uninterrupted) is matched to the possibly bursty or packetized traffic over bus or NOC. Next to the main benefit, viz. the reuse of an existing communication infrastructure, there is a side benefit, as their wrapper design proposal slightly reduces the test length compared to a conventional dedicated TAM.

Analysis of TAM bandwidth utilization for modular SOC testing was first published by Goel and Marinissen [10, 20]. For dedicated TAMs, they classify under-utilized bandwidth into three types of idle bits. Their first type of idle bits is caused by different completion times of the various TAMs in an SOC. If a TAM is not of Pareto-Optimal width for a particular core that is assigned to it, this causes the second type of idle bits. The third type of idle bits is due to imbalanced wrapper chain lengths per core. Hussin et al. [16] identified another, fourth type of idle bits, specific to test wrappers that reuse functional interconnect. Their paper also proposes a modification to the wrapper design of [2], that eliminates these idle bits, but adds significant (but in the paper unquantified) area costs. As a continuation of that work in [17] two heuristically devised wrapper designs (under maximum bandwidth and test application time constraints) were presented that slightly reduce the test time by 7.8%.

This paper presents a holistic test bandwidth analysis for systems where existing functional interconnect, such as a bus or NOC, is reused as TAM. First we start by analyzing the streaming-data requirements from scan tests and test equipment, and how they can be mapped onto existing functional interconnect. Subsequently, we present a precisely quantified analysis of the idle bits that occur in such systems, and that cause under-utilization of the available bandwidth between ATE and core under test, hence contributing to a long-than-strictly-necessary test length. Our study is based on an optimized version of the test wrapper design

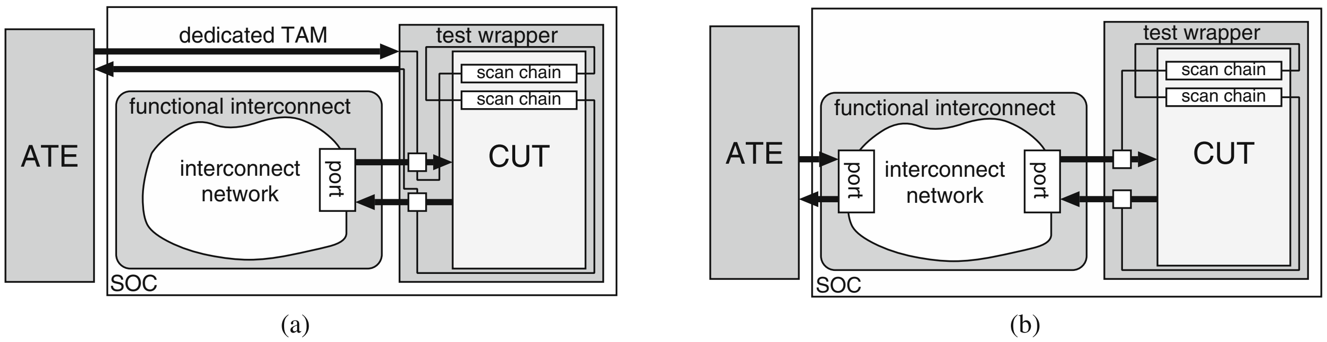


Fig. 1 Test set-ups in which (**a**) a conventional dedicated TAM is used, and (**b**) the existing functional interconnect is reused as TAM

proposed in [2]; without loss of generality we focus on testing a single core in isolation. Our paper demonstrates that test length reductions can be achieved through design modifications that are suggested by our analysis.

3 Functional Interconnect as TAM

In conventional modular SOC test approaches, dedicated TAMs are used to transport test data from the ATE to the core-under-test and vice versa. Reusing existing functional interconnect as TAM avoids dedicated TAMs and their associated design and area costs. Figure 1 shows, at a conceptual level, the difference between the two approaches. The new approach requires a customized wrapper design modified in comparison to conventional wrappers [2]: it lacks the dedicated-TAM input and output of conventional wrappers, but instead is equipped with ports that "speak" the communication protocol of the overall system functional interconnect and convert periodically-arriving functional data into streaming scan data and vice versa.

Although different in the details of exact signal names and semantics, functional port protocols as AXI [4] and DTL [25] typically have a similar structure, which consists of three signal groups: command, write, and read. We distinguish between *initiator* and *target* ports. An initiator port sends out the command and hence initiates the communication; a target port receives the command. A write port communicates data in the same direction as the command (i.e., from initiator to target), while a read port communicates data in the opposite direction; read-write ports can communicate data in both directions.

Figure 2 illustrates the above for DTL's Memory-Mapped Block-Data (MMBD) profile which is the most complex profile of the DTL protocol [25]. The signal names indicate the partitioning of the port signals into three groups; command, write, and read. All three signal groups have their own dedicated valid and accept signals that regulate the handshake process for data transfer. In addition, the command group has three more signals, that indicate address, read/write direction, and block size.

To transport test data over the functional interconnect to a core, that core needs to have at least one port that can serve as a test stimulus input and at least one port that can serve as a test response output. The test stimulus input role can be enacted by an initiator read or read-write port, or a target write or read-write port. Similary, a test response output role can be performed by an initiator write or read-write port, or by a target read or read-write port.

In our approach, we have selected to restrict ourselves to work with two disjoint ports per core, one serving as test stimulus input port and the other one serving as test response output port. In principle, it would be possible to unite these two functions in a single bi-directional (read-write) port. We have avoided this, in order to allow simultaneous operation of scan-in and scan-out operations during test. Also, in general,

