A New Structure for Interconnect Offline Testing

SomayehSadeghi-Kohan, ShahrzadKeshavarz, FarzanehZokaee, FarimahFarahmandi,
Zainalabedin Navabi
School of Electrical and Computer Engineering
College of Engineering, University of Tehran, Iran
{somayeh, shahrzad-k, f-zokaee, farimah, navabi}@cad.ut.ac.ir

Abstract

Multigigahertz range of working frequency, shrinking of technology and loss of signal integrity put circuits' interconnection at a higher risk of permanent or more frequent transient faults. These faults reduce overall reliability and performance of the circuit. Because of this, testing interconnects becomes an important issue. This paper presents an offline interconnect testing method that improves test time compared to some other earlier methods. The proposed method is implemented by a simple hardware structure, which has low hardware overhead and can detect crosstalk and other types of interconnect faults.

1. Introduction

With the advances in VLSI technology and decrease of the feature-size and moving towards deep sub-micron technologies, interconnects play an important role in determining the overall performance of a system.

Reaching to higher speed is one of our major concerns and since parallel interconnects and multiple links transfer different data bits, they provide higher speed and as a result are widely used in on-chip interconnects. As the complexity of circuits grow, parallel interconnect testing becomes more important in terms of its test time, hardware overhead and a complete diagnosis.

Shrinking feature size, increasing interconnect wires length, rapid increase of the SoCs' functional frequency and using deep sub-micron SoCs magnifies the capacitive coupling between adjacent wires. This increases the probability of crosstalk noise and along with that it provides glitch and delay that affect signal integrity and performance of a circuit. These considerations show the importance of finding an efficient way to test parallel interconnects.

This paper addresses the issue of interconnect testing in offline mode. In this paper a structure for testing parallel links is introduced. The proposed structure produces the required test data and activates all critical states resulting in crosstalk faults in offline mode. This structure is a self-test circuit that generates the test data automatically without any external circuitry and equipment.

This paper is organized as follows: Section II presents some of the related works. Section III illustrates the fault model used in our method. Section IV discusses details of our proposed method. Section V demonstrates experimental results in terms of test time and hardware overhead, and Section VII concludes the paper.

2. Related Works

Several researches has been done on interconnect testing. A general structure is proposed in [1] to detect and locate delay, transient faults, and crosstalk faults that impact interconnects of a bus. Although this method can be used for fault detection and location, it cannot detect delays with long width. Additionally, it cannot detect stuck-at and bridging faults.

In [2], an online test method is provided. It can detect multi-source errors, but the fault coverage for crosstalk faults is about 40%. Also, this method cannot detect stuck-at faults.

There are also several works on designing Built-in Self-Test for interconnects. In many researches [3], [4] a combination of Linear Feedback Shift Register and Multiple Input Signature Register is used for testing interconnects. These methods have very low fault coverage. In [5] and [6], some methods are proposed for BIST-based test improvement. In [6] and [7], graph coloring and genetic search algorithms are used and an LFSR/MISR structure with better fault coverage is introduced for interconnect testing.

The method presented in [8] uses IETD-MISR (Internal Exclusive-OR linear feedback MISR composed of T-type and D-type flip-flops) for signature compaction. The authors claim that the fault coverage is higher if the interconnects' test data are

978-1-4799-2096-9/13/$31.00 ©2013 IEEE

compressed using T flip-flops. They implemented a MISR using D and T flip-flops, and introduced a test structure using this MISR that detects, locates, and identifies interconnect faults.

Authors in [9] and [10] decrease test time by generating two independent test strings: one for negative glitches and the other for positive glitches. Since the suggested data strings can test faults on all interconnects, there is no need to test each line independently for all faults, which, in effect, reduces test time.

A self-test structure for SoC interconnects is introduced in [11], that is based on the MT (Multiple Transition) fault model. This structure generates test data with fault coverage of 100%. However, the hardware overhead of this method is very high.

There are also methods that use the IEEE 1149.1 standard to perform interconnect testing. In [12], parallel interconnects are tested for six states of the possible crosstalk faults. This is done by modifying the boundary scan cells and generating eight test data. This method adds two multiplexers, one for T flip-flop and one for AND gate to the standard boundary scan cells. Two additional instructions are also added to the IEEE Boundary Scan 1149.1standard instruction set. This method is activated by an external test circuit. In order to test each of these wires and choose them as a victim line, the Boundary Scan shift operation must be done in the shift/capture flip-flops. Also, a counter is used to count the number of generated test data. After applying the first four data strings to the interconnect lines, the first string from the second set of data is shifted into the shift/capture flip-flops and loaded into the update flip-flops. Then the data for selecting the victim line must be shifted into shift/capture flip-flops. The process of shifting data for initial values and choosing the victim line is time consuming, and therefore results in a high test time. In this paper with some modification on this method, we proposed a low test time, and low hardware overhead method of interconnect offline testing.

