

Optimization of test accesses with a combined BIST and external test scheme

Sugihara, Makoto

Department of Computer Science and Communication Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

Yasuura, Hiroto

Department of Computer Science and Communication Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

<https://hdl.handle.net/2324/6794456>

出版情報 : IEICE Transactions on Fundamentals of Electronics. E84-A (11), pp.2614-2622, 2001-11. the Institute of Electronics, Information and Communication Engineers(IEICE)

バージョン :

権利関係 :



Optimization of Test Accesses with a Combined BIST and External Test Scheme

Makoto SUGIHARA[†], *Nonmember* and Hiroto YASUURA[†], *Member*

SUMMARY External pins for test are precious hardware resources because the number of them are strongly restricted. In this paper, an optimization method of test accesses with a combined BIST and external test (CBET) scheme is proposed. The method can minimize test application time and eliminate the wasteful usage of external pins considering the trade-off between test application time and the number of external pins. Our ideas consist of two parts. One is to determine the optimum groups each of which consists of cores to simultaneously share mechanisms for external test. The other is to determine the optimum bandwidth of external input and output for external test. The ideas are basically used for the purpose of eliminating the wasteful external pin usage. The ideas make external test part to be under full bandwidth of external pins under consideration of the trade-off between test application time and the number of external pins. This is achieved only with CBET scheme because CBET permits test sets for both BIST and external test to be elastic. Taking test bus architecture for instance, a formulation for minimization of test application time and experimental results are shown. Experimental results show that our optimization can achieve a 51.9% reduction of test application time of conventional test scheduling and our proposals are surely very effective to reduce test application time of SOC.

key words: *Test Application Time, BIST, External Test, CBET, Test Scheduling, Test Bus*

1. Introduction

Recent remarkable advances of LSI technologies have been increasing the number of transistors on a chip dramatically. System designers can now build a large system on a single chip as an SOC (System-On-a-Chip). They often 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 details are invisible due to the protection of intellectual property (IP) information.

The number of transistors exponentially increases as LSI technologies are improved. The increase of transistors leads to that of test vectors to be applied. The increase of test vectors leads to the serious increase of test application time and therefore enhances test cost per chip. CBET (Combination of BIST and External Test) test approach has been proposed and researched in [2–5], in order to shorten test application time of SOCs and minimize test application time of them which include black-boxed cores. In these literatures, CBET

test approach is introduced and experiments are done for several virtual core-based SOCs. Weaknesses of these proposals is in the neglect of the difference of primary inputs and outputs ports between cores. There are assumptions that external pins are sequentially occupied among cores for external test and test patterns are serialized if the number of ports for CUT is higher than the number of external pins. These assumptions lead to the wasteful usage of external pins. In [2], a method to share BIST circuit among cores is discussed, still sharing simultaneously test buses among cores for external test is not discussed. It is essential that wasteful usage of external pins are removed for reducing test application time and saving external pins for test.

When system designers use test bus architecture in their design, it is necessary to serialize test patterns to apply them to corresponding cores because there are differences between the width of test bus and the number of input and output ports of CUT. In this paper, we propose an optimization method of test accesses, especially for test bus architecture, in which all of cores are assigned to the optimum group and test buses are sequentially occupied among groups (not cores). The group means a set of cores to simultaneously share test buses for external test. Sharing test buses by several cores enhances the usage rate of external pins to deliver test pattern between SOC and tester and therefore test application time reduction can be achieved. And moreover the optimum test bus architecture is searched under constraints of test application time and the number of external pins.

The rest of this paper is organized as follows: In Section 2, test application time minimization by optimization of I/O bandwidth is discussed. In Section 3, the problem defined in Section 2 is extended to our idea of sharing simultaneously test buses among cores. In Section 4, computational complexity and algorithm of test access optimization is discussed. In Section 5, experimental results are shown to validate our proposal. And Section 6 concludes this paper with a summary.

