

# On Reliable Modular Testing with Vulnerable Test Access Mechanisms

Lin Huang, Feng Yuan and Qiang Xu  
Department of Computer Science & Engineering  
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong  
Email: {lhuang,fyuan,qxu}@cse.cuhk.edu.hk

## ABSTRACT

*In modular testing of system-on-a-chip (SoC), test access mechanisms (TAMs) are used to transport test data between the input/output pins of the SoC and the cores under test. Prior work assumes TAMs to be error-free during test data transfer. The validity of this assumption, however, is questionable with the ever-decreasing feature size of today's VLSI technology and the ever-increasing circuit operational frequency. In particular, when functional interconnects such as network-on-chip (NoC) are reused as TAMs, even if they have passed manufacturing test beforehand, failures caused by electrical noise such as crosstalk and transient errors may happen during test data transfer and make good chips appear to be defective, thus leading to undesired test yield loss. To address the above problem, in this paper, we propose novel solutions that are able to achieve reliable modular testing even if test data may sometimes get corrupted during transmission with vulnerable TAMs, by designing a new "jitter-aware" test wrapper and a new "jitter-transparent" ATE interface. Experimental results on an industrial circuit demonstrate the effectiveness of the proposed technique.*

## Categories and Subject Descriptors

B.7.3 [Integrated Circuits]: Reliability and Testing

## General Terms

Reliability, Design.

## Keywords

Modular Testing, Test Access Mechanisms, Reliable Test

## 1. INTRODUCTION

Modular testing is the most popular test strategy for large system-on-a-chip (SoC) as it reduces the complexity of SOC test problem in a "divide and conquer" manner [25]. In modular testing, an embedded core is isolated from its surrounding logic using a test wrapper, and test data is transported via test access mechanisms (TAMs). Various TAM designs have been proposed in the literature, and they can be broadly categorized as either *dedicated bus-based access* or *functional access*. In dedicated bus-based access scheme, specific test access structure are introduced for test data transfer; while

in functional access scheme, functional interconnects as reused as TAMs and test data are transported along them. The latter scheme has become increasingly popular since embedded cores are already well-connected (e.g., functional buses or network-on-chip (NoC) [6]) and this strategy saves the design effort and silicon cost associated with dedicated TAMs [2].

To the best of our knowledge, all existing work in modular testing assumes the TAMs to be error-free. While this might be valid for the dedicated bus-based TAMs when its operational speed is slow. For the case when at-speed functional interconnects are reused as TAMs or when the test buses operate at hazardous rate, however, with the ever-decreasing feature size and ever-increasing operational frequency, the "error-free" assumption is questionable. This is because, even if these TAMs have passed the manufacturing test beforehand, failures caused by crosstalk, IR drop, and even alpha particle hits may still happen during the test data transfer process. While various fault tolerance techniques have been proposed to address the reliability issues of on-chip bus (e.g., [19]) or on-chip network (e.g., [11, 21, 23]) in normal functional mode, they are not readily applicable for reliable modular testing with vulnerable TAMs. This is mainly due to the stringent timing requirements of the automatic test equipments (ATEs). Although these fault tolerant schemes are able to eliminate data transmission errors, the associated traffic jitter and variable test bandwidth will invalidate the testing of embedded cores, if the manufacturing test process does not understand the TAM's fault-tolerant features. *Therefore, good chips may fail manufacturing test when vulnerable TAMs are used for test data transfer, leading to undesired test yield loss.*

We propose novel solutions to address the above problem, which are able to achieve reliable testing even if test data may sometimes get corrupted during transmission with vulnerable TAMs. The main contributions of this paper are as follows:

- we propose a "jitter-aware" test wrapper design for embedded cores, which is able to manage the traffic jitter during test data transfer, by taking the fault-tolerant features of TAMs into account.
- we also present an on-chip "jitter-transparent" ATE interface that is able to accommodate the bandwidth mismatch between the ATE with constant test data transfer rate and the chip under test with variable TAM operational rate.

*To the best of our knowledge, this is the first work that considers the reliability of the test access mechanisms in modular testing of SoC devices.* The remainder of this paper is organized as follows. Section 2 reviews prior work and presents the motivation. Section 3 then discusses the impact of fault tolerance schemes on test reliability. Next, the proposed "jitter-aware" test wrapper design and "jitter-transparent" ATE interface design are detailed in Section 4. Section 5 presents experimental results and discussions. Finally, Section 6 concludes this paper.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2008, June 8–13, 2008, Anaheim, California, USA.  
Copyright 2008 ACM ACM 978-1-60558-115-6/08/0006 ...\$5.00.