3. Fault Model

The fault model that is used in this work is the Maximum Aggressor (MA) fault model introduced in [13]. It is the most widely used method in similar works. This model can cover most faults resulted from coupling capacitance and crosstalk noises. The modeling of interconnect lines with resistors, capacitors and inductors shows that more glitches and crosstalk faults occur as a result of decrease in size (Figure 1).It is also proved thatas the length of wires increase, the probability of occurrence of these faults increases. The authors have considered six faults:positive and negative glitches, delay in positive and negative edges, and speedy in positive and negative edges. They tried to model these faults such that they require a minimum set of test data. This fault modeling is shown in Figure 2.

According Figure 2, Authors proposed a fault simulation method that activates crosstalk faults. The procedure of fault simulation is simplified with some assumptions. In each step of test, one of the wires is considered as the victim wire and the others are considered as aggressive wires. For activating glitch faults, the value of victim wire under test is held unchanged, and the values of the others are changed (in the same direction). For delay faults, the voltage of this wire is changed at the opposite direction of the other wires. For speedy faults, the voltage of victim wire is changed at the same direction of the other wires. It is proved that this model covers a high percentage of the crosstalk faults.

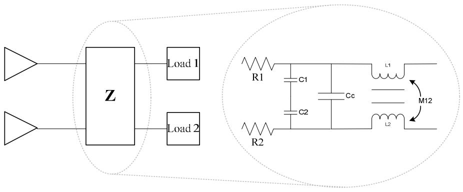
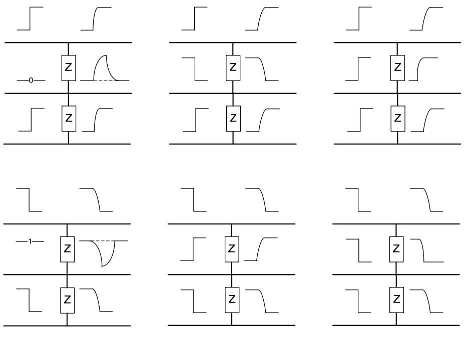


Figure 1. Physical structure of interconnect wires [13]



(a) glitch fault (b) delay fault (c) speedy fault

Figure 2. Possible fault models for interconnects [13]

4. Proposed Method

The method proposed in this paper performs some simple modifications on the work done in [12]. With these modifications, the test time is decreased and parallel interconnects can be automatically tested without the need of an external test circuits. Our method has less hardware overhead than the method from [12], and also it tests more faults. The structure of this method is depicted in Figure 3. Test data which areapplied to the interconnect lines are shown in Figure 4. In the procedure of testing, first, string number 1 is applied to wires and after that by resetting the flip-flops, which generate the test data and activating the *comp* signal, string number 2 will be applied to the wires.

Flip-flop number 2 is used for generating the test data for the victim wire and flip-flop number 1 generates the data for the aggressive lines. These flip-flops invert their stored data on each input clock edge.

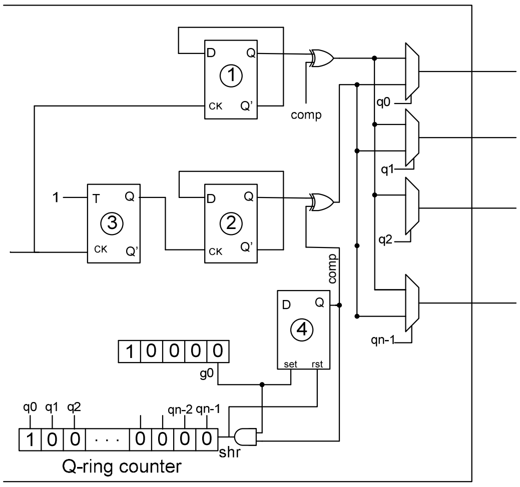


Figure 3. The structure of proposed method for offline testing of parallel interconnects

