Analysis and Minimization of Test Time in a Combined BIST and External Test Approach

Makoto Sugihara† Hiroshi Date‡ Hiroto Yasuura†

†Department of Computer Science
and Communication Engineering,
Kyushu University
6–1 Kasuga-koen, Kasuga,
Fukuoka 816-8580 Japan
{sugihara,yasuura}@c.csce.kyushu-u.ac.jp

‡Institute of Systems &
Information Technologies/KYUSHU
Fukuoka SRP Center Building 7F,
2-1-22, Momochihama Sawara-ku,
Fukuoka City 814-0001, Japan
date@k-isit.or.jp

Abstract

In this paper, an analysis of test time by CBET (which is an acronym for Combination of BIST and External Test) test approach is presented. The analysis validates that CBET test approach can achieve shorter testing time than both external test and BIST in many situations. An efficient test time minimization algorithm for CBET-based LSIs is also proposed. It uses several characteristics of CBET test approach derived by the analysis to reduce computation time to find the optimum test sets. The algorithm helps designers to save their precious design time.

1 Introduction

Recent remarkable advances of LSI technology have been increasing the number of transistors on a chip dramatically. System designers can build a large system on a single chip as an SOC (System-On-a-Chip). They come to use multiple pre-designed and pre-verified blocks, which are called cores in the rest of this paper, to shorten time for design and verification. These cores include black-boxed cores whose detail is invisible due to the protection of intellectual property (IP).

The increase of transistors leads to that of test vectors to be applied. The increase of test vectors leads to that of test time and therefore enhances test cost per chip. Many studies to reduce test time have been presented. In the area of test generation, the main concern of researchers is test set compaction technique [3–5]. The goal of these studies is achievement of as small test set as possible. In the area of Built-In Self Test (BIST), researchers have studied techniques of easy detection for random-pattern-resistant faults [6]. Recently, several researches on test scheduling to reduce test time have been presented. Test scheduling for scan-based test is shown in [1]. This research is effective in test time reduction but not sufficient for LSIs in which the frequency gap between on chip and off chip is salient. In [7, 8], a test scheduling is proposed in which the combination of BIST and external test is used. Several test sets each of which has a difference combination from the others are generated for each core using the combination and the optimum test set for each core are selected. This methodology necessitates serious computation time to derive the optimum test scheduling. As denoted the above, test methods are demanded by which designers can achieve a short test time.

In this paper, testing time reduction by CBET test approach is discussed. CBET is an acronym for Combination of BIST and External Test. The main contribution of this paper is twofold. One part of our contribution is an analysis of testing time by CBET test approach. CBET test approach was proposed in [7, 8] but the reason why the approach can reduce testing time has never been shown yet (here note that the term “CBET” is defined not in [7,8] but in this paper for the first time). And only core-based systems are concentrated on in [7,8], that is, test time reduction of general circuits is not discussed. In this paper, testing time for general circuits by CBET test approach is analyzed qualitatively and experimentally. The other part of our contribution is an efficient test time minimization algorithm for the problem [7, 8]. In [7, 8], testing time minimization problem for core-based LSIs is defined. The problem is minimization of core-based LSIs by selecting the most suitable CBET test set for each core. The problem is a combinatorial optimization one and the algorithm presented in [7, 8] derives the optimum solution by searching almost exhaustively. In this paper, an efficient test time minimization algorithm for the problem considering CBET characteristics derived by the analysis.

In this paper, an overview of test time reduction by CBET test approach is shown in Section 2. Testing time of CBET test approach is analyzed qualitatively and experimentally in Section 3. The analysis clarifies several characteristics of CBET test approach. And moreover such characteristics can alleviate computation of the problem in [7, 8]. An efficient algorithm for the problem is discussed in Section 4. Section 5 concludes this paper with a summary.

2 CBET

The CBET test approach uses BIST and external test to reduce testing time and pin memory size on an LSI tester. CBET test approach does not restrict the structure of propagating test vectors in both BIST and external testing.

How do speed to apply a test vector in BIST and that in external test affect test time? In this paper, a test vector is defined as a value of primary inputs and primary outputs of a CUT (Circuit Under Test) to test it. The structure of BIST and external test is shown in Figure 1 in general [9]. The rectangle drawn by the dashed line represents a boundary between a chip and an LSI tester in external test and the one drawn by the solid line represents that in BIST. There are mainly two differences depend on the boundaries. The one is the place where test vectors are laid. Test vectors in BIST are laid on the chip and those in external test off chip. This may lead to the difference of clock frequency. Of course, there is no difference of clock frequency between in BIST and in external test when test vectors for external test are laid on chip. The other is the number of clock cycles to propagate a test vector. Test vectors in BIST can be laid nearer to inputs and outputs of a CUT than those in external test. Test vector in BIST can be therefore applied within smaller clock cycles than those in external test.

