SOC 테스팅을 위한 효율적인 부분 분리 링
(An Efficient Partial Isolation Ring Technique for SOC Testing)

김 문 준 † 이 영 균 †† 김 석 윤 ††† 장 훈 ††††
(Moon-Joon Kim) (Young-Gyun Lee) (Seok-Yoon Kim) (Hoon Chang)

요 약 코어를 기초로 설계된 칩을 테스팅 하기 위해서는 각 코어로의 테스트 데이터 접근을 위해 완전 분리 링이 필요하다. 부분 분리 링은 코어를 둘러싸고 있는 분리 링의 크기를 줄이고 분리 링으로써의 역할을 모두 수행한다. 본 논문에서는 기존의 방법들보다 부분 분리 링의 크기를 더욱 작게 하고 해법을 찾는 데 걸리는 시간도 보다 줄일 수 있는 효율적인 방법을 제시한다. 이를 위해 테스트가능도에 기반을 둔 효율적인 순위 결정 기법을 적용하였으며 SOC 테스팅의 특성을 살린 정렬 기법을 적용하였다. 실험 결과에서 기존의 방법들보다 본 논문에 제시된 방법이 실제로 유용하다는 것을 보여준다.

Abstract Testing a core-based designed chip requires a full isolation ring to provide for core test data access to each core. A partial isolation ring replaces the full isolation ring reducing total isolation ring size surrounding cores. This paper proposes an efficient method to reduce the size of the partial isolation ring and shorten the time to acquire the final solution. For this, a reasonable ordering technique according to testability is introduced and a sorting technique is adopted to reduce the total solution time. Experimental results show that the proposed method can be useful in practice.

1. Introduction

Recently a SOC(System On a Chip) design technique has been popularly adopted to integrate chips with pre-designed cores. There were many research papers on the embedded core testing[1, 2,3,4]. One simple approach for testing embedded cores is to use multiplexing to make the inputs and outputs of the core accessible to the chip pins[5]. However, this approach does not help with testing the logic surrounding the core. Then another approach applying the test sets into the cores by adapting a full isolation ring technique[6] is appeared. Figure 1 shows the full isolation ring technique. UDL(User Defined Logic) is a logic that integrators(User) would put between cores. A full isolation ring surrounding a core provides an accessibility from PIs(Primary Inputs) to the core[7,8].

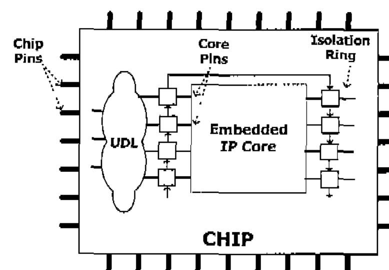


Fig. 1 Architecture of Full Isolation Ring

The full isolation ring, also called "Wrapper" in the P1500 standardizing working group[9], is very similar to IEEE 1149.1 boundary scan cells. But

· 본 연구에 사용된 H/W 및 S/W는 부분적으로 IDEC의 지원에 의한 것임.
† 학생회원 : 숭실대학교 컴퓨터학과
mjkim@watt.ssu.ac.kr
†† 비 회 원 : 숭실대학교 컴퓨터학과
yklee@watt.ssu.ac.kr
††† 정 회 원 : 숭실대학교 컴퓨터학과 교수
ksy@comp.ssu.ac.kr
†††† 비 회 원 : 숭실대학교 컴퓨터학과 교수
hoon@comp.ssu.ac.kr
논문접수 : 2001년 2월 16일
심사완료 : 2001년 7월 27일

applying the full isolation ring technique into high-performance chips is difficult to be accepted in practice because there is so much area and performance overhead. To address this problem, a partial isolation ring technique has been proposed. Four approaches are introduced in[10]. They are Hill-Climbing, Clique Hill-Climbing, Clique Greedy, and Branch and Bound. Among them the first and second are reasonable to be applied in practice. If n represents the number of the inputs of a core, the time complexity of Hill-Climbing method is only $O(n)$. It is simple to implement, but not clever since it could not lead to an optimal result. The Clique Greedy is a method of $O(n^2)$ complexity. Despite of its big time complexity, this method is not exhaustive, either. So it is hardly useful in cases where n is large.

The Hill-Climbing method could have several different partial isolation ring solutions according to the selecting order of the candidate cells. Fig. 2 shows such a situation. When the selecting order of the scan cells is determined like the left picture in Fig. 2(a), only 1 scan cell could be eliminated. On the other hand if the order is determined as shown in Fig. 2(b), 3 scan cells could be removed.

