논문 2004-41SD-10-2

테스팅 및 저전력을 고려한 상태할당 기술 개발

(A New State Assignment Technique for Testing and Low Power)

조 상 욱*, 박 성 주**

(Sangwook Cho and Sungju Park)

요 약

유한상태기의 상태할당은 이로부터 구현되는 순차회로의 속도, 면적, 테스팅 및 소비전력에 큰 영향을 미친다. 본 논문에서는 상태변수 그룹들 사이에 상호 의존성(dependency)을 최소화하여 테스팅 및 전력소모를 개선하기 위한 m-블록 분할을 이용한 새로운 상태할당 기술을 소개한다. 제안된 알고리듬은 상태들을 그룹으로 나누어 상태변수의 상호의존성을 줄이고, 상태천이 확률에 의해 결정된 무게인자에 따라 상태 간 상태변수의 변화를 최소로하는 코드를 할당하여 상태 천이시 스위칭 횟수를 줄인다. 즉 피드백 순환의 길이와 수는 상태 변수들 간에 최소 전환 활동으로 감소됩니다. 벤치마크 회로에 대한 실험결과는 테스팅 및 소비전력이 현저히 개선되었음을 확인하였다.

Abstract

The state assignment for a finite state machine greatly affects the delay, area, power dissipation, and testabilities of the sequential circuits. In order to improve the testabilities and power consumption, a new state assignment technique based on m-block partition is introduced in this paper. The algorithm minimizes the dependencies between groups of state variables are minimized and reduces switching activity by grouping the states depending on the state transition probability. In the sequel the length and number of feedback cycles are reduced with minimal switching activity on the state variables. Experiment shows significant improvement in testabilities and power dissipation for benchmark circuits

Keywords: scan design, low power design, state assignment, m-block partition, partition pairs

Ⅰ. Introduction

As the density of the SoC(System on a Chip) becomes extensively high, the testing and power consumption are of great concerns for various applica-tions. In order to alleviate the expense of the SoC design especially for the testability and low power consumption, the optimization has to be considered at the very early stage of the design such as logic synthesis level. A number of state encoding techni-ques have been developed for testable design [1-5]. Instead of analyzing only the gate level circuit infor-mation, implicit techniques for FSM (Finite State Machine) traversal is used to identify non controllable state registers to be included in partial scan flip-flops [1]. State bi-partitioning technique is introduced to minimize the dependencies among state variables and thus hopefully to reduce the number of partial scan flip-flops [4]. An m-block state partitioning technique, which is more general than bi-partitioning technique, has been developed to maximize the testabilities and reduce the area overhead [6,7]. A few state encoding techniques have been addressed to minimize the power consumption [8-11]. Testability was considered in re-encoding the states for the low power [10],

* 한양대학교 컴퓨터공학과
(Dept. of Computer Science & Engineering, Hanyang Univ.)
** 한양대학교 전자컴퓨터공학부
(Dept. of Electrical Engineering Computer Science, Hanyang Univ.)
※ 본 논문은 한국과학재단 특정기초과제 (R01-2003-000-10150-0) 로부터 지원을 받아 진행하였습니다.
접수일자: 2003년7월28일, 수정완료일: 2004년10월4일

however no paper has tried to optimize the testability and low power simultaneously at the state assign -ment stage, although it is inherently contradictory problem. In this paper we introduce m-block state partitioning technique, which is more general and efficient than bi-partitioning technique in [4], to maximize the testabilities and minimize the power consumption. State transition probability is exten -sively adopted in partitioning the states to minimize the dependencies of the state variables.

This paper is organized as follows: After introdu -cing the state encoding for testability technique in section 2, state encoding for low switching transitions is described in section 3. Our new state encoding techniques aimed to improving both testability and power consumption is presented in Section 4, follow -ed by experimental results and conclusions.

II.State Encoding for Testability Technique

One of the important issues in the logic synthesis for sequential circuits is to assign binary code values to each state of a state table extracted from a state diagram. In general, the binary code must be assig -ned such that the functional dependencies of state and output variables are minimized and the circuits are constructed with least number of memory elemen -ts. However cases for state assignments grows exponentially with regard to number of states n, hence state encoding algorithm is a complicated NP -complete problem.