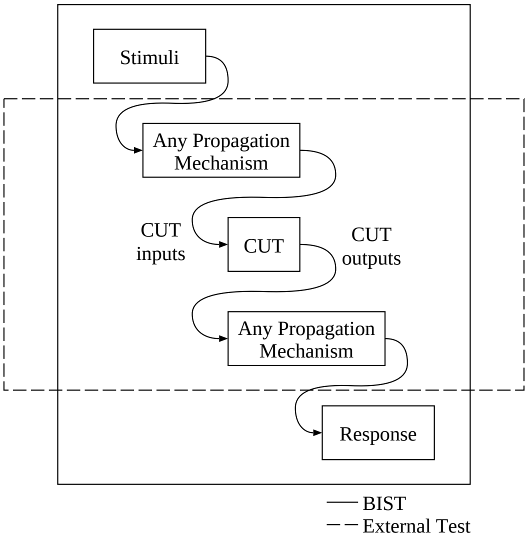

Figure 1. Test structure.

How does the test quality per vector in BIST and external test affect test time? In this paper, test quality per test vector at fault coverage c, $Q(c)$, is defined as the number of faults detected by the test vector at c. Let us discuss the test quality of both. To avoid the influence of test structure on test time, assume that test structure have no effect on test time, that is, time to apply a test vector of ATPG is identical to that of BIST, in this paragraph. ATPG generally generates utmost test vectors in the test quality as there exist many good techniques which enhance the test quality of vectors. But BIST generally generates poor test vectors in the test quality as those attribute to pseudo-randomness. Figure 2 and 3 show the average of test quality per vector in applying the vector at any fault coverage. Figure 3 is a part zoomed in high fault coverage part of Figure 2. The horizontal axis is fault coverage and the vertical axis is the average of test quality per vector. The average is calculated using two values. The first value is the number of test vectors previously applied and test vectors next applied to detect more faults. The second one is the number of faults detected by such test vectors. Figure 2 shows that test quality of ATPG at a fault coverage c, $Q_E(c)$, is higher than that of BIST, $Q_B(c)$. Therefore ATPG achieves a smaller test set than BIST does as a result of their own test quality. Figure 3 shows that test vectors of ATPG can moderately keep test quality whereas those of BIST cannot keep test quality. Test quality of BIST is extremely poor late in test than that of ATPG. Almost all test vectors of BIST late in test detect no faults. Both test methods achieve 99.97% of fault coverage. The numbers of required test vectors in ATPG and BIST are 64 and 39667 respectively to achieve the fault coverage.

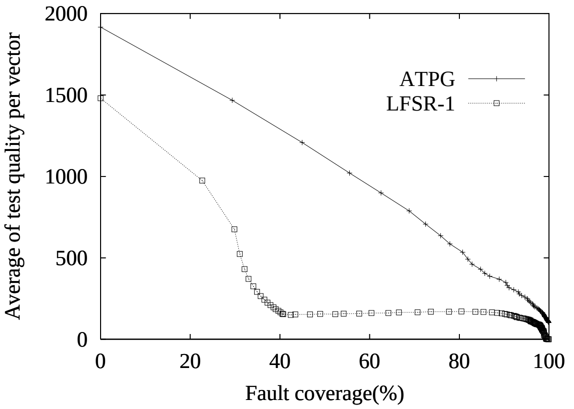

Figure 2. Fault coverage vs. test vectors.

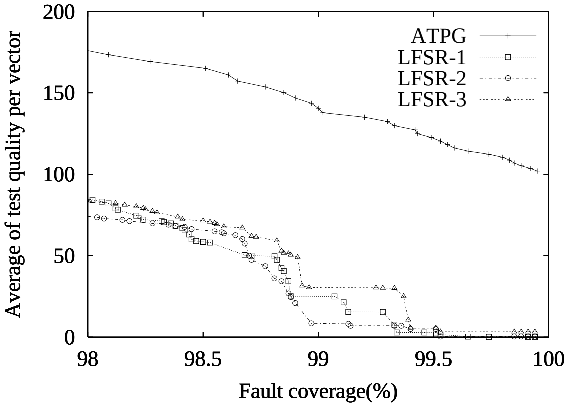

Figure 3. Fault coverage vs. test vectors.

As is stated above, BIST has the advantage of clock frequency and/or clock cycles to apply a vector and external test is advantageous in test quality. Frequency for BIST, F_B, is often higher than that for external test, F_E, and clock cycles to apply a test vector in BIST, AC_B, is smaller than those in external test, AC_E. Frequency to apply a vector in BIST, F_B/AC_B, is defined as test speed of BIST, S_B, and that in external test, F_E/AC_E, is defined as test speed of external test, S_E. If $S_B = S_E$ and $Q_E(c) > Q_B(c)$ in any c, you have only to adopt external test in testing your designs. If $S_B \gg S_E$, there may be no need to use external test in detecting stuck-at faults. Intuitively speaking, if there is one fault coverage that satisfies

$Q_B(c) \cdot S_B = Q_E(c) \cdot S_E$, BIST should be applied until fault coverage c is achieved and external test until a desired fault coverage. In CBET test approach, the most suitable test method is applied in conformity with test process.