2. Optimization of I/O Bandwidth

In this paper, we assume that test bus architecture is used for a mechanism for external test and all cores have their own BIST circuits. Our ideas can be applied to the other mechanisms such as full scan, partial scan

[†]Department of Computer Science and Communication Engineering, Graduate School of Information Science and Electrical Engineering, Kyushu University

and boundary scan design but those are not discussed for brief explanation. In this section, a CBET test approach [2–5] is extended to the test bus architecture and an optimization of test accesses for the architecture is discussed. Here it is assumed that cores have their own BIST circuits and test buses are used for external test, that is to say, all of cores are tested with CBET. In CBET test approach, all of cores have their own several test sets, whose fault coverages are same. It, however, differs from one another with regard to the number of test patterns for both BIST and external test. The optimum test set for each core is searched to minimize test application time of the SOC.

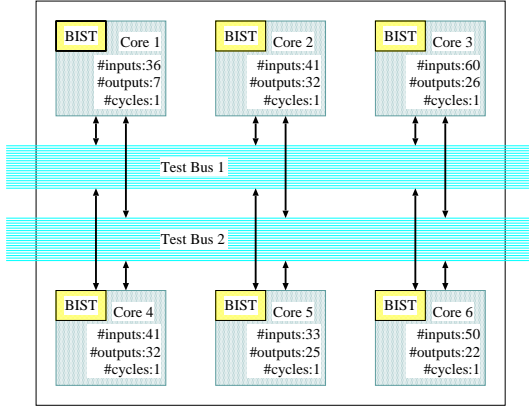


Fig. 1 Test Architecture for an SOC.

The test architecture for an SOC which we tackle is shown in Figure 1. It is assumed that an SOC has two test buses. However, it can be extended to SOC's which have more test buses. Flip-flops are inserted in input and output ports of CUTs (Circuits Under Test) for external test and BIST circuits also inserted there. In CBET test approach for the SOC, it is important that two test buses are connected to all cores because they permit external test to be pipelined. No extra flip-flops for pipelining are necessary because they can be substituted for by flip-flops of BIST circuits which all input and output ports of CUTs have. Here note that pipelining of external test is not a novelty of our proposals (ex. scan design). Our novelty is in reduction of the invalid usage of external pins under pipelining. External pins are precious hardware resources and so must be used without loss. Here note that our proposal does not enhance power consumption because it only lessens the wasteful usage of external pins. The relationship between our proposal and test scheduling for low power is discussed in Section 6.

The test bus width, W , which system designers can use is the sum of the widths of two test buses, W_1 and W_2 . External test generally consists of three operations: Operations of input, execution and output. The input operation means that values of input ports of a CUT are set from somewhere to control. The execution operation means that the values are applied to

the CUT. The output operation means that the resultant values of output ports of the CUT are transported to somewhere to observe. Core i has its own input and output ports as CUT. It also has the number of cycles to execute a test pattern (ordinarily one). Here the number of the input and output ports of the CUT are I_i and O_i respectively. And the number of cycles to execute test is CE_i . If I_i or O_i is larger than the width of the corresponding test buses, it is necessary to serialize the value of inputs/outputs of CUT to transport from/to tester. Therefore the serialization dominates the number of cycles for input and output operation. Test buses of which the widths are WI_i and WO_i are assigned for input and output operations respectively. The number of cycles to control the input ports, CI_i , is

$$CI_i = \lceil I_i / WO_i \rceil,$$

and the number of cycles to observe the output ports, CO_i , is

$$CO_i = \lceil O_i / WI_i \rceil$$

with the proviso that

$$W = W_1 + W_2,$$

$$1 \leq W_1 \leq W_2 \leq W$$

$$\text{if } I_i \leq O_i \text{ then, } WI_i = W_1, WO_i = W_2 \quad (1)$$

$$\text{else } WI_i = W_2, WO_i = W_1. \quad (2)$$

In the CBET test approach, each core has several test sets. A vector of m_i test sets, $\mathbf{v}_i = (v_1, v_2, \dots, v_{m_i})$, is given to core i and the 0-1 integer variable a_{ij} stands for the usage the j th test set. If the j th test set is used, $a_{ij} = 1$, otherwise $a_{ij} = 0$. A test set used for core i , v_i , is shown as follows.

$$\mathbf{v}_i = \mathbf{v}_i \cdot \mathbf{a}_i^T, \quad (3)$$

with the proviso that

$$\sum_{j=1}^{m_j} a_{ij} = 1.$$

When core i is tested by a test set v_i , $V_E(v_i)$ and $V_B(v_i)$ stand for the number of test patterns for external test and BIST respectively. Then CI_i , CO_i , CE_i and $V_E(v_i)$ dominate clock cycles for external test part. The number of total clock cycles for external test, E_i , is shown as follows.

