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

出版情報：Proceedings of the 2002 conference on Asia South Pacific design automation/VLSI Design, pp.683-688, 2002-01. IEEE Computer Society
バージョン：
権利関係：

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

Makoto Sugihara and Hiroto Yasuura
Department of Computer Science and Communication Engineering,
Graduate School of Information Science and Electrical Engineering,
Kyushu University
6–1 Kasuga-koen, Kasuga, Fukuoka 816-8580 Japan
{*sugihara,yasuura*}*@c.csce.kyushu-u.ac.jp*

Abstract

External pins for test are precious hardware resources because this number is strongly restricted. Cores are tested via test access mechanisms (TAMs) such as a test bus architecture. When cores are tested via test buses which have constant bit widths, test stimuli and test responses for a particular core have to be transported over these test buses. The core might require more widths for input and output than test buses, and hence, for some part of the test, the TAMs are idle; this is a wasteful usage of the TAMs. In this paper, an optimization method of test accesses with a combined BIST and external test (CBET) scheme is proposed for eliminating the wasteful usage of test buses. This method can minimize the test time and eliminate the wasteful usage of external pins by considering the trade-off between test time and the number of external pins. Our idea consists of two parts. One is to determine the optimum groups, each of which consists of cores, to simultaneously share mechanisms for the external test. The other is to determine the optimum bandwidth of the external input and output for the external test. Our idea is basically formulated for the purpose of eliminating the wasteful external pin usage. We make the external test part to be under the full bandwidth of external pins by considering the trade-off between the test time and the number of external pins. This is achieved only with the CBET scheme because it permits test sets for both the BIST and the external test to be elastic. Taking test bus architecture as an example, a formulation for test access optimization and experimental results are shown. Experimental results reveal that our optimization can achieve a 51.9% reduction in the test time of conventional test scheduling and our proposals are confirmed to be effective in reducing the test time of system-on-a-chip.

Keywords: test time, CBET, TAM, external pins

1. Introduction

Recent significant advances in 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 a system-on-a-chip (SOC). They often use multiple pre designed and pre verified blocks, hereafter called cores, to reduce the time required for design and verification. These cores include black-boxed cores whose details are unknown due to the protection of intellectual property (IP) information.

The number of transistors on a chip increases considerably as LSI technologies are improved. This increases the number of faults and test vectors. The increase in the number of test vectors leads to the serious increase in test time and therefore increases the test cost per chip. Test time reduction is one of research challenges. Test scheduling has been researched to reduce the test time of digital systems [1–3]. These researches mainly specialize in BIST. Parallelized BISTs increase the power consumption of the test more than that for normal operations and test scheduling is carried out to satisfy a power constraint. This research does not use combination of BIST and external test (CBET) scheme which is a powerful tactic to reduce both the test time and power consumption. The CBET approach has been proposed and researched in [5–9], in order to reduce the test time of SOCs and minimize the test time of those which include black-boxed cores. In this research, the CBET scheme is introduced and experiments are carried out for several virtual core-based SOCs. The weaknesses of these proposals involve the neglect of the difference in primary input and output ports between cores. There exist assumptions that external pins are sequentially occupied among cores for the external test and test patterns are serialized if the number of ports for the core under test (CUT) is larger than the number of external pins. These assumptions lead to the wasteful usage of TAMs. In [5,6], a method to share the BIST circuit among cores is discussed, it is not, however, discussed whether the simultaneous sharing of test buses among cores for the external test is valid for test time reduction or not. It is essential that the wasteful usage of TAMs is eliminated for the reduction of both test time and the number of external pins required for test.

When system designers use test bus architecture in their designs, 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 CUTs. In this paper, we propose an optimization method of test accesses, particularly 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 indicates a set of cores that simultaneously share test buses for the external test. Sharing test buses among several cores enhances the usage rate of external pins to deliver test patterns between an SOC and a tester, and therefore, test time reduction can be achieved by eliminating the wasteful usage of external pins. Moreover, the optimum test bus architecture is sought under the constraints of test time and the number of external pins.