3 An Analysis of Testing Time

In this section, the validity of the CBET test approach is qualitatively analyzed. The analysis in this section is useful to estimate CBET test time of circuits but the estimation techniques should be studied as future work as the estimation process takes much computation time. And the analysis is also useful to reduce computation time to minimize test time of Core-Based ICs such as an algorithm described in Section 4.

3.1 Assumptions

We use the following assumptions to explain CBET testing time reduction mechanism briefly.

- No redundant fault in target systems.
- A characteristic polynomial of an LFSR used for BIST is primitive. The LFSR generates M-sequences.
- The bit-width of LFSR, l, equals to the number of inputs of a CUT.
- Faults to be detected only by all-0-input vector do not exist.
- The number of test vectors by ATPG is equal to that of elements in the maximal independent fault set [2].
- A circuit can be tested by only one BIST structure, that is, AC_B is constant.
- A circuit can be tested by only one external test structure, that is, AC_E is constant.
- A test vector in BIST is applied within a cycle, that is, $AC_B = 1$.

3.2 Notations

The following notations are used to analyze testing time reduction by CBET test approach.

- $\mathbb{F}$: A set of all faults to be detected.
  $\mathbb{F} = \{f_1, f_2, ..., f_{|\mathbb{F}|}\}$
- $\mathbb{TV}_f$: A set of all test vectors which can detect a fault f.
- tv_f : A test vector in $\mathbb{TV}_f$.
- $\mathbb{IF}$: An independent fault set
  $\mathbb{IF} = \{if_1, if_2, ..., if_{|\mathbb{IF}|}\}$
- $F(v)$: A set of faults detected by a test vector v.
- $P(A_i)$: A probability of emergence of A_i in terms of an event A.
- D_x^f : An event that a fault f has been detected within x cycles.
- N_x^f : An event that a fault f has not been detected within x cycles.
- $P_{A_i}(B_j)$: A conditional probability of emergence of B_j in terms of an event B when an event in terms of A is A_i.

3.3 Analysis

In this section, an analysis on fault coverage versus the number of test vectors in BIST, external test and CBET is discussed.

Firstly, an analysis on fault coverage versus the number of test vectors in BIST is shown. In detecting a fault f by BIST, a probability that the fault f cannot be detected within x cycles is formulated as follows.

$$\begin{aligned} P(N_x^f) &= P(N_1^f) \cdot P_{N_1}(N_2^f) \cdot \; \cdots \; \cdot P_{N_{x-1}}(N_x^f) \\ &= \prod_{i=1}^{x} \frac{2^l - 1 - (i-1) - |\mathbb{TV}_f|}{2^l - 1 - (i-1)} \end{aligned} \quad (1)$$

Therefore, a probability that the fault f can be detected within x cycles is formulated as follows.

$$\begin{aligned} P(D_x^f) &= 1 - P(N_x^f) \\ &= 1 - \prod_{i=1}^{x} \frac{2^l - 1 - (i-1) - |\mathbb{TV}_f|}{2^l - 1 - (i-1)} \end{aligned} \quad (2)$$

By the formula (2), the fault coverage by BIST at the x cycles, $FC_B(x)$, is formulated as follows.

$$FC_B(x) = \frac{1}{|\mathbb{F}|} \sum_{\forall f \in \mathbb{F}} P(D_x^f) \quad (3)$$

Of course, the formula (3) potentially includes errors which attribute to modeling. In reality, detection for each fault is represented by either zero or one but in the model detection for each fault is continuous value between zero and one. If a fault is detected within x inputs, an error of fault coverage is $1 - P(D_x^f)$. If the fault is not detected within x inputs, an error of the number is $P(D_x^f)$. Probabilities of these errors are $P(D_x^f)$ and $1 - P(D_x^f)$, respectively. So from the viewpoint of probability, Formula (3) can include error between

$$\pm \sum_{\forall f \in \mathbb{F}} P(D_x^f)\{1 - P(D_x^f)\}. \quad (4)$$

Figure 4 presents fault coverage versus clock cycles in C1355 in ISCAS'85 benchmark circuits. The solid line denotes theoretical fault coverage computed by the formula (3) and the dotted lines show fault coverage achieved by miscellaneous LFSRs whose initial value is all-1. Error range is large early in BIST but is small late in BIST as the formula (3).