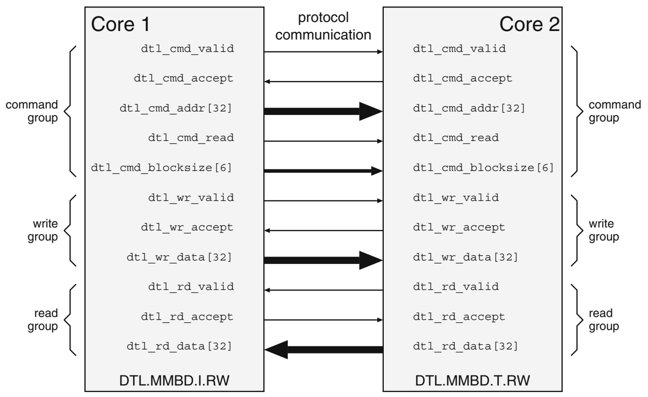


Fig. 2 DTL Memory-Mapped Block-Data (MMBD) functional port protocol

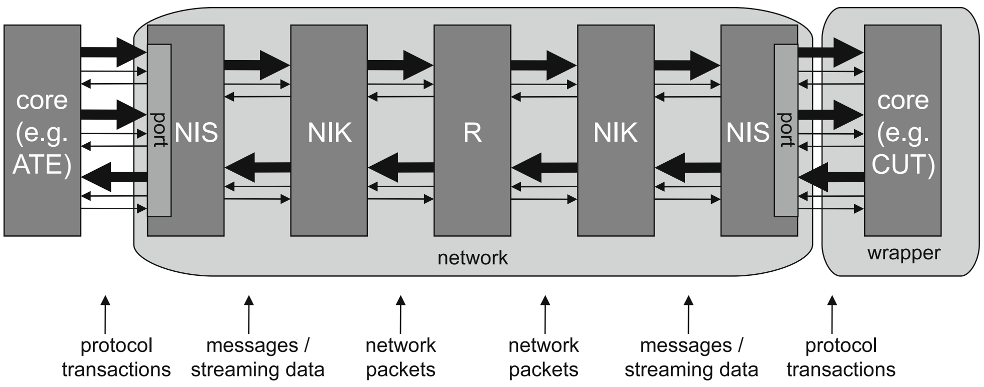


Fig. 3 NOC, network interface shells (NIS) and kernels (NIK)

it would be possible to use multiple input or output ports for test purposes in cases such ports are available; this would possibly increase the available accumulative bandwidth for test data transport. Again, we have decided not to do this, in order to perform our study in isolation from issues related to the scheduling and synchronization of multiple simultaneous test streams through the functional interconnect.

It is conceivable that test data would not only be transported through the normal data lines (in Fig. 2: `dtl_wr_data[32]` and `dtl_rd_data[32]`), but also via wires of the command group (e.g., `dtl_cmd_addr[32]`). We have decided *not* to do so, and restrict ourselves to test data transport via the normal data words only, in order to use the functional interconnect in its normal mode of operation only.

Note that our choices imply a significant restriction in bandwidth available for test data transport purposes. The example DTL MMBD ports in Fig. 2 consist of 109 wires each, of which only 32 are effectively used for (test) data transport. This was done to present a fair comparison of our proposal instead of reporting the maximal improvements possible.

When the functional interconnect is implemented as a bus or a crossbar, the protocol used between the interconnect and the core is also used within the interconnect. As a result, the bandwidth reserved between the cores is equal to the bandwidth used inside the interconnect. When NOCs are used, this is no longer the case. Figure 3 shows that for an NOC, transactions between cores and interconnect are transported over the network as packets [26]. Network interfaces (NI) packetize and depacketize transactions. NIs are often split in a NI Shell (NIS) and a NI Kernel (NIK). NI shells convert the multiple signal groups that make up a transaction to a single signal group (streaming data), which is a serialized transaction. A transaction is made up of a request message (command and write-data signal groups) and a response message (read-data signal group). The links inside the NOC are likely to be of different width, and run at a higher speed than those between cores and NOC. NI shells also perform this conversion. The NI kernels take care of (de)packetization, i.e., the conversion between messages and packets. Figure 4 illustrates the message and packet formats of the Æthereal NOC [12] used here. Although most cores communicate through distributed shared memory and use MMBD, some use streaming-data communication and are connected directly to the NI kernel. In the latter case, messages are not used but the raw data of the core is packetized directly.

In our approach, we aim at decoupling the design and test generation flows. To the NOC, it makes no difference whether it is used in mission mode transporting regular data between cores, or in test mode acting as TAM transporting test stimuli and responses. In both cases, the NOC is in its normal operation mode and configured to transport data between a set of cores, as specified in the application use cases (a) and (b), respectively, of Fig. 5. Similarly, core-level tests are generated independently from the question how they will be transported to the ATE: directly through chip pins, via a conventional TAM, via a reused functional bus, or via a reused functional NOC.