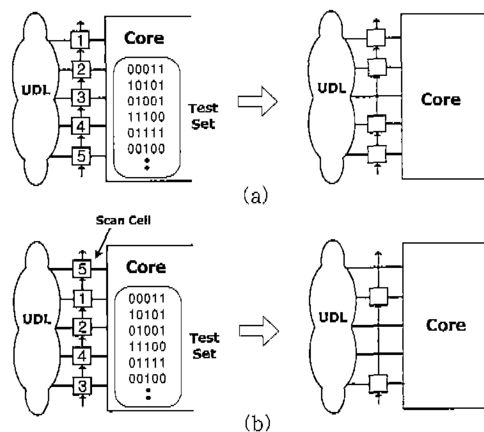


Fig. 2 Different F/F Results Depending on Selecting Order

This paper proposes another sequential method resulting in smaller size of the partial isolation ring solution. It can be accomplished by employing a kind of weighing technique with a testability analysis of UDL. A heuristic is used to reduce the total execution time although the time complexity is still $O(n)$.

In Section 2, a process for the partial isolation ring solution is explained. The proposed sequential method is introduced and explained in Section 3. Section 4 shows and compares the experimental results, and Section 5 concludes this paper.

2. Process for Partial Isolation Ring Solution

2.1 Selecting Procedure with Random Order

This section describes an overview of the conventional process extracting the partial isolation ring solution. Test sets given from a core vendor satisfy a certain fault coverage. The rationale of partial isolation ring solution is that it has to yield the same fault coverage though certain scan cells are removed. To verify that this condition holds true, it has to be shown that all test vectors in the test set are to be justified through the UDL placed before the core. The procedure of the sequential method is as follows.

1. Eliminate an arbitrary scan cell on the unexamined scan cell list.
2. Examine all the test vectors whether they can be justified through the UDL.
3. Keep the scan cell eliminated if every test vector has been justified through the UDL
4. Bring and place the eliminated scan cell back if any test vector has not been justified.
5. Go back to 1 if unexamined scan cells is left.
6. If all the scan cells are examined, the left over cells become the partial isolation ring solution.

This approach examines each cell sequentially and checks whether all the vectors in the test set can be justified through the UDL. The time complexity of this method is $O(n)$ where n represents the number of core inputs. In this procedure the priority among the cells has not been considered and the results can be varied every time if the order determining which cell to be eliminated is changed.

2.2 Checking Procedure to Justify Test Vectors by ATPG

To test an embedded core with a partial isolation ring inserted, a part of a test vector is to be inserted and shifted into the partial isolation ring and the other part of the vector is applied to the UDL to generate the rest of the inputs to the isolation ring. The UDL input sequences are obtained from the process checking the test vectors of the test set. The checking process can be implemented in various ways.

In[10], an ATPG(Automatic Test Pattern Generation) technique is used to check the given test vectors to be justified through the UDL. Every vector has to go through this ATPG checking process. Every time a scan cell is tested, all the test vectors have to be justified through the UDL since all the test vectors have to be applied in the real testing. The following is the flow to implement this checking process adopting an ATPG process.

1. Identify which core inputs do not have a scan cell.
2. Fetch a vector from the test set and confirm which bit positions do not have scan cells.
3. Identify the values of the non scan cell positions.
4. If 0 is the value of the vector on that position, place an inverter on that core input.
   (Nothing is required to core inputs to be justified to 1)
5. Bind all the core inputs which have no scan cell to an AND gate.
6. Execute the ATPG process on the output line of the AND gate with a s-a-0 fault.
7. If a test pattern is generated for the fault, the test vector is justified through the UDL.
8. If not, the corresponding cell can not be eliminated.

For example, if the set of scan cells left is {0, 2, 5} as shown in Fig. 3 and a test vector to be justified is 101101, then the circuit needs to be modified as shown in Fig. 4. ATPG is carried out on a s-a-0 fault at the output of the AND gate. The result from here is the vector for the UDL to generate 010 on {1, 3, 4}. Now we know the vector can be justified without 3 scan cells.