By the equation (2), test quality per vector in BIST is formulated as follows.

$$Q_B(FC_B(x)) = \sum_{\forall f \in \mathbb{F}} \left\{ P(D_{x+1}^f) - P(D_x^f) \right\} \quad (5)$$

Secondly, an analysis on the fault coverage versus test vectors in external test is shown. In our analysis, the notion of fault independence is used as assumed in Section 3.1. The number of test vectors for external test is equal to the number of elements included in the independent fault set. When a test vector tv_{if} in $\mathbb{TV}_{if}$ is applied to a circuit at fault coverage of zero, the increment of the fault coverage is $|F(tv_{if})|/|\mathbb{F}|$. When test vectors are applied in order such that $tv_{if_1}, tv_{if_2}, ..., tv_{if_n}$, the fault coverage

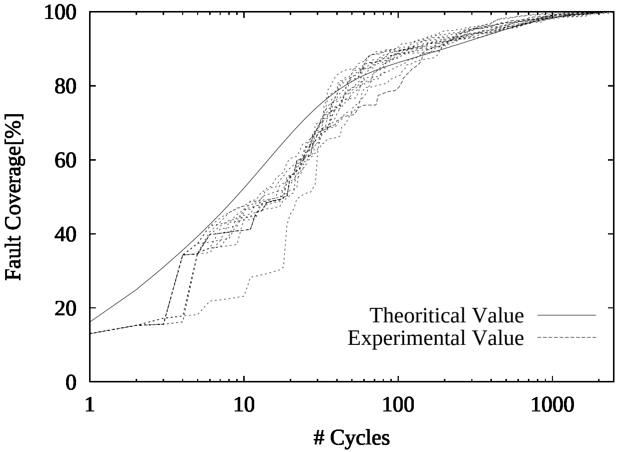

Figure 4. FC vs. test vectors in BIST.

achieved within x cycles, $FC_E(x)$, is formulated as follows.

$$FC_E(x) = \begin{cases} 0 & (0 \le x < AC_E) \\ \dfrac{1}{|\mathbb{IF}|}\left|\displaystyle\bigcup_{i=1}^{\lfloor x/AC_E \rfloor} F(tv_{if_i})\right| & \left(1 \le \dfrac{x}{AC_E} \le |\mathbb{IF}|\right) \end{cases} \tag{6}$$

Finally, we analyze test time of CBET test approach. For convenience of our analysis, let us assume that external test is performed after BIST is done by x cycles. A probability $P(D_x^{if})$ that the independent fault if is detected within x cycles is formulated as follows.

$$\begin{aligned} P(D_x^{if}) &= 1 - P(N_x^{if}) \\ &= 1 - \prod_{i=1}^{x} \frac{2^l - 1 - (i-1) - |\mathbb{TV}_{if}|}{2^l - 1 - (i-1)} \end{aligned} \tag{7}$$

By the above formula, the number of the remains of independent faults, $R(x)$, is formulated as follows.

$$R(x) = |\mathbb{IF}| - \sum_{\forall if \in \mathbb{IF}} P(D_x^{if}) \tag{8}$$

From the formula (7), we can see that the larger $|\mathbb{TV}_{if}|$ is, the more easily an independent fault if is detected. So the remains of independent faults after x cycles of BIST consist of $R(x)$ faults which are the most difficult to detect among all the faults in $\mathbb{IF}$. Here it is assumed that if $i < j$ then $|\mathbb{TV}_{if_i}| < |\mathbb{TV}_{if_j}|$. Test quality of external test after x cycles of BIST is formulated as follows.

$$Q_E(FC_B(x)) = \left| F(tv_{if_{R(x)}}) - \bigcup_{i=R(x)+1}^{|\mathbb{IF}|} F(tv_{if_i}) \right| \tag{9}$$

Testing time achieved by CBET test approach, $T(x)$, is formulated as follows.

$$T(x) = \frac{x}{S_B} + \frac{R(x)}{S_E} \tag{10}$$

If BIST cycle x exists which satisfies

$$Q_E(x) \cdot S_E > Q_B(x) \cdot S_B, \tag{11}$$

test time can be reduced by CBET test approach. The analysis in this section can be applied to designs with small extension, in which test vectors of external test are implemented in ROM. All you have to do is to manage both F_B and F_E as the same.

3.4 Experimental Results

In this section, test time by CBET test approach is experimentally analyzed. Characteristics of circuits used for experiments are shown in Table 1. Circuits are of several circuits of ISCAS'85 benchmark. Figure 5 shows CBET various test sets which achieve the same of fault coverage of 100%. The horizontal axis is the number of test vectors for BIST and the vertical axis is that for external test. It can be understood that test quality of BIST late in test is extremely poor and so test quality of external test is much higher than that of BIST late in test.

Table 1. Circuit characteristics.