At first we will apply the conventional random state encoding algorithm and check the dependencies among state variables. The state table shown in figure 1 can be synthesized to the figure 2 circuit by taking the state assignment α of the figure 1(b). It is noted that three flip-flops Y1(y1), Y2(y2), Y3(y3) (Y: next state, y: current state) in figure 2 have complete dependencies among themselves. Existence of simple cycles in the scan graph was further investigated in [6] and there are two simple cycles in figure 2.

On the other hand, if the circuit is implemented as

	NS				Assignment α , β					
PS\x_1x_2	00	01	11	10	y_1	y_2	y_3	Y_1	Y_2	Y_3
a	e	c	d	e	0	0	0	0	0	0
b	g	a	b	g	1	0	0	1	1	0
c	a	c	h	e	0	1	0	0	1	0
d	c	a	f	g	0	1	1	1	0	0
e	e	c	d	e	1	1	1	0	0	1
f	g	a	b	g	1	0	1	1	1	1
g	a	c	h	e	1	1	0	0	1	1
h	c	a	f	g	0	0	1	1	0	1

(a) (b)

그림 1. 상태천이표 및 상태할당
Fig. 1. State Transition Table and Assignment.

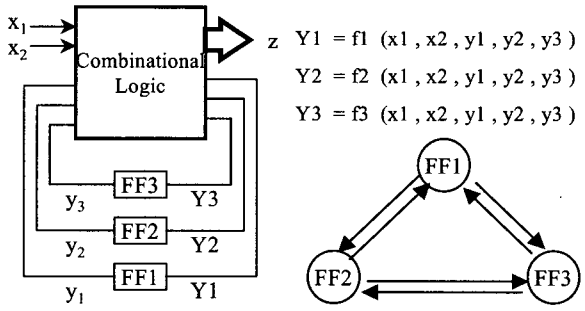

그림 2. 상태할당α에 의한 회로구조 및 스캔 그래프
Fig. 2. Circuit structure and Scan Graph by State Assignment α.

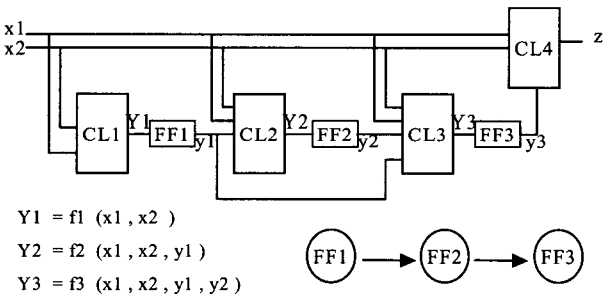

그림 3. 상태할당β에 의한 회로구조 및 스캔 그래프
Fig. 3. Circuit structure and Scan Graph by State Assignment β.

figure 3 using the different state assignment β of the figure 1(b), the dependencies among three state variables Y1(y1), Y2(y2), Y3(y3) are unidirectional, and it is more efficient for partial scan design than the previous state assignment. Since no simple cycle exist in figure 3 circuit, we may not need any scan flip-flop for testing. It can be further notified that the bit changes between states in state assignment β is 6 bits less than the assignment α. Therefore we can conclude that the assignment β is better than α both in area and testability senses.

A few terms are defined as followings for detailed description of optimal state assignment for high testability and low power consumption.

Definition 1] A partition consists of blocks such that there is no common state symbol among the subsets and the union of all subsets constitutes the state set S of a FSM.

Definition 2] Partition pair (p1,p2) is an ordered pair of partitions p1 and p2 in which any state in a block of the partition p1 is transited to the same block of the partition p2.

Definition 3] Closed partition: A partition P on the set of states of a sequential machine is said to be closed if, for every two states s1 and s2 which are in the same block of P and any input I, the next states for s1 and s2 are in common block of P.

Definition 4] $\pi(0)$: every block in the product contains only a single state.

Definition 5] m-partition: the smallest partition containing all the successors of the blocks of predecessor partition.

Definition 6] M-partition: the largest partition the successors whose blocks are contained in the blocks of m-partition.

Definition 7] Mm pairs: an ordered pair of partitions such that, if states s1 and s2 are in the same block of M-partition, then for every input I, the next states for s1 and s2 are in the same block of m-partition.