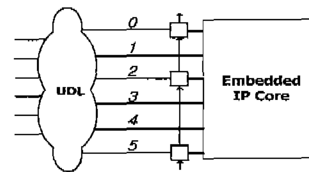


Fig. 3 Partial Isolation Ring with Some Cells Removed

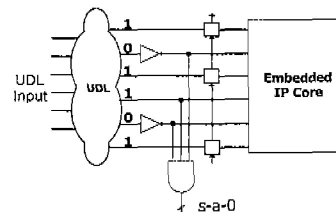


Fig. 4 Modification for ATPG to Figure Out Whether a Test Vector can be Justified

3. An Efficient Method for Partial Isolation Ring Solution

3.1 Reasonable Selecting Order Based On Testability

This section proposes an efficient sequential algorithm. When using a sequential algorithm, the order determined which one is to be tested ahead of others is very important. This is because a certain wire, whose scan cell is eliminated earlier, might prevent another wire from being set a desired value whose scan cell is attempted to be tested next. A desired fault coverage is supposed to be satisfied when the optimal solution is achieved.

In this paper, to remove scan cells as many as possible, testability of UDLs is calculated to determine the order of them. Testability consists of controllability and observability. But only the controllability is required for the proposed method. The wires which need to be analyzed are the input lines of the core. They are output lines of the UDL at the same time. So the observability is not needed since they are treated as POs of the UDL. If the controllability analysis procedure is

completed, the selecting order can be determined as follows. The scan cell which is corresponding to the easily-controllable wire has a higher priority than the harder ones. This is because the previously tried scan cell would be removed easily. But if the scan cells of harder-controllable wires are eliminated in advance, the scan cells of easily-controllable wires would be hardly removed. It means that if the scan cell of the hardest controllable wire is kept removed, the probability that many other scan cells of easily-controllable wires could be removed becomes very low. Because the harder-controllable wire disturbs the other wires from being set of a desired value.

Controllability consists of 0-controllability and 1-controllability. The probability that both 0 and 1 appear on a certain bit position is 0.5 in fact. It would not be acceptable for investigating all the test set to count how many times 0 or 1 on that bit position appears to adopt the proper controllability. Even if the probabilities of all vectors are decided, not all core inputs are required to justify given test set. It would be waste of time to count them and even a mistake putting wrong priorities on the core inputs. Therefore it is the reasonable method to calculate priorities by the combined value of 0 and 1 controllability and decide the sequence in the ascending order of controllability in order to check the easily-controllable wires ahead.

3.2 Controllability Calculation Procedure

0(or 1)-controllability means the degree how difficult to feed 0(or 1) on that wire. In this paper, a heuristic is used to calculate the controllability as in[11]. The algorithm is basically based on SCOAP [12]. Depth influences on the controllability and the value is shown in Table 1.

Table 1 Depth Value of Cell Type

Number of Inputs	Depth
1 or less	0
2 or more	1

The first column of Table 1 represents the number of input lines of a cell. If the value of a input line propagates to the output of the cell without any interference of the other inputs, that belongs to the third row. The way to calculate controllability is as follows.

1. Identify how many input lines are required to set the output line of the gate.
2. Add the corresponding controllabilities to the sum.
3. Add the depth and fanout number of the gate to the sum.
4. The sum is the value of controllability of that node.

If only one input line is required to feed a certain value on the output line of the gate, only the value of controllability of the most easiest line is required(e.g. 0 at output line of AND gate). In case that every input line is necessary for setting a certain value on the output line, all the values of controllability of those wires are to be added. A simple example shows the controllability calculation procedure in Table 2 for a sample circuit shown in Fig. 5.

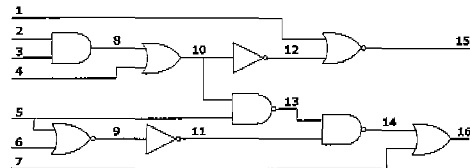


Fig. 5 Example Circuit for Controllability Calculation

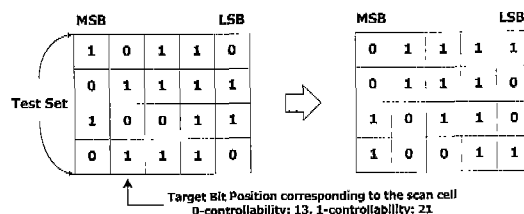
Table 2 Controllability Calculation Result

Node	0-Controllability	1-Controllability
1	0	0
2	0	0
3	0	0
4	0	0
5	1	1
6	0	0
7	0	0
8	1	1
9	1	2
10	3	2
11	2	1
12	2	3
13	4	2
14	4	3
15	1	3
16	5	1

3.3 Vector Sorting Procedure

Typical sequential algorithm for the partial isolation ring solution has $O(n)$ complexity. However the total execution time varies in reality because a counter vector can appear early or late occasionally if no particular action is taken. The counter vector is a vector that determines the scan cell fixed to remain. If the counter vector appears late, the time spent for justifying previous vectors becomes useless. Hence the earlier the counter vector appears, the total execution time can be reduced because no scan cell can be removed unless all the vectors pass the ATPG process. Therefore sorting the test vectors in descending order of the controllability helps the ATPG process complete earlier and total execution time shorten.