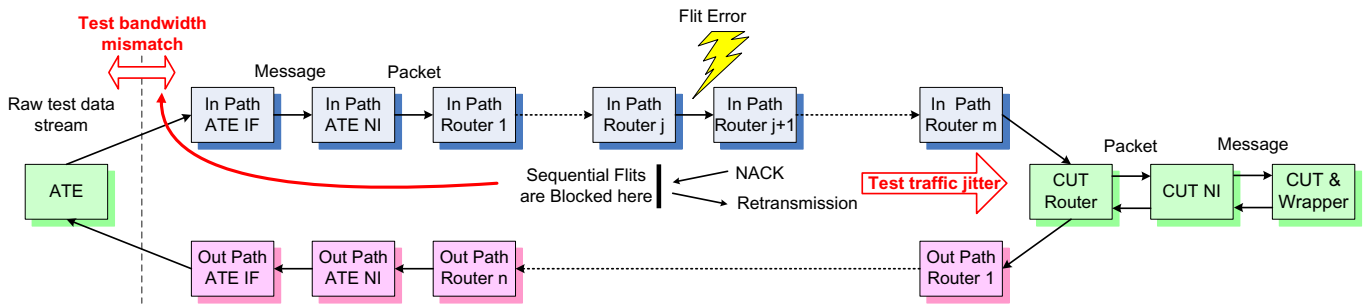


Figure 1: Test Data Journey when NoC is Reused as TAM.

## 2. PRIOR WORK AND MOTIVATION

One of the main tasks in modular SOC testing is to design an efficient TAM to link the test sources and sinks to the core under test (CUT) [28]. While the most widely used solution is still to introduce dedicated bus-based TAMs (e.g., Test Bus [24] or TestRail [20]) nowadays, with more functional interconnect resources available on-chip for today’s communication-centric design [9], it is becoming more popular to reuse them as TAMs to save the silicon area cost and routing overhead associated with dedicated bus-based TAMs [26]. For example, almost all existing work on testing NoC-based systems advocated to reuse the on-chip network itself as test access mechanism (e.g., [1, 2, 8, 14, 17, 18]) and some researchers also proposed to reuse the on-chip buses for test data transfer (e.g., [12, 15]). Moreover, with embedded processors widely utilized in SoCs, a number of software-based testing approaches that utilize the computational power of such processors and their easy access to other embedded cores have been presented in the literature (e.g., [3, 13, 16]). These methods also rely on the functional paths (e.g., buses) to deliver test data. For the functional interconnect, a “core test” is simply treated as a regular application running on it.

During manufacturing test, the tester typically inject/collect test data to/from the chip at constant rate, and the test data are expected to be transported to the CUTs in a lossless and zero-jitter manner. Existing work assumes that TAMs are error-free during the test data transfer process once they pass the manufacturing test. With the relentless scaling of CMOS technology, however, guaranteeing error-free information transfer on global on-chip wires becomes increasingly difficult due to the following reasons [10]: (i) energy and device reliability concerns impose designs with small logic swings and hence circuits are designed with less noise margin inherently; (ii) electrical noise due to crosstalk, electromagnetic interference, and radiation-induced charge injection becomes more severe with technology advancement. Therefore, while the “error-free” TAM assumption may still hold true in most cases when dedicated bus-based TAMs are used and they operate at low speed, it is questionable when at-speed functional interconnects such as NoC are reused as TAMs or the dedicated TAMs transfer test data at high frequency.

To deal with the situation that transmitting digital values on wires will be inherently unreliable and nondeterministic, various fault tolerance techniques have been presented in the literature for on-chip buses (e.g., [19]) and on-chip network (e.g., [11, 21, 23]), in which the mainstream technique is the retransmission scheme. While the above technique effectively eliminates data transmission errors in functional mode, it can cause other problems for reliable modular testing with vulnerable TAMs. That is, the associated test traffic jitter and variable test bandwidth can violate the stringent timing requirements of the ATE (detailed in Section 3).

For ease of discussion, in this paper, we focus on the case of testing NoC-based systems when the vulnerable NoC itself is reused as TAM<sup>1</sup>. In such case, new design-for-test (DfT) modules need to be developed to transfer test data through the on-chip network. Embedded cores typically use standard protocols (e.g., OCP [22]) to communicate with each other. Since ATE does not understand such protocols, Amory *et al.* [1] introduced a so-called *ATE Interface* DfT module on-chip to conduct the protocol translation. Test wrapper design is also different from the one when dedicated TAMs are used, since it also needs to do protocol conversion and buffering in addition to balancing the wrapper scan chains. Several wrapper designs (e.g., [2, 14, 17]) have been presented in the literature with different area cost and testing time implications. With the help of the above DfT structures, the path that the test data take between the ATE and a CUT is depicted in Fig. 1. On the input path, the test stimuli coming out of ATE are broken into packets and subsequently into flits and transferred along the on-chip network via the ATE interface, the network interfaces (NIs) and several routers to the CUT; while on the output path, the test responses are integrated and delivered to the tester to compare with the expected values.

When flit errors occur during the test data transfer process, with retransmission scheme, the receiver detects the error and signals the sender to request a retransmission. This brings the following problems (detailed in Section 3): (i) The erroneous test flit, together with its following flits, cannot arrive at the test wrapper at the expected time. If the test wrapper is not aware of this traffic jitter, test data will be corrupted during core test; (ii) The retransmission scheme also results in variable on-chip test data transfer rate. Since many ATEs simply inject/collect test data to/from the CUTs at constant rate without any handshaking capabilities, the possible bandwidth mismatch may also lead to corrupted test data.