Circuit Name	Input Lines	Output Lines	Total Cells	Faults
C432	36	7	157	513
C499	41	32	202	750
C880	60	26	383	942
C1355	41	32	546	1566
C1908	33	25	878	1862
C3540	50	22	1620	3126

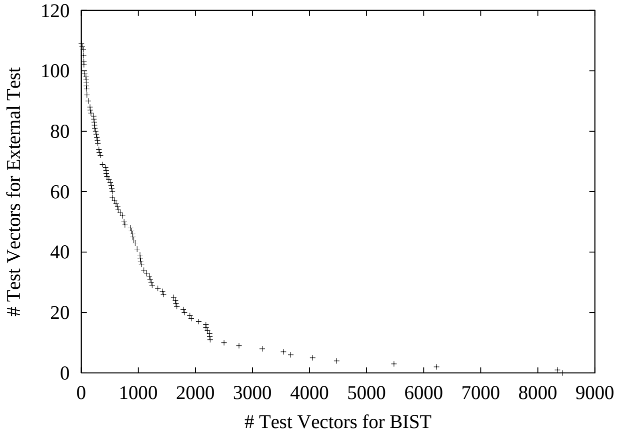

Figure 5. Test sets by CBET in C1908.

Figure 6 shows test time achieved by various CBET test sets in C1908 when $S_B/S_E = 100$. The ratio is reasonable because external test necessitates many clock cycles even if $S_B = S_E$. The horizontal axis is the number of BIST test vectors included in a test set and the vertical axis is test time. The minimum test time is $33.57 \times S_E{}^{-1}$. CBET test approach achieved a 60.17% reduction in test time of BIST only. It also achieved a 70.03% reduction in test time of external test only. As mentioned in Section 2, there exists the upper limit of S_B/S_E for test time reduction by CBET. The upper limit of S_B/S_E for the circuits are shown in Table 2.

Table 2. The upper limit of speed ratio.

	C432	C499	C880	C1355	C1908	C3540
S_B/S_E	455.0	261.0	567.4	394.0	1101.5	578.9

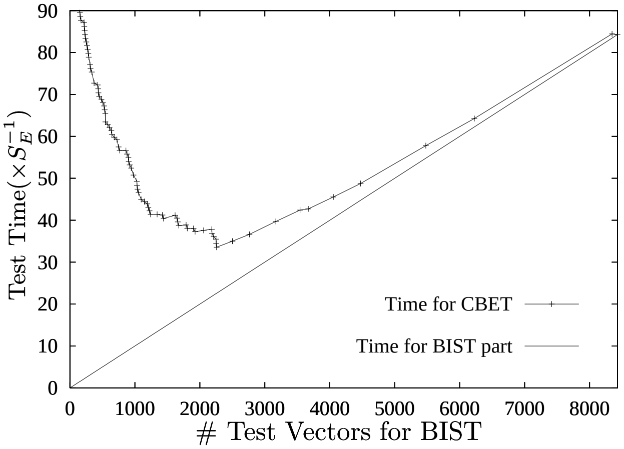

Figure 6. CBET test time in C1908.

4 A Test Time Minimization Algorithm for Core-Based LSIs

In this section, an approximation algorithm for the problem in [7, 8] is shown.

4.1 Definitions

- n : The number of cores.
- $\mathbb{V}_i$: A set organized by several sets, which are based on CBET, of test vectors for core i.
- v_i : A set of test vectors for core i. v_i is an element in $\mathbb{V}_i$, which consists of BIST part and external testing part and satisfies required fault coverage.
- $C_B(v_i)$: The number of cycles to apply a vector in BIST for a set v_i.
- $C_E(v_i)$: The number of cycles to apply a vector in external testing for a set v_i.
- $T_E(v_i)$: Test time for an external test part of v_i.
- $T_B(v_i)$: Test time for an BIST part of v_i.
- $T_S(v_i)$: Testing time for a core i by v_i, that is, $T_E(v_i) + T_B(v_i)$.
- $T(\mathbf{v})$: Testing time for a core-based LSI where test set $\mathbf{v} = (v_1, \ldots, v_n)$ is given.

4.2 Problem Description

Test time minimization problem for core-based LSIs is defined as bellow: [7, 8]
For given $\mathbb{V}_i$ for all i, find v_i in $\mathbb{V}_i$ which minimizes the testing time $T(\mathbf{v})$, where

$$T(\mathbf{v}) = \max\left\{\sum_{i=1}^{n} T_E(v_i), \max_{i=1}^{n} T_S(v_i)\right\} \qquad (12)$$

4.3 Notations

In this section, the notations shown in Section 3.2 are used. Extended notations for a proposed algorithm are shown as below.

- $\mathbb{BT}_i = \left\{\dfrac{C_B(v_i)}{F_B} \mid \forall v_i \in \mathbb{V}_i\right\}$
- $bt_i \in \mathbb{BT}_i$
- $ET(bt_i)$: Time for external test when BIST time is bt_i.
- $ST(bt_i) = bt_i + ET(bt_i)$
- $\mathbb{M}_i = \left\{x \mid \forall\, x \in \mathbb{V}_i, T_S(x) = \min_{\forall v_i \in \mathbb{V}_i} T_S(v_i)\right\}$
- $m_i \in \mathbb{M}_i$
- Right(v) : The least BISTed test set among test sets which are more BISTed than the test set v. If such a test set does not exist, it returns ε.
- Left(v) : The most BISTed test set among test sets which are less BISTed than the test set v. If such a test set does not exist, it returns ε.