Let us consider two partitions p′ = (a, d : b : c, e : f) and p″ = (a, e : b, d, c, f) from the figure 4 state transition diagram. The p′ and p″ are 4-block and 2-block partitions respectively, and an ordered pair (p′,p″) is a partitioning pair. The relation between partition and state assignment can be explained through the figure 4.a. From the state assignment, each state variable generate a 2-block partition, that is, y1 variable produces p1 = (a, b, d : c, e, f) partition, y2 produces p2 = (a, c, d, e : b, f), and y3 produces p3 = (a, e : b, d, c, f). Furthermore more than one variables can generate m-block partitions such as y1 and y2 produce p12=(a, d : b : c, e : f) 4-block partition. Hence state assignment and block partitioning can be considered as a similar problem, and possibly the dependencies among memory elements of a sequential circuit may be estimated from the block partition. In figure 4, it can be seen that next state variable Y3 depends on the current state variables y1 and y2, and two memory element pairs of (FF3, FF2) and (FF3, FF1) do not include any feedback loop between flip-flops within each pair. By assigning state codes through the block partitioning, not only the great reduction in area is achievable but also the plagued test generation problem for sequential circuits can be drastically simplified. In this paper, we adopt generalized m-block partitioning method in assigning state codes and develop an efficient heuristic to minimize the dependencies among state variables. Although 2-block partition can consider dependencies among single state variables only, our approach takes into account the dependencies among set of state variables with m

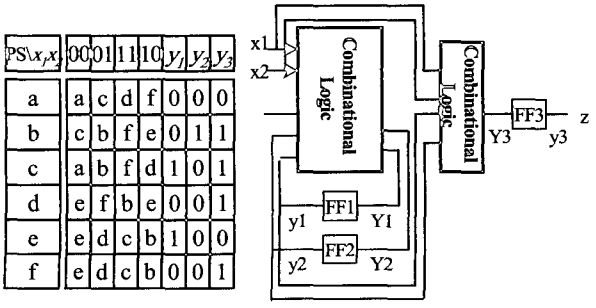

그림 4. 분할쌍에 의한 회로 구조
Fig. 4. State Assignment and Circuit Structure.

Step 1. k= $\log_n$ state variables for n state encoding
Step 2. While (states > 2) {
  Find M and m from the state table
  IF (M, m = ((partition pairs & Closed partition)) {
    Find Mm pairs
    Select $\pi(0)$ among Mm pairs
    Select yr($1 \le r < k$) state variables from the M partition
    Assign state value to each state variable
    Select yk-r state variables from the m partition
    Assign state value to each state variable }
  }
Step 3. Repeat ***Step 2*** until the partitions are partition *pair relations*

그림 5. 상태할당 알고리듬
Fig. 5. State Assignment Algorithm.

-block partition, thus the complex feedback cycles are greatly simplified.

In other word, by considering m-block partition our approach can identify the dependencies among set of state variables more globally, and thereby find an optimal state assignment for partial scan. In [4], non-controllability of flip-flops is evaluated by a systematic analysis of the state transitions and the encoding of the underlying FSM. In contrast to the method in [4], which is for selecting an optimal set of flip-flops for partial scan at gate-level, our method considers m-block partition to find an optimal state encoding, which could keep the number of non-controllable flip-flops minimal during the state assignment. The state assignment algorithm proposed in this paper for the minimal mutual dependencies among memory elements can be summarized as the figure 5.

III. State Encoding for Low Switching Transitions

In order to reduce the power consumption, it is very crucial to assign the states so that the transi -tions among state variables occur least frequently. After mathematically summarizing the power consumption model, state assignment technique for low power is described in this section.

(1) power consumption model

In general, the average power consumption can be modeled as follows[11]:

$$P_{ave} = \frac{1}{2} V_{dd} \bullet f \sum_{i=1}^{n_g} C_i P_i \qquad (1)$$

P_{ave} : average power consumption

f : operating frequency

V_{dd} : power supply voltage

C_J : load capacitance

n_g : the number of gates

P_i : switching probability

P_i(switching probability) in equation (1) is the key requirement for reducing power consumption. The above V_{dd}, f, and C_i parameters are constantly given by a design process and in this paper the focus is given to reduce switching count.

