On Applying Circular Self-Test Path (CSTP) Technique to Circuits*

Charles A. Njinda, Rajagopalan Srinivasan and Melvin A. Breuer

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089-0781, U.S.A.

Abstract — The use of a circular self-test path is an attractive built-in self-test technique due to the advantages of a single test session and ease of control. In this paper we present a procedure to select registers for circular path so that each combinational block of the circuit is tested functionally exhaustively. High-level structural details like identity modes (I-modes) of circuit are used to reduce the test time and area overhead at the expense of slightly more complex control. A mixed mode test strategy is suggested to improve the pattern coverage for the circuit. Analytical expressions are derived for estimating test time.

Keywords: Circular self-test path, functional exhaustive testing, identity modes

1 Introduction

The circular self-test path (CSTP) technique proposed by Krasniewski and Pilarski [1] is a method for designing self-testable circuits with a low area overhead. A subset of registers in the design are selected to form the CSTP. Their strategy is to select at least one register from every loop and all input/output registers of the circuit. The path is a circular shift register with the output of the last storage element connected to the first one. The path serves simultaneously as a test pattern generator, with characteristic polynomial $1 + x^k$, where k is the length of the circular path, and a response compactor. Circuits designed with the CSTP technique usually have area overhead slightly exceeding that of scan designs but less than comparable BILBO designs. Equations have been provided in [1, 2] for computing test time and expected fault coverage. These equations are based on the probability of applying all possible input patterns to each combinational block *(kernel)* in the design. However to apply CSTP to real circuits, several important factors have not been accounted so far. In this paper we shall address some of these problems.

In [1] no unified approach is provided for selecting registers for the circular path. The registers selected will affect the test efficiency and fault coverage of the design. We will show that by using the *identity modes (I-modes)* [3] of kernels, the area and/or test time required to make a design CSTP testable can be reduced. To guarantee adequate pattern coverage, some registers in the design have to be configured as maximal length pattern generators. Equations are provided to determine the test time of a design. A procedure is provided for selecting registers for the circular path so that the test time can be reduced.

The paper is organized as follows. In Section 2 we present the theory and equations for computing test time of a CSTP testable design. In Section 3, the influence of of I-modes in the circuit on register selection is presented. A graph model is used to represent a design and a procedure using this graph model is presented for selecting the CSTP registers. In Section 4 we compare the CSTP and BILBO strategies. The necessity for mixed mode testing is discussed. We conclude in Section 5 with a brief summary of the issues.

2 Theory

To ensure complete randomness of patterns, the output bits of a pattern generator should have a 1-probability (i.e. probability of having 1) of 0.5. It is proved in [1] that if at least one input bit of a circular path has a 1-probability of p, where $0 < p < 1$, then the 1-probability of each output bit will approach 0.5 as time proceeds. One important question is how sensitive is this convergence to 0.5 as a function of p? The following theorem answers part of this question.

Theorem 1 *For a given circular path if all inputs are mutually independent and each has a 1-probability of p, then the expected number of clock cycles required for the 1-probability of each output bit to approach* $0.5 \pm \delta$ *is*

$$T_0 = \frac{ln(2\delta)}{ln(|1-2p|)}. \tag{1}$$

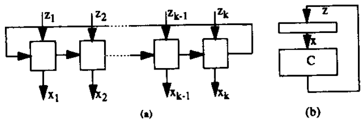

Figure 1: (a)A circular path and (b) its model

Proof : First we shall prove that the 1-probability of each CSTP output will converge to 0.5. Consider the circular path shown in Figure 1(a). Let z_i and x_i denote ith input and output bits respectively. The 1-probabilities of input bit z_i and output bit x_i at time t are denoted as $P(z_i,t)$ and $P(x_i,t)$ respectively. We shall assume that initially all outputs have a 1-probability of 0. In other words,

$$P(z_i,t) = p, \quad 1 \leq i \leq k \text{ and } \forall\, t \geq 0 \tag{2}$$

$$P(x_i,0) = 0, \quad 1 \leq i \leq k \tag{3}$$

After t clock cycles,

$$P(x_i,t) = P(x_j,t-1)[1-P(z_i,t)] + [1-P(x_j,t-1)]P(z_i,t)$$
$$\text{where } j = (i-1)\ mod\ k, \quad 1 \leq i,j \leq k \tag{4}$$

