Online Fault Detection, Diagnosis and Repair using RA

Ankita Tripathi

Electronics & Telecommunication Department, Hyderabad Institute Technology and Management, Hyderabad, India

Abstract: ***Built-in self-test (BIST) is a commonly used design technique that allows a circuit to test circuit itself. But for this circuit have to switch in off-line mode which unnecessary waste time and power to avoid this best method is online bist In this paper BIST architecture is implemented for Detection, Diagnosis and Repair of various faulty circuits. A complete and versatile online test solution based on reconfigurable test architecture is presented in this paper. Reconfigurable test architecture works alongside the controllers for online concurrent fault detection. The output vectors of the controllers are concurrently monitored and any fault present is detected in a few cycles from the sensitization of the fault. The architecture is then reprogrammed to a similar set of diagnostic hardware to locate a sub block which is the cause for the fault. The same architecture is then reprogrammed to replace the faulty block thereby completing repair. The test architecture is designed based on configurable logic blocks. The design has several advantages viz. (i) it works well for critical VLSI controllers where shutting down or suspending the operation of a controller for testing is not possible and where the fault needs to be detected at the earliest, during the run time of the system, (ii) after a fault is detected, diagnosis can be performed online, (iii) once a faulty block is located, repair is also done online. Since fault detection, diagnosis and repair are completed online with one test hardware, the effective hardware overhead is negligible and the system can resume its function within a brief period. The applicability of the architecture is demonstrated for the control blocks in OC8051.***

Keywords: concurrent test, multiple controllers, online test, output vector monitoring, programmable architecture, reconfigurable architecture

1. Introduction

N most of the embedded system applications, real time computing is used. The operations execute within the strict constraints called system deadlines. The anti-lock brakes on a car is a simple example of a real-time computing system. The controllers that manage anti-lock brakes must continue to perform its intended real-time function during its entire lifetime. The controllers which are very large scale integration (VLSI) circuits can become faulty in the due course of their operation. Some faults that arise later in the lifetime, because of electro-migration, stress, time-dependant dielectric breakdown or thermal cycling, make estimation of mean-time-to-failure (MTTF) very difficult at design time and consequently, failure detection at runtime [1]. The faults that occur during the lifetime of an integrated circuit (IC) can be classified as follows [2]

1) **Permanent faults** are those which remain indefinitely in the system. Most of these are manufacturing faults or design errors. A few of these are caused by major environmental changes or physical damage of the chip. This effect may remain for the entire lifetime.
2) **Intermittent faults** may appear for a brief period of time, then disappear and reappear after a relatively longer period. As there are no fixed parameters that cause these faults, predicting them seems to be an impossible task. The two major reasons for their occurrence are marginal dimensions in manufacturing and tight constraints during the design. Since the system works well for most of the time and under most of the conditions, testing and their diagnosis is a major concern.
3) **Transient faults** mostly appear for a much shorter period and disappear quickly. Their appearances are rare and are mainly caused due to environmental variations.

One of the common methods of testing circuits after these are placed in the field is using built-in-self-tests (BISTs). These are both effective and practical for most offline tests. The advantages of BIST include the capability of performing at-speed testing, high fault coverage, elimination of test generation effort and less reliance on expensive external testing equipment for applying and monitoring test patterns [3]. BISTs which are programmable offer much larger flexibility for deeply embedded components [4]. Owing to these advantages, BIST offers a very cost effective test package. The test methods are further divided into offline and online BISTs. When the system is shutdown completely or the circuit is detached from the field and the inputs/outputs are captured by the BIST circuitry are called offline BIST. Online BIST is where the operation of the circuit might be temporarily suspended to change the mode into test mode and run a BIST. Here a test generator (TG) applies the test vectors either in a random or in a deterministic way to the circuit. A response verifier (RV) verifies the captured output. The compiled response is finally analysed to determine if a fault is present. There are several proposals to tackle both online and offline BIST. However these methods are not sufficient to handle concurrent online testing. In the present work a complete online test solution is presented. It concurrently detects faults in controllers, locates the design block that is faulty and replaces the faulty block with a fault free one. The entire test hardware is built by configurable logic blocks (CLBs). Concurrent online testing is carried out by simultaneously monitoring the outputs of multiple controller blocks and dynamically generating signatures, which are in turn monitored for faults. For the purpose of diagnosis and repair, each of the controllers is divided into blocks of similar gate counts. The hardware used for fault detection is rerouted to target each block and diagnosis is performed. Once a faulty block is detected, then the function of the block is programmed on the CLBs which have been used for test, which replaces the faulty block, thus completing repair. The paper is organized as follows. In Section II, the various challenges for concurrent online testing and diagnosis are discussed. In Section III, the principle behind the design of the complete online test solution i.e. the Reconfigurable Architecture for Online- Detection, Diagnosis and Repair (RAO-DDR) is presented. Section IV presents, the Concurrent Online Test Architecture for Multiple Controllers (COTA-MC). In Section V the online fault diagnosis process is given. Section VI deals with the fault repair procedure. In Section VII, the proposed design is validated by implementing it for the control blocks of OC8051. Finally, conclusions are drawn in Section VIII.