From the above, good chips may fail manufacturing test and appear to be defective when vulnerable TAMs are used to transport test data. To see the significance of the problem, we estimate the potential test yield loss caused by flit errors by doing some simple calculations. Given  $N$  is the number of flits in the entire test data volume,  $\lambda$  is the flit error rate, the potential yield loss can be expressed as

$$YieldLoss = 1 - (1 - \lambda)^N \quad (1)$$

When  $\lambda = 10^{-9}$  and flit size is 32 bits, the test yield loss for the the chip contains 21.5M gates [4] is 11.47%. This problem is exacerbated with the increase of flit error rate and/or the test data volume. Therefore, how to achieve reliable modular testing with vulnerable TAMs is an interesting and relevant research problem, which motivates this work.

<sup>1</sup>Similar techniques can be applied/adapted for the case when on-chip buses or other functional interconnects are reused as TAMs or for the case when test data are transferred on dedicated TAMs at hazardous frequency.

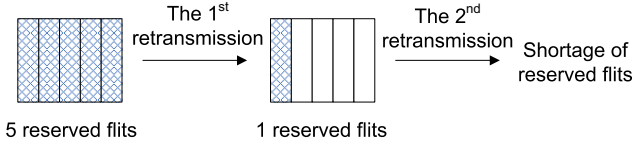


Figure 2: Problem with Buffer-Only Solution.

### 3. THE IMPACT OF FAULT TOLERANCE SCHEMES ON TEST RELIABILITY

To address the reliability issues in on-chip interconnects caused by channel disturbances such as crosstalk and transient faults, designers mainly resort to two techniques: correction or retransmission. It is obvious that test data cannot be corrupted if the error correction scheme is capable to fully recover the error. However, the associated cost of such techniques (e.g., forward error correction [27]) in terms of area and energy-efficiency is quite high and hence retransmission scheme is the mainstream technique utilized nowadays. Also, even for the case when single error correction is employed, since the probability of a double (or higher) error within a single flit may not be insignificant due to crosstalk, typically a hybrid technique that provide both error correction and retransmission is utilized [23]. We therefore mainly focus on the impact of retransmission scheme on reliable testing of SoC devices in this work.

The main problem that the retransmission scheme brings is test traffic jitter, which violates the strict timing requirement for manufacturing test. Suppose a flit error occurs during transmission between two routers on the input access path, with end-to-end or hop-by-hop retransmission scheme, the receiver detects the error and signals the sender to request a retransmission. This test flit, together with its following flits, cannot arrive at the test wrapper at the expected time. If the test wrapper is not aware of the traffic jitter caused by retransmission scheme, it will suffer from input starvation and will load one or more flits consisting of meaningless bits from the input access path, and hence the test stimulus will be corrupted. Consequently, the corresponding test response after application, is most likely to be different from the correct one, and the tester will mark the chip as defective despite the fact that the chip may be functionally correct. Similarly, when a flit error happens at the output access path, while the error in this particular test flit will be corrected with retransmission, the remaining flits can be overwritten during the scan process when the scan out path is blocked by this particular flit while the following test stimulus are scanned in without knowing this problem. Again, the test data are corrupted, resulting in possible test yield loss.

While introducing buffers at the test wrapper (e.g., [1]) can alleviate the traffic jitter problem, it costs more silicon area and it is also not a scalable solution for this problem (i.e., more buffers are required with the increase of flit error rate and test data volume). To see this clearly, let us examine an example as follows: Given the extra delay caused by one retransmission is 40 cycles and the flit injection rate is 0.1 flits per cycle, and a 5-flit buffer is implemented in the test wrapper, as shown in Fig. 2. Suppose a flit error is detected on the input access path, the arrival of all the following flits at the test wrapper will be postponed. As the test data are initially loaded when the buffer is filled up, we can tolerate one flit error, but the number of buffered flits is reduced to  $5 - 40 \times 0.1 = 1$  after one retransmission and it is not recoverable because the ATE injects test data to the CUT at constant rate. When flit error rate is

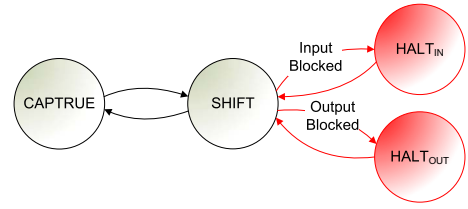


Figure 3: State Diagram for the Wrapper in Test Mode.

high and/or the total number of test flits required to test this core is large, another flit error may occur and require retransmission, then it is unavoidable that corrupted test data are loaded and applied, leading to undesired test yield loss.