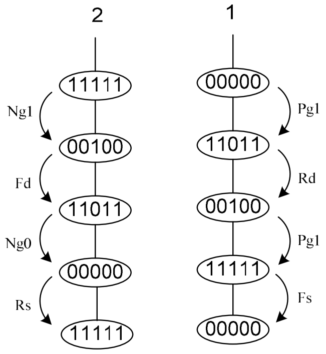


Figure 4. The generated bit patterns by the proposed test structure

For each wire, a multiplexer is used to select input of wires, which is driven from a ring counter. When the bit corresponding to a wire becomes '1' in the Q-ring counter (will be described in the later paragraphs), output of the flip-flop number 2 is sent to the interconnect wire and the wire is chosen as a victim line. If the corresponding bit of a wire is '0' in the ring counter, the wire is chosen as an aggressive and outputof the flip-flop number 1 is sent to this interconnect line.

Flip-flop number 3 is a T flip-flop, which activates the clock signal of flip-flop number 2. In fact, this flip-flop causes the clock of flip-flop number 2 to have a clock frequency equal to 1/2 of the original clock frequency. Therefore, the inversion of the output bit of this flip-flop happens every other clock cycle and as a result, the test data of string number 1 is generated.

In this structure, a 5-bit ring counter is used to count the number of data which is being applied to the interconnect lines. Initially, this counter is set to "10000". When the test procedure starts, this counter is shifted to right with every rising edge of the clock. When the last bit of this counter (g0) becomes '1', flip-flop number 4 is activated (the *set* signal of this flip-flop is activated), and therefore the *comp* signal becomes activated. Activation of this flip-flop means the end of applying data string number 1. In this state, flip-flops number 1 and 2 also receive the reset signal. The test procedure continues with all flip-flops containing zero and flip-flops number 1 and 2 generate test data. In this state, since the*comp* signal is active, the bits generated by these flip-flops are inverted and as a result, string number 2 is applied to the interconnect lines.

The Q-ring counter (n-bit ring counter: n is the number of interconnect lines) shifts after testing each wire (simultaneous activation of flip-flop number 4 and g0), and test procedure for the next wire begins. At the beginning of the test procedure, just one bit of this counter is '1' and the other bits are '0'. When a bit is '1', it means that its corresponding wire is a victim line. In this case other wires are considered as aggressive lines. If the circuit layout is available, we can extract the necessary information to find the maximum aggressive wires for a victim wire. This number is used to set the initial value of the n-bit ring counter. In this case, more bits of this counter can be '1' and the test procedure can be done concurrently.

On the receiver side, we can use noise and skew recognition cells. It is also possible to detect stuck-at, bridging, and state-dependent faults in addition to crosstalk faults by copying the structure of Figure 3 and comparing its data with the received data.

5. Experimental Results

In this section, the experimental results of our method will be discussed.

5.1. Fault Coverage

Test strings shown in Figure 4 detect delay and glitch faults, as well as speedy faults in the rising and falling edges. Testing these kinds of faults is important in today’s circuits because of the increasing frequency. The test strings also test glitches related to overshoot and undershoot (like in [12]). Although these faults do not cause logical faults, they have a great impact on power consumption and drivers corruption.

Given the applied data strings, it can be concluded that the crosstalk fault coverage in the proposed structure is 100%. Table 1 shows the comparison between the proposed method and other methods for crosstalk faults they cover. As shown, our proposed structure covers all crosstalk faults.

5.2. Test Time

The time for testing each victim line is equal to 10 clock cycles and it is n*10 for all wires. Comparisons between the proposed method and methods from [12] and [11]are done in Table 2. As shown, the time required by the proposed method is less than the time required by [12] and more than the time required by [11]. The reason for higher test time is that our method considers more crosstalk faults and therefore, increases the number of test data.

5.3. Hardware Overhead

Hardware overhead of the proposed method in comparison with [12] and [11] is shown in Table 3. As shown, this method has less hardware overhead in comparison with the other two mentioned methods.

Also, the hardware overhead of this method for Networks on Chip is relatively high. This is because of the fact that in these systems there is a large number of connection links, and therefore the ring counter of the design should be large. If the time of offline test in NoCs is not critical, we can only use one counter and data generating flip-flop structures in each switch. In this case, testing each switch should be done independently and when testing one port of the switch is finished, testing the other port of the switch could begin. In this case, hardware overhead of this method for a 5-port switch becomes 1/5 and duration of connection test procedure increases 5 times.