2. Challenges for Concurrent Online Test and Diagnosis

Online testing addresses the detection of faults that emerge during the operation of the system. These are mainly the intermittent and transient faults. Online testing is especially important for critical applications and those applications which are in high demand. These systems are not expected to fail without warning. Online testing provides an option to avoid catastrophe, if a system fails. Once the test detects an error, the system performs one or a few of the following tasks to adjust to the error: i) it saves the critical data, ii) issues a warning or switches to a different module, iii) steps down the performance of the system, iv) starts a repair sequence, v) starts a reconfiguration mechanism and/or vi) shuts down the system. Online test can be done with a setup outside the system either with the help of software or hardware alone. But the external setup does not have sufficient external pins to monitor the entire complex hardware within. Also all the internal faults do not show up on the pins and external monitoring is expensive. Internal online testing is the alternative method to test ICs on the system. Testing is internal if it takes place on the same substrate as the design-under-test (DUT) within the system-on-chip (SoC).

Online testing can be further divided into concurrent and non-concurrent testing. In non-concurrent testing the DUT is tested while the normal operation of the circuit is temporarily suspended or during the shutdown or boot sequence. For critical applications where the operation of the circuit cannot be suspended, testing has to be carried out during the normal functioning of the circuit. This kind of testing is called concurrent online testing. Normal online testing methods do not work for concurrent testing, nor do the external online testing schemes. The major parameters to be considered while designing a concurrent online test scheme are:

Concurrent Test Latency (CTL): It is the number of normal functional inputs that must be applied to the circuit under test (CUT) while the CUT operates normally in order to complete the concurrent test process. This is an important factor as it determines how quickly the functional vectors achieve the expected fault coverage. If this measure is high, then probably the CUT has to wait for a higher number of cycles for all the targeted faults to be covered [13].

Fault Latency (FL): It is the time taken for the concurrent test to detect the fault from the time it actually appeared. A related parameter is error latency (EL) defined as the time taken to detect an error from the time the error gets activated by the input vector. FL is the most important factor since it gives the information about the number of cycles that go by without the fault being detected. EL helps to assess and design the monitor circuit in a better way to capture the effect of the fault as soon as possible, after it gets activated.

Fault Coverage (FC): It is the fraction of the targeted faults for a particular CUT that are detected by a specific test or a test set. Circuits that are critical, require very high fault coverage in each of the fault categories.

Area Overhead (AO): It is the number of gates that are needed to complete online testing over and above the gate count of the original design. Even though area overhead is not a major factor, it affects scalability. If area scales proportionally then area overhead becomes important. Concurrent online testing was initially carried out by using watchdog timers [5]. Watchdog timers alone proved to be inefficient, because these only confirm if control flow traverses properly. Later, redundancy has been introduced. In one case duplication with comparison (DWC) [6], where the outputs of the two copies of the same circuit which operate in tandem, is compared. These can only detect a single error but with 100% area overhead and are still inefficient because it is difficult to synchronize both. The method has been further improvised by comparing the outputs of three identical circuits receiving the same inputs. The result is interpreted based on majority.