4.4 Assumptions

In combinatorial optimization problem, tree pruning is very important to solve it efficiently. On the purpose of tree pruning, it is assumed the following assumptions :

- $|\mathbb{M}_i| = 1$
- $ST(bt_i)$ monotonously decreases, where $0 \leq bt_i \leq T_S(m_i)$.
- $ST(bt_i)$ monotonously increases, where $T_S(m_i) \leq bt_i \leq$ the upper limit of bt_i.
- $ET(bt_i)$ monotonously decreases.

4.5 Problem Solutions

Lemma 1 Minimized testing time is $\max_{i=1}^{n} T_S(m_i)$ or longer.

Proof. By the third assumption, the following formula can be derived.

$$\begin{aligned} T(\mathbf{v}) &= \max\left\{ \sum_{i=1}^{n} T_E(v_i), \max_{i=1}^{n} T_S(v_i) \right\} \\ &\geq \max_{i=1}^{n} T_S(v_i) \\ &\geq \max_{i=1}^{n} T_S(m_i) \end{aligned}$$

Theorem 1 If $\exists v_i \in \{v | v \in \mathbb{V}_i, T_B(v) < T_B(m_i)\}$ makes a solution for testing time minimization problem, the v_i can be substituted by m_i.

Proof. Let us assume that $\mathbf{v} = (v_1, ..., v_i, ..., v_n)$ can minimize $T(\mathbf{v})$ and that the v_i is included in the set $\{v | v \in \mathbb{V}_i, T_B(v) < T_B(m_i)\}$. By the second and fourth assumptions, the following formulas can be derived.

$$T_S(\forall x \in \{v | v \in \mathbb{V}_i, T_B(v) < T_B(m_i)\}) > T_S(m_i) \quad (13)$$

$$T_E(\forall x \in \{v | v \in \mathbb{V}_i, T_B(v) < T_B(m_i)\}) > T_E(m_i) \quad (14)$$

By the formula (13), the following formula is derived.

$$\begin{aligned} \sum_{i=1}^{n} T_E(v_i) &= T_E(v_1) + ... + T_E(v_i) + ... + T_E(v_n) \\ &> T_E(v_1) + ... + T_E(m_i) + ... + T_E(v_n) \end{aligned} \quad (15)$$

And by the formula (14), the following formula is derived.

$$\begin{aligned} \max_{i=1}^{n} T_S(v_i) &= \max\{T_S(v_1), ..., T_S(v_i), ..., T_S(v_n)\} \\ &\geq \max\{T_S(v_1), ..., T_S(m_i), ..., T_S(v_n)\} \end{aligned} \quad (16)$$

By the formula (15) and (16), the following formula is derived.

$$\begin{aligned} T(\mathbf{v}) &= \max\Big\{\sum_{i=1}^{n} T_E(v_i), \max_{i=1}^{n} T_S(v_i)\Big\} \\ &\geq \max\Big[T_E(v_1) + ... + T_E(m_i) + ... + T_E(v_n) \\ &\quad , \max\big\{T_S(v_1), ..., T_S(m_i), ..., T_S(v_n)\big\}\Big] \\ &= T\big((v_1, ..., m_i, ..., v_n)\big) \end{aligned} \tag{17}$$

By the assumption, $T(\mathbf{v})$ is the minimum test time and therefore $T(\mathbf{v}) = T\big((v_1, ..., m_i, ..., v_n)\big)$.

Theorem 2 If testing time of $Time$ can be achieved, $\exists\, v_i \in \big\{x \mid x \in \mathbb{V}_i, T_S(x) \leq Time\big\}$ makes solution for testing time minimization problem.

Proof. Let us assume that $\exists\, v_i \in \big\{x \mid T_S(x) > Time\big\}$ makes solution for the testing time minimization problem if testing time T can be achieved. Then,

$$\begin{aligned} T &= \max\Big\{\sum_{i=1}^{n} T_E(v_i), \max_{i=1}^{n} T_S(v_i)\Big\} \\ &\geq \max_{i=1}^{n} T_S(v_i) \\ &> Time \end{aligned}$$

This is a contradiction and the theorem is proved.

Theorem 3 If testing time of the most BISTed test set in a core among given test sets is $\max_{i=1}^{n} T_S(m_i)$ or shorter, the test set makes solution for testing time minimization problem.

Proof. When the minimized test time is $\sum_{i=1}^{n} T_E(v_i)$, it can be understood that the most BISTed test set whose test time is the minimized test time or shorter can minimize $\sum_{i=1}^{n} T_E(v_i)$ by the fourth assumption. When the minimized test time is $\max_{i=1}^{n} T_S(v_i)$, the most BISTed test set is $\max_{i=1}^{n} T_S(v_i)$ or shorter by Lemma 1 and the above condition.

