Paper 2B.1 at
IEEE European Test Symposium (ETS'08)
Verbania, Italy – May 2008

Bandwidth Analysis for Reusing Functional Interconnect as Test Access Mechanism

Ardy van den Berg[2] Pengwei Ren[2*] Erik Jan Marinissen[1] Georgi Gaydadjiev[2] Kees Goossens[1,2]

[1] NXP Semiconductors
Corporate Innovation & Technology
High Tech Campus 37
5656AE Eindhoven, The Netherlands
{erik.jan.marinissen, kees.goossens}@nxp.com

[2] Delft University of Technology
Department of Computer Engineering
Mekelweg 4
2628CD Delft, The Netherlands
a.j.vandenberg-et@student.tudelft.nl g.n.gaydadjiev@ewi.tudelft.nl

Abstract

Test data travels through a System-on-Chip (SOC) from the chip pins to the module-under-test and vice versa via a Test Access Mechanism (TAM). Conventionally, a TAM is implemented with dedicated wires. 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 the silicon area. For a given module, its test set, and maximal bandwidth that the functional interconnect can offer between ATE and module-under-test, our approach designs a test wrapper for the module-under-test such that the test length is minimized. Unfortunately, it is unavoidable that with the test data also unused (idle) bits are transported. This paper presents a TAM bandwidth utilization analysis and techniques for idle bits reduction, to minimize the test length. We classify the idle bits into four types which explain the reason for 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.

1 Introduction

Rapid improvements in the semiconductor industry allow the design and manufacturing of increasingly complex chips, often referred to as SOCs. SOCs are composed of multiple, often heterogeneous modules. Each module is tested individually using on-chip isolation hardware called a wrapper. Stimuli and responses travel through the chip to and from the embedded module using a Test Access Mechanism (TAM) [1]. Conventionally, dedicated wires are used to implement this TAM.

Recently, it has been proposed to reuse existing functional interconnect, such as a bus or a NOC [2, 3], as TAM [4–12]. 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 [13], but via reused functional ports instead.

The length of an SOC test dictates the required vector storage 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 which cause under-utilization of the available bandwidth and hence contribute to a longer-than-strictly-necessary test length. These idle bits are unavoidable for a given TAM and module 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 summarizes 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 4, and describe how to reduce the amount of idle bits in Section 5. Experimental results are provided and discussed in Section 6, while Section 7 concludes this paper.

2 Prior work

Examples of papers that propose to handle on-chip transport of test data via a reused functional bus are [4, 5, 7, 9, 14]. Unfortunately, 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. [14] does

*Pengwei Ren is currently with ASML in Veldhoven, The Netherlands, e-mail: pengwei.ren@asml.com.

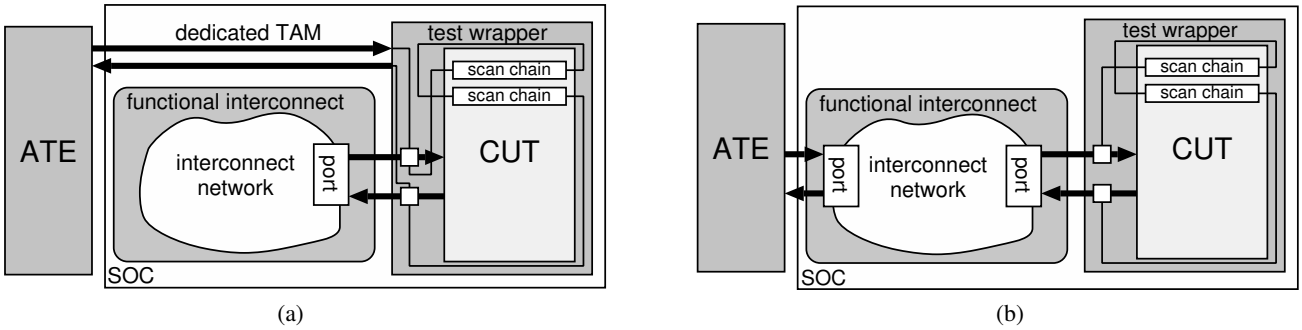

Figure 1: Test set-ups in which (a) a conventional dedicated TAM is used, and (b) the existing functional interconnect is reused as TAM.

apply structural scan-based tests via the ARM bus, but in a rather cumbersome way. Nahvi and Ivanov [6] 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 unacceptably high. Cota et al. [8] 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. [11] 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 latency [2, 15]. They describe how the streaming nature of scan testing is matched to the possibly bursty or packetized bus or NOC traffic. 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 [16, 17]. 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 module. Hussin et al. [12] 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 [11], that eliminates these idle bits, but adds significant (but in the paper unquantified) area costs.