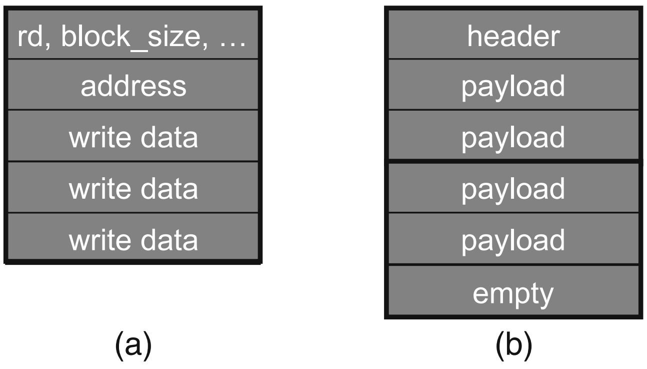


Fig. 4 Examples of Æthereal NOC (**a**) write message and (**b**) packet

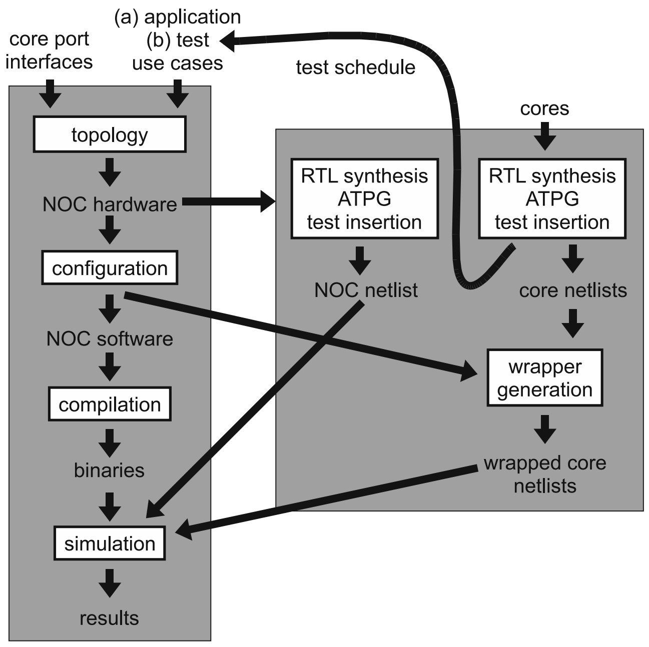


Fig. 5 NOC and core-wrapper design flow

In order to transparently reuse a NOC as TAM, we want to abstract from the NOC data transportation and implementation details, and instead be able to use a NOC as a set of "virtual wires" between two ports. This requires a guaranteed bandwidth and latency between cores. In our case, this is achieved through time-division multiplexing (TDM) in the NI kernel [12, 13]. The NOC hardware (RTL of the topology of routers and NIs, etc.) and NOC software (the C program to configure the NOC at run time) are generated by a dedicated NOC design flow, as shown in Fig. 5 [11]. The hardware and software together can be co-simulated; additionally the performance can be verified analytically. The RTL of the NOC is entered in a conventional synthesis and test flow. Cores are synthesized, have scan chains inserted, and are connected to the functional interconnect which acts as a TAM, in conjunction with the wrapper described in the next section. More information on the above topic is available in [27].

4 Test Wrapper

Figure 6 shows a simplified example of a wrapper which allows the reuse of the functional interconnect as TAM [2]. As in conventional test wrappers, all core terminals are equipped with a wrapper cell, and wrapper chains are formed by concatenating wrapper cells and core-internal scan chains. However, ports to connect to dedicated TAM wires, common in conventional wrappers, are absent. The functional core terminals are

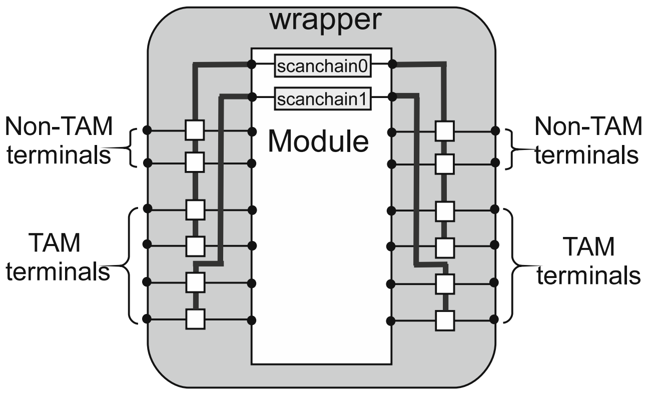


Fig. 6 Simplified example wrapper

partitioned into the protocol ports that are selected as TAM terminals, and all other non-TAM terminals. Test stimuli periodically arrive over the functional interconnect that serves as TAM; in our simplified example in Fig. 6 with word width $w_{in} = 4$. The four bits are divided over $wc = 2$ wrapper chains, and hence each wrapper chain receives two bits every period of $p_{in} = 2$ clock cycles. The test stimuli are shifted into the core-under-test through the wrapper chains. The last word that arrives over the TAM terminals does not need to be shifted in, but can be applied directly to the core-under-test. After the actual testing launch and capture takes place, test responses are transported away from the core in a similar fashion.

Figure 7a shows the typical ordering of elements in a wrapper chain for a conventional wrapper, which uses a dedicated TAM. As defined in [21], input wrapper cells are followed by internal scan chains, which are followed by output wrapper cells. Such a wrapper chain receives one stimulus bit every clock cycle; subsequently it takes s_{in} scan cycles to fill the wrapper chain with stimuli. Similarly, it takes s_{out} scan cycles to offload the responses from the wrapper chain. Figure 7b shows the typical wrapper chain ordering for our new wrapper design. Also here the ordering is: input wrapper cells, followed by internal scan chains, followed by output wrapper cells. However, at the extreme input side, those input wrapper cells are positioned, which periodically every p_{in} clock cycles receive a new parallel word with stimulus bits. Similarly, at the extreme output side, the output wrapper cells are positioned, which periodically every p_{out} clock cycles send out a new parallel word with response bits.

Let B_{in} be the bandwidth over the functional interconnect from ATE to core-under-test, and let B_{out} be the bandwidth in the reverse direction. The maximum number of wrapper chains wc that can be supplied through the functional interconnect with streaming

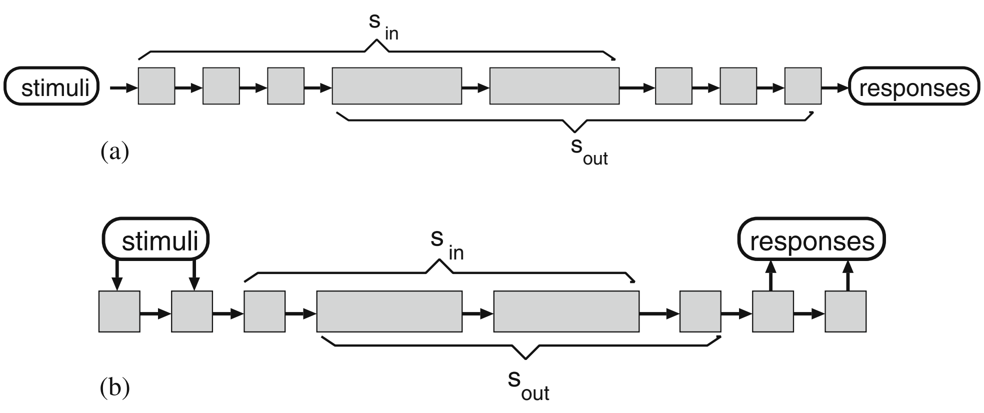


Fig. 7 Typical wrapper chain for (**a**) a wrapper with a dedicated TAM, and (**b**) a wrapper which reuses functional interconnect as TAM

scan test data (i.e., one bit per clock cycle per wrapper chain) is given by $wc = \left\lfloor \frac{\min(B_{\text{in}}, B_{\text{out}})}{f} \right\rfloor$, where f is the test frequency of the core-under-test. Stimulus bits arrive periodically in words of w_{in} bits and are divided over the wc wrapper chains. This process is repeated every p_{in} cycles, with $p_{\text{in}} = \left\lfloor \frac{w_{\text{in}}}{wc} \right\rfloor$. The responses are handled likewise, with $p_{\text{out}} = \left\lfloor \frac{w_{\text{out}}}{wc} \right\rfloor$.