However, the area overhead increases to 200%. The other method is to do the same operation twice and compare between the two results. This method is called double-execution or retry. Transient faults are likely to be detected. Although this technique is area efficient, it introduces a lot of time redundancy. For applications where run time is critical, these methods cannot be used. But even otherwise, the time penalty it imposes is too high. Another similar method is re-computing with shifted operands (RESO). In this a coding based method of parity checking is used especially for detecting memory and data transmission errors [6].

The initial work on vector monitoring concurrent BIST (C-BIST) was reported by Saluja [7]. The test generator of C-BIST is a linear feedback shift register (LFSR) and the active test set consists of exactly one active test vector i.e. the current value of the LFSR. C-BIST has low hardware overhead but very high CTL. This is because in every clock cycle the input vector is compared with only one active test vector. To drive down the CTL so far four techniques have been reported in literature viz. i)Multiple Hardware Signature Analysis Technique (MHSAT) [8], ii) Order Independent Signature Analysis Technique (OISAT) [9], iii) windowed-Comparative Concurrent BIST (w-CBIST) [10] and iv) RAM-based Concurrent BIST (R-CBIST) [11]. These decrease the CTL by increasing the number of active test vectors. Built-In Concurrent Self-Test (BICST) has been proposed by Sharma and Saluja [12]. When BICST is applied to an n-input m-output combinational CUT that can be tested with T vectors, it utilizes a T-line X (n+m)-column PLA. In [13], an input vector monitoring concurrent BIST technique for monitoring input vectors for concurrent testing based on a pre-Computed test SET (MICSET) is given. This scheme is based on a test set stored in a mapping logic module which can be implemented with either random logic or a ROM whose address inputs are driven by a subset of the input bits of the CUT. This scheme again suffers from a very high CTL and hence very high fault latency. Since the hardware overhead scales along with the size of the CUT, this scheme is not a workable solution. For systems whose continuous functioning is of utmost importance, online concurrent testing with minimum area overhead and minimum fault latency that is presented here is the best solution. Diagnosis has been almost completely offline, since online diagnosis is expensive and the online diagnostic resolution achieved has been very low. Moreover, there is nothing much one can do after online diagnosis because repair of logic blocks has again been an almost impossible task. One method of diagnosis is using external hardware or a reconfigurable FPGA connected to the chip, which is a very long and tedious process [19]. As mentioned earlier, after completing diagnosis there are no efficient repair methods to replace logic. An efficient diagnosis and repair method is also proposed which solves most of the problems mentioned above.