The remainder of this paper is organized as follows: In Section 2, the optimization of I/O bandwidth without groupage of cores is discussed. In Section 3, the problem defined in Section 2 is extended to our idea of the simultaneous sharing of test buses among cores and the formulation is shown. In Section 4, the computational complexity and algorithm of the test access optimization are discussed. In Section 5, our proposal is applied to a virtual SOC which comprises 10 ISCAS benchmark circuits. Experimental results are shown to validate our proposal. 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 the external test and that all cores have their own BIST circuits. Our ideas can be applied to other mechanisms such as full scan, partial scan and boundary scan design, but those are not discussed in our brief explanation. In this section, a CBET scheme [5–9] is extended to 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 the external test, that is to say, all cores are tested with the CBET scheme. In the CBET scheme, all cores have several their own test sets, whose fault coverages are the same. They, however, differ from one another with regard to the number of test patterns for both the BIST and the external test. The optimum test set for each core is sought in order to minimize the test time of SOCs.

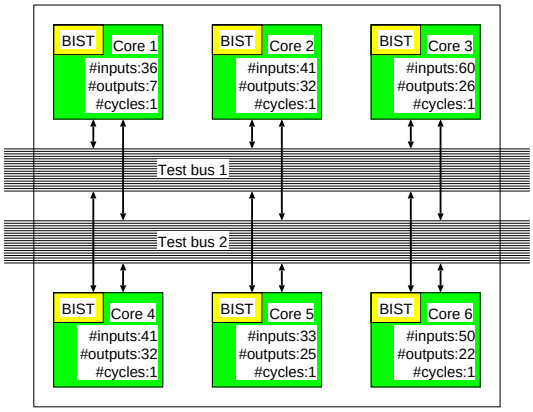

Figure 1. Test architecture for SOCs.

The test architecture for the SOCs which we examine is shown in Figure 1. It is assumed that an SOC has two test buses. However, this test architecture can be easily extended to SOCs which have more test buses. Flip-flops are inserted in input and output ports of CUTs for the external test and BIST circuits are also inserted there. In the CBET scheme for the SOCs, it is important that the two test buses are connected to all cores because they permit the external test to be pipelined. No extra flip-flops for pipelining are necessary because they can be substituted for by the flip-flops of BIST circuits, which all CUTs have. Note that the pipelining of the external test is not novel in our proposals (e.g. scan design). The novel thing here involves the reduction of the invalid usage of external pins under pipelining. External pins are precious hardware resources and must be used without any loss. Here note that our proposal does not increase the 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 use is the sum of the widths of two test buses, W_1 and W_2. The external test generally consists of three operations: input, execution and output. The input operation means that values of the 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 the output ports of the CUT are transported somewhere for observation. Core i has its own input and output ports for the external test as the CUT. It also has a number of cycles to execute a test pattern (an ordinarily one). Here, the number of input and output ports of the CUT is I_i and O_i, respectively. The number of cycles required to execute the 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 the CUT for transport to/from the tester. Therefore, the serialization dominates the number of cycles for input and output operations. Test buses whose 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

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

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

$$CO_i = \lceil O_i / WO_i \rceil ,$$

with the proviso that

$$\begin{gathered} W = W_1 + W_2, \\ 1 \le W_1 \le W_2 \le W \\ \text{if } I_i \le O_i \text{ then, } WI_i = W_1, WO_i = W_2 \\ \text{else } WI_i = W_2, WO_i = W_1. \end{gathered} \tag{1}$$

In the CBET scheme, each core has several test sets. A vector of m_i test sets, $\boldsymbol{v_i} = (v_1, v_2, \ldots, v_{m_i})$, is given to core i. The 0-1 integer variable a_{ij} stands for the usage of the jth test set. If the jth test set is used, $a_{ij} = 1$, otherwise $a_{ij} = 0$. A test set used for core i, v_i, is shown as follows.

$$v_i = \boldsymbol{v_i} \cdot \boldsymbol{a_i}^T, \tag{2}$$

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 the external test and the BIST, respectively. Then, CI_i, CO_i, CE_i and $V_E(v_i)$ dominate the clock cycles for the external test part. The number of total clock cycles for the external test, E_i, is shown as follows.

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

An example of the pipelining of the external test based on the above equation is shown in Figure 2. When the clock frequency for the external test is F_E, the time for the external test of core i, TE_i, is shown as follows.

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

When a test pattern for the BIST can be applied within a clock cycle, the number of test patterns for the BIST, $V_B(v_i)$, is equal to that of clock cycles for the BIST, $B_i(v_i)$.