5 Idle Bits Classification

Ideally, every clock cycle every wire of the TAM transports either a test stimulus bit or a test response bit. However, it is often unavoidable that some non-useful bits are transported together with the useful test data. These non-useful bits are referred to as *idle bits*. They may increase the test data volume to be stored on the ATE and consume part of the available bandwidth for test data transport. Idle bits occur in (1) traditional monolithic scan testing, (2) conventional modular SOC testing with dedicated TAMs, as well as in (3) a modular SOC test approach that reuses functional interconnect as TAM. This section describes and classifies four types of idle bits that arise in the latter case.

- Type-1: different scan chain lengths within a core [10, 20];
- Type-2: scan-in (scan-out) length is not an exact multiple of the input (output) period;
- Type-3: maximum scan-in and scan-out lengths are different;
- Type-4: the word width of the functional interconnect is not an exact multiple of the number of wrapper chains [2, 16].

In the sections below we discuss each type in more detail.

5.1 Type-1 Idle Bits

Shifting bits into the wrapper chains completes when the wrapper chain i with the longest scan-in length $s_{\text{in},i}$ is filled with test stimuli. Other, shorter wrapper chains require less time to shift in valid stimuli and receive therefore dummy bits before their valid test stimulus bits are sent. A similar situation exists at the test response side. These dummy bits are Type-1 idle bits. (Note: Type-1 idle bits were introduced in [10] as Type-3 idle bits.) The bigger the difference between the average scan-in (-out) length and the maximum scan-in (-out) length, the more Type-1 idle bits there are. Figure 8 shows two wrapper chains of unequal length and the corresponding Type-1 idle bits for this example.

There are one or more *tests* for a core, where for each test i, pat_i patterns exist to test the core. Type-1 idle bits, if present, occur in every pattern of every test. Hence:

$$ib^1_{\text{in}} = \sum_{i=1}^{\text{tests}} \sum_{j=1}^{wc} \left((S_{\text{in}} - s_{\text{in},j}) \cdot pat_i \right) \tag{5.1}$$

$$ib^1_{\text{out}} = \sum_{i=1}^{\text{tests}} \sum_{j=1}^{wc} \left((S_{\text{out}} - s_{\text{out},j}) \cdot pat_i \right) \tag{5.2}$$

where $S_{\text{in}} = \max_{1 \le x \le wc}(s_{\text{in},x})$ and $S_{\text{out}} = \max_{1 \le x \le wc}(s_{\text{out},x})$.

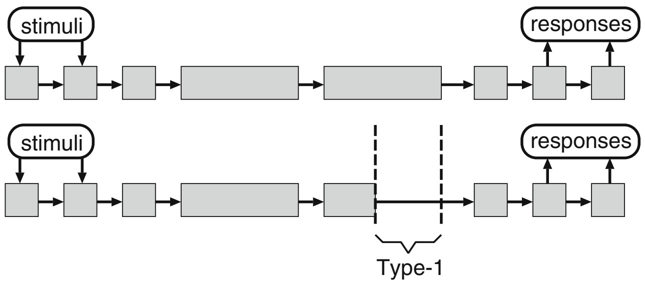


Fig. 8 The cause of Type-1 idle bits: multiple wrapper chains with unequal scan-in and/or scan-out length

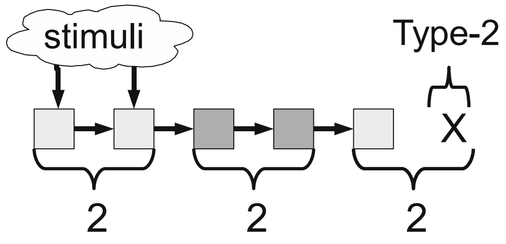


Fig. 9 Type-2 idle bits at the input side, caused by the period at which stimuli arrive

5.2 Type-2 Idle Bits

Stimuli are loaded into the wrapper periodically, in order to keep the chain shifting continuously. The shift-in length of the longest wrapper chain S_{in} divided by the period p_{in} determines the number of words needed to fill the wrapper chain with stimuli. If the shift-in length is not a multiple of the period, one or more idle bits are shifted in; they are referred to as Type-2 idle bits. For example: $S_{\text{in}} = 5$ and $p_{\text{in}} = 2$ results in one idle bit of Type-2. This example is visualized in Fig. 9. These type of idle bits occur at the output side for responses as well.

Type-2 idle bits, if present, occur in every pattern of every test for all wrapper chains.

$$ib^2_{\text{in}} = \sum_{i=1}^{\text{tests}} \left(S'_{\text{in}} - S_{\text{in}}\right) \cdot pat_i \cdot wc \tag{5.3}$$

$$ib^2_{\text{out}} = \sum_{i=1}^{\text{tests}} \left(S'_{\text{out}} - S_{\text{out}}\right) \cdot pat_i \cdot wc \tag{5.4}$$

where $S'_{\text{in}} = \left\lceil \frac{S_{\text{in}}}{p_{\text{in}}} \right\rceil \cdot p_{\text{in}}$ and $S'_{\text{out}} = \left\lceil \frac{S_{\text{out}}}{p_{\text{out}}} \right\rceil \cdot p_{\text{out}}$

5.3 Type-3 Idle Bits

In scan testing, it is common practice to overlap the shift-out of the responses of test pattern n with the shift-in of the stimuli for the next pattern $n + 1$. This process repeats for all patterns of the test set; only when the responses of the last test pattern are shifted out, no new stimuli are shifted in again. We also use this so-called *pipelined scan* in our approach, as it can save up to 50% of the test application time.

In conventional scan testing, scan-in and scan-out lengths are equal, i.e., $S'_{\text{in}} = S'_{\text{out}}$. This is not necessarily true for wrapper-based modular testing, in which S'_{in} can be different from S'_{out}, due to different numbers of input wrapper cells and output wrapper cells. For example, if $S'_{\text{out}} < S'_{\text{in}}$, shifting out responses takes fewer clock cycles than shifting in stimuli; the idle bits shifted out after the valid responses are referred to as Type-3 idle bits. If $S'_{\text{in}} < S'_{\text{out}}$, Type-3 idle bits are shifted in before the valid stimuli.