Substituting the values for $P(z_i,t)$ from equation 2 in equation 4, we get

$$P(x_i,t) = p + (1-2p)P(x_j,t-1) \tag{5}$$

Equation 5 implies that for all $t \geq 0$,

$$P(x_1,t) = P(x_2,t) = \ldots = P(x_k,t) \tag{6}$$

Therefore, for $1 \leq i \leq k$

$$P(x_i,0) = 0; \quad P(x_i,1) = p; \quad and \quad P(x_i,2) = 2p - 2p^2$$

In general

$$P(x_i,t) = \sum_{j=0}^{t-1} (-2)^j \binom{t}{j+1} p^{j+1} \text{ for } t \geq 1$$

Since $(1+x)^n = \sum_{i=0}^{n} \binom{n}{i} x^i$, then

$$P(x_i,t) = \sum_{i=1}^{t} (-2)^{-1} \binom{t}{i} (-2p)^i$$

---

[0]*This work was supported by the Defense Advanced Research Projects Agency and monitored by the Federal Bureau of Investigation under Contract No. JFBI90092. The views and conclusions considered in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U. S. Government.

$$
\begin{aligned}
&= \left[(1-2p)^t - (2p)^0 \binom{t}{0}\right](-2)^{-1} \\
&= \frac{1}{2} - \frac{1}{2}(1-2p)^t \quad \text{and} \qquad (7) \\
P(x_i, t) &\rightarrow 0.5 \quad \text{as } t \rightarrow \infty
\end{aligned}
$$

Using equation 7, it is easy to show that for a given p, the number of clock cycles T_0 required for the probability of each output bit of the circular path to approach $0.5 \pm \delta$ is

$$T_0 = \frac{ln(2\delta)}{ln(|1-2p|)} \quad (p \neq 0.5)$$

Note that this model does not exactly match the actual operation of the circuit, as shown in Figure 1(b). Clearly the distribution of input values (x) to C may directly affect the distribution of input values (z) to the CSTP. We assume that these distributions are independent. Thus even if the initial state of the CSTP is all zeros and many of these zeros persist for some time before a stationary state distribution is achieved we assume this transient behavior will not affect the distribution of the output values of C. This is equivalent to assuming the minterms of the function realized by C are uniformly distributed.

From equation 1, the closer p is to 0.5, the shorter it takes for the 1-probability at each output bit of the circular path to approach 0.5. When p is 0.5, it takes exactly one clock cycle for this to be achieved. Consider the worst case where only one input bit of the circular path has 1-probability of p while the rest have 1-probability of 0. In this situation, the CSTP acts as a shift register (except for one bit) and only after every k clock cycles will the 1-probability of each output bit but one be equal. Hence for all output bits to have a 1-probability of $0.5 \pm \delta$, it takes

$$k\left[\frac{ln(2\delta)}{ln(|1-2p|)}\right] \qquad (8)$$

clock cycles. For the case when the 1-probability of all inputs bits are different, we approximate p as $0.5 - \sum_{i=1}^{k} |p_i - 0.5|$, where p_i is the 1-probability of input z_i, for all $i = 1, \ldots, k$. In this case,

$$T_0 = \frac{ln(2\delta)}{ln(\frac{2}{k}\sum_{i=1}^{k} |p_i - 0.5|)} \qquad (9)$$