(2) Probabilistic Approach using Transition probability

First of all, the transition relations according to inputs need to be expressed as transition probabili -ties. The Markov Chain used as a probability model has a relation to dynamic variations in sequential characteristics. Given the information on a system and probabilities for inputs, the transition probabilities for state transition diagram can be calculated. The transition condition from the current state to the next state shall be decided only by the currents state. Figure 6 is a state transition diagram based on a state transition table. The conditional transition probability(CTP) from the current state to the next state is defined as:

$$P_{i,j} = P(Ns_j | Ns_i) \qquad (2)$$

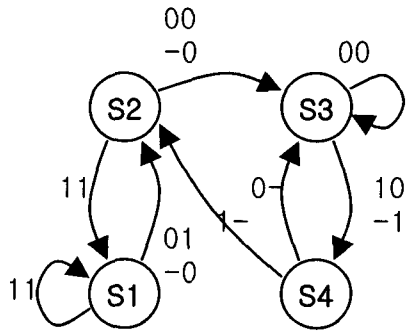

그림 6. 상태천이도(STD)
Fig. 6. State Transition Diagram.

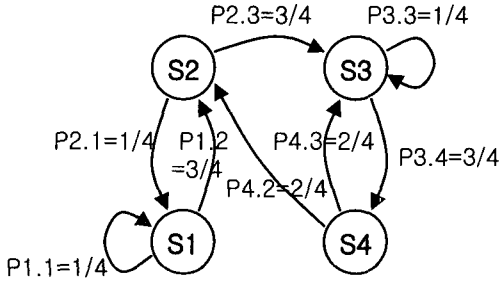

그림 7. 천이 조건 확률로 표시된 상태천이도
Fig. 7. Conditional Transition Probability of STD.

For example, the CTP from s2 to s3 is 3/4 since the state transition occurs on 01, 00, 10 of 4 possible inputs. The CTPs among all the states are shown in Figure 7. However the CTP itself, which only considers the input values, is not enough to show switching variations precisely. To extract more accurate switching variations, the equation is augmented as follows considering current state probability.

$$Gp_{i,j} = p_{i,j} \circ Ps_i \tag{3}$$

Here, Ps_i is the current state probability of Si and can be calculated with Markov chain characteristics. The conditional transition probability $p_{i,j}$ is expre -ssed as a matrix B, which can be calculated with the following equations.

$$P = B^T \circ P \tag{4}$$

$$\sum_{i=0}^{Sn} Ps_i = 1 \tag{5}$$

Figure 8 shows the calculation procedure of the current state probabilities and all of the transition probabilities. The power consumption can be reduced by assigning the states so that the variation of the state values are minimized among the states with high residency and transitions[12]. The highest transi -tion probability from the current state si to the next state sj implies that the switching variation between si and sj occurs most frequently. Figure 9 is a state transition diagram, considering weight transition probability and normalizing the Figure 8 as integers. Figure 9 expresses each transition between two states by one integer and rules out self-transitions.

To reduce the power, the remaining job is only to assign the states to keep the rate of the flip-flop switching very low looking at the weighted graph of the Figure 9. In this example the highest transition occurs between s3 and s4. Therefore the s3 and s4 states must be encoded to minimize the Hamming distance among them. Next consideration must go to the states pair (s2, s3) and (s2,s4) of which the transition is equally 9.

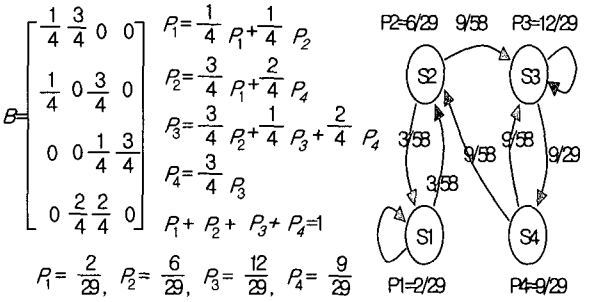

그림 8. 전체 천이 확률로 표시된 상태천이도
Fig. 8. Global Transition Probability of STD.