Figure 10 shows for a small example with only two test patterns the idle bits of Type-1, -2, and -3. $S'_{\text{in}} = 12$, while $S'_{\text{out}} = 8$, and hence $S'_{\text{out}} < S'_{\text{in}}$. Four Type-3 idle bits are injected at the response side for each wrapper chain and for each test pattern except the last pattern.

Type-3 idle bits are quantified as follows:

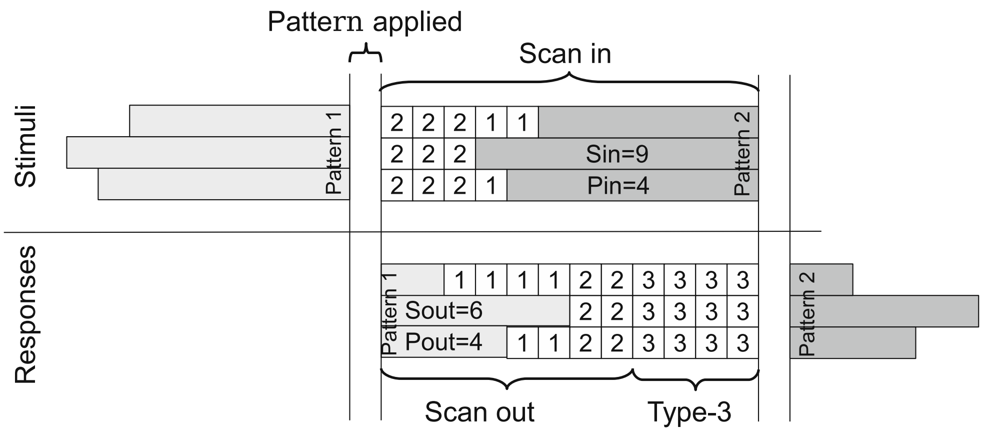
$$ib^3_{\text{in}} = \sum_{i=1}^{\text{tests}} \max\left(0, S'_{\text{out}} - S'_{\text{in}}\right) \cdot (pat_i - 1) \cdot wc \tag{5.5}$$

$$ib^3_{\text{out}} = \sum_{i=1}^{\text{tests}} \max\left(0, S'_{\text{in}} - S'_{\text{out}}\right) \cdot (pat_i - 1) \cdot wc \tag{5.6}$$

5.4 Type-4 Idle Bits

The number of wrapper chains wc should be as large as possible, to make maximum use of the available

Fig. 10 Example of Type-3 idle bits: shifting in stimuli does not take the same time as shifting out responses



TAM bandwidth and hence reduce the corresponding test application time: $wc = \left\lfloor \frac{\min(B_{\text{in}}, B_{\text{out}})}{f} \right\rfloor$. With period $p_{\text{in}} = \left\lfloor \frac{w_{\text{in}}}{wc} \right\rfloor$ a parallel word of w_{in} bits arrives, which is divided over the wc wrapper chains. Due to rounding differences, for every pattern in each such parallel word except for the last one, (w_{in} mod wc) bits are wasted; we refer to them as Type-4 idle bits. A similar situation occurs at the test response side.

Note that these Type-4 idle bits arrive in dedicated wrapper cells, which in [2] were referred to as RSDI and RSDO cells. In [2, 16], these RSDI and RSDO wrapper cells are placed in the middle of the wrapper chains. In contrast, we put them at the extremes of the wrapper chains, such that they do not unnecessarily contribute to the scan lengths, and hence we obtain a (minor) test length improvement over [2, 16].

An example of Type-4 idle bits is shown in Fig. 11. An existing functional input port that serves as TAM has functional word width $w_{\text{in}} = 32$. Given the bandwidths, in this example we could afford to make $wc = 10$ wrapper chains, and hence words are delivered with a period of $p_{\text{in}} = 3$ clock cycles. The 32 input bits are divided over 10 wrapper chains and hence we have (w_{in} mod wc) = 2 RSDI wrapper cells; in Fig. 11 these wrapper cells are shaded dark. For all TAM words, except for the last word of each pattern, these cells carry Type-4 idle bits.

Type-4 idle bits are present for all patterns of all tests. They occur for every word delivered, apart for the last word of each pattern on the stimulus side and the first word of each pattern on the response side.

$$ib^4_{\text{in}} = \sum_{i=1}^{\text{tests}} \left(\left\lceil \frac{\max(S'_{\text{in}}, S'_{\text{out}})}{p_{\text{in}}} \right\rceil - 1 \right) \cdot pat_i \cdot (w_{\text{in}} \bmod wc) \tag{5.7}$$

$$ib^4_{\text{out}} = \sum_{i=1}^{\text{tests}} \left(\left\lceil \frac{\max(S'_{\text{in}}, S'_{\text{out}})}{p_{\text{out}}} \right\rceil - 1 \right) \cdot pat_i \cdot (w_{\text{out}} \bmod wc) \tag{5.8}$$

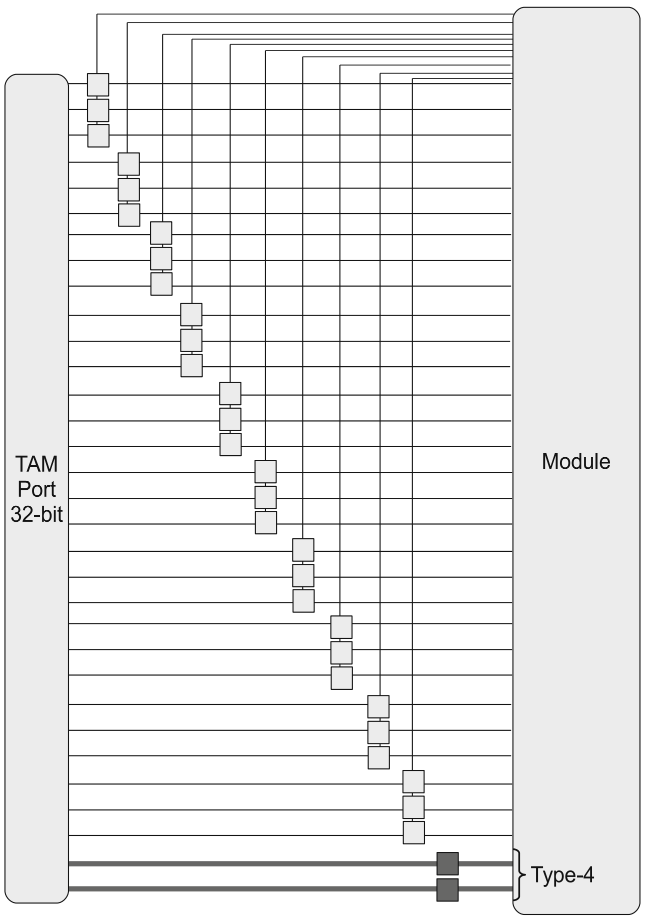


Fig. 11 Type-4 idle bits: data going to or coming from selected data cells which are not in a wrapper chain (*dark cells*)

6 Idle Bit Reduction

Idle bits increase the test length and the number of bits stored on the ATE. To increase the bandwidth utilization, we need to reduce the number of idle bits. For each type of idle bits, we discuss how to reduce them.