Inputs 1-probabilities $p_i, i = 1, \ldots, 16$	*Time to reach* 0.5 ± 0.0001	
	simulated	computed
0.1,0.2,0.3,0.4,0.6,0.8,0.4,0.7 0.4,0.9,0.3,0.2,0.5,0.3,0.2,0.4	9	10
0.4,0.6,0,0.7,0,0.8,0.9,0,0.55 0.6,0.8,0.9,0.7,0.8,0.9,0.95	15	16
0.4,0.3,0.2,0.1,0.45,0.15 0.1,0.25,0.01,0.25,0.02 0.03,0.04,0.15,0.25,0.0625	18	20
0.01,0.02,0.0625,0.01,0.04 0.002,0.03,0.14,0.005,0.006,0.02 0.004,0.02,0.035,0.045,0.025	128	129
0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8 0.8,0.8,0.8,0.8,0.8,0.8,0.8,0.8	17	17

Table 1: Results on a 16-bit circular path

Table 1 gives a summary of the simulated and computed results of the number of clock cycles required to achieve an equiprobable distribution at the outputs of a 16-bit circular path. In the experiment it is assumed that each input bit has a constant 1-probability whose value is shown in column 1, and the probability of each output bit is observed at every clock cycle. If the probability of each output bit is within the range 0.5 ± 0.0001, the number of clock cycles is recorded in column 2. The 1-probability for each output bit to reach 0.5 ±0.0001 is computed from equation (9) and shown in column 3. The close match between the computed and simulated values implies equation (9) can be used as an approximation in predicting the number of clock cycles required for convergence.

From the above discussion, it is guaranteed that the patterns generated from the circular path become equiprobable after a relatively short period after the onset of testing. To compute the test time, the next important question is how many clock cycles are necessary to generate a given number of unique test patterns. This problem has been addressed in [4] in dealing with multiple input signature registers (MISRs) as test pattern generators. Since the circular path is also a MISR, the same results are applicable here. It is shown in [4] that if all 2^k input patterns to a k-bit MISR are equiprobable, the number of clock cycles to generate r unique patterns is

$$T = 2^k\left[ln\left(\frac{2^k}{2^k - r}\right)\right] \qquad (10)$$

Let α be the number of unique patterns generated in T_0 clock cycles. During the time to generate r unique patterns (equation (10)), it is possible that some of the α patterns may be repeated. In most cases since $T_0 \ll T$, T can be used as an approximation for the expected number of clock cycles required to test a design. However, for the case when the 1-probabilities of the inputs to the CSTP are either close to 0 or 1, T_0 can be significant and hence should be taken into consideration.

3 Selection of CSTP Registers

Given a design without global reset capability, the CSTP registers must be able to initialize the circuit to a known state. This implies all registers at the primary inputs and a set of registers containing one from each loop must be selected. Other registers are selected based on the test efficiency and the test time required. To guarantee adequate testing of each kernel, we should be able to apply all possible patterns that would occur in normal mode [5]. This implies that the test set for each kernel should be a superset of all patterns that would occur in normal mode. This is referred to as *functional exhaustive testing* [5] and illustrated in example 1.

Example 1 Consider the circuit in Figure 2(a). The circuit consists of kernels C1 through C4 and registers R1 through R5. Using the register selection rules in [1], it suffices to select only R1 to make the circuit CSTP testable. During testing, since R1 acts as a pseudo-input and output, the resulting block K, (composed of C1,R2,R3,C2,C3,R4,R5,C4) depends on $n1$ inputs. If we assume T_0 is infinitesimal, then the expected number of clock cycles required to apply all $2^{n1} - 1$ different patterns to K is $2^{n1} n1 ln(2)$. It can be observed from equation (10) that an infinite time is required to apply all 2^{n1} patterns. As the number of clock cycles increases, the number of unique vectors applied to K asymptotically approaches 2^{n1}. We say that the block K is asymptotically exhaustively tested. During this time C1 is tested asymptotically exhaustively while C2, C3 and C4 are tested functionally exhaustively. □

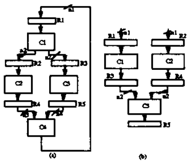

Figure 2: Example circuit (a) with a closed loop (b) with data selector

The basic steps required to make a design CSTP testable are: (1) Select a minimal set of registers R_a such that all loops are broken. (2) Add to R_a, all registers which act as primary inputs or outputs. (3) Connect the registers selected to form a circular path.