Our paper presents a bandwidth utilization analysis for reusing functional interconnect as TAM. We work with a slightly improved version of the test wrapper design of [11], and focus on testing a single core in isolation. We provide an extensive idle-bit classification, which puts the prior work into perspective, and extends the number of idle-bit types with two novel, previously unpublished, classes. Our paper demonstrates that further test length reductions are possible through design modifications suggested by our bandwidth utilization analysis.

3 Functional interconnect as TAM

In conventional modular SOC test approaches, dedicated TAMs are used to transport test data from ATE to the module-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 wrapper design modified in comparison to conventional wrappers [11]: it lacks the dedicated-TAM input and output of conventional wrappers, but instead is equipped with ports that "speak" the communication protocol of the functional interconnect (such as AXI [18]) and convert periodically-arriving functional data into streaming scan data and vice versa.

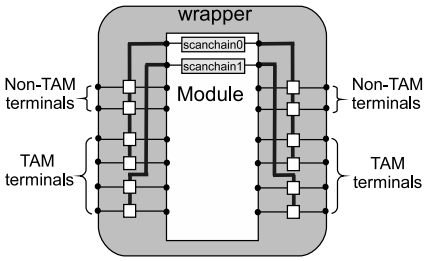

Figure 2: Simplified example wrapper.

Figure 2 shows a simplified example of a wrapper which reuses functional interconnect as TAM [11]. As in conventional test wrappers, all core terminals are equipped with a wrapper cell, and wrapper chains are formed by concatenating wrapper cells and module-internal scan chains. However, ports to connect to a dedicated TAM, common in conventional wrappers, are absent. The functional core terminals are 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 Figure 2 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 module-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 module-under-test. After the actual testing launch and capture takes place, test responses are transported away from the module in a similar fashion.

Figure 3(a) shows the typical ordering of elements in a wrapper chain for a conventional wrapper, which uses a dedicated TAM. As defined in [19], 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 3(b) 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.

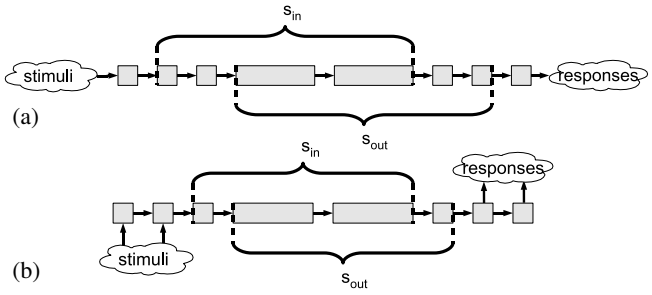

Figure 3: Typical wrapper chain for (a) a wrapper with a dedicated TAM, and (b) a wrapper which reuses functional interconnect as TAM.

Let B_{in} be the bandwidth over the functional interconnect from ATE to module-under-test, and let B_{out} be the bandwidth vice versa. The maximum number of wrapper chains wc that can be supplied through the functional interconnect with streaming scan test data (i.e., one bit per clock cycle per wrapper chain) is given by $wc = \left\lfloor \frac{\min(B_{in}, B_{out})}{f} \right\rfloor$, where f is the test frequency of the module-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 $p_{in} = \left\lfloor \frac{w_{in}}{wc} \right\rfloor$. The responses are handled likewise, with $p_{out} = \left\lfloor \frac{w_{out}}{wc} \right\rfloor$.

4 Idle bits classification

It is often unavoidable that some non-useful bits are transported together with the useful test data. These non-useful bits are called *idle bits*; they 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. For the latter case, this section describes and classifies four types of idle bits.

- Type-1: different scan chain lengths within a module [17];
- Type-2: scan-in (scan-out) length is not an exact multiple of the input (output) period;
- Type-3: 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 [11, 12].

4.1 Type-1 idle bits

Shifting bits into the wrapper chains completes when the wrapper chain i with the longest scan-in length $s_{in,i}$ 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 [17] 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 4 shows two wrapper chains of unequal length and the corresponding Type-1 idle bits for this example.

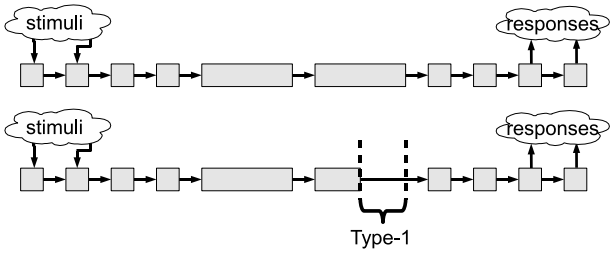

Figure 4: The cause of Type-1 idle bits: multiple wrapper chains with unequal scan-in and/or scan-out length.