Type-1 idle bits are minimal for wrapper chains with balanced scan-in/out lengths. We employ the COMBINE algorithm [21] for this purpose. Obviously, cores with *hard* scan chains limit the possibilities to balance the scan-in/out lengths; typically, better results can be achieved if the scan chains in a core are *soft*, such that they can be re-designed and adapted to wrapper and TAM design.

There are no Type-2 idle bits if (S_{in} mod p_{in}) = 0 and (S_{out} mod p_{out}) = 0. p_{in} and p_{out} are determined by the number of wrapper chains, the available bandwidth, and the test frequency. S_{in} and S_{out} are preferably as low as possible to reduce the test length. No solution is available yet to reduce Type-2 idle bits.

Type-3 idle bits are caused by a difference in S'_{in} and S'_{out}. These idle bits can be reduced by creating wrapper chains with either multiple inputs or multiple outputs, in order to reduce the largest of the two variables. In regular scan testing this does not pay off; however, we postulate that this can pay off for wrapper-based modular SOC testing.

When a functional interconnect is reused as a TAM, it may introduce Type-4 idle bits, depending on the

Table 1 Bandwidth analysis for 26 cores of the ITC'02 SOC Test Benchmarks [22]

SOC	Core	Total idle bits (bit)	T1 (%)	T2 (%)	T3 (%)	T4 (%)	Bandwidth efficiency (%)
g1023	1	24260	1	2	7	0	90
g1023	2	15540	11	0	0	11	78
g1023	3	8721	19	2	0	6	73
g1023	4	34304	9	3	0	3	84
g1023	10	9222	9	6	0	19	66
g1023	11	1168	4	8	8	0	80
g1023	12	784	5	9	0	0	86
g1023	14	204640	8	0	31	0	61
p93791	1	312067	5	0	0	0	95
p93791	5	330858	5	12	0	5	79
p93791	6	409186	1	0	0	2	96
p93791	10	194260	4	3	25	0	67
p93791	11	91256	5	0	0	21	74
p93791	13	195552	5	0	0	0	95
p93791	14	195552	5	0	0	0	95
p93791	17	284472	3	0	0	6	91
p93791	19	700350	2	0	0	25	74
p93791	23	221364	6	0	0	0	94
p93791	29	795500	26	0	0	0	74
p93791	32	511816	4	3	25	0	68
a586710	1	35937835	7	0	0	18	74
a586710	2	297445750	7	0	12	2	78
a586710	3	1284242480	5	4	16	3	72
a586710	4	19200840	4	10	0	3	83
a586710	5	1584410	1	0	0	0	99
a586710	7	329282348	3	7	29	0	62
Average	–	75855174	6	3	6	5	80

functional data width. They can be prevented by buffering all stimuli and responses. Hussin et al. [16] propose a solution in which load and shift registers are used to buffer. This solution requires a significant, but in their paper unquantified, amount of extra silicon area.

7 Experimental Results

We have automated the wrapper design to generate wrappers for cores using the approach of Amory et al. [2] and calculated the bandwidth under-utilization for each core due to idle bits. The wrapper generator uses

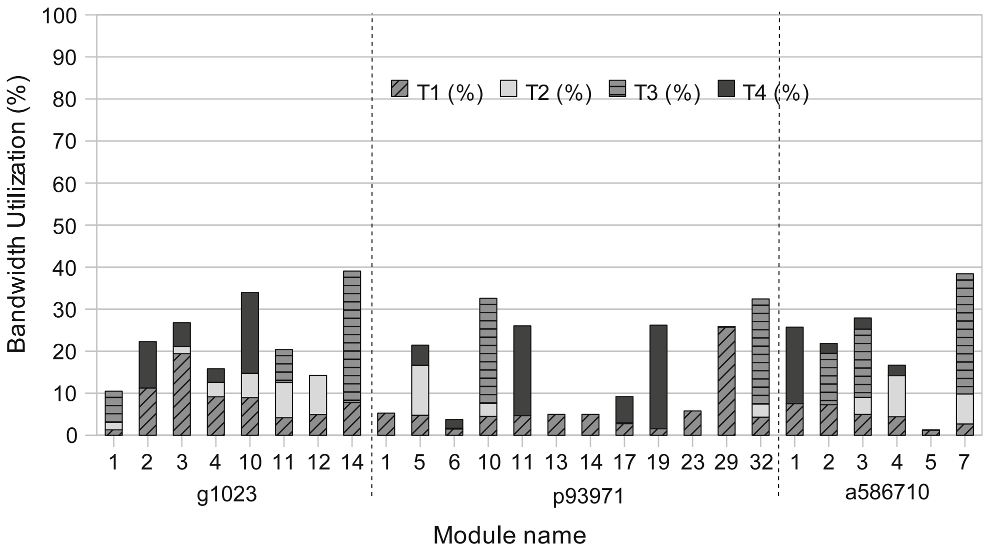


Fig. 12 Bandwidth under-utilization due to idle bits

as many ports as possible and tries to generate a wrapper design with minimal test length.