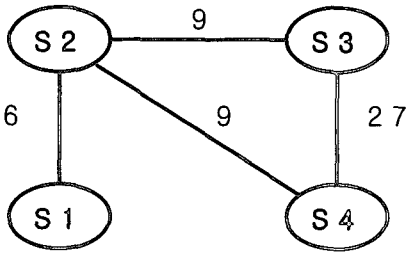

그림 9. 무게인자(weight)로 표시된 상태천이도
Fig. 9. weight transition probability of STD.

IV. New State Encoding Technique for Testability and Low Power

This section introduces a new state assignment technique that reduces the power dissipation consi -dering state transition probability as well as partitioning the states into m-block to minimize the dependencies among memory elements. There can be many partition pairs satisfying $\pi(0)$ condition which is necessary condition for each state to be uniquely assigned. Among the m-block partitions, the more the blocks containing states with high weight transition probability, globally the switching occurs less frequently, hence the power consumption can be highly reduced.

Observation : In partitioning the states into m-blocks, the power can be saved by taking the states with high transition probability into the same block.

Justification : Since the states within the same block are likely to get more same bits than the states in different blocks, the flip-flop transitions can be

reduced while the dependencies are kept low. There -fore this m-block partition considering state transi -tion probability can improve both the testability and power consumption.

The state assignment for m-block partition pairs satisfying the condition for assigning unique value for each state is carried out by 2 stages. At the first stage, the upper bits of states belong to the same block of current states are assigned by the same value and in the same way the lower bits of states belong to the same block of next states are assigned by the same value. At the second stage, state values are partly reassigned so that Hamming distance among states with the highest weight transition probability in state transition diagram can be the minimum. In the state which keeps low dependencies by m-block partition, selecting block partition pairs and assigning state values can cause the cost and power consumption for testing to be minimized. Formula(6) is a function measuring the cost of weight transition probabilities in the blocks with $\pi(0)$ partitions.

$$Cost(\tau_t, \tau_t') = \sum_{t}^{block} Weight(m_t) \quad (6)$$

Example: Figure 4 can be reconstructed with 8 partition pairs like Figure 10. Of the partition pairs, (r_2, r_2'), (r_3, r_3'), (r_6, r_6'), (r_8, r_8') satisfy $\pi(0)$ condition and their costs are 42, 18, 30, and 23, respectively. Thus (r_2, r_2') of the highest cost is selected and the states are assigned by the block partition algorithm. In the first stage, the current state partitions $\{\overline{AC} : \overline{B} : \overline{DE} : \overline{F}\}$ are encoded by Ya and Yb bits and the next state partitions $\{\overline{A} : \overline{BC} : \overline{DF} : \overline{E}\}$ are by Yb and Yc bits. Initial values assigned to 6 states can be {001, 100, 000, 111, 110, 011} respectively. Although the states within the same block get the same upper or lower bits, the discrepancy among states in different blocks can be too high. For example, the B and F states with weight transition probability of 14 are assigned as 100 and 011 respectively, hence the Hamming distance becomes 3

$(r_1, r_1') = (\{\overline{ABC} : \overline{DEF}\}, \{\overline{ABC} : \overline{DEF}\})$
$(r_2, r_2') = (\{\overline{AC} : \overline{B} : \overline{DE} : \overline{F}\}, \{\overline{A} : \overline{BC} : \overline{DF} : \overline{E}\})$
$(r_3, r_3') = (\{\overline{AD} : \overline{B} : \overline{CE} : \overline{F}\}, \{\overline{AE} : \overline{BD} : \overline{CF}\})$
$(r_4, r_4') = (\{\overline{AE} : \overline{BF} : \overline{CD}\}, \{\overline{AE} : \overline{BF} : \overline{CD}\})$
$(r_5, r_5') = (\{\overline{ACF} : \overline{B} : \overline{DE}\}, \{\overline{ABCDF} : \overline{E}\})$
$(r_6, r_6') = (\{\overline{A} : \overline{BD} : \overline{C} : \overline{E} : \overline{F}\}, \{\overline{A} : \overline{BF} : \overline{CE} : \overline{D}\})$
$(r_7, r_7') = (\{\overline{AC} : \overline{BDE} : \overline{F}\}, \{\overline{A} : \overline{BCDEF}\})$
$(r_8, r_8') = (\{\overline{A} : \overline{B} : \overline{CF} : \overline{D} : \overline{E}\}, \{\overline{AD} : \overline{BF} : \overline{C} : \overline{E}\})$