The retransmission scheme also involves another type of test reliability concern besides the test traffic jitter problem. That is, when retransmission happens, the test data transfer is stalled and hence the total test data volume that can be processed on-chip is reduced. If the ATE injects/collects test data at constant rate without considering this problem, the bandwidth difference will also result in test data corruption. Unfortunately, many ATEs work in a rigid “streaming” mode and do not have any handshaking mechanism to stop transferring test data. Even for those high-end ATEs that are able to conduct handshaking with the CUTs, designers may select not to use this feature because it typically costs more testing time due to the more complex communication protocol and designers need to introduce extra test pins for these handshaking signals (could be a problem for those pin-constrained designs). Again, introducing buffers at the ATE interface can alleviate this bandwidth mismatching problem, but it is not a scalable solution with the increase of flit error rate and/or test data volume.

From the above, we can conclude that the most widely-used on-chip fault tolerance scheme, i.e., the retransmission scheme, has a significant impact on test reliability, and simply employing more buffers to mitigate this problem is not a good solution. We therefore propose novel DFT solutions to tackle this problem, including a “jitter-aware” test wrapper design and a “jitter-transparent” ATE interface design, as described in the following section.

## 4. PROPOSED SOLUTION FOR RELIABLE MODULAR TESTING

### 4.1 “Jitter-Aware” Test Wrapper Design

Different from mitigating traffic jitter by using buffers, the basic idea behind the proposed test wrapper design is to halt the scan process by gating the scan clock signal when necessary. That is, when compared with the previous designs, for scan tested cores, besides the normal *SHIFT* and *CAPTURE* states, we introduce two extra states: *HALT<sub>IN</sub>* state for the case that the test vector flits come late to the wrapper due to retransmission on input path, and *HALT<sub>OUT</sub>* state for the case that new test response flits cannot be transmitted because the test traffic is blocked on the output access path (as shown in Fig. 3). By using this method, suppose we are able to notify the ATE to start/stop the continuous test data stream, the test yield loss can be completely eliminated.

The proposed core wrapper architecture for reliable modular testing is depicted in Fig. 4, assuming an embedded core in NoC-based systems connect to its NI via bidirectional channels and it communicates with other cores using OCP protocol<sup>2</sup> [22]. Inside the test

<sup>2</sup>In OCP, the master issues a command (e.g., read, write) by using *MCmd* signal, and the slave responds this command by issuing *SCmdAccept* signal.

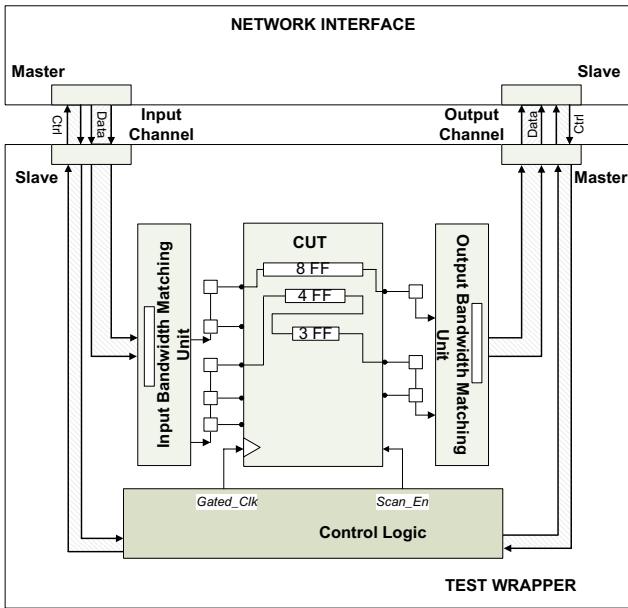


Figure 4: Wrapper Architecture for Reliable Modular Testing.

wrapper, the *data terminals* of the *input channel* are connected to a so-called *input bandwidth matching unit*, used to match the difference between the NoC bit-width assigned for testing this particular core and the number of wrapper scan chains. Similarly, the *data terminals* of the *output channel* are connected to a so-called *output bandwidth matching unit* for data width conversion. Existing designs such as the one proposed in [14] can be utilized to conduct this data width conversion.

As the key element of the proposed wrapper design, we use a new *control logic block* to generate the control signals inside the test wrapper. Note that, because these control signals are generated locally, the probability of error occurrence in the control logic block is negligible. As can be observed in Fig. 5, this logic block is composed of three main components: input & output control, wrapper kernel control, and clock division (optional). We present them in details as follows.

**Input & Output Control:** The input & output control units are used to monitor the test traffic and determine whether the wrapper should transit to *HALT<sub>IN</sub>* state or *HALT<sub>OUT</sub>* state. In normal scan shift process, for the input control unit, it keeps the *IPath\_Blocked* signal to be low and asserts *SCmdAccept\_In* when the NI issues a write request *MCmd\_In* to write test stimuli to the CUT via the *input channel*; Similarly, for the output control unit, it keeps the *OPath\_Blocked* signal to be low and asserts *MCmd\_Out* to notify the NI to write test response out via the *output channel*. When flit error occurs and retransmission happens in either input path or output path, the corresponding unit detects the temporary stall of test data by monitoring the buffer status of the input/output bandwidth matching unit and the *MCmd\_In* or *SCmdAccept\_Out* signal, and asserts *IPath\_Blocked* or *OPath\_Blocked* signal to notify the wrapper kernel control unit. Later, after receiving the confirmed *Block\_Ctrl* signal, it de-asserts the *SCmdAccept\_In* or *MCmd\_Out* signal to stop test data transfer from/to the NI. When the retransmission completes, similarly, the input/output control unit detects this event and de-asserts *IPath\_Blocked* or *OPath\_Blocked* signal, and the wrapper kernel control unit then controls the test to go back to normal scan shift or capture mode.