Table 1. Comparison of detected faults

	Crosstalk faults							
	Delay on rising edge	Delay on falling edge	Positive glitch	Negative glitch	overshoot	undershoot	Speedy rising edge	Speedy falling edge
Method [12]	✓	✓	✓	✓	✓	✓	×	×
Method [11]	✓	✓	✓	✓	×	×	✓	✓
Proposed Method	✓	✓	✓	✓	✓	✓	✓	✓

Table 2. Comparison of test time for different methods

Used Method	Number of BUS links		
	8 links	16 links	32 links
Method [12]	264	520	1032
Method [11]	64	128	256
Proposed method	80	150	320

Table 3. Comparison of hardware overhead for different methods in sample circuits

Circuit under test	Used method		
	Method [12]	**Method** [11]	**Proposed method**
SAYEH Processor	~4.3%	2.4%	~2%
Adding Machine	8.7%	6.8%	4.7%
NOC 8*8	~11.5%	10.4%	7.9%
NOC 16*16	~11.5%	10.4%	7.9%

6. Conclusion

We proposed an offline interconnect test structure that can test all crosstalk faults with low hardware overhead. This structure improves test time in comparison with other interconnect test methods. Moreover the proposed method is also capable of detecting other types of interconnect faults such as bridging and stuck-at faults.

7. Acknowledgement

This project was partially funded by Iran Telecommunication Research Center (ITRC).

8. References

[1]C. Metra, M. Favalli, and B. Ricco, “Self-checking detection and diagnosis of transient, delay, and crosstalk

faults affecting bus lines," *Computers, IEEE Transactions on*, vol. 49, no. 6. pp. 560-574, 2000.

[2]Y. Zhao, S. Dey, and L. Chen, "Double sampling data checking technique: an online testing solution for multisource noise-induced errors on on-chip interconnects and buses," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 12, no. 7. pp. 746-755, 2004.

[3]J. C. Chan, "An improved technique for circuit board interconnect test," *Instrumentation and Measurement, IEEE Transactions on*, vol. 41, no. 5. pp. 692-698, 1992.

[4]J. Koeter and S. Sparks, "Interconnect testing using BIST embedded in IEEE 1149.1 designs," *ASIC Conference and Exhibit, 1991. Proceedings., Fourth Annual IEEE International*. pp. P11-2/1-4, 1991.

[5]K. Sekar and S. Dey, "LI-BIST: a low-cost self test scheme for SOC logic cores and interconnects," *electronic testing: theory and applications*, pp. 113-123, 2003.

[6]C. P. Ravikumar and S. Chopra, "Testing interconnects in a system chip," *VLSI Design, 2000. Thirteenth International Conference on*. pp. 388-391, 2000.

[7]U. Pillkahn, "Structural test in a board self test environment," *Test Conference, 2000. Proceedings. International*. pp. 1005-1012, 2000.

[8]M. Kopec, T. Garbolino, K. Gucwa, and A. Hlawiczka, "Test-per-Clock Detection, Localization and Identification of Interconnect Faults," *Test Symposium, 2006. ETS '06. Eleventh IEEE European*. pp. 233-238, 2006.

[9]R. Nourmandi-Pour, A. Khadem-Zadeh, and A. Masoud Rahmani, "An IEEE 1149.1-based BIST method for at-speed testing of inter-switch links in network on chip," *Microelectron. J.*, vol. 41, no. 7, pp. 417-429, Jul. 2010.

[10]R. Nourmandi-Pour, N. Mousavian, and A. Khadem-Zadeh, "BIST for network on chip communication infrastructure based on combination of extended IEEE 1149.1 and IEEE 1500 standards," *Microelectronics Journal*, vol. 42, no. 5, pp. 667-680, 2011.

[11]X. Bai, S. Dey, and J. Rajski, "Self-Test Methodology for At-Speed Test of Crosstalk in Chip Interconnects," in *Design Automation Conference*, 2000, pp. 619-224.

[12]M. H. Tehranipour, N. Ahmed, and M. Nourani, "Testing SoC interconnects for signal integrity using extended JTAG architecture," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 23, no. 5. pp. 800-811, 2004.

[13]M. Cuviello, S. Dey, X. Bai, and Y. Zhao, "Fault modeling and simulation for crosstalk in system-on-chip interconnects," *Computer-Aided Design, 1999. Digest of Technical Papers. 1999 IEEE/ACM International Conference on*. pp. 297-303, 1999.