As input we use the SOCs g1023, p93791, and a586710 of the ITC'02 SOC Test Benchmark Set [22]. For each core we assume for every 100 inputs and 100 outputs one input and one output port with a word width $w = 32$ using the AXI protocol [4]; cores with a big amount of i/o-terminals will therefore have a bigger bandwidth compared to cores with a small amount of i/o-terminals. The test frequency $f_{\text{test}} = 100$ MHz. Today's functional interconnects can work at a frequency of 500 MHz [12]. A 32-bit port delivers 32-bit per cycle minus overhead, which is assumed to be 20%. The ITC'02 benchmarks are five years old, and, scaling with Moore's Law, we assume their functional interconnects were working at 1/8 of today's bandwidth. These assumptions result in $b = 32 \cdot 500 \cdot 0.8 \cdot \frac{1}{8} = 1{,}600$ Mbit/s per port.

For 26 cores of the ITC'02 benchmarks, we have calculated the bandwidth under-utilization due to idle bits. Table 1 lists the results. For each core, the absolute number of idle bits has been calculated, as well as the relative percentages of Type-1, -2, -3, and -4 idle bits. The last column of Table 1 lists the bandwidth efficiency, which is reduced due to the idle bits. For example, Core 1 of SOC p93791 requires 312067 idle bits to transport all stimuli and responses to and from the core. These idle bits reduce the useful bandwidth from 100% to 95%. The 5% reduction was due to Type-1 idle bits. Average results over all 26 cores are given in the bottom row of the table.

Figure 12 shows a graphical representation of the results of Table 1. On average 80% of the available bandwidth is used for actual stimuli and responses. 20% is idle bits and causes under-utilization of bandwidth. The idle bits are more or less equally spread over all four categories.

8 Conclusion

Reusing the existing functional interconnect as a TAM cancels the need for a dedicated TAM. In this paper we analyzed the bandwidth utilization for wrappers which reuse the existing functional interconnect as a TAM. We defined four types of idle bits to explain the under-utilization of the available bandwidth between the ATE and core under test. Since reduction of idle bits improves the bandwidth utilization and minimizes the required ATE vector storage, several solutions to reduce idle bits were discussed.

We automatically generated wrappers for 26 cores of the ITC'02 SOC Test Benchmarks and calculated the bandwidth utilization by useful test data and idle bits. Idle bits can use up to 39% of the available bandwidth, with an average of 20%. All four types of idle bits were found to contribute to the bandwidth under-utilization.

Using the proposed bandwidth under-utilization analysis, wrappers which reuse the existing functional interconnect can be efficiently modified to reduce the overall test length of cores. This, along with the silicon area savings obtained by omitting the conventional dedicated TAM infrastructure, make the proposed method suitable for a wide range of modern SOCs.

Acknowledgments We thank NXP colleagues Martijn Coenen and Andreas Hansson for help with Æthereal NOC architecture and design flow, and NXP colleagues Jo Gunnes, Mario Konijnenburg, and Ronald de Leeuw van Weenen for help with CAT test tools and flow. The work in this project was partly supported by the MEDEA+ 2A702 project NanoTest.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Amory AM, Cota É, Lubaszewski M, Moraes FG (2004) Reducing test time with processor reuse in Network-on-Chip based systems. In: Proceedings Brazilian symposium on integrated circuits and system design (SBCCI). Pernambuco, Brazil, pp 111–116
2. Amory AM, Goossens K, Marinissen EJ, Lubaszewski M, Moraes FG (2007) Wrapper design for the reuse of a bus, Network-on-Chip, or other functional interconnect as Test Access Mechanism. IET Comput Dig Tech 1(3):197–206
3. Amory AM, Oliveira LA, Moraes FG (2003) Software-based test for non-programmable cores in bus-based system-on-chip architectures. In: Proceedings IFIP international conference on very large scale integration (VLSI-SOC). Darmstadt, Germany, pp 174–179
4. ARM (2003) AMBA AXI protocol specification
5. Cota E, Carro L, Lubaszewski M (2004) Reusing an on-chip network for the test of core-based systems. ACM Transact Des Automat Electron Syst 9(4):471–499
6. Cota E, Kreutz M, Zeferino CA, Carro L, Lubaszewski M, Susin A (2003) The impact of NoC reuse on the testing of core-based systems. In: Proceedings IEEE VLSI test symposium (VTS). Napa, CA, USA, pp 128–133
7. DaSilva F (ed) (2005) IEEE Std 1500$^{\text{TM}}$-2005. IEEE standard testability method for embedded core-based integrated circuits. IEEE, New York
8. De Micheli G, Benini L (eds) (2006) Networks on chips: technology and tools. The Morgan Kaufmann series in systems on silicon. Morgan Kaufmann
9. Feige C, ten Pierick J, Wouters C, Tangelder R, Kerkhoff HG (1999) Integration of the scan-test method into an architecture specific core-test approach. J Electron Test: Theory and Applications 14(1–2):125–131