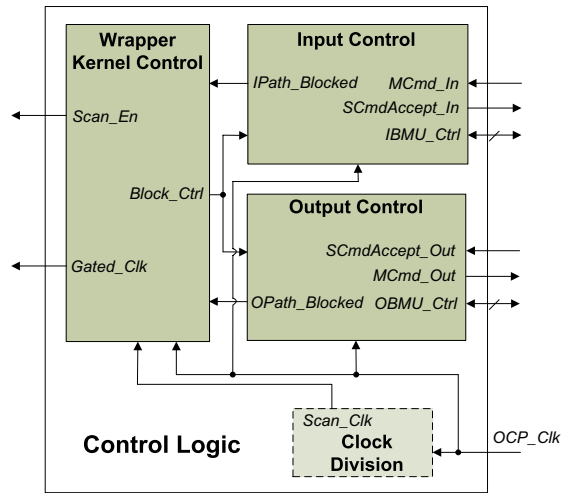


Figure 5: Block Diagram of Control Logic.

**Wrapper Kernel Control:** The wrapper kernel control unit implements the finite state machine shown in Fig. 3 and generates the scan enable signals (*Scan\_En*) and the gated scan clock signal (*Gated\_Clk*) (see the timing diagram in Fig. 6). When *IPath\_Blocked* or *OPath\_Blocked* is asserted, it enters the corresponding *HALT<sub>IN</sub>* or *HALT<sub>OUT</sub>* state and asserts *Block\_Ctrl* to confirm the state transition. The *Scan\_Clk* signal is then gated to generate the *Gated\_Clk* to halt the scan shift process (as shown in Fig. 6).

**Clock Division:** Typically the test data are shifted at lower frequency during scan test. This optional clock division block is used to provide this low-speed scan clock signal *Scan\_Clk*, generated by dividing the high-frequency OCP clock signal *OCP\_Clk*.

## 4.2 “Jitter-Transparent” ATE Interface Design

If the ATE is provided with handshaking capability with the chip under test and is able to start/stop test data injection/collection in real-time, with the help of the proposed wrapper design presented in section 4.1, the test yield loss due to vulnerable TAMs can be completely eliminated. Unfortunately, many ATEs operate in a stream mode and do not have this capability. Even for those high-end ATEs that have this feature, designers may not use it for testing time and/or pin count considerations. While it is impossible to avoid test yield loss completely in such case, in this section, we present an ATE interface design to minimize it at the cost of a slight increase in testing time.

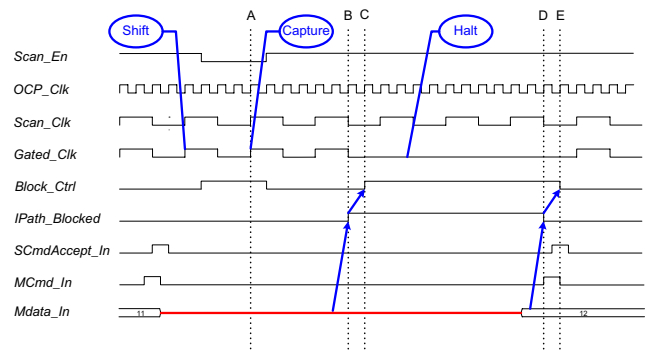


Figure 6: Timing Diagram when Retransmission Happens.

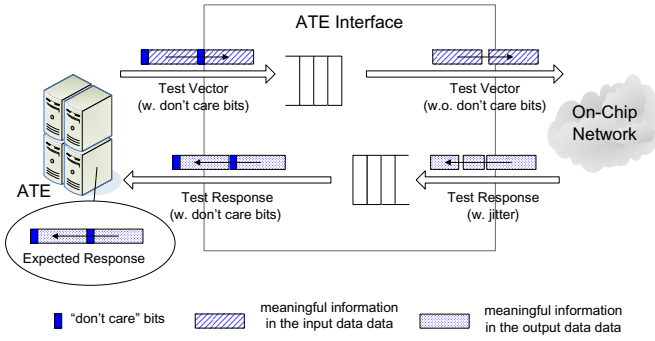


Figure 7: ATE Interface in Operation.