In CSTP, the test time is related to the maximum number of inputs that any kernel can have in the test mode. Therefore, it is always appropriate to select the registers such that this value is minimized. For the example circuit in Figure 2(a), if only R1, is selected for the circular path, the expected test time is $2^{n1} n1 ln(2)$. However, if R3 and R4 are selected, the expected test time is $2^{n2+n3}(n3 + n4) ln(2)$.

Typically, circuits may contain multiplexers and/or busses. These kernels are referred to as *data selectors (DS)*. Kernels which are not data selectors are referred to as *data processors (DP)*. Data selectors have I-modes associated with them. For a circuit containing data selectors it may be possible to reduce the test time and/or the area overhead at the expense of a more complicated test schedule. This is illustrated by example 2.

Example 2 Consider the circuit in Figure 2(b) where R1, R2 and R5 are used in the circular path. The expected test time for the design is approximately $2^{2n1+1}n1ln2$. However, if C3 is a multiplexer, then the circuit can be tested in two test sessions. In the first test session the path {R1,C1,R3,C3,R5} is tested. The expected number of clock cycles required in the session is $2^{n1}n1ln2$. During this period C1 is tested assymptotically exhaustively and C3 functionally exhaustively. In the second test session the path {R2,C2,R4,C3,R5} is tested in $2^{n1}n1ln2$ clock cycles. The total test time is reduced to $2^{n1+1}n1ln2$. ☐

Apart from exploiting the structure of a design, the test time can also be reduced by adding more registers to the circular path. Given the kernels obtained when all the CSTP registers have been replaced by primary inputs and outputs, additional registers can be selected so that the largest kernel has m or fewer inputs. If this modification is achievable with the set of registers in the design, then the expected number of clock cycles required to test the design is approximately $2^m mln2$. It should be noted that no additional registers are added to the original circuit.

Example 3 Consider the circuit in Figure 2(b) and assume $n1 > n2$. If the registers selected are R1, R2 and R5, then the expected number of clock cycles required to test the design is approximately $2^{2n1}2n1ln(2)$. Suppose if both R3 and R4 are also included in the circular path, the maximum input dependency (number of inputs on which any kernel depends) reduces to $2n2$, and hence the expected number of clock cycles required to test the design reduces to $2^{2n2}2n2ln(2)$. Apart from reducing the test time, because of the longer length of the circular path, the aliasing probability may also be reduced. ☐

3.1 Selection Procedure

The process for selecting registers can be based on one of the following criteria: (1) minimal area overhead; (2) reduced test time. It is also necessary to provide the designer with a set of feasible solutions and the flexibility to choose solutions to suit his/her needs. To generate feasible solutions, an objective (cost) function needs to be specified. We will illustrate this by considering minimal area overhead solutions.

Before describing such cost functions, we present a circuit model to represent designs. A circuit is modeled as a directed graph. The nodes represent kernels and registers and the edges represent the data paths. There are three types of nodes. They represent data selectors (DS), data processors (DP), and registers (A). The circuit in Figure 3(a) is represented by the graph in Figure 3(b).

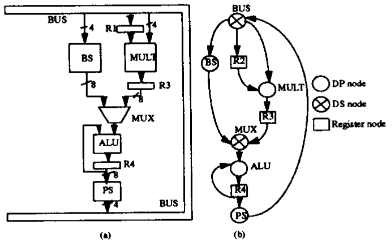

Figure 3: (a) A circuit and (b) its graph model.

The process of selecting CSTP registers involves breaking all the loops in the circuit. This implies simplifying the graph representation of the circuit by considering each setting of the DS node (I-mode) in turn. During a test session, only one I-mode can be selected for every DS node. After the selection of I-modes for each DS node, all reachable and accessable nodes are grouped into partitions. The procedure to determine the graph partitions is illustrated in example 4.