$$E_i(v_i) = \begin{cases} 0 & (V_E(v_i) = 0) \\ CI_i + CE_i + CO_i & (V_E(v_i) = 1) \\ CI_i + \max(CE_i, CI_i) \\ \quad + \max(CO_i, CE_i) + CO_i & (V_E(v_i) = 2) \\ CI_i + V_E(v_i) \\ \quad \cdot \max(CI_i, CE_i, CO_i) \\ \quad + CO_i & (V_E(v_i) \geq 3) \end{cases}$$

An example of the pipelining of external test based on

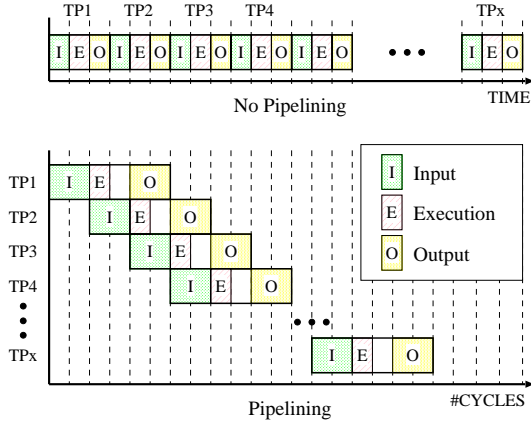


Fig. 2 An example of pipelining.

the above equation is shown in Figure 2.

When clock frequency for external test is F_E , time for external test in core i , TE_i , is shown as follows.

$$TE_i(v_i) = E_i(v_i)/F_E$$

When a test pattern for BIST can be applied within a clock cycle, the number of test patterns for BIST, $V_B(v_i)$, is equal to that of clock cycles for BIST, $B_i(v_i)$. When clock frequency for BIST is F_B , time for BIST, $TB_i(v_i)$ is given by the following formula.

$$TB_i(v_i) = B_i(v_i)/F_B = V_B(v_i)/F_B$$

If there is no dead time to test core i , test application time for core i , $TC_i(v_i)$, is shown in the following formula.

$$TC_i(v_i) = TE(v_i) + TB_i(v_i)$$

According to [3–5], total test application time for the SOC, $T(v)$ is therefore shown as follows.

$$T(v) = \max \left\{ \sum_{i=1}^n TE_i(v_i), \max_{i=1}^n TC_i(v_i) \right\} \quad (4)$$

Test application time minimization for test bus architecture in CBET test approach is generally solved by searching the variables \mathbf{a}_i for all i , W_1 and W_2 to minimize the equation (4). An optimization of test accesses is done by searching W which minimizes test application time and conform to a constraint of the number of external pins considering the trade-off between test application time and the number of external pins. An example of test scheduling for an SOC is shown in Figure 3.

3. Groupage of Cores

Several factors make the optimization of the width of test buses difficult. The factors are the number of input and output ports and the number of cycles to execute a test pattern. If there are no difference in them among cores, we can easily deduce the width of test buses. It is

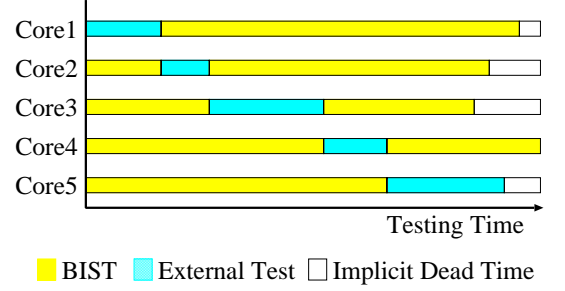


Fig. 3 An example of test scheduling.

intuitively understood that the minimum width of test bus should be searched which minimizes the pipelining time slot, $\max(CI_i, CE_i, CO_i)$, in order to minimize test application time in most of cases. Nonetheless, $(I_i \bmod W)$ and $(O_i \bmod W)$ bits are wasted to input and output a test pattern, respectively. For example, the wasteful use of test buses is shown in Figure 4.