4.6 Algorithm

The algorithm based on the assumptions in Section 4.4 is presented in Figure 7. By Theorem 1, candidates for searching the optimal test set in a core are m_i or the ones BISTed more than m_i. The algorithm control flow is decided by the relation between $\sum_{i=1}^{n} T_E(v_i)$ and $\max_{i=1}^{n} T_S(v_i)$. When $\sum_{i=1}^{n} T_E(v_i)$ is larger than $T_S(v_i)$, test sets which BISTed higher than currently given test sets are searched in order to reduce $\sum_{i=1}^{n} T_E(v_i)$. When $\sum_{i=1}^{n} T_E(v_i)$ is smaller than $\max_{i=1}^{n} T_S(v_i)$, test sets which BISTed lower than currently given test sets are searched in order to reduce $\max_{i=1}^{n} T_S(v_i)$. When $\sum_{i=1}^{n} T_E(v_i)$ is equal to $\max_{i=1}^{n} T_S(v_i)$, the above two operations are done.

4.7 Experimental Results

In this section, minimized testing time and computation time are presented. The algorithm computed on Intel Pentium Pro processor (200 MHz). The algorithm is implemented with C programming language.

Test Time Minimization Algorithm

```
Procedure Minimize ($\mathbf{v}$)
Input : $\mathbf{v} = (v_1, v_2, ..., v_n)$
Output : The optimal test sets
begin
  repeat
    $T$ := Testtime($\mathbf{v}$);
    if $\sum_{i=1}^{n} T_E(v_i) > \max_{i=1}^{n} T_S(v_i)$ then
      for $i := 1$ to $n$ do
        if Right($v_i$) $\neq \varepsilon$ then
          $\mathbf{v}' := (v_1, ..., \mathrm{Right}(v_i), ..., v_n)$ ;
          if Testtime($\mathbf{v}'$) $< T$ then
            $T$ := Testtime($\mathbf{v}'$) ;
            $\mathbf{v}_{sol} := \mathbf{v}'$ ;
          endif
        endif
      endfor
      if $T <$ Testtime($\mathbf{v}$) then
        $\mathbf{v} := \mathbf{v}_{sol}$ ;
      else
        return $\mathbf{v}$ ;
      endif
    else if $\sum_{i=1}^{n} T_E(v_i) < \max_{i=1}^{n} T_S(v_i)$ then
      for $i := 1$ to $n$ do
        if $v_i \neq m_i$ then
          $\mathbf{v}' := (v_1, ..., Left(v_i), ..., v_n)$ ;
          if Testtime($\mathbf{v}'$) $< T$ then
            $T$ := Testtime($\mathbf{v}'$) ;
            $\mathbf{v}_{sol} := \mathbf{v}'$ ;
          endif
        endif
      endfor
      if $T <$ Testtime($\mathbf{v}$) then
        $\mathbf{v} := \mathbf{v}_{sol}$ ;
      else
        return $\mathbf{v}$ ;
      endif
    else /* $\sum_{i=1}^{n} T_E(v_i) = \max_{i=1}^{n} T_S(v_i)$ */
      $\mathbf{v}' := \mathbf{v}$ ;
      for $i := 1$ to $n$ do
        if $T_S(v_i) = T$ then
          $v_i' :=$ Left($v_i'$) ;
        endif
      endfor
      if Testtime($\mathbf{v}'$) $> T$ then
        return $\mathbf{v}$ ;
      else
        $\mathbf{v}$ := $\mathbf{v}$' ;
      endif
    endif
  until the optimal test sets are found
end
Procedure Testtime ($\mathbf{v}$)
Input : $\mathbf{v} = (v_1, v_2, ..., v_n)$ ;
Output : $T$ ; /* $T$ is test time achieved by $\mathbf{v}$. */
begin
  $T := \max\{\sum_{i=1}^{n} T_E(v_i), \max_{i=1}^{n} T_s(v_i)\}$;
end
```

Figure 7. Test Time Minimization Algorithm.

About 30 million problems were solved. We used 10 ISCAS'85 benchmark circuits for our experiments and the circuits characteristics are shown in Table 4. The number of cores for minimization is from 10 to 20. The number of test sets in each core is shown in Table 3. The second line denotes the number of given test sets and the third line denotes the number of test sets which satisfy the assumptions in Section 4.4. The BIST frequency is 32.0 MHz and the test frequency is 6.4 MHz.

CPU time of the algorithm and reduction ratio

Table 3. The number of test sets in ISCAS '85 benchmark circuits.

Circuit	C432	C499	C880	C1355	C1908	C2670	C3540	C5315	C6288	C7552
# sets (not ideal)	30	46	33	83	96	27	95	63	18	66
# sets (ideal)	29	41	23	68	53	19	60	47	14	61
# cycles per vector in ex.test	2	2	2	2	2	5	2	6	1	7

achieved by CBET minimization method compared with external test are shown in Table 5. CPU time includes time of the idealization action that selects test sets which satisfy the assumptions from the given test sets. If the worst comes to the worst, CPU time is 0.41 seconds. The average of CPU time is 4.21 milli-seconds in minimizing 20 cores.

