

# A New State Assignment Technique for Testing and Low Power

Sungju Park, Sangwook Cho  
Dept. of CSE,  
Hanyang University at Ansan, Korea  
{parksj,swcho}@mslab.hanyang.ac.kr

Seiyang Yang  
Dept. of Computer Engineering,  
Pusan University, Korea  
syyang@pusan.ac.kr

Maciej Ciesielski  
Dept. of ECE,  
University of Massachusetts  
ciesiel@ecs.umass.edu

**ABSTRACT**

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 length and number of feedback cycles are reduced with minimal switching activity on the state variables. Experiment shows significant improvement in power dissipation and testabilities for benchmark circuits.

**Categories and Subject Descriptors**

B.8.1 [Performance and Reliability]: Reliability, Testing and Fault-Tolerance.  
J.6 [Computer-Aided Engineering]: Computer-aided design (CAD)

**General Terms**

Algorithm, Logic Synthesis, Design.

**Keywords**

State Encoding, Fault Coverage, Low power, Scan design.

**1. 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 applications. 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 techniques have been developed for testable design [1,2]. Instead of analyzing only the gate level circuit information, 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 [2]. 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 [3]. A few state encoding techniques have been addressed to minimize the power

\*This work was supported by grant no. R01-2003-000-10150-0 from Korea Science & Engineering Foundation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2004, June 7–11, 2004, San Diego, California, USA  
Copyright 2004 ACM 1-58113-828-8/04/0006...\$5.00.

consumption [4-6]. Testability was considered in re-encoding the states for the low power [5], however no paper has tried to optimize the testability and low power simultaneously at the state assignment 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 [2], to maximize the testabilities and minimize the power consumption. State transition probability is extensively adopted in partitioning the states while preserving the dependencies of the state variables minimal.

This paper is organized as follows: After introducing 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 targeted to improve both testability and power consumption is presented in section 4 followed by experimental results and conclusions.

**2. 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.

PS\X,x	NS				Assignment $\alpha, \beta$					
	00	01	11	10	$y_1$	$y_2$	$y_3$	$v_1$	$v_2$	$v_3$
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

Figure 1. State Transition Table and Assignment

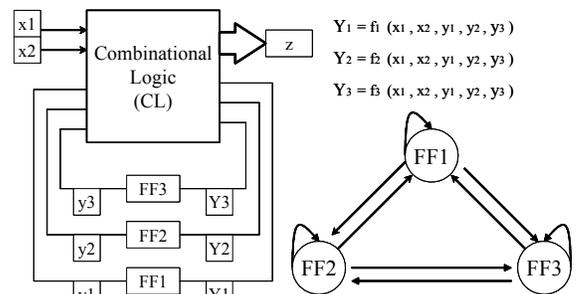


Figure 2. Circuit structure and Scan Graph by State Assignment  $\alpha$

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, where PS and NS represent present states and next states respectively, can be synthesized to the figure 2 circuit by taking the state assignment  $\alpha$  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.

On the other hand, if the circuit is implemented as figure 3 using the different state assignment  $\beta$  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  $\beta$ . Since no simple cycle exist in figure 3 circuit, we may not need any scan flip-flop for testing.

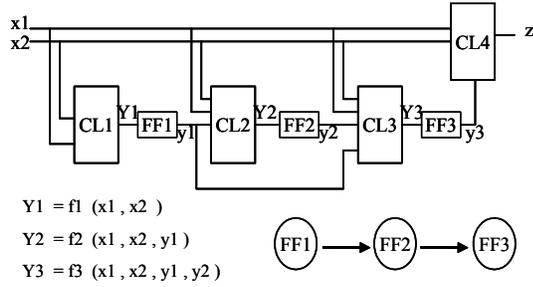


Figure 3. Circuit structure and Scan Graph by State Assignment  $\beta$

A few terms are defined as followings for detailed description of optimal state assignment for high testability and low power consumption [3].

**Definition 1]** A partition consists of blocks such that there is no common state symbol among the blocks and the union of all blocks 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.

**Theorem 1]** Let M be a sequential machine with K state variables,  $y_1, y_2, \dots, y_k$ . If there exists a closed partition  $\Pi$  on the states of M and if r state variables, where  $r = \lceil \log_2 \#(\Pi) \rceil$ , are assigned to

the blocks of  $\Pi$ , so that all the states contained in each block are assigned the same values of  $y_1, y_2, \dots, y_r$ , then the next state variables,  $Y_1, Y_2, \dots, Y_r$  are independent of the remaining k-r variables.