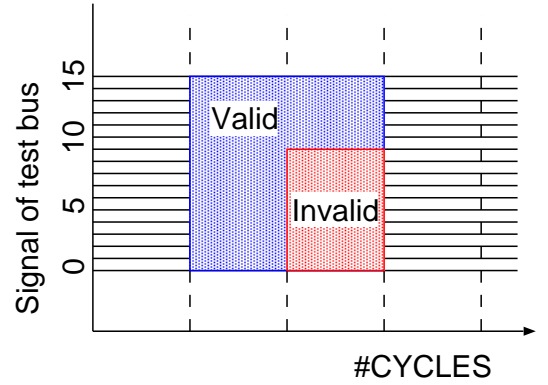


Fig. 4 Wasteful use of test buses.

It is very difficult to derive the optimum test mechanism and scheduling for SOCs and therefore it is very challenging to make them optimum with regard to test application time. Our idea is in the reduction of the wasteful test bits and the parallelization of execution of test patterns. Cores are grouped in order to achieve the reduction and parallelization. Cores in the same group simultaneously share test buses to control input ports and observe output ports. And executions of external test for the cores are simultaneously done. A simple example of our idea is shown in Figure 5. In this example, the group consists of Core 1, Core 2 and Core 3. A test pattern for the group necessitates six clock cycles regardless of pipelining. If test buses are not shared among the cores, the number of clock cycles to apply a test pattern for the group is nine. Test application time minimization with several cores simultaneously sharing test buses is achieved by searching the optimum groups, test sets and the width of test buses.

Now let us define the test application time minimization problem for test bus architecture. In the CBET test approach, each core has several test sets. A test set used for core j , v_j , is formulated as in Equation (3). v_j is shown as follows.

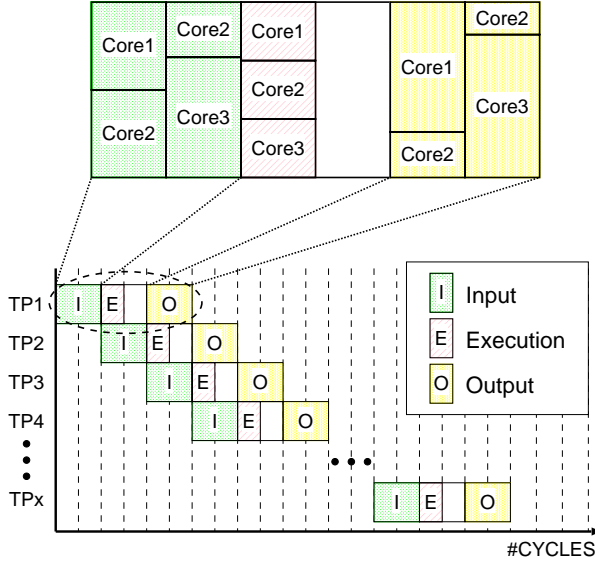


Fig. 5 External test for a group by our idea.

$$v_j = \mathbf{v}_j \cdot \mathbf{b}_j^T,$$

with the proviso that

$$\sum_{k=1}^{m_j} b_{jk} = 1.$$

An assignment of core j to group i is represented by a 0-1 integer variable c_{ij} . If core j is included in group i , then $c_{ij} = 1$, otherwise $c_{ij} = 0$. The variable c_{ij} has the following constraint which means that core i belongs to only a group.

$$\sum_{i=1}^n c_{ij} = 1$$

If no external test for core j is done by test set v_j , $d(v_j) = 0$, otherwise $d(v_j) = 1$. The number of clock cycles for the operation to input a test pattern of group i , CI_i' , is

$$CI_i' = \left\lceil \left\{ \sum_{j=1}^n c_{ij} d(v_j) I_j \right\} / WI_i \right\rceil,$$

and that to output it, CO_i' , is

$$CO_i' = \left\lceil \left\{ \sum_{j=1}^n c_{ij} d(v_j) O_j \right\} / WO_i \right\rceil,$$

with the same proviso as Equation (2). And the number of cycles to execute a test pattern, CE_i' , is shown as follows.