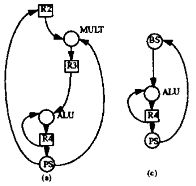

Figure 4: Graph partitioning

Example 4 Consider the graph in Figure 3(b). We start with the node MUX and select the I-mode R3 → ALU and group all reachable (ALU,R4,PS) and accessable (MULT,R2) nodes to it. This is represented in Figure 4(a). The other partition (Figure 4(b)) is obtained by selecting the second I-mode BS → ALU for the node MUX. ☐

A minimal set of partitions is selected to cover the original graph. In example 4, the two partitions shown in Figure 4 cover the original graph. A set of register nodes in each partition is selected such that the resulting circuit is CSTP testable. Each partition of the circuit is tested in a different test session.

In making a circuit CSTP testable, the area overhead is proportional to the sum of the widths of the registers used in the circular path. The registers selected include input registers, output registers and registers such that each loop in the design is broken. Since the sum of the widths of input and output registers is a constant for a circuit, the area overhead is proportional to sum of the widths of registers selected such that each loop is broken. Let the ith partition of the circuit be denoted by the graph $G_i(DP_i, A_i)$, where DP_i represent the set of data processor nodes and A_i a set of register nodes. To obtain a minimal area overhead solution for this partition it is necessary to remove a set of register nodes $R_i \subseteq A_i$ such that the graph $G_i(DP_i, A_i - R_i)$ has no loops, and $\sum_{r_x} l_x$ is minimized, where l_x is the length of r_x and the summation is over all $r_x \in R_i$. For a design having m partitions the cost function is $\sum_{r_x} l_x$, where the summation is over all $r_x \in \{R_1 \cup R_2 \cup \ldots \cup R_m\}$. The minimal area overhead solution can therefore be obtained using the following steps:

procedure minoverhead ():

1. Transform the circuit into the graph representation $G(DS, DP, A)$.
2. Partition the graph $G(DS, DP, A)$ into n subgraphs, where $G_i(DP_i, A_i)$ represents ith subgraph, $i = 1, \ldots, n$.
3. Obtain a minimal set of partitions (say m) that cover all nodes and edges in the original graph.
4. Remove a set of register nodes from $G_i(DP_i, A_i)$ such that each loop is broken and $\sum_{r_x} l_x$ is minimized, where the summation is over all $r_x \in \{R_1 \cup R_2 \cup \ldots \cup R_m\}$ and R_i is the set of registers removed from $G_i(DS_i, A_i)$.
5. Add to $R_1 \cup R_2 \ldots \cup R_m$ the set of registers that act only as primary inputs and outputs.

It must be noted that if m subgraphs cover the entire circuit then the circuit can be tested in at most m test sessions. The test time of the design is related to the number of inputs to the largest kernel when the design is CSTP testable. Reducing the test time is simply a process of adding more registers to the circular path such that the input dependency of each kernel is minimized. Assume that for the minimal area overhead solution each kernel depends on at most k inputs. To obtain the minimal test time solution, we continually reduce x by adding more registers to the circular path until no other solution is possible. The last solution obtained is the minimal test time solution with a condition that no new registers are added to the circuit. The number of clock cycles required to test the design can be computed from equation (10). Any other solution obtained in this process is an intermediate solution.

Example 5 Considering the graphs in Figure 4(a) and (b) for a minimal area overhead solution, only R4 is required to make the circuits modeled by both graphs testable. Two test sessions are required to test the design. In the first session the circuit represented by the graph in Figure 4(a) is tested. Since the resulting kernels when all CSTP registers are replaced by primary inputs and outputs depends on 8 inputs, the circular path has to be clocked for approximately 1500 cycles. The rest of the circuit is tested in the second session. 1500 clock cycles are required in this case. The expected number of clock cycles required to test the design using the CSTP technique is 3000. The CSTP testable version of the circuit in Figure 3(a) is illustrated in Figure 5(a).