**Proof]** Please refer to the [3].

It can be observed from the theorem 1 that if the states can be partitioned into m-block satisfying closed partitioning condition, the dependencies among state variables can be greatly simplified. Since the number of partial scan flip-flops are highly dependent on the complexity of the dependencies among state variables[1], the partial scan testabilities can be also highly improved. In figure 4 state transition diagram, let us consider two partitions  $p' = \{\overline{ad} : \overline{b} : \overline{ce} : \overline{f}\}$  and  $p'' = \{\overline{ae} : \overline{bdcf}\}$ , where each block is separated by ":" symbol. The p' and p'' are 4-block and 2-block partitions respectively, and an ordered pair (p',p'') is a partitioning pair.

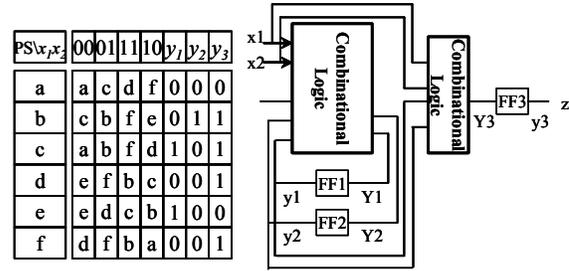


Figure 4. State Assignment and Circuit Structure

The relation between partition and state assignment can be explained through the figure 4. From the state assignment, each state variable generates a 2-block partition, that is, y1 variable produces  $p1 = \{\overline{abd} : \overline{cef}\}$  partition, y2 produces  $p2 = \{\overline{acde} : \overline{bf}\}$ , and y3 produces  $p' = \{\overline{ae} : \overline{bdcf}\}$ . Furthermore more than one variables can generate m-block partitions such as y1 and y2 produce  $p12 = \{\overline{ad} : \overline{b} : \overline{ce} : \overline{f}\}$  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 among flip-flops in 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 contrast to the method in [2], 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.

**Example:** By applying our state assignment algorithm based on m-block partitioning to the figure 4 state machine, we have found that the state machine includes a few different state partitioning pairs as shown in figure 5. If the partition results in similar dependencies among state variables, which pair will be better choice for low power consumption? The following section will describe a solution for this question.

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

Figure 5. Different partitioning pairs

### 3. State Encoding for Low Switching Transitions

In order to reduce the power consumption, it is very crucial to assign the states so that the transitions among state variables occur least frequently. First of all, the transition relations according to inputs need to be expressed as transition probabilities. 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 current state. Figure 6(a) is a state transition

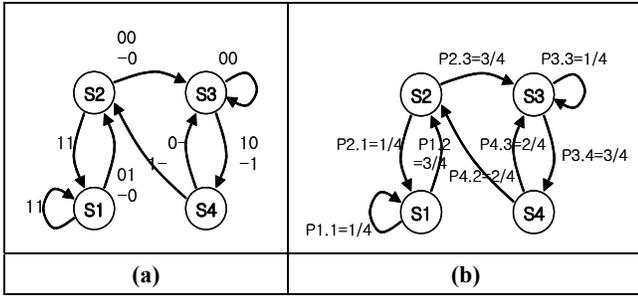


Figure 6. State Transition Diagram (STD) and Conditional Transition Probability of STD

diagram based on a state transition table.

The conditional transition probability(CTP) from the current state to the next state is defined as:

$$P_{i,j} = P(Ns_j | Ns_i) \quad (1)$$

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 6(b). 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.

$$Gp_{i,j} = p_{i,j} \cdot P_{S_i} \quad P = B^T \cdot P \quad \sum_{i=0}^n P_{S_i} = 1 \quad (2)$$

Here,  $P_{S_i}$  is the current state probability of  $S_i$  and can be calculated with Markov chain characteristics. The conditional transition probability is expressed as a matrix B.

Figure 7 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 [6].

The highest transition probability from the current state  $s_i$  to the next state  $s_j$  implies that the switching variation between  $s_i$  and  $s_j$  occurs most frequently.

$$B = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} & 0 & 0 \\ \frac{1}{4} & 0 & \frac{3}{4} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{3}{4} \\ 0 & \frac{2}{4} & \frac{2}{4} & 0 \end{bmatrix}$$

$$\begin{aligned}
P_1 &= \frac{1}{4} P_1 + \frac{1}{4} P_2 \\
P_2 &= \frac{3}{4} P_1 + \frac{2}{4} P_4 \\
P_3 &= \frac{3}{4} P_2 + \frac{1}{4} P_3 + \frac{2}{4} P_4 \\
P_4 &= \frac{3}{4} P_3 \\
P_1 + P_2 + P_3 + P_4 &= 1
\end{aligned}$$

$$P_1 = \frac{2}{29}, P_2 = \frac{6}{29}, P_3 = \frac{12}{29}, P_4 = \frac{9}{29}$$

Figure 7. Global Transition Probability of STD

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 normalized integer values of the figure 7. In this example the most frequent transition occurs between s3 and s4 with integer value of 27, thus they 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.