$$CE_i' = \max_{j=1}^n \{c_{ij} d(v_j) CE_i\}$$

The number of test patterns for external test of group i , $V_{E_i}(\mathbf{v})$, is shown as follows.

$$V_{E_i}(\mathbf{v}) = \max_{j=1}^n \{c_{ij} V_{E_i}(v_j)\}$$

The number of clock cycles for external test for group i , E_i' , is formulated as follows.

$$E_i'(\mathbf{v}) = \begin{cases} 0 & (V_{E_i}(\mathbf{v}) = 0) \\ CI_i' + CE_i' + CO_i' & (V_{E_i}(\mathbf{v}) = 1) \\ CI_i' + \max\{CE_i', CI_i'\} \\ \quad + \max\{CO_i', CE_i'\} + CO_i' & (V_{E_i}(\mathbf{v}) = 2) \\ CI_i' + V_{E_i}(\mathbf{v}) \\ \quad \cdot \max\{CI_i', CE_i', CO_i'\} \\ \quad + CO_i' & (V_{E_i}(\mathbf{v}) \geq 3) \end{cases}$$

Clock cycles for BIST for group i , B_i' , is formulated as follows.

$$B_i'(\mathbf{v}) = \max_{j=1}^n \{c_{ij} \cdot V_B(v_j)\}$$

Time for external test and BIST for group i , TE_i' and TB_i' , is shown as follows respectively.

$$TE_i' = E_i' / F_E$$

$$TB_i' = B_i' / F_B$$

If there is dead time to test group i , test application time for group j , TC_i' , is the sum of TE_i' and TB_i' . Therefore total test application time for SOC with several cores sharing test bus is shown as follows according to [3-5].

$$T' = \max \left\{ \sum_{i=1}^n TE_i', \max_{i=1}^n TC_i' \right\} \quad (5)$$

Test application time minimization problem for test bus architecture is solved by find the variables \mathbf{b}_j for all cores, \mathbf{c}_i for all groups, W_1 and W_2 which minimize the equation (5). An optimization of test accesses is done by searching W which minimizes test application time and conform to a constraint of the number of external pins considering the trade-off between test application time and the number of external pins. Computational complexity of test access optimization is proportional

to Bell number, $B_n = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k!}{k^n}$.

4. Test Access Optimization

Computational complexity of test time minimization without test access optimization is $O(c^n)$ where c is a constant value and n is the number of cores. The minimization can be done within less computational time, linear to the number of cores at its best, if we use the test time minimization algorithm introduced in [5]. Test access optimization proposed in this paper is independent of such test time minimization.

It is necessary to discuss a sophisticated algorithm for test access optimization because it is a very time consuming process. Computational complexity of test access optimization is proportional to Bell number,

Table 1 Characteristics of cores.

	C432	C499	C880	C1355	C1908	C2670	C3540	C5315	C6288	C7552
C_i	36	41	60	41	33	157	50	178	32	206
O_i	7	32	26	32	25	64	22	123	32	107
CE_i	1	1	1	1	1	1	1	1	1	1

$B_n = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k!}{k^n}$, where n is the number of cores. Tree pruning is hopeful to reduce computational time for optimization of test accesses. The tree pruning is with regard to the wasteful test bits on test buses. Several cores should be grouped and test time minimization for the groups should be done if the usage rate which the groups use valid test bits on test buses is higher than that for temporal solution. The increase of the wasteful test bits on test buses lead to low efficient usage of test buses for external test. An overview of an algorithm for test access optimization is shown in Algorithm 1. Initial solution is the group each of which has only one core. The maximum number of cores in a group should be a low number when a solution for the optimum test accesses cannot be derived within practical time.

Algorithm 1 Test access optimization

Procedure Optimize(\mathbf{v})
Input: $\mathbf{v} = (v_1, v_2, \dots, v_n)$
Output: The optimal test sets and test bus architecture
for all width of test buses **do**
 // for the trade-off between test time and # of external pins
 for all groupage assignment **do**
 // for reduction of the wasteful test bits on test buses
 Compute the wasteful test bits of test buses for both all groups.
 if the wasteful test bits are reduced by grouping, comparing with temporal solution **then**
 Find the test sets which minimize test application time for the groups using fast algorithm such as [5].
 end if
 end for
end for