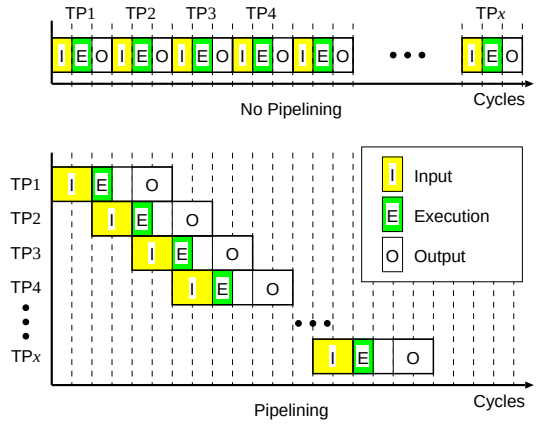

Figure 2. An example of pipelining.

When clock frequency for the BIST is F_B, the 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, the test 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 [7–9], the total test time for the SOC, $T(\boldsymbol{v})$, is therefore shown as follows.

$$T(\boldsymbol{v}) = \max\left\{\sum_{i=1}^{n} TE_i(v_i), \max_{i=1}^{n} TC_i(v_i)\right\} \qquad (3)$$

Test time minimization for test bus architecture in the CBET scheme is generally solved by searching the variables $\boldsymbol{a_i}$ for all i, W_1 and W_2 to minimize Equation (3). An optimization of test accesses is performed by searching W which minimizes the test time and conforms to a constraint of the number of external pins by considering the trade-off between the test time and the number of external pins. An example of test scheduling for an SOC is shown in Figure 3.

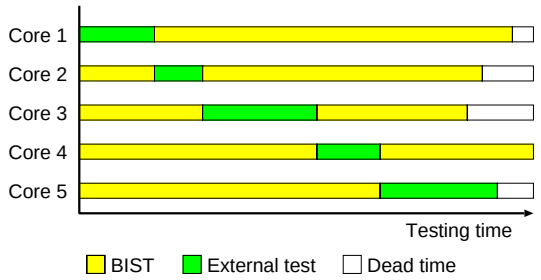

Figure 3. An example of test scheduling.

3. Grouping of Cores

Several factors make the optimization of the width of test buses difficult. These factors include the number of input and output ports and the number of cycles required to execute a test pattern. If there are no differences in them among cores, we can easily deduce the width of test buses. It can be intuitively understood that the minimum width of test bus, which minimizes the pipelining time slot, $\max(CI_i, CE_i, CO_i)$, should be searched in order to minimize the test time in most cases. Nonetheless, $(I_i \bmod W)$ and $(O_i \bmod W)$ bits are wasted to input and output a test pattern, respectively. The wasteful use of test buses is shown in Figure 4.

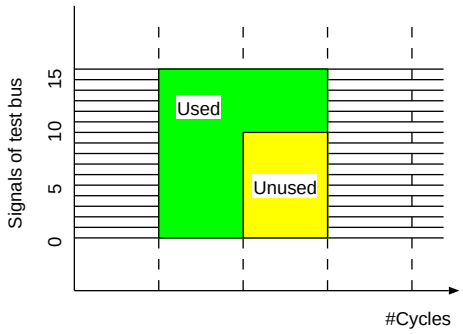

Figure 4. Wasteful use of test buses.

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 time. Our idea involves the reduction of the wasteful test bits and the parallelization of the execution of test patterns. Cores are grouped in order to achieve the reduction and parallelization. Cores in the same group simultaneously share external test resources to control input ports and observe output ports. Executions of the external test for the cores are simultaneously carried out. 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 time minimization with several cores simultaneously sharing test buses is achieved by identifying the optimum groups, test sets and width of test buses.

Now let us define the test time minimization problem for test bus architecture. In the CBET scheme, each core has several test sets. A test set used for core j, v_j, is formulated in Equation (2). An assignment of core j to group i is rep-

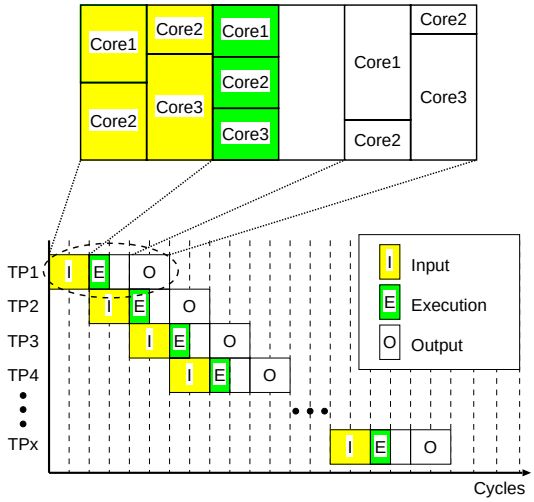

Figure 5. External test for a group by our idea.

resented 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 constraint, $\sum_{i=1}^{n} c_{ij} = 1$, which means that core i belongs to only a group. If no external test for core j is performed 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 (1). 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 the external test of group i, $V_{Ei}(\boldsymbol{v})$, is shown as follows.