10. Goel SK, Marinissen EJ (2003) SOC test architecture design for efficient utilization of test bandwidth. ACM Transact Des Automat Electron Syst 8(4):399–429
11. Goossens K, Dielissen J, Gangwal OP, Pestana SG, Rădulescu A, Rijpkema E (2005) A design flow for application-specific networks on chip with guaranteed performance to accelerate SOC design and verification. In: Proceedings design, automation, and test in Europe (DATE), Munich, Germany, pp 1182–1187
12. Goossens K, Dielissen J, Rădulescu A (2005) The Æthereal network on chip: concepts, architectures, and implementations. IEEE Des Test Comput 22(5):414–421
13. Hansson A, Goossens K, Rădulescu A (2007) A unified approach to mapping and routing on a network on chip for both best-effort and guaranteed service traffic. VLSI Des 2007: Article ID 68432. Hindawi Publishing Corporation, 16 pp
14. Harrod P (1999) Testing reusable IP—a case study. In: Proceedings IEEE international test conference (ITC). Atlantic City, NJ, USA, pp 493–498
15. Huang J-R, Iyer MK, Cheng K-T (2001) A self-test methodology for IP cores in bus-based programmable SOCs. In: Proceedings IEEE VLSI test symposium (VTS). Marina del Rey, CA, USA, pp 198–203
16. Hussin FA, Yoneda T, Fujiwara H (2007) Optimization of NoC wrapper design under bandwidth and test time constraints. In: Proceedings IEEE European test symposium (ETS). Freiburg, Germany, pp 35–42
17. Hussin FA, Yoneda T, Fujiwara H (2008) NoC-compatible wrapper design and optimization under channel-bandwidth and test-time constraints. IEICE Trans Inf Syst E91-D(7): 2008–2017
18. Hwang S, Abraham JA (2001) Reuse of addressable system bus for SOC testing. In: Proceedings IEEE international ASIC/SOC conference, Arlington, VA, USA, pp 215–219
19. Liu C, Link Z, Pradhan DK (2006) Reuse-based test access and integrated test scheduling for Network-on-Chip. In: Proceedings IEEE European test symposium (ETS), Southampton, UK, pp 303–308
20. Marinissen EJ, Goel SK (2002) Analysis of test bandwidth utilization in test bus and TestRail architectures for SOCs. In: Proceedings IEEE design and diagnostics of electronic circuits and systems workshop (DDECS). Brno, Czech Republic, pp 52–60
21. Marinissen EJ, Goel SK, Lousberg M (2000) Wrapper design for embedded core test. In: Proceedings IEEE international test conference (ITC). Atlantic City, NJ, USA, pp 911–920
22. Marinissen EJ, Iyengar V, Chakrabarty K (2002) A set of benchmarks for modular testing of SOCs. In: Proceedings IEEE international test conference (ITC). Baltimore, MD, USA, pp 519–528
23. Nahvi M, Ivanov A (2001) A packet switching communication-based Test Access Mechanism for system chips. In: Digest of papers of IEEE European test workshop (ETW). Saltsjöbaden, Sweden, pp 195–200
24. Nolen JM, Mahapatra RN (2008) Time-division-multiplexed test delivery for NoC systems. IEEE Des Test Comput 25(1):44–51
25. Philips Semiconductors (2002) Device transaction level (DTL) protocol specification. Version 2.2
26. Rădulescu A, Dielissen J, Pestana SG, Gangwal OP, Rijpkema E, Wielage P, Goossens K (2005) An efficient on-chip network interface offering guaranteed services, shared-memory abstraction, and flexible network programming. IEEE Trans Comput-Aided Des Integr Circuits Syst 24(1): 4–17
27. van den Berg A (2007) Automation of wrapper design for the reuse of a bus, Network-on-Chip, or other functional interconnect as Test Access Mechanism in a chip. MSc Thesis, Delft University of Technology, The Netherlands
28. van den Berg A, Ren P, Marinissen EJ, Goossens K, Gaydadjiev G (2008) Bandwidth analysis for reusing functional interconnect as Test Access Mechanism. In: Proceedings IEEE European test symposium (ETS). Lago Maggiore, Italy, pp 21–26
29. Zorian Y, Marinissen EJ, Dey S (1998) Testing embedded-core based system chips. In: Proceedings IEEE international test conference (ITC). Washington, DC, USA, pp 130–143

Ardy van den Berg received a BSc degree in Electrical Engineering and an MSc degree in Computer Engineering from Delft University of Technology, The Netherlands. He carried out his MSc graduation project at NXP Semiconductors in Eindhoven, The Netherlands. Currently he follows a traineeship at Essent, a power company in Arnhem, The Netherlands.

Pengwei Ren received a BSc degree in Mechanics from Chongqing University, China (2000), and a second BSc degree in Information Technology (2004) as well as an MSc degree in Computer Engineering (2006) from Delft University of Technology, The Netherlands. He carried out his MSc graduation project at NXP Semiconductors in Eindhoven, The Netherlands. Currently, he works at ASML in Veldhoven, The Netherlands.

Erik Jan Marinissen is Principal Scientist at IMEC vzw in Leuven, Belgium. Previously, he worked at NXP Semiconductors and Philips Research, both in Eindhoven, The Netherlands. Marinissen holds an MSc degree in Computing Science (1990) and a PDEng degree in Software Technology (1992), both from Eindhoven University of Technology. Marinissen's research interests include all topics in the domain of test and debug of microelectronics. He is co-author of over 125 journal and conference papers and co-inventor on nine granted US and EP patent families. Marinissen is recipient of the ITC 2008 Most Significant Paper Award and Best Paper Awards at the Chrysler-Delco-Ford Automotive Electronics Reliability Workshop 1995 and the IEEE International Board Test Workshop 2002. He served as Editor-in-Chief of IEEE Std. 1500. He is a founder of workshops on 'Diagnostic Services in Network-on-Chips' (DSNOC), '3D Integration', and '3D-Test'. He serves on numerous conference committees, including ATS, ETS, DATE, ITC, and VTS, and on the editorial boards of IEEE Design & Test of Computers, IET Computers and Digital Techniques, and Springer's Journal of Electronic Testing: Theory and Applications (JETTA). Marinissen is Senior Member of IEEE and Golden Core Member of Computer Society.

Georgi Gaydadjiev is currently a faculty member at the Computer Engineering Laboratory, Microelectronics and Computer Engineering Department of Delft University of Technology, The Netherlands. His research and development experience includes more than 20 years in hardware and software design. Before joining TU Delft he worked with System Engineering Ltd. in Pravetz, Bulgaria and Pijnenburg Microelectronics and Software B.V. in Vught, The Netherlands. His research interests include embedded systems design, advanced computer architectures, reconfigurable computing, hardware/software co-design, VLSI design, and computer systems testing. He is a member of the IEEE

Computer Society and ACM. Georgi Gaydadjiev served as a program chair of SAMOS 2006, ICCD 2008, and Computing Frontiers 2009. He is the coordinator of the European Union funded integrated project SARC and a member of the HiPEAC (FP6 and FP7) European network of excellence Steering Committee.

Kees Goossens received his BSc in Computer Science from the University of Wales in 1988, and obtained his PhD from the University of Edinburgh in 1993. In his thesis he investigated the formal verification of hardware, in particular by using semi-automated proof systems in conjunction with formal semantics of hardware description languages such as ELLA and VHDL. He continued this work at several other universities before joining Philips Research in the Netherlands in 1995. At Philips, he worked on behavioral synthesis for high-throughput video processing, then on on-chip communication protocols and memory management. Until 2010, at Philips/NXP Semiconductors Research he led the team that defined the Aethereal network-on-chip for consumer electronics, where real-time performance and low cost are major constraints. He was also part-time full professor at the Delft University of Technology from 2007 to 2010, and is currently full professor at the Eindhoven University of Technology, where his research focuses on composable (virtualized), predictable (real-time), low-power embedded systems, and the systematic debugging of those systems.