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

$$ib_{in}^{1} = \sum_{i=1}^{tests} \sum_{j=1}^{wc} ((S_{in} - s_{in,j}) \cdot pat_i) \tag{4.1}$$

$$ib_{out}^{1} = \sum_{i=1}^{tests} \sum_{j=1}^{wc} ((S_{out} - s_{out,j}) \cdot pat_i) \tag{4.2}$$

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

4.2 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: $S_{in} = 5$ and $p_{in} = 2$ results in one idle bit of Type-2. This example is visualized in Figure 5. These type of idle bits occur at the output side for responses as well.

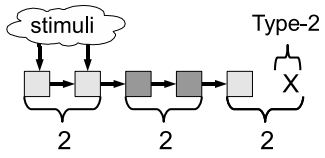

Figure 5: Type-2 idle bits at the input side, caused by the period at which stimuli arrive.

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

$$ib_{in}^{2} = \sum_{i=1}^{tests} \left(S'_{in} - S_{in}\right) \cdot pat_i \cdot wc \tag{4.3}$$

$$ib_{out}^{2} = \sum_{i=1}^{tests} \left(S'_{out} - S_{out}\right) \cdot pat_i \cdot wc \tag{4.4}$$

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

4.3 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., $S'_{in} = S'_{out}$. 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 $S'_{out} < S'_{in}$, 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 $S'_{in} < S'_{out}$, Type-3 idle bits are shifted in before the valid stimuli.

Figure 6 shows for a small example with only two test patterns the idle bits of Type-1, -2, and -3. $S'_{in} = 12$, while $S'_{out} = 8$, and hence $S'_{out} < S'_{in}$. Four Type-3 idle bits are injected at the response side for each wrapper chain and for each test pattern except the last pattern.

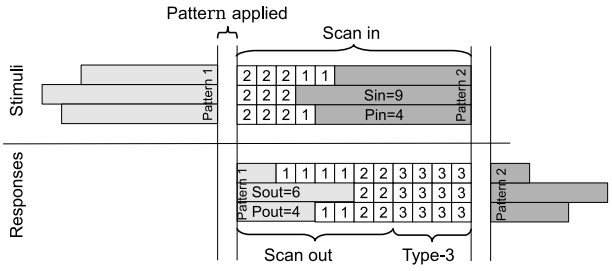

Figure 6: Example of Type-3 idle bits: shifting in stimuli does not take the same time as shifting out responses.

Type-3 idle bits are quantified as follows:

$$ib_{in}^{3} = \sum_{i=1}^{tests} \max\left(0, S'_{out} - S'_{in}\right) \cdot \left(pat_i - 1\right) \cdot wc \tag{4.5}$$

$$ib_{out}^{3} = \sum_{i=1}^{tests} \max\left(0, S'_{in} - S'_{out}\right) \cdot \left(pat_i - 1\right) \cdot wc \tag{4.6}$$

4.4 Type-4 idle bits

The number of wrapper chains wc is as large as possible, in order to make maximum use of the available 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 $p_{in} = \left\lfloor \frac{w_{in}}{wc} \right\rfloor$ 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, $(w_{in} \bmod wc)$ 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 [11] were referred to as RSDI and RSDO cells. In [11, 12], 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 [11, 12].

An example of Type-4 idle bits is shown in Figure 7. An existing functional input port that serves as TAM has word width $w_{in} = 32$. Given the bandwidths, in this example we could afford to make $wc = 10$ wrapper chains, and hence words are delivered with a period of $p_{in} = 3$ clock cycles. The 32 input bits are divided over 10 wrapper chains and hence we have $(w_{in} \bmod wc) = 2$ RSDI wrapper cells; in Figure 7 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.