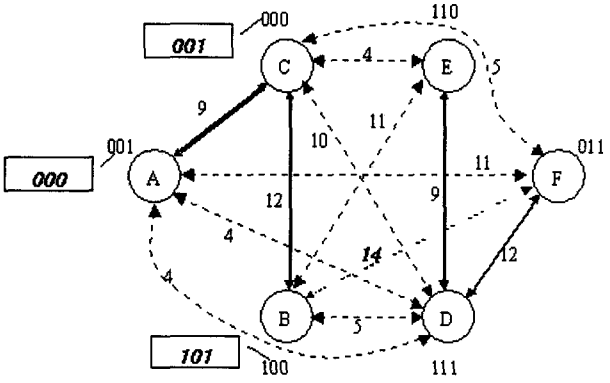

그림 10. 무게천이확률에 대한 m-블록 분할 쌍 및 m-블록 상태천이도

Fig. 10. m-block partition pair and m-block state assign -ment graph for weight transition probability.

which is the worst case. The augmenting algorithm, shuffling subset of state codes keeping the basic rule of the first stage, is applied in the second stage. In this example the state values of B, C and A are changed into 101, 001, 000, thus {000, 101, 001, 111, 110, 011} are finally assigned to each states. Note that the Hamming distance between state B and F is reduced to 2 from 3, and globally bit transitions are reduced as well.

To evaluate a new state assignment technique proposed in this paper a Minimum Transition Function (MTF) is defined as follows:

$$MTF = \sum_{t=1}^{Sn} Weight_t * HD(Ps_t \bullet Ns_t) \quad (7)$$

This equation of summing the state transitions on each connectivity is used to evaluate the switching frequency of a state assignment. In the above example, the MTF equals to 172 for the dependencies only and goes down to 168 for both the dependencies and transition probabilities.

표 1. 상태할당을 통한 고장점검도
Table 1. Fault Coverages upon different State Assign -ments.

Circuit	Ns/Nb	fault coverage(%)					
		Jedi	Random	2-block	m-block	New m-block	One-hot
bbsse	16/4	98.24	90.66	98.12	98.85	99.13	97.80
mark1	16/4	98.10	94.47	98.85	98.85	98.85	97.12
keyb	19/5	91.50	95.04	93.66	96.88	92.03	97.62
s832	25/5	97.56	98.88	45.61	97.92	98.66	86.43
tbk	32/5	96.97	98.59	98.98	98.98	99.14	97.38
s1494	121/7	96.81	96.34	98.26	94.87	95.98	59.77
scf	218/8	95.76	44.59	97.66	96.72	96.18	98.09

표 2. 상태할당을 통한 전력비교표
Table 2. Power Consumptions upon Different State Assignments.

Circuit	Ns/Nb	power(μW)					
		Jedi	Random	2-block	m-block	New m-block	One-hot
bbsse	16/4	551.0	527.5	568.1	572.9	519.8	823.9
mark1	16/4	360.8	436.6	383.0	383.0	383.0	600.1
keyb	19/5	824.2	1030.7	620.5	765.5	531.1	1362.2
s832	25/5	1161.4	1381.1	1181.0	1189.1	1071.4	2147.0
tbk	32/5	711.9	721.1	717.8	717.8	676.1	1051.4
s1494	121/7	2039.9	2698.2	2143.8	2043.3	1899.4	3683.7
scf	218/8	2451.4	2550.1	2473.8	2440.5	2286.1	4263.9

V. Experimental Results

For the experiments, synthesis tool SIS from U. C. Berkeley, Blif-to-Bench script, and automatic test pattern generation tool HITEC from U. C. Illinois are extensively used with our new state encoding algori -thm coded in C. The fault coverages for the sequential circuits synthesized by applying one-hot, random, Jedi, and our method are compared in 1. Especially m-block gives the high fault coverage than any other for most of the benchmarks. For keyb circuit, although our method results in 0.74% lower coverage than the one-hot, but 5.78% and 1.84% improvements than the Jedi and random respectively. As expected the m-block partition always gives better fault coverages than the 2-block partition, and the new m-block results in less switching activities than the conventional m-block algorithm. As shown in table 2, our new m-block state encoding technique promises the best power consumption as well as high fault coverages shown in table 1.