When retransmission due to flit error occurs, the test data volume that can be processed on-chip is reduced. If the ATE is not aware of it and keeps on injecting test data at constant rate, the test data blocked on the TAM will be overwritten and become corrupted. To deal with this situation, designers can rely on introducing buffers at the ATE interface solely to mitigate this problem. However, with the increase of flit error rate and/or test data volume, the required buffer size increases dramatically and hence it is not a scalable solution. In our design, we keep the buffer size to the minimum that is able to tolerate the extra delay caused by one retransmission. Then, given the flit error rate  $\lambda$ , the test yield loss in such case can be calculated as follows:

$$YieldLoss_B = 1 - (1 - \lambda)^N - N\lambda(1 - \lambda)^{N-1} \quad (2)$$

where,  $N = \frac{V_{test}}{flit_{size}}$  is the number of flits required to transport the entire test data volume  $V_{test}$  with the size of a flit equals  $flit_{size}$ .

Based on the above equation, suppose the flit error rate  $\lambda = 10^{-7}$ , for a large circuit with 6.3M gates in [4], the test yield loss could be as high as 44.2%. To further decrease the test yield loss, instead of adding more buffers, we propose to divide the whole input test data flow into  $r$  segments and add a small section of “don’t-care” (useless) data at the end of each segment in the ATE channel memory. When such a mixed data flow is injected into the ATE interface, only the meaningful bits (i.e., the actual test data) are forwarded to the associated network interface; the meaningless bits (i.e., “don’t-care” bits), in contrast, are discarded (see Fig. 7). Because we add the “don’t-care” bits at regular interval, a simple counter can be utilized to control the useless data not to be written into the ATE interface buffer. The purpose of the above strategy is to create a test data flow gap to cancel the influence of the extra delay caused by potential retransmission. To accommodate the delay caused by one retransmission, the size of each “don’t-care” section is set to be equal to that of the buffer in ATE interface. Therefore, as long as the test data transfer in each segment does not incur more than one flit error, reliable testing can be achieved. As the number of flits in a segment is much smaller than that in the entire test, the test yield loss can be significantly reduced and is given as:

$$YieldLoss_{B\&S} = 1 - \left( (1 - \lambda)^{\frac{N}{r}} + \frac{N}{r} \lambda (1 - \lambda)^{\frac{N}{r} - 1} \right)^r \quad (3)$$

With the above strategy, test stimuli are injected from ATE with “don’t-care” bits at regular interval. To accommodate this test bandwidth reduction at the input side, similarly, for the test responses collection and comparison at the output side, as can be seen in Fig. 7, we also need to introduce “don’t-care” bits in the ATE channel memory and we need a buffer that is able to tolerate one retransmission to shape the test response so that it can be compared with the expected value at the exact time.

## 5. EXPERIMENTAL RESULTS

To evaluate the DfT area cost of the proposed “jitter-aware” wrapper, we implement the wrapper design for the core shown in Fig. 4 and synthesize it using a commercial 90nm CMOS technology, the total size of the wrapper is 838 two-input NAND equivalent gates. It should be noted that the input & output bandwidth matching units in this wrapper are designed according to the type II wrapper in [14] for data width conversion, which occupies a large portion of the wrapper area.

If the ATE is able to handshake with the CUT to start/stop test data transfer, we can effectively eliminate the test yield loss caused by vulnerable TAMs with the help of the proposed wrapper design, without the necessity to introduce other DfT logic. However, since this handshaking capability is not available in most cases, we need the help of the on-chip ATE interface to mitigate test yield loss due to vulnerable TAMs. To evaluate the proposed solution in Section 4.2, given flit error rate and test data volume, we analyze the design tradeoffs in terms of DfT area overhead, yield loss mitigation, and testing time. To demonstrate our results, we use an industrial circuit shown in [5] with 2.6M gates, 274K scan cells and 106M compressed scan test data volume as an embedded core to be tested with vulnerable TAMs. The test data is assumed to be injected from ATE to the chip at 0.25 flits per cycle with flit size  $flit_{size} = 32$  bits, and the on-chip traffic is protected by the end-to-end retransmission scheme. According to [21, 23], the NoC communication latency is in the range of tens of clock cycles (sometimes even more than 100). In our experiments, the extra delay caused by a retransmission is assumed to be 40 cycles.

As shown in Fig. 8, when the NoC is designed with a relatively high flit error rate of  $10^{-7}$  for energy savings [7], if we do not introduce any DfT logic to mitigate test traffic jitter, the test yield loss can be as high as 28.20%. Apparently, this is not acceptable. If the proposed wrapper is utilized and a buffer that is able to tolerate one retransmission is introduced in the ATE interface, the test yield loss reduces to 4.41%, which is still quite large. If we divide the entire test into 100 segments and add “don’t-care” bits at the end of each segment, the yield loss becomes roughly 0.05%, at the cost of slightly increased testing time associated with the “don’t care” bits (around 0.03%).