3. Reconfigurable Architecture for Online Detection, Diagnosis and Repair

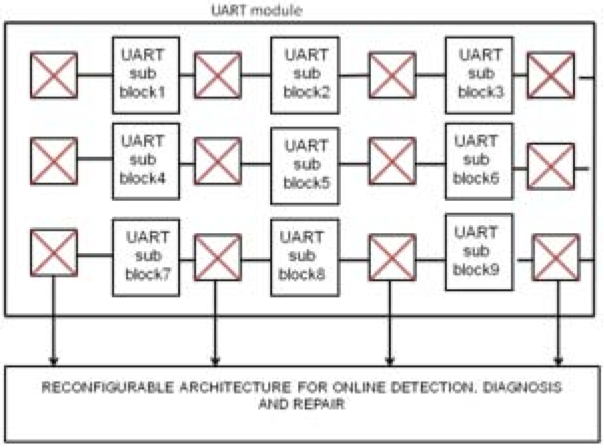

Figure 1: Sub blocks of a module connected with RAO-DDR through switch matrices

In order to facilitate a complete test solution, the test hardware is a reconfigurable logic; lookup table (LUT) based configurable logic blocks (CLBs). A CLB can be made up of sub-components called slices and each slice can have one or two 6-input LUTs, a full adder (FA) and one or two D flip-flops. Test architecture of 15 CLBs is considered. The proposed test architecture is configured first for concurrent fault detection of either one or multiple controllers. This is achieved by routing the outputs of the controller to the COTA-MC. The outputs are monitored and dynamic signatures are generated. When an unexpected signature is detected, a fail signal is asserted. Fault detection is explained in detail in the next section. In order to make diagnosis possible, the controller under test is segmented into a number of smaller blocks, which have approximately half the gate count as in one CLB. A controller, like the one shown in the Fig. 1 is bisected into sub-blocks. Each block has a switch matrix at its input and output. The inputs go through the switch matrix and the outputs come through the switch matrix. A switch matrix consists of a few pass transistors and has high speed. The states of the pass transistors are set by programming SRAM cells. These switch matrices create a tap to the inputs and outputs of each of the blocks. These are also capable of disassociating a block from its inputs and outputs. The taps to the outputs help in diagnosis. The disassociating option of a block from its inputs and outputs helps in repair of the block.

4. Concurrent Online Test Architecture for Multiple Controllers

The Concurrent Online Test Architecture for Multiple Controllers (COTA-MC) [18] is used here for fault detection in RAO-DDR. Its effectiveness has been established based on its implementation on the controllers of OC8051 [17] as shown in Fig. 2. OC8051 is chosen because in most embedded system applications at least a single microcontroller is used. The method of testing used is non-intrusive and adds no performance overhead to the circuit under test. The normal program execution is the necessary input required for the CUT. Furthermore, there is no requirement of an extra test pattern generator like most other BIST methods. The outputs of the controllers are passed through a set of scramblers and then through a series of XOR gates. It is then fed to two compactors viz. the multiple input signature register (MISR) and the accumulator based compactor (ABC). These are further processed by a set of rule sets implemented on a PLA which gives the final Pass/ Fail signal. The entire COTA-MC is housed on the CLBs of RAO-DDR.

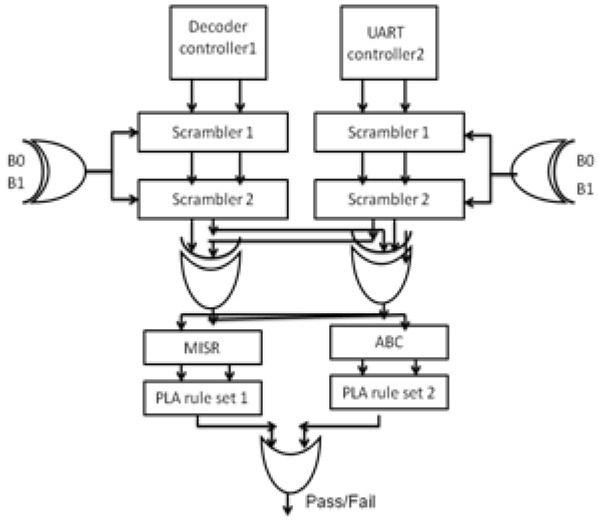

Figure 2: COTA-MC

The registers of COTA-MC are reset when either the system reset is given or the program counter (PC) reaches a particular address or COTA-MC's reset is provided. For many of the embedded systems and systems with critical applications there are two facts which are exploited in the present architecture.

One is that there is a specific program cycle that gets executed repeatedly. The other is that the program is loaded once in the system and is not changed unless the normal operation of the system is suspended. Therefore, it is a safe assumption that during the normal operation of a circuit, it is sufficient if the system works fine for the current program that is loaded into the system. A reset signal is generated when PC reaches a pre-programmed value. COTA-MC's reset can be multiplexed with an external input pin. To demonstrate the architecture and its ability to simultaneously test multiple controllers, two of the controllers of OC8051 are chosen. One is the decoder and the other is the universal asynchronous receiver transmitter (UART). The outputs of the decoder and the UART, each pass through a two stage scrambler. The function of a scrambler is to shuffle the data bits in a predetermined manner. One way for this is to shuffle them based on the XOR-ed output of two constantly changing bits. For example, two bits of the opcode have been used here.. The 4-bit input A, B, C and D are shuffled based on the parity generated (Y) by the two opcode bits B(0) and B(1). If Y=0, the outputs of the scrambler A', B', C' and D' will be A, C, B and D; and if