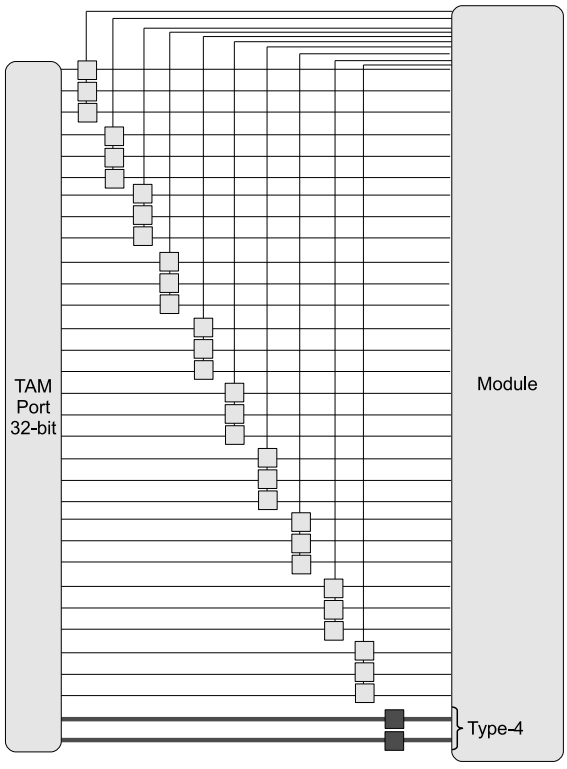

Figure 7: Type-4 idle bits: data going to or coming from selected data cells which are not in a wrapper chain (dark cells).

Type-4 idle bits are present for all patterns of all tests. They occur

for every word delivered, apart for the very last word of each pattern on the stimulus side and the very first word of each pattern on the response side.

$$ib_{in}^{4} = \sum_{i=1}^{tests} \left(\left\lceil \frac{\max(S'_{in}, S'_{out})}{p_{in}} \right\rceil - 1 \right) \cdot pat_i \cdot (w_{in} \bmod wc) \tag{4.7}$$

$$ib_{out}^{4} = \sum_{i=1}^{tests} \left(\left\lceil \frac{\max(S'_{in}, S'_{out})}{p_{out}} \right\rceil - 1 \right) \cdot pat_i \cdot (w_{out} \bmod wc) \tag{4.8}$$

5 Idle bit reduction

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 [19] for this purpose. Obviously, modules 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 module are *soft*, such that they can be re-designed and adapted to wrapper and TAM design.

There are no Type-2 idle bits if $(S_{in} \bmod p_{in}) = 0$ and $(S_{out} \bmod p_{out}) = 0$. 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 preferable 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 more inputs or more 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.

There are no Type-4 idle bits if all stimuli from the TAM and responses to the TAM are buffered. Hussin et al. [12] propose a solution in which load and shift registers are used to buffer. Unfortunately, this solution requires a significant, but in their paper unquantified, amount of extra silicon area.

6 Experimental results

We have automated the wrapper design to generate wrappers for modules using the approach of Amory et al. [11] and calculated the bandwidth under-utilization for each module due to idle bits. The wrapper generator uses 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 [20]. For each module 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 [18]; modules with a big amount of i/o-terminals will therefore have a bigger bandwidth compared to modules with a small amount of i/o-terminals. The test frequency $f_{test} = 100$ MHz. Today's functional interconnects can work at a frequency of 500 MHz [2]. 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 $b = 32 \cdot 500 \cdot 0.8 \cdot \frac{1}{8}$ = 1600 Mbit/s per port.

For 26 modules of the ITC'02 benchmarks, we have calculated the bandwidth under-utilization due to idle bits. Table 1 lists the results. For each module, 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, Module 1 of SOC p93791 requires 312067 idle bits to transport all stimuli and responses to and from the module. 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 modules are given in the bottom row of the table.

SOC	Module	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%**

Table 1: Bandwidth analysis for 26 modules of the ITC'02 SOC Test Benchmarks [20].

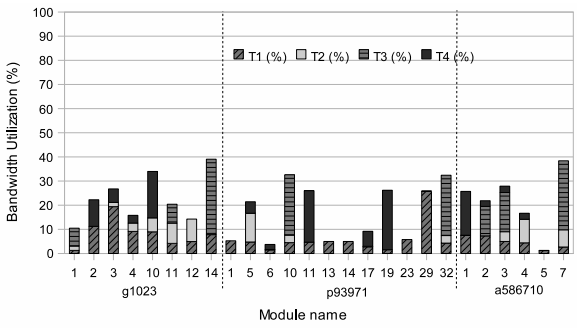

Figure 8: Bandwidth under-utilization due to idle bits.

Figure 8 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.

7 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 functional interconnect as a TAM. We defined four types of idle bit to explain the under-utilization of bandwidth. Reduction of idle bits improves the bandwidth utilization and reduces the required ATE vector storage. Several solutions to reduce idle bits are discussed.

We automatically generated wrappers for 26 modules 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. On average, 20% of the available bandwidth is consumed by idle bits. All four types of idle bits contribute to the bandwidth under-utilization.

Using the bandwidth under-utilization analysis, wrappers which reuse the existing functional interconnect can be modified to reduce the test length of modules.

Acknowledgements

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.