The test set has hundreds of or thousands of test vectors. The execution time of ATPG for each vector is not quite short. The point here is that if any vector is not justified, the cell trying to eliminate can not be removed. Therefore considering the counter vector first will be better to shorten the total execution time as a result. Whether targeted cell remains or not can be known early if the counter vector is tested ahead of other vectors. Hence unnecessary time to test a number of remained vectors can be saved giving the chance to the next cell to be tested earlier at the same time. On the contrary, if the counter vector is at the end of the list of the test set, the whole time consumed to test the vectors is just a waste of time. To prevent this, the priorities are to be assigned to vectors of the test set to sort the test set. The priorities depend on the controllability calculated before. The test set is sorted just once at every time a new cell is selected. The sorted test set must rank the counter vector at the top of the list. To do this, it has to be found which controllability of that wire, whose scan cell is targeted this time, is measured bigger, 0 or 1. As an example, if 0-controllability of a wire is 12 and 1-controllability is 15, the latter is selected for that wire as a weight to sort the test set. Then sort the whole test set to put the vectors ahead which have 1 at that bit position. Fig. 6 shows the sorting process.



(a) Before Sorting Process (b) After Sorting Process Target Bit Position

Fig. 6 Sorting Process for Test Set to Reduce Total Execution Time

It is enough to perform the sorting process just once at a time when checking a new scan cell. The step is as follows.

1. When a cell is started to be checked, perform the sorting process on that bit position.
2. Perform the ATPG process.
3. Save and maintain that order of test set if the cell successes to be eliminated.
   Or, recover the previous order.
4. Move to the next target cell.

Sorting time depends on the number of vectors in test set. That is treated as a constant when considering the time complexity. Therefore the time complexity of the proposed method is still $O(n)$.

4. Experimental Results

ISCAS85 bench marking circuit is used to experiment, and they act as UDLs. The core is a black box, so there is no need to prepare a circuit for it. The test vectors for core testing is given from the vendor. We don't know a thing about the core, so in this paper the test set is made by random function of *C* language. Important thing is not what the core is, but what the test vectors are. The purpose is eliminating scan cells as many as possible. We extract experimental results from 3 methods, shown in Table 3. They are the random selecting method, the proposed controllability considered method, and the opposite of proposed

method to show the worst case. Random means that the order scan cell is selected is determined by the random function. It represents the Hill Climbing method. Table 3 shows the result of the 3 methods for the partial isolation ring solution. The partial isolation ring size smaller is the better. Note that the ring size in table 3 is the partial isolation ring size at the inputs of a core. The partial isolation ring size at the outputs of a core is not considered in this paper.

Table 3 Number of F/Fs in Partial Isolation Rings

UDL Name	# of Gates	# of Vectors	Full Ring Size	Partial Isolation Ring Size		
				worst Case	Random (Hill Climbing)	controllability conlsidered Method
c432	160	500	7	1	2	0
c499	202	1000	32	25	24	24
c880	383	1000	26	13	11	7
c1355	546	1000	32	25	25	24
c1908	880	1000	25	19	13	5
c3540	1669	1000	22	18	16	13
c5315	2307	1000	123	94	81	63
c6288	2416	1000	32	26	26	26
c7552	3512	1000	108	102	80	46

Table 4 shows the enhancement degree of total processing time at each circuit. The unit of time is a second. The proposed method in this paper, giving priorities on selecting the order of scan cells, is used to extract both the results. There is no vertical relationship in table 4. The second column means how many vectors are dealt with for ATPG process. They are reduced because a counter vector appears at the higher rank in the test set by sorting based on controllability. The number of considered vectors during ATPG is reduced by about 34% in average. Moreover, the total processing time is also reduced because of the smaller ATPG job amount than unsorted. The actual time for sorting is nearly nothing as shown in the table. In Table 4, the minimum number of vectors justified through ATPG is $n-1$, where n is the number of the core inputs. The maximum is $\sum_{k=2}^{n} 2^k$. The maximum value is achieved only when all the F/Fs of the isolation ring can be eliminated.