The reduction ratio is between 37.1% and 97.4%. The more cores are tested in parallel, the more reduction ratio can be achieved. In most of low reduction ratio cases, several C2670 circuits are included and minimized testing time is about 600 micro-seconds. BIST cycles of given test sets for C2670 had bad resolution around 600 micro-seconds and so minimization did not go well. The selection by the algorithm in such cases also achieved lower reduction ratio than that in the other cases. Of course, the optimum solution can be derived by searching exhaustively using the optimal solution and the theorems, if you want not the optimal solution but the optimum one.

Table 4. ISCAS '85 Circuit Characteristics.

Circuit Name	Total Cells	Input Lines	Output Lines	Cycles		Faults
				Ctrl	Obsrv	
C432	157	36	7	2	1	513
C499	202	41	32	2	1	750
C880	383	60	26	2	1	942
C1355	546	41	32	2	1	1566
C1908	878	33	25	2	1	1862
C2670	961	157	64	5	2	1990
C3540	1620	50	22	2	1	3126
C5315	2298	178	123	6	4	5252
C6288	2399	32	32	1	1	7638
C7552	3397	206	107	7	4	7041

Table 5. CPU time and reduction ratio.

# of cores	CPU Time [*ms*]		Reduction Ratio[%]		
	Max	Avr.	Min	Max	Avr.
10	10.0	1.72	37.1	96.0	76.3
11	20.0	1.93	37.1	96.3	76.8
12	20.0	2.13	37.1	96.4	77.3
13	20.0	2.34	37.1	96.5	77.7
14	20.0	2.59	37.1	96.7	78.0
15	20.0	2.82	37.1	96.9	78.3
16	30.0	3.07	37.1	97.1	78.5
17	30.0	3.31	37.4	97.3	78.7
18	40.0	3.58	37.7	97.3	78.9
19	40.0	3.85	38.0	97.4	79.0
20	410.0	4.21	38.3	97.4	79.2

5 Conclusions and Future Works

In this paper, the analysis of test time by CBET test approach is shown. Test speed to apply a vector and test quality per vector in both BIST and external test are introduced. Test speed and test quality of a test vector gave us an analysis of test time by CBET test approach. In our analysis, it is understood that CBET test approach can reduce test time within the valid range, which is determined by both test quality, Q_B and Q_E, and speed ratio S_B/S_E. It was shown that CBET test approach achieved a 60.17% reduction in test time of BIST only and a 70.03% reduction in test time of external test only.

The approximation algorithm based on the CBET characteristics are shown. The maximum reduction ratio by the algorithm is more than 96% compared with external test and the average of the reduction ratio is 76.5%. And computation time was permissible. The average of computation time to minimize testing time in 10 cores was 0.2 seconds. Proposed algorithm is very efficient to minimize testing time for core-based LSIs.

There exists several future works. The one is CBET test circuit synthesis. If a technique which can analyses information in Section 3 with permissible computation time, it helps CBET test circuit synthesis at logic level.

References

[1] J. Aerts and E. J. Marinissen. Scan Chain Design for Test Time Reduction in Core-Based ICs. In *Proc. of International Test Conference*, pages 448–457, October 1998.

[2] S. B. Akers. On the Role of Independent Fault Sets in the Generation of Minimal Test Sets. In *Proc. of International Test Conference*, pages 465–472, October 1987.

[3] J. S. Chang and C. S. Lin. Test Set Compaction for Combinational Circuits. *IEEE Trans. on Computer-Aided Design*, pages 1370–1378, November 1995.

[4] I. Hamzaoglu and J. H. Patel. Test Set Compaction Algorithms for Combinational Circuits. In *Proc. of Int. Conf. on CAD*, pages 283–289, November 1998.

[5] S. Kajihara, I. Pomeranz, K. Kinoshita, and M. Reddy. Cost Effective Generation of Minimal Test Sets for Stuck at Faults in Combinational Logic Circuits. In *Proc. of the Design Automation Conf.*, pages 102–106, June 1998.

[6] H. D. Schnurmann, E. Lindbloom, and R. F. Carpenter. The Weighted Random Test-Pattern Generator. *IEEE Trans. on Computers*, C-24,No.7:695–700, July 1975.

[7] M. Sugihara, H. Date, and H. Yasuura. A Novel Test Methodology for Core-Based System LSIs and a Testing Time Minimization Problem. In *Proc. of International Test Conference*, pages 465–472, October 1998.

[8] M. Sugihara, H. Date, and H. Yasuura. A test methodology for core-based system lsis. *IEICE Trans. Fundamentals*, E81–A(12):2640–2645, December 1998.

[9] Y. Zorian, E. J. Marinissen, and S. Dey. Testing embedded-core based system chips. In *Proc. of International Test Conference*, pages 130–143, October 1998.