### 4. New State Encoding Technique for Testability and Low Power

**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. Therefore this m-block partition considering state transition 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(3) is a function measuring the cost of weight transition probabilities in the blocks with  $\pi(0)$  partitions.

$$Cost(\tau_i, \tau_i') = \sum_i^{block} Weight(m_i) \quad (3)$$

**Example:** Figure 4 can be reconstructed as 8 partition pairs of figure 5 and m-block state assignment graph of figure 8. 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, 23, and 23, respectively. Thus  $(r_2, r_2)$  of the highest cost is selected and the states are assigned by the block partition algorithm.

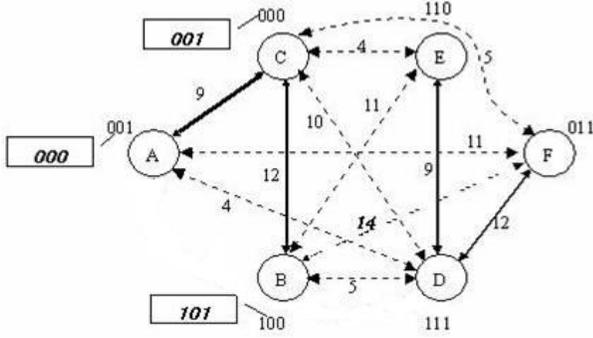


Figure 8. m-block state assignment graph for weight transition probability

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(HD) becomes 3 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 HD 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:

$$MTF = \sum_{i=1}^{S_n} Weight_i * HD(Ps_i \bullet Ns_i) \quad (4)$$

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.

## 5. Experimental Results

For the experiments, synthesis tool SIS from U.C. Berkeley, Blifto-Bench script, and automatic test pattern generation tool HITEC from U.C. Illinois have been extensively used with our new state encoding algorithm. The new m-block in the table is the algorithm proposed in this paper adding HD heuristics among states for low power to the m-block algorithm. The single stuck-at fault coverages for the sequential circuits synthesized by applying one-hot, random, Jedi, and our method are compared in table 1, where Ns and Nb represent the number of states and flip-flops respectively. Although our new m-block does not guarantee the highest fault coverages for all the benchmarks, it has shown that comparable coverages can be achievable. However, as far as the power consumption is concerned, the table 2 shows that our new m-block method produces the least power consumption for most of the benchmark circuits. Both tables show that our approach

shown the 5<sup>th</sup> columns achieves almost highest average fault coverage and power consumption at the same time

Table 1. Fault Coverage(FC) upon different State Assignments

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	48/6	96.81	96.34	98.26	94.87	95.98	59.77
Average FC		96.42	88.37	90.16	97.58	97.14	90.60

Table 2. Power Consumption(PC) upon Different State Assignments

Circuit	Ns/Nb	power( $\mu W$ )					
		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	48/6	2039.9	2698.2	2143.8	2043.3	1899.4	3683.7
scf	121/7	2451.4	2550.1	2473.8	2440.5	2286.1	4263.9
Average PC		1157.2	1335.0	1155.4	1158.9	1052.4	1990.3

## 6. Conclusion

In this paper a new m-block partitioning technique for the state assignment is proposed to reduce the number of feedback cycles and keep low switching activities among state variables. Experiment shows significant improvement in power dissipation and testabilities for benchmark circuits. By synthesising SoC cores with our state encoding technique it is expected that the test power can also be highly saved.

## 7. 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] K. T. Cheng, and V. D. Agrawal, "Design of Sequential Machines for Efficient Test Generation", in Proc. of ICCAD, pp. 358-361, 1989..
- [3] Z. Kohavi, Switching and Finite Automata Theory, McGraw-Hill, 1978.
- [4] E. Olson, S.M. Kang, "Low-Power State Assignment for Finite State Machines", proc. IEEE Intl. Workshop on Low Power Design, pp. 63-68, April 1995.
- [5] 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.
- [6] L.Benini and G. De. Micheli, "State assignment for low power dissipation", IEEE Journal of Solid-State Circuits, vol. 30. March 1995.