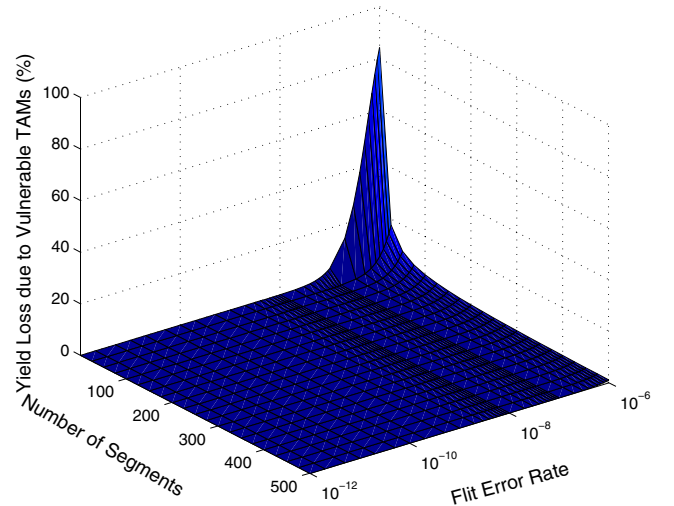


Figure 8: Test Yield Loss for a Core with 2.6 Million Gates.

$\lambda$	Yield Loss $\leq 1\%$			Yield Loss $\leq 0.5\%$		
	$n_b$ (flits)	$n_p$ (flits)	$\Delta T_p$	$n_b$ (flits)	$n_p$ (flits)	$\Delta T_p$
$1 \times 10^{-8}$	20	20	0.0003%	20	20	0.0003%
$5 \times 10^{-8}$	40	20	0.0006%	40	20	0.0009%
$1 \times 10^{-7}$	40	20	0.0018%	40	20	0.0033%
$5 \times 10^{-7}$	100	20	0.0411%	120	20	0.0824%
$1 \times 10^{-6}$	160	20	0.1642%	180	20	0.3297%

$\lambda$ : Flit error rate.

$n_b$ : The buffer size for barely using buffers to handle the latency.

$n_p$ : The buffer size for the proposed design.

$\Delta T_p$ : The testing time extension ratio for the proposed design.

**Table 1: Comparison between the Proposed Technique and the Buffer-Only Solution.**

Table 1 compares the proposed solution and the one that solely rely on introducing buffers to mitigate the influence of vulnerable TAMs (denoted as the buffer-only solution). As can be observed clearly from the table, to satisfy the same test yield loss requirements, if the flit error rate is relatively high, the proposed ATE interface design costs much less buffer size when compared to the buffer-only solution, which significantly reduces DfT area overhead, at the cost of slightly increased testing time. This is because, in the proposed solution, ATE interface is designed with minimized buffer size that is able to tolerate only one retransmission in each test segment. The buffer size required for the buffer-only solution, however, has to be large enough to be able to tolerate several times of retransmissions when the flit error rate is high. For example, when the flit error rate is smaller than  $10^{-8}$ , the two solutions are essentially the same; when the flit error rate is as high as  $10^{-6}$  and the test yield loss is required to be less than 0.5%, the buffer-only solution requires nine times larger buffers to store 180 flits or 5760 bits when compared to the proposed solution, while the testing time increase is less than 0.33%.

It should be noted that the above discussion is based on a circuit with 2.6 million gates [5], for a SoC device containing tens of or even hundreds of large cores, the total test data volume can be much larger and hence the benefits of the proposed solution will be more evident.

## 6. CONCLUSION

Existing work assumes TAMs to be error-free during test data transfer, which will not be true with the relentless scaling of CMOS technology, especially when reusing on-chip network or functional bus as TAMs. While fault tolerance techniques eliminate data transmission error, they bring test traffic jitter and test bandwidth variation and invalidate the entire test process. Therefore using such vulnerable TAMs to transfer test data in modular testing may lead to significant undesired test yield loss. In this paper, we propose novel DfT solutions to tackle this problem, including a “jitter-aware” test wrapper design and a “jitter-transparent” ATE interface design with acceptable DfT cost, which facilitate reliable modular testing of SoC devices even if test data may sometimes get corrupted during transmission. Experimental results on an industrial circuit demonstrate the effectiveness of the proposed technique.

## 7. ACKNOWLEDGEMENT

This work was supported in part by the Hong Kong SAR RGC Earmarked Research Grant 417406 and 417807, and in part by the National High Technology Research and Development Program of China (863 program) under grant no. 2007AA01Z109.