References

[1] Yervant Zorian, Erik Jan Marinissen, and Sujit Dey. Testing Embedded-Core Based System Chips. In *Proceedings IEEE International Test Conference (ITC)*, pages 130–143, Washington, DC, USA, October 1998.

[2] Kees Goossens, John Dielissen, and Andrei Rădulescu. The Æthereal network on chip: Concepts, architectures, and implementations. *IEEE Design & Test of Computers*, 22(5):414–421, September-October 2005.

[3] Giovanni De Micheli and Luca Benini, editors. *Networks on Chips: Technology and Tools*. The Morgan Kaufmann Series in Systems on Silicon. Morgan Kaufmann, July 2006.

[4] Peter Harrod. Testing Reusable IP - A Case Study. In *Proceedings IEEE International Test Conference (ITC)*, pages 493–498, Atlantic City, NJ, USA, September 1999.

[5] Jing-Reng Huang, Madhu K. Iyer, and Kwang-Ting Cheng. A Self-Test Methodology for IP Cores in Bus-Based Programmable SOCs. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 198–203, Marina del Rey, CA, USA, May 2001.

[6] Mohsen Nahvi and Andre Ivanov. A Packet Switching Communication-Based Test Access Mechanism for System Chips. In *Digest of Papers of IEEE European Test Workshop (ETW)*, pages 195–200, Saltsjobaden, Sweden, May 2001.

[7] Sungbae Hwang and Jacob A. Abraham. Reuse of Addressable System Bus for SOC Testing. In *Proceedings IEEE International ASIC/SOC Conference*, pages 215–219, Washington, DC, USA, September 2001.

[8] Érika Cota et al. The Impact of NoC Reuse on the Testing of Core-based Systems. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 128–133, Napa, CA, USA, April 2003.

[9] Alexandre M. Amory, Leandro A. Oliveira, and Fernando G. Moraes. 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)*, pages 174–179, Darmstadt, Germany, December 2003.

[10] Alexandre M. Amory and Érika Cota and Marcelo Lubaszewski and Fernando G. Moraes. Reducing Test Time With Processor Reuse in Network-on-Chip Based Systems. In *Proceedings Brazilian Symposium on Integrated Circuits and System Design (SBCCI)*, pages 111–116, Pernambuco, Brazil, September 2004.

[11] Alexandre M. Amory, Kees Goossens, Erik Jan Marinissen, Marcelo Lubaszewski, and Fernando Moraes. Wrapper Design for the Reuse of a Bus, Network-on-Chip, or other Functional Interconnect as Test Access Mechanism. *IET Computers and Digital Techniques*, 1(3):197–206, May 2007.

[12] Fawnizu Azmadi Hussin and Tomokazu Yoneda and Hideo Fujiwara. Optimization of NoC Wrapper Design under Bandwidth and Test Time Constraints. In *Proceedings IEEE European Test Symposium (ETS)*, pages 35–42, Freiburg, Germany, May 2007.

[13] Francisco DaSilva, editor. *IEEE Std 1500™-2005, IEEE Standard Testability Method for Embedded Core-based Integrated Circuits*. IEEE, New York, NY, USA, August 2005.

[14] Chris Feige, Jan ten Pierick, Clemens Wouters, Ronald Tangelder, and Hans G. Kerkhoff. Integration of the Scan-Test Method into an Architecture Specific Core-Test Approach. *Journal of Electronic Testing: Theory and Applications*, 14(1–2):125–131, February 1999.

[15] John Mark Nolen and Rabi N. Mahapatra. Time-division-multiplexed test delivery for NoC systems. *IEEE Design & Test of Computers*, 25(1):44–51, April 2008.

[16] Erik Jan Marinissen and Sandeep Kumar Goel. 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)*, pages 52–60, Brno, Czech Republic, April 2002.

[17] Sandeep Kumar Goel and Erik Jan Marinissen. SOC Test Architecture Design for Efficient Utilization of Test Bandwidth. *ACM Transactions on Design Automation of Electronic Systems*, 8(4):399–429, October 2003.

[18] ARM. *AMBA AXI Protocol Specification*, June 2003.

[19] Erik Jan Marinissen, Sandeep Kumar Goel, and Maurice Lousberg. Wrapper Design for Embedded Core Test. In *Proceedings IEEE International Test Conference (ITC)*, pages 911–920, Atlantic City, NJ, USA, October 2000.

[20] Erik Jan Marinissen, Vikram Iyengar, and Krishnendu Chakrabarty. A Set of Benchmarks for Modular Testing of SOCs. In *Proceedings IEEE International Test Conference (ITC)*, pages 519–528, Baltimore, MD, USA, October 2002.