5. Experimental Results

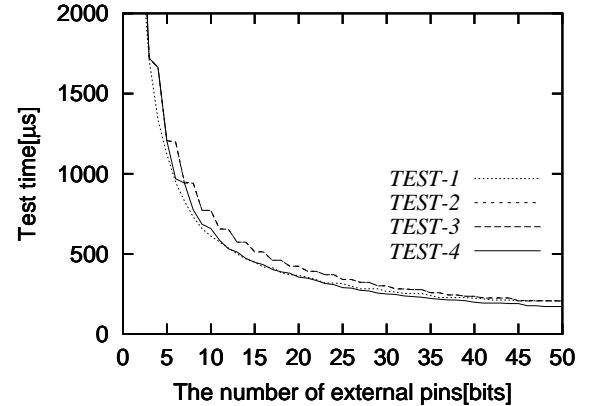
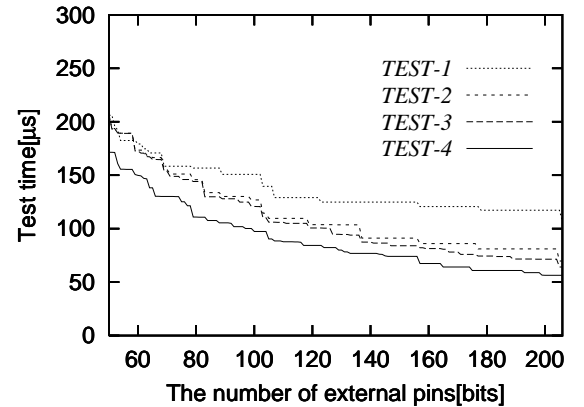
In this section, it is experimentally shown that the simultaneous share of test buses by several cores for external test part in CBET test approach is effective to reduce test application time. And it is also shown that we can design our SOC's under the trade-off between test application time and the number of external pins. We used ISCAS'85 benchmark circuits as cores for a virtual SOC. The circuits are described in Table 1. The number of groups is limited between 6 and 10 for the reason of getting the optimal solution within practical computational time. It assumed that clock frequency for both external test and BIST is 10MHz in our experiments. It is however easy to adopt multi-frequencies to be accommodated to cores.

To validate our proposals, four test methods are used to minimize test application time. The four test methods are described in Table 2. The second column

of “Pipelined” shows whether external test is pipelined or not. The third column of “Grouped” shows whether cores are grouped to reduce invalid bits of external test or not. The fourth column of “Optimization of test buses” shows whether the widths of two test buses is optimized to minimize test application time or not. If not pipelined, only one test bus is used for external testing and all external pins are assigned to the bus. If pipelined without optimization, the widths of test buses are half of the number of external pins. If the number of external pins is an odd number, the width of a test bus is one more than the width of the other test bus.

Table 2 Characteristics of test methods.

Method	Pipelined	Grouped	Optimization of test buses
TEST-1	NO	NO	NO
TEST-2	YES	NO	NO
TEST-3	YES	YES	NO
TEST-4	YES	YES	YES

**Fig. 6** Test application time in the small # of pins.**Fig. 7** Test application time in the large # of pins.

Test application time derived by the four methods is shown in Figure 6 and 7. Figure 6 is in case of the small number of external pins and Figure 7 is in case

of the large number. Normalizing with test application time of *TEST-1*, reduction ratios by *TEST-2*, *TEST-3* and *TEST-4* are shown in Figure 8. According to our experiments, *TEST-2* is more effective than *TEST-1* if the number of external pins is larger than almost 48. The numbers of *TEST-3* and *TEST-4* are more than almost 44 and 11 respectively. It can be understood that test bus architecture should not be pipelined if external pins of which the number is small are available. The maximum reduction ratios by *TEST-2*, *TEST-3* and *TEST-4* are 40.5%, 45.3% and 51.9% respectively. Normalizing with test application time of *TEST-2*, reduction ratios by *TEST-3* and *TEST-4* are shown in Figure 9. Our proposals of both the test bus width optimization and groupage of cores are effective for test time reduction. The test bus width optimization is especially effective. The maximum reduction ratios by *TEST-3* and *TEST-4* are 12.0% and 30.5% respectively. The effectiveness of our optimization increases as the number of external pins increases. It is by reason of that the wasteful bits of test buses increases as the number of external pins increases.