Y=1, the outputs will be B, D, C and A. This shuffling based on the opcode bits increases the probability of error detection. A two stage scrambler is used to thoroughly shuffle the data bits. Both the decoder and the UART have their respective two-stage scramblers. The data bits are preserved at the output of the scrambler, but these appear shuffled. In the next stage, the data bits from the decoder are XOR-ed with the data bits from the UART. This is done for two reasons. Firstly, to reduce the hardware overhead in having separate signatures generated. Secondly, an error in one of the bits will generate a different output at the XOR gate. The XORed bits are then fed to two different compactors, the ABC and the MISR. ABC [14, 15] is chosen since it is proven to have negligible aliasing, provides extremely high fault coverage, has very little hardware overhead and can work effectively for a very large number of cycles with little or no error cancellation

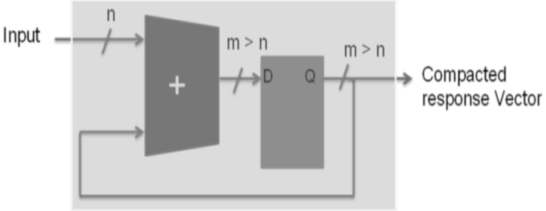

Figure 4: Accumulator Based Compactor (ABC)

The ABC used is shown in Fig. 4. Each output word of the scrambler (n-bits) is added to the contents ofthe register of the accumulator (m-bits; m>n) and the result is in turn added to the next word of the output and so on. The ABC is usually used to produce a unique signature. MISR is the other well known compactor chosen because of its small area and acts as an additional signature register along with ABC. The probability of error escaping both the signature registers, is extremely low. COTA-MC does not wait until the entire program cycle is over to read out the signature registers. The outputs of the signature registers are supplied to the next stage where a set of rules are implemented using either a PLA structure or one of the CLBs. It is noticed that some combinations of the output bits of the ABC and MISR do not occur during the fault free execution of the program cycle. Those bit combinations are chosen as rule sets and are checked in every valid program cycle. The rule sets are combinational functions that check the cumulative validity of the signatures upto the previous program cycle. Three rule functions for each of the signature registers are chosen in the present work and the total six rule function outputs are ORed together to generate the Fail function. If a violation is found, then the fail signal is asserted. If there is a fault in either the decoder or the UART, the signatures in both ABC and MISR change and violate at least one rule set in the next few cycles. The present COTA-MC can be used for both online and offline testing and is fully customizable according to the test requirements of the CUT. Online testing will be completed while the CUT is performing its normal operation. All the faults in the false paths i.e. the non-functional paths are naturally excluded. Since it is based on output vector monitoring, the architecture is scalable and hence hardware overhead becomes negligible for larger controllers.

5. Online Fault Diagnosis

For online fault diagnosis, the controller has to temporarily suspend operation. As explained earlier, the controller is divided into several smaller sub blocks for easy isolation of a faulty block. The objective of diagnosis is not to locate the gate or a transistor that is faulty. But the objective is to isolate the block that is faulty and replace that block with a fault free one. The controller that is to be diagnosed is considered one a regular cycle. The switch matrices are used to tap the outputs of the block under consideration. The outputs are fed to the COTA-MC as explained in the previous section. The COTA-MC is configured into the reconfigurable test architecture. In each cycle, the dynamic signatures are watched to decide if the fault occurs in this particular sub block. When a fault occurs, the fail signal is asserted. A little simpler option can also be adopted for diagnosis. Instead of watching the dynamic signatures, the final signatures can be compared after the full run of the instruction sequence. In both the cases the faulty block is located but in the later case the detection will only happen when the test cycle ends. If no fault is detected in this sub block, then this sub block is released and the next sub block's output is similarly connected to the COTA-MC through the switch matrices. The controller is executed for a new set of test cycle. This process is repeated for each of the blocks until the faulty block is detected. If no fault is detected, then there is no fault in any of the sub blocks. Thus diagnosis is completed with the chip on board.