## 8. REFERENCES

- [1] A. M. Amory, F. Ferlini, M. Lubaszewski, and F. Moraes. DfT for the Reuse of Networks-on-Chip as Test Access Mechanism. In *Proc. IEEE VLSI Test Symposium (VTS)*, pp. 435–440, 2007.
- [2] A. M. Amory, et al. Wrapper Design for the Reuse of a Bus, Network-on-Chip, or Other Functional Interconnect as Test Access Mechanism. *IET Computers & Digital Techniques*, 1(3):197–206, May 2007.
- [3] A. Apostolakis, M. Psarakis, D. Gizopoulos, and A. Paschalis. Functional Processor-Based Testing of Communication Peripherals in Systems-on-Chip. 15(8):971–975, Aug. 2007.
- [4] C. Barnhart, et al. Extending OPMISR Beyond 10x Scan Test Efficiency. *IEEE Design & Test of Computers*, 19(5):65–73, Sept.–Oct. 2002.
- [5] C. Barnhart, et al. OPMISR: The Foundation for Compressed ATPG Vectors. In *Proc. IEEE International Test Conference (ITC)*, pp. 748–757, Nov. 2001.
- [6] L. Benini and G. de Micheli. Networks on chips: A new SoC paradigm. *Computer*, 12(1):70–78, January 2002.
- [7] D. Bertozzi, L. Benini, and G. D. Micheli. Error Control Schemes for On-Chip Communication Links: the Energy-Reliability Tradeoff. *IEEE Transactions on Computer-Aided Design*, 24(6):818–831, 2005.
- [8] E. Cota, L. Carro, and M. Lubaszewski. Reusing an on-chip network for the test of core-based systems. *ACM Transactions on Design Automation of Electronic Systems*, 9(4):471–499, 2004.
- [9] W. J. Dally and B. Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pp. 18–22, 2001.
- [10] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks: An Engineering Approach*. IEEE CS Press, 1997.
- [11] T. Dumitras, S. Kerner, and R. Marculescu. Towards On-Chip Fault-Tolerant Communication. In *Proc. IEEE Asia South Pacific Design Automation Conference (ASP-DAC)*, pp. 225–232, 2003.
- [12] P. Harrod. Testing Reusable IP - A Case Study. In *Proc. IEEE International Test Conference (ITC)*, pp. 493–498, Atlantic City, NJ, Sept. 1999.
- [13] J.-R. Huang, M. K. Iyer, and K.-T. Cheng. A Self-Test Methodology for IP Cores in Bus-Based Programmable SOCs. In *Proc. IEEE VLSI Test Symposium (VTS)*, pp. 198–203, 2001.
- [14] F. A. Hussin, T. Yoneda, and H. Fujiwara. Optimization of NoC Wrapper Design under Bandwidth and Test Time Constraints. In *Proc. IEEE European Test Symposium (ETS)*, pp. 35–42, 2007.
- [15] F. A. Hussin, T. Yoneda, A. Orailoglu, and H. Fujiwara. Core-Based Testing of Multiprocessor System-on-Chips Utilizing Hierarchical Functional Buses. In *Proc. IEEE Asia South Pacific Design Automation Conference (ASP-DAC)*, pp. 720–725, 2007.
- [16] A. Krstic, et al. Embedded Software-Based Self-Testing for SoC Design. In *Proc. ACM/IEEE Design Automation Conference (DAC)*, pp. 355–360, 2002.
- [17] J. Li, Q. Xu, Y. Hu, and X. Li. Channel Width Utilization Improvement in Testing NoC-Based Systems for Test Time Reduction. In *Proc. IEEE International Symposium on Electronic Design, Test & Applications (DELTA)*, pp. 26–31, 2008.
- [18] C. Liu, E. Cota, H. Sharif, and D. K. Pradhan. Test Scheduling for Network-on-Chip with BIST and Precedence Constraints. In *Proc. IEEE International Test Conference (ITC)*, pp. 1369–1378, 2004.
- [19] T. Lv, J. Henkel, H. Lekatsas, and W. Wolf. Enhancing Signal Integrity through a Low-Overhead Encoding Scheme on Address Buses. In *Proc. Design, Automation, and Test in Europe (DATE)*, pp. 542–547, 2003.
- [20] E. J. Marinissen, et al. A Structured And Scalable Mechanism for Test Access to Embedded Reusable Cores. In *Proc. IEEE International Test Conference (ITC)*, pp. 284–293, Washington, DC, Oct. 1998.
- [21] S. Murali, T. Theocharides, N. Vijaykrishnan, M. J. Irwin, L. Benini, and G. D. Micheli. Analysis of Error Recovery Schemes for Networks on Chips. *IEEE Design & Test of Computers*, 22(5):434–442, Sept.–Oct. 2005.
- [22] Open Core Protocol Specification. <http://www.ocpip.org>.
- [23] D. Park, et al. Exploring Fault-Tolerant Network-on-Chip Architectures. pp. 93–104, 2006.
- [24] P. Varma and S. Bhatia. A Structured Test Re-Use Methodology for Core-Based System Chips. In *Proc. IEEE International Test Conference (ITC)*, pp. 294–302, Washington, DC, Oct. 1998.
- [25] Q. Xu and N. Nicolici. Resource-Constrained System-on-a-Chip Test: A Survey. *IEE Proc., Computers and Digital Techniques*, 152(1):67–81, January 2005.
- [26] F. Yuan, L. Huang, and Q. Xu. Re-Examining the Use of Network-on-Chip as Test Access Mechanism. In *Proc. Design, Automation, and Test in Europe (DATE)*, pp. 808–811, 2008.
- [27] H. Zimmer and A. Jantsch. A Fault Model Notation and Error-Control Scheme for Switch-to-Switch Buses in a Network-on-Chip. In *Proc. IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, pp. 188–193, 2003.
- [28] Y. Zorian. Test Requirements for Embedded Core-Based Systems and IEEE P1500. In *Proc. IEEE International Test Conference (ITC)*, pp. 191–199, Washington, DC, Nov. 1997.