Table 4 Number of Considered Vectors during ATPG & Total Processing Time

UDL Name	Number of Vectors		Improvement	Total Processing Time		Improvement
	Unsorted	Sorted		Unsorted	Sorted	
c499	9509	4144	57%	97	40	59%
c880	21305	15663	26%	207	166	20%
c1355	4354	2900	33%	49	32	35%
c1908	13099	8926	32%	152	108	29%
c3540	1969	1376	30%	27	20	26%
c5315	38828	29678	24%	643	517	20%
c6288	2560	1321	48%	44	21	52%
c7552	39070	30194	23%	799	631	21%

5. Conclusions

This paper described an efficient sequential method for the partial isolation ring solution. It guarantees the same fault coverage for the given test set relieving the area overhead of the partial isolation ring solution by determining the selecting order based on controllability of UDL. Using a property of the SOC testing, we sorted the test vectors based on the controllability calculated, and thus we reduced the total processing time. The time complexity of the proposed method is $O(n)$. Experimental results indicate a significant reduction in the number of F/Fs compared with the previous method.

References

[1] Lee Whetsel, *"Addressable Test Ports An Approach to Testing Embedded Cores," Proc. International Test Conference*, pp. 1055-1064, 1999.

[2] Erik Jan Marinissen, Yervant Zorian, Rohit Kapur, Tony Taylor, Lee Whetsel, *"Towards a Standard for Embedded Core Test: An Example," Proc. International Test Conference*, pp. 616-627, 1999

[3] Peter Harrod, *"Testing Reusable IP - A Case Study," Proc. International Test Conference*, pp. 493-498, 1999.

[4] Rochit Rajsuman, *"Testing a System-On-a-Chip with Embedded Microprocessor," Proc. International Test Conference*, pp. 499-508, 1999.

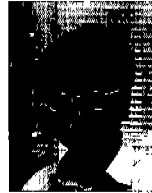
[5] Immaneni, V., and S.Raman, *"Direct Access Test Scheme Design of Block and Core Cells for Embedded ASICS," in Proc. International Test Conference*, pp. 488-492, 1990.
[6] Yervant Zorian, Erik Jan Marinissen, and Sujit Dey, *"Testing Embedded-Core Based System Chips," Proc. International Test Conference*, pp. 130-243, 1998.
[7] E. J. Marinissen, R, Arendsen, G. Bos, H. Dingemanse, M. Lousberg, and C. Wouters, *"A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores," in Proc. International Test Conference*, pp. 284-293, 1998.
[8] *"Preliminary Outline of the IEEE P1500 Scalable Architecture for Testing Embedded Cores," Proc. VLSI Test Symposium*, 1999.
[9] IEEE P1500 General Working Group website, *"IEEE P1500 Standards For Embedded Core Test,"* http://grouper.ieee.org/groups/1500/.
[10] Nur A. Touba and Bahram Pouya, *"Testing Embedded Cores Using Partial Isolation Rings." IEEE Design & Test of Computers*, Vol. 14, No. 4, pp. 52-59, 1997.
[11] Jongwook Park, Sang-Hoon Shin, Sungju Park, *"A Partial Scan Design by Unifying Structural Analysis and Testabilities," Journal of KISS(A): Computer Systems and Theory*, Vol. 26, No. 9, pp 1177-1184, September. 1999.
[12] L. H. Goldstein, *"Controllability/Observability Analysis of Digital Circuits," IEEE Trans. on Circuits and Systems*, Vol. 26, pp. 685-693, 1979.

김 문 준

2000년 숭실대학교 컴퓨터학부 졸업(B.S). 2000년 ~ 현재 숭실대학교 대학원 컴퓨터학과 석사과정. 관심분야는 컴퓨터구조, VLSI 설계 및 테스팅, CAD

이 영 균

1999년 숭실대학교 컴퓨터학부 졸업(B.S). 1999년 ~ 현재 숭실대학교 대학원 컴퓨터학과 석사과정. 관심분야는 컴퓨터구조, VLSI 설계 및 테스팅, CAD



김 석 윤

1980년 서울대 공대 전기공학과 학사. 1990년 University of Texas at Austin 전기, 컴퓨터공학과 석사. 1993년 University of Texas at Austin 전기, 컴퓨터공학과 박사. 1982년 ~ 1987년 한국 전자통신 연구소 연구원. 1993년 ~ 1995년 Motorola Inc., Senior Staff Engineer. 1995년 ~ 현재 숭실대학교 컴퓨터학부 교수. 주관심 분야는 설계 자동화, VLSI 회로해석 및 설계, 통신 시스템

장 훈

정보과학회논문지 : 시스템 및 이론
제 28 권 제 6 호 참조