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

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

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

Clock cycles for the BIST of group i, $B_i{}'$, is formulated as follows.

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

The time for the external test and the BIST for group i, $TE_i{}'$ and $TB_i{}'$, are shown as follows, respectively.

$$TE_i{}' = E_i{}' / F_E, \quad TB_i{}' = B_i{}' / F_B.$$

If there is dead time to test group i, test time of group i, $TC_i{}'$, is the sum of $TE_i{}'$ and $TB_i{}'$. Therefore, the total test time for the SOC with several cores sharing the test bus is shown as follows according to [7–9].

$$T' = \max \left\{ \sum_{i=1}^{n} TE_i{}', \max_{i=1}^{n} TC_i{}' \right\} \tag{4}$$

The test time minimization problem for test bus architecture is solved by determining the variables $\boldsymbol{a_j}$ for all cores, $\boldsymbol{c_i}$ for all groups, and W_1 and W_2 which minimize Equation (4). Optimization of test accesses is carried out by determining W which minimizes the test time and conforms to a constraint of the number of external pins by considering the trade-off between the test time and the number of external pins.

4. Computational Complexity and Algorithm

The 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 carried out within a shorter amount of computation time, linear to the number of cores at best, if we use the test time minimization algorithm introduced in [9]. *The 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. The computational complexity of *test access optimization* is expected to be proportional to the Bell number, $B_n = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k!}{k^n}$, where n is the number of cores. Tree pruning is useful in reducing the computation time required for test access optimization. Tree pruning is carried out to reduce the wasteful test bits on test buses. Several cores should be grouped and test time minimization for the groups should be performed if the usage rate at which the groups use valid test bits on test buses is higher than that for a temporal solution. The increase in the wasteful test bits on test buses lead to low efficient usage of test buses for the external test. An overview of an algorithm for test access optimization is shown in Algorithm 1. In an initial solution, each group has only one core. The number of groups to be searched should be reduced to less than B_n when a solution for the optimum test accesses cannot be derived within a practical computation time.