6. Online Fault Repair

Once a faulty sub block is identified, the corresponding switch matrices are programmed to isolate the sub block from the controller and the inputs are rerouted to the test architecture. The test architecture is reconfigured to perform the same function as that of the faulty sub block that needs replacement. The outputs of this reconfigured block are routed back through the output switch matrix of the faulty sub block. So the faulty sub block is replaced with the reconfigured test architecture. This repair is feasible only when the faulty block is not closed very tight for timing. As the pass transistors are fast their delay can be neglected. The delay from the CLBs prove to be a few extra gate delays. In worst cases, if the clock frequency can be lowered a little bit, this repair becomes better. There is a small compromise on performance but the system can continue to work.

7. Design Validation

The RAO-DDR scheme is implemented within the OC8051 microcontroller. A set of 15 CLBs are used to implement the COTA-MC for online concurrent fault detection and fault diagnosis. The CLBs are then reprogrammed to replace the faulty block as explained. The hardware is the same for fault detection, diagnosis and repair. There is no extra area overhead. The same architecture works for all controllers whose sub-blocks can fit within the 15 CLBs and the outputs are limited to the allocated CLB inputs. Each CLB is approximately equivalent to 230 gates. The architecture is scalable for larger circuits. It only depends on the number of output lines and the size of the sub-blocks. The minimum overhead required for the test architecture comes from the COTA-MC.

8. Conclusion

In the present work, an all-inclusive test architecture is presented and shown feasible for controller like modules. The proposed RAO-DDR is capable of concurrently detecting faults during system operation. Once the fault is detected, it is able to locate the fault up to a sub-block and repair the faulty sub-block. All these are achieved without any additional hardware overhead because of the reprogrammable logic used for the test architecture. The practical effectiveness of the method is demonstrated by applying the scheme to the controllers of OC8051 and is compared with the concurrent online fault detection methods. RAO-DDR proves to be far better in most aspects. It is a complete online test solution and is one of its kind. Methods to improve diagnostic resolution and efficiency in repair methods are being investigated as further scope of this research work.

References

[1] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, “Exploiting structural duplication for lifetime reliability enhancement,” *Proc. 32nd Int. Sym.Comput. Arch. (ISCA)*, 2005, pp. 520–531.

[2] H. Al-Asaad and M. Shringi, “On-Line Built-In Self-Test for Operational Faults,” *Proc. of Conf. Systems Readiness Technology*, 2000, pp. 168-174.

[3] M. Abramovici, M. Breuer, and A. Friedman, Digital Systems Testing and Testable Design. *Computer Science Press*, 1990.

[4] P. Philemon Daniel and Rajeevan Chandel, “A Flexible Programmable Memory BIST Architecture,” *IETE Journal of Education*, vol. 51, pp.67-74, Dec 2010.

[5] A. Mahmood and E. McCluskey, “Concurrent error detection using watchdog processors-A survey”, *IEEE Trans. on Computers,* vol. C-37, no.2, pp. 160-174, February 1988.

[6] B. W. Johnson, *Design and Analysis of Fault Tolerant Digital Systems,* Addison-Wesley, Reading, Massachusetts, 1989.

[7] K.K. Saluja, R. Sharma, and C.R. Kime, “A Concurrent Testing Technique for Digital Circuits,” *IEEE Trans. Computer-Aided Design*, vol. 7, no. 12, pp. 1250-1260, Dec. 1988.

[8] K.K. Saluja, R. Sharma, and C.R. Kime, “Concurrent Comparative Testing Using BIST Resources,” *Proc. Int. Conf. Computer Aided Design*, Nov. 1987, pp. 336-339.

[9] K.K. Saluja, R. Sharma, and C.R. Kime, “Concurrent Comparative Built-In Testing of Digital Circuits,” *Technical Report ECE-8711*, Dept. of Electrical and Computer Eng., Univ. of Wisconsin, 1986.

[10] I. Voyiatzis and C. Halatsis, “A Low Cost Concurrent BIST Scheme for Increased Dependability,” *IEEE Trans. Dependable and Secure Computing*, vol. 2, no. 2, pp. 150-156, April-June 2005.

[11] I. Voyiatzis, A. Paschalis, D. Gizopoulos, N. Kranitis, and C.Halatsis, “A Concurrent Built-In Self Test Architecture Based on a Self-Testing RAM,” *IEEE Trans. Reliability*, vol. 54, no. 1, pp. 69-78, Mar. 2005. of VSI..

[12]