4 CSTP vs. BILBO

Using the BILBO technique, a testable version of the circuit shown in Figure 3(a) with minimal area overhead is illustrated in Figure 5(b). Compared to the CSTP testable version, two additional registers {Re1, Re2} and a multiplexer are required. Three test sessions are necessary. The details are listed in Table 2. The total test time for the design, ignoring the time to switch between the test sessions is 66048 clock cycles. Table 3 lists the differences between BILBO and CSTP techniques.

session	RPG	SA	kernels tested	test time
1	Re1, R2	Re2, R3	MULT	256
2	R3, R4	Re2	ALU	65536
3	R4	Re1	PS	256

Table 2: BILBO test sessions

The main difficulty with the CSTP technique is to ensure that all possible vectors that occur in normal operation are applied to each kernel in the design. If this condition is not achieved then the fault coverage obtained can be very low. This is especially important when kernels are driven from primary inputs as illustrated in example 6.

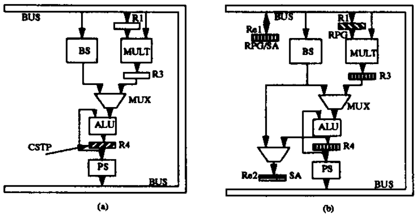

Figure 5: Testable versions (a) CSTP (b) BILBO

Parameter	*BILBO*	*CSTP*
Area overhead	High	Low
Test time	Low	High
Fault coverage	≃ 100 %	Circuit dependent
Implementation complexity	Difficult. Registers have different operation modes	Easy. Registers have same operation modes
Test Control	Complex	Simple
Performance degradation	Can be high. Circuit dependent.	Usually low.
Extra I/O pins	> 3 (Depends on controller design)	3 (2 for scan path and 1 for control)

Table 3: Comparison between CSTP and BILBO

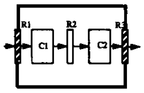

Figure 6: Example circuit

Example 6 In Figure 6, though the input from the circular path to R1 has a 1-probability of 0.5, at each clock cycle only two state changes are possible as oppose to up to 2^m for an m-bit wide register driven by a combinational kernel. Suppose, for example, that R1 is 4-bits wide and the present state is 0101, then the next state can either be 0010 or 1010. Suppose the state 1010 appears, then a possible vector sequence that may be applied to C1 is 0101, 1010, 0101, 1010, 0101, ... This implies that during the computed test time, we cannot guarantee that all possible test vectors are applied to the kernels driven from the primary inputs. To circumvent this problem, we propose a mixed mode test strategy where all the registers at the primary inputs are configured into maximal length pattern generators. For the example in Figure 6, we suggest that R1 be made an RPG while only R3 is used in the circular path. For this example, the test time will be reduced to 2^4 clock cycles, since during this period C1 is tested exhaustively and C2 functionally exhaustively. □

5 Conclusion

In this paper we have presented a modified CSTP technique to enhance fault coverage as a function of test application time. Equations are provided to compute the test time for CSTP testable designs. A procedure is presented for selecting CSTP registers such that each combinational kernel can be tested functionally exhaustively. A comparison of the BILBO and CSTP techniques is provided. The necessity for mixed mode testing where some registers in the design are configured as maximal length pattern generators is also discussed.

Future work involves approaches to reduce test time by using weighted patterns (1-probabilities of outputs of the circular path need not necessarily be 0.5). The effect of fault coverage due to kernels that do not produce all possible output patterns will be investigated.

Acknowledgements: The authors would like to thank Amitava Majumdar for helpful discussions.

References

[1] A. Kraśniewski and S. Pilarski. Circular Self-Test Path: A Low-Cost BIST Technique for VLSI Circuits. *IEEE Trans. on CAD*, 8(1):46–55, January 1989.

[2] S. Sastry and A. Majumdar. Test-Efficiency Analysis of Random Self-Test of Sequential Circuits. *To appear in IEEE Trans. on CAD*, 1991.

[3] M. S. Abadir and M. A. Breuer. A Knowledge-Based System for Designing Testable VLSI Chips. *IEEE Design & Test*, 2(4):56–68, August 1985.

[4] M. Rudolph. Feedback-Testing by using Multiple Input Signature Registers. *Journal of Electronic Testing: Theory and Applns.*, 1(3):213–219, October 1990.

[5] A. Kraśniewski and A. Albicki. Automatic Designing of Exhaustively Self-Testing Chips with BILBO Modules. In *Proc. Int'l Test Conf.*, pages 362–371, November 1985.