Algorithm 1 Test access optimization

```
Procedure Optimize(v)
Input: v = (v_1, v_2, ..., v_n)
Output: The optimal test sets and test bus architecture
for all widths of test buses do
  // for the trade-off between test time and # of external pins
  for all grouping assignments do
    // for reduction of the wasteful test bits on test buses
    Compute the number of wasteful test bits of test buses for
    all groups.
    if the wasteful test bits are reduced by grouping, compared
    with the temporal solution then
      Find the test sets which minimize test time for groups
      using the fast algorithm such as that in [9].
    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 scheme is effective to reduce test time. And it is also shown that we can design our SOCs under the trade-off between test time and the number of external pins. We used ten ISCAS'85 benchmark circuits [4] as cores for a virtual SOC for the experiments. The circuits are described in Table 1. CE_i is equal to 1 for all cores in the experiments. This does not lose the generality of the proposal because CE_i is usually 1 in practical test designs. Of course, the optimization will be well even if CE_i is another number.

Table 1. Characteristics of cores.

	C432	C499	C880	C1355	C1908	C2670	C3540	C5315	C6288	C7552
I_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
#Test sets	38	52	37	80	61	96	72	16	76	95

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 the experiments.

It is however easy to adopt multi-frequencies to be accommodated to cores. Test sets were generated for all cores on the basis of CBET scheme. The number of test sets for all cores are shown in Table 1. Test patterns for external test were generated by Synopsys Test Compiler. LFSRs were used for BIST pattern generation and primitive polynomials are used for the LFSRs. Multiple LFSRs were used for pattern generations in several cores because no primitive polynomial exists for the number of input ports of such cores. It was assumed that initial values of all LFSRs are all-1. All primitive polynomials are shown in Table 2.

To validate the proposals, five test methods are used to optimize test accesses. The five test methods are described in Table 3. 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 time or not. If not pipelined, only single 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. Primitive polynomials for LFSRs.

Width	Primitive polynomial
206	None. This is substituted by LFSRs whose widths are 127, 31, 19, 17, 7 and 5.
157	None. This is substituted by LFSRs whose widths are 127, 17, and 13.
127	$x^{127} + x + 1$
60	$x^{60} + x^{49} + 1$
50	$x^{50} + x^6 + x^3 + x^2 + 1$
41	$x^{41} + x^{21} + 1$
36	$x^{36} + x^{11} + 1$
33	$x^{33} + x^{13} + 1$
32	$x^{32} + x^8 + x^6 + x^5 + x^4 + x + 1$
31	$x^{31} + x^3 + 1$
19	$x^{19} + x^{17} + x^{16} + x^{14} + x^{13} + x^9 + x^7 + x^4 + 1$
17	$x^{17} + x^3 + 1$
13	$x^{13} + x^{11} + x^9 + x^8 + x^5 + x^4 + x^2 + x + 1$
7	$x^7 + x + 1$
5	$x^5 + x^2 + 1$

Table 3. 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	NO	YES
TEST-5	YES	YES	YES

Test application times derived by the four methods are shown in Figures 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*, *TEST-4* and *TEST-5* 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 51. The numbers of *TEST-3*, *TEST-4* and *TEST-5* are more than almost 44, 11 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*, *TEST-4* and *TEST-5* are 40.5%, 45.3%, 42.3% and 51.9%, respectively. Next, normalizing with test application time of *TEST-2*, reduction ratios by *TEST-3*, *TEST-4* and *TEST-5* are shown in Figure 8. The maximum reduction ratios by *TEST-3*,*TEST-4* and *TEST-5* are 12.0%, 21.1% and 30.5%, respectively. 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 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

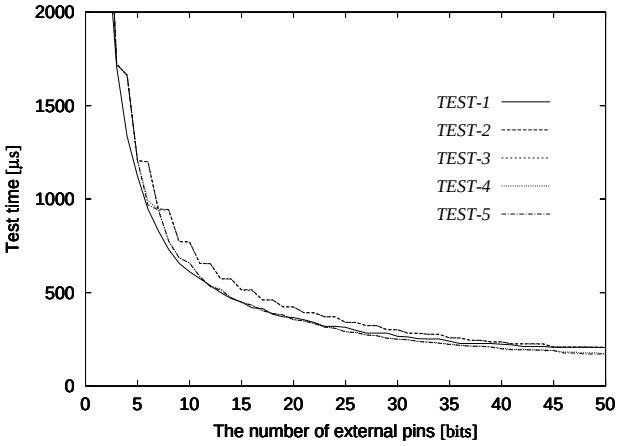

Figure 6. Test time in the small # of pins.

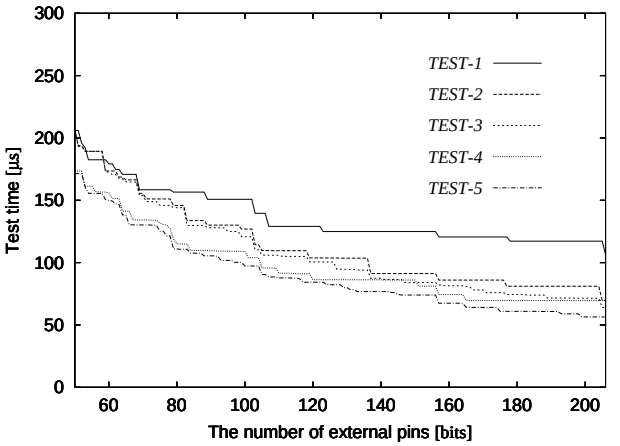

Figure 7. Test time in the large # of pins.

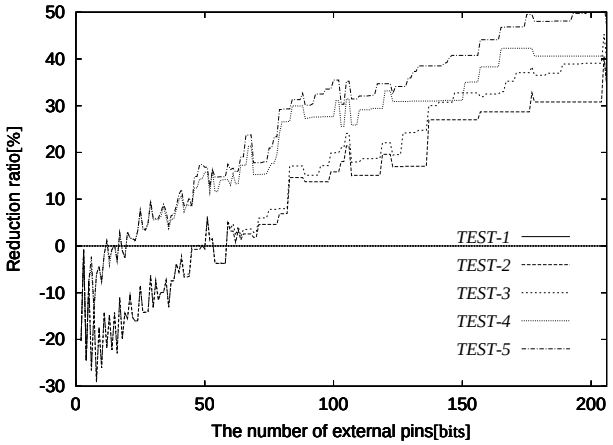

Figure 8. Reduction ratios of test time normalized with *TEST-1*.