VI. Conclusions

In this paper a new m-block partitioning technique for the state assignment has been proposed with agoal to improve circuit testability and minimize power consumption. Our new state assignment techni -que is known to achieve high fault coverage by reducing the number of feedback cycles, while kee -ping the power consumption low by maintaining the low switching activities among the state variable. Experiment shows significant improvement in testabilities and power dissipation for benchmark cir -cuits. Recently test power for a SoC with multiple cores is of great concern. By synthesizing each core with our state encoding technique it is expected that the test power can also be improved.

References

[1] P. Kalla and M. J. Ciesielski., "A Comprehensive Approach to the Partial Scan Problem using Implicit State Enumeration," Proc. Int'l. Test Conf., pp.651-657, Nov. 1998.

[2] Xuejun Du, Gary Hachtel, Bill Lin, and A. Richard Newton, "MUSE: A MUltilevel Symbolic Encoding Algorithm for State Assignment," IEEE Trans on CAD., Vol. 10, NO. 1, pp. 28-38, January 1991.

[3] S Yang and M. J. Ciesielski, "Optimum and Suboptimum Algorithms for Input Encoding and Its Relationship to Logic Minimization," IEEE Trans. on CAD., Vol 10. No. 1. pp. 4-12, Jan. 1991.

[4] K. T. Cheng, and V. D. Agrawal, "Design of Sequential Machines for Efficient Test Genera -tion," in Proc. of ICCAD, pp. 358-361, 1989.

[5] G. De Micheli, "Symbolic Design of Combina -tional Sequential Logic Circuits Implemented by Two-level Logic Macros," IEEE TCAD, Vol. CAD-5, pp. 597-616, Oct. 1986.

[6] R. K. Brayton, G. D. Hatchel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli, Logic Minimi -zation Algorithms for VLSI Synthesis, Norwell, MA: Kluwer Academic, 1984.

[7] Z. Kohavi, Switching and Finite Automata Theo -ry, McGraw-Hill, 1978.
[8] E. Olson, S.M. Kang, "Low-Power State Assign -ment for Finite State Machines," in Proc. IEEE Intl. Workshop on Low Power Design, pp. 63-68, April 1995.
[9] V. Veeramachaneni, A. Tyagi, S. Rajgopal, "Re -encoding for Low Power State Assignment of FSMs," in Proc. IEEE Intl. Symposium on Low Power Design, pp. 173-178, April 1995.
[10] Chiusano S, Corno F, Prinetto P, Rebaudengo M, Sonza Reorda M, "Guaranteeing testability in re-encoding for low power," Test Symposium (ATS '97) Proceedings, Sixth Asian, pp. 30-35, 1997.
[11] L.Benini and G. De. Micheli, "State assignment for low power dissipation", IEEE Journal of Solid-State Circuits, vol. 30, March 1995.
[12] T. Villa et al., Synthesis of FSMs: Logic Optimi -zation. New York: Kluwer Academic, 1997.

저 자 소 개

박 성 주(정회원)
1983년 한양대학교 전자공학과 학사졸업.
1983년~1986년 금성사 소프트웨어 개발.
1992년 Univ. of Massachusetts 전기 및 컴퓨터공학과 석·박사졸업.
1992~1994년 IBM Microelec- tronics 연구스텝.
1995~현재 한양대학교 전자컴퓨터공학부 부교수
<주관심분야: 테스트 합성, Built-In Self Test, Scan Design, ATPG, ASIC 설계, 고속 신호처리 시스템 설계, 그래프이론 등>

조 상 욱(정회원)
1997년 한양대학교 전자계산학과 학사 졸업.
1999년 한양대학교 전자계산학과 석사 졸업.
2000년~현재 한양대학교 컴퓨터 공학과 박사과정.
<주관심분야: 테스트 합성, 저전력, Scan Design, VLSI 시스템 & 테스팅, SOC 테스팅>