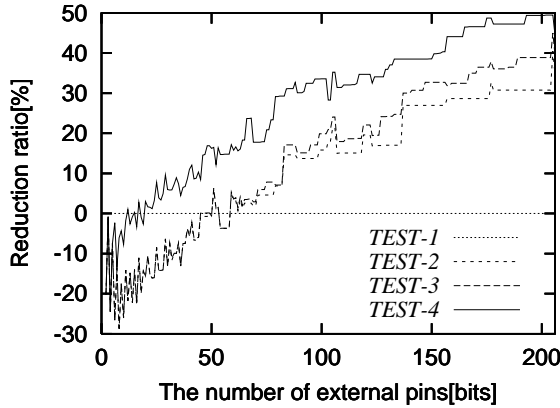


Fig. 8 Reduction ratios of test application time normalized with *TEST-1*.

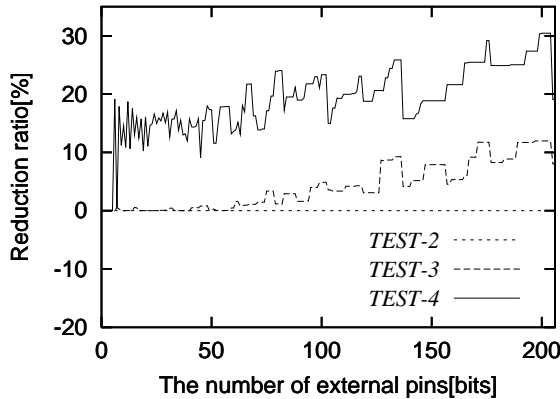


Fig. 9 Reduction ratios of test application time normalized with *TEST-2*.

Experimental results shows that our proposals of grouping cores and optimizing test buses are effective

to reduce test application time of SOC's. It can be also understood that we can design our SOC's considering the trade-off between test application time and the number of external pins such as Figure 6 and 7. External pins for test in future SOC designs will be more than that in the present and so our proposals are effective for test application time reduction of future SOC's.

6. Concluding Remarks

In this paper, optimization of test accesses with CBET scheme was proposed to minimize test application time considering the trade-off between test application time and the number of external pins. The ideas for optimization are to determine the optimum bandwidth of external I/O for external test and to determine the optimum groups each of which consists of cores which simultaneously share mechanisms for external test. And test bus width can be searched conforming to the external pin count constraints by system designers. The optimization of test accesses based on our ideas reduced 51.9% of test application time of conventional method. It was experimentally validated that test scheduling and optimization of test mechanism are very effective for test application time reduction. It is a hopeful test time minimization method in the next SOC era in which system designers will be able to use more external pins for test in their designs.

Our proposal does not enhance power consumption because it only lessens the wasteful usage of external pins. It is expected that our proposals can be easily extended to test scheduling for low power consumption. Power for BIST is higher than that for external test because BIST can be under higher clock frequency than external test and therefore it is natural to partially halt BIST operation to conform to power consumption constraint. Our proposal is independent of the test scheduling for low power, but can indeed enhance fault coverage for external test per power.

References

- [1] J. Aerts, and E. J. Marinissen, "Scan Chain Design for Test Time Reduction in Core-Based ICs," *Proc. of International Test Conference(ITC)*, pp.448–457, October 1998.
- [2] K. Chakrabarty, "Test Scheduling for Core-Based Systems," *Proc. of International Conference on Computer Aided Design(ICCAD)*, pp.391–394, November 1999.
- [3] M. Sugihara, "A Test Methodology for Core-Based LSIs," Master's thesis, Dept. Computer Science and Communication Engineering, Kyushu University, March 1998.
- [4] M. Sugihara, H. Date, H. Yasuura, "A Novel Test Methodology for Core-Based System LSIs and a Testing Time Minimization Problem," *Proc. of International Test Conference(ITC)*, pp.465–472, October 1998.
- [5] M. Sugihara, H. Date, and H. Yasuura, "Analysis and Minimization of Test Time in a Combined BIST and External Test Approach," *Proc. of Design, Automation and Test in Europe(DATE)*, pp.134–140, March 2000.