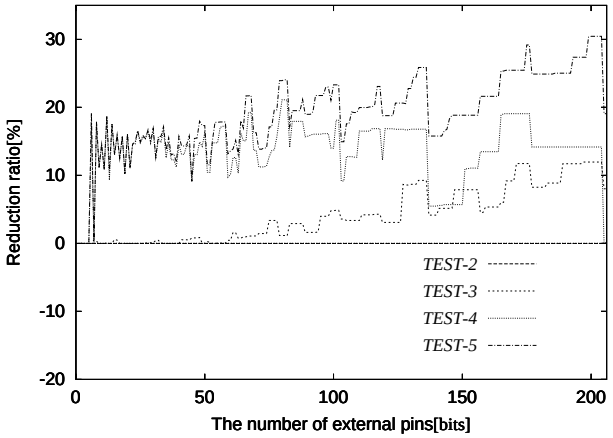

Figure 9. Reduction ratios of test time normalized with *TEST-2*.

external pins increases. At a glance, the groupage of cores seems to have a little impact on test time reduction. But it is indispensable for test bus optimization because only changing the widths of test buses cannot reduce the wasteful test bits fully .

Experimental results show that the proposals of grouping cores and optimizing test buses are effective to reduce test time of SOCs. It can be also understood that the we can design our SOCs considering the trade-off between test time and the number of external pins such as Figures 6 and 7. External pins for test in future SOC designs will be more than that in the present and so the proposals are effective for test time reduction of future SOCs.

6. Concluding Remarks

In this paper, the optimization of test accesses with the CBET scheme was proposed to minimize the test time by considering the trade-off between the test time and the number of external pins. Our ideas for test access optimization involve determining the optimum bandwidth of external I/O for the external test and determining the optimum groups each of which consists of cores which simultaneously share mechanisms for the external test. Test bus widths can be sought by conforming to the external pin count constraints of system designers. The test access optimization based on our ideas reduced the test time by 51.9% of the conventional method. It was experimentally conformed ed that test scheduling and optimization of the test mechanism are very effective for test time reduction. Thus, we propose a test access optimization method for use in the next SOC era in which system designers will be able to use more external pins for testing in their designs.

The optimization requires lengthy computation. In our experiments, the number of groups was between 6 and 10 so that the optimal solution might be obtained within a practical computation time. It is necessary that an efficient heuristic for the optimization is developed. The heuristic, which can search for stricter solutions within a shorter time, should be developed in the future. This is one of our planned future works.

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

Acknowledgment

The authors would like to thank the VDEC (VLSI Design and Education Center) for providing CAD tools. This research was supported by the Japan Society for the Promotion of Science under a Grant-in-Aid for Scientific Research B(2) (No.#11450143).

References

[1] Y. Zorian, “A distributed BIST control scheme for complex VLSI devices,” Proc. VLSI Test Symposium, pp.4–9, 1993.

[2] R. M. Chou, K. K. Saluja and V. D. Agrawal, “Scheduling tests for VLSI systems under power constraints,” IEEE Trans. on VLSI Systems, Vol. 5, No. 2, pp.175–185, 1997.

[3] V. Muresan, X. Wang, V. Muresan and M. Vladutiu, “A comparison of classical scheduling approaches in power-constrained block-test scheduling,” Proc. ITC, pp. 882-891, 2000.

[4] F. Brglez and H. Fujiwara, “A Neutral Netlist of 10 Combinational Benchmark Designs and a Special Translator in Fortran,” Proc. International Symposium on Circuits and Systems, pp.695–698, June 1985.

[5] K. Chakrabarty, “Test scheduling for core-based systems,” *Proc. of International Conference on Computer Aided Design (ICCAD)*, pp.391–394, November 1999.

[6] K. Chakrabarty, “Test scheduling for core-based systems using mixed-integer linear programming,” IEEE Trans. CAD, vol.19, no. 10, October 2000.

[7] M. Sugihara, “A test methodology for core-based LSIs (in Japanese),” Master’s thesis, Dept. Computer Science and Communication Engineering, Kyushu University, March 1998.

[8] M. Sugihara, H. Date and H. Yasuura, “A novel test methodology for core-based system LSIs and a testing time minimization problem,” *Proc. International Test Conference (ITC)*, pp.465–472, October 1998.

[9] M. Sugihara, H. Date and H. Yasuura, “Analysis and minimization of test time in a combined BIST and external test approach,” *Proc. Design, Automation and Test in Europe (DATE)*, pp.134–140, March 2000.