

논문 2004-41SD-1-11

테스팅 및 저전력을 고려한 최적화된 상태할당 기술 개발

(Development of Optimized State Assignment Technique for Testing and Low Power)

조 상 욱*, 이 현 빈*, 박 성 주**

(Sangwook Cho, Hyunbean Yi, and Sungju Park)

요 약

유한상태기의 상태할당은 이로부터 구현되는 순차회로의 속도, 면적, 테스트 및 소비전력에 큰 영향을 미친다. 본 논문에서는 상태변수 그룹들 사이에 상호 의존성(dependency)을 최소화하여 테스트 및 전력소모를 개선하기 위한 m-블록 분할을 이용한 새로운 상태할당 기술을 소개한다. m-블록 분할 알고리즘에 의해 상태도로부터 상태들을 그룹으로 나누어 상태변수의 상호의존성을 줄이고, 상태전이 확률에 의해 결정된 무개인자에 따라 상태간 상태변수의 변화를 최소화하는 코드를 할당하여 상태전이시 스위칭 횟수를 줄인다. 상태변수 의존성을 줄임으로써 순차회로 사이클이 줄어들어서 부분스캔 및 테스트 생성이 용이하게 되고, 상태변수간의 스위칭 횟수를 줄임으로써 소비전력이 줄어들게 된다. 즉, 본 상태할당 기술은 서로 상반 관계에 있는 테스트와 저전력 문제를 동시에 해결할 수 있는 새로운 기술이다. 벤치마크 회로에 대한 실험결과 기존의 방법보다 고장점검도 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. By the m-block partition algorithm, the dependencies among groups of state variables are minimized and switching activity is further reduced by assigning the codes of the states in the same group considering the state transition probability among the states. In the sequel the length and number of feedback cycles are reduced with minimal switching activity on state variables. It is inherently contradictory problem to optimize the testability and power consumption simultaneously, however our new state assignment technique is able to achieve high fault coverage with less number of scan flip flops by reducing the number of feedback cycles while the power consumption is kept low upon the low switching activities among state variables. Experiment shows drastic improvement in testabilities and power dissipation for benchmark circuits.

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

I. 서 론

칩의 집적도의 증가에 따라 설계점검 및 칩제작 후의

* 정회원, 한양대학교 컴퓨터공학과
(Dept. of Computer Science & Engineering, Hanyang Univ.)

** 정회원, 한양대학교전자 컴퓨터공학과
(Dept. of Electrical Engineering Computer Science, Hanyang Univ.)

※ 본 논문은 한국과학재단 특정기초과제(R01-2003-000-101-50-0)로부터 지원을 받아 진행하였습니다.

접수일자 : 2003년8월12일, 수정완료일 : 2003년12월26일

기능점검 등은 더욱 더 어려운 문제로 부각되고 있다. 이러한 테스트 문제를 해결하기 위하여 체계적인 테스트설계 기술이 널리 개발되고 있다. 본 논문에서는 회로의 면적 최소화와 함께 테스트가능도 및 저전력을 최적화 할 수 있는 상태할당 기술을 연구하였다. 이제까지 발표된 상태할당 연구는 면적의 최소화나 성능 또는 저전력에 대해서는 어느 정도 활발히 진행되고 있으나 테스트와 저전력을 동시에 고려한 상태할당 연구는 거의 전무한 상태이다. 상태할당 방법에는 One-hot, Random,

Jedi 와 같은 방법이 개발되었다^{[1],[2]}. One-hot은 n 개의 상태들로 구성되어진 유한 상태기에 정확하게 n 개의 플립플롭을 사용해서 상태할당을 하는 것이다. 이는 가장 많은 플립플롭을 필요로 하지만 다음 상태와 output logic을 줄인다는 면에 있어서 효율적이다. Random은 $\log(N)$ 개의 플립플롭을 사용하여 상태할당을 한다. 그리고 Jedi 역시 $\log(N)$ 개의 플립플롭을 사용하여 상태할당을 하는데, greedy 알고리즘을 사용하여 멀티 레벨을 위한 일반적인 symbol encoding에 목표를 두고 있다. 이 방법은 심볼들의 쌍들간에 무게를 적용하는 휴리스틱 방법을 적용하고 있다. 이 방법은 또한 literal saving 면에서 효율적인 방법이다. 이와 같이 상태할당은 구현되는 동기 순차회로의 면적에 큰 영향을 미치므로 이론적인 면에서나 실질적인 면에서 상당히 중요한 문제로 대두되어 왔다^{[3],[4],[5],[6],[7]}. 그러나 상태할당 문제는 구현되는 회로의 면적뿐만 아니라 성능/테스트 가능성도/전력소비량 등에도 큰 영향을 미칠 수 있다.

특히 회로의 테스트 가능성에 상태할당이 지대한 영향을 미칠 수 있다는 것이 최근의 연구들을 통하여 알려지고 있다^{[4],[8]}. 이는 구현된 동기 순차회로에 테스트를 수행하는 경우에 있어 회로의 메모리 소자들의 상호 의존성은, 이들 테스트 수행의 난이도에 결정적인 영향을 미칠 수 있기 때문인데, 이러한 메모리 소자들간의 상호 의존성은 오직 상태할당에 의하여 결정된다. 이러한 사실을 이용하여^[9]에서는 2-블록 분할(2-block partition)을 이용하여 순차 테스트를 용이하게 하면서 면적최소화가 가능하게 하는 상태할당 방법을 제안하였다. 본 논문의 전반부에서는^[9]에서의 방법을 일반화시켜 m -블록 분할을 이용한 상태할당을 통하여 2-블록 보다 더욱 전체적인 관점에서 메모리 소자들간의 상호 의존성을 최소화하여 회로 면적을 최소화하고 테스트가능도도 높이도록 한다.

지금까지 저전력을 고려한 상태할당 기술은 몇 가지 있지만^[10-15] 저전력과 테스트를 동시에 고려한 상태할당 기술^[17]은 거의 전무한 상태이다. 전력소비를 최소화하는 방법 중 하나는 유한 상태기에 대한 상태할당 방법이다. 저전력을 위한 상태할당은 상태천이 그래프에서 각 상태천이에 대해 평균 게이트 스위칭 횟수를 줄이는데 초점을 둔다. ^[10]은 각 상태에 대해서 방문되는 확률을 구하여 가장 빈번히 방문되는 상태에 대해서는 가장 짧은 코드가 할당되도록 Huffman 인코딩을 사용한다.^[20]은 회로의 입력 조건이 주어지면 이를

바탕으로 하여 각 신호 천이의 관계를 확률적으로 측정하고 스위칭 확률이 많이 일어나는 곳에 비트의 변화가 최소가 되도록 encoding하여 전력소비를 감소시킨다. 최근에는 알고리즘들의 정확도를 높이기 위해서 상태들에 대한 확률과 각 변화하는 플립플롭에 붙어있는 조합회로 부분을 함께 평가한다. 상태할당은 또한 조합회로에 대한 불린 식을 변화시키고 플립플롭간의 의존 관계를 변화시킴으로써 회로의 게이트 수준 구조에 변화를 주게되므로 테스트가능도에도 많은 영향을 주게 된다. 그런데, 저전력을 위한 최적화 과정의 목표는 테스트가능도를 높이기 위한 최적화 과정과 보편적으로 상반되는 특징을 갖는다. 저전력 회로는 각 입력의 변화에 대해서 게이트 이벤트가 최소화 되어야하지만 테스트하기 쉬운 회로는 입력의 이벤트가 출력에 쉽게 전달되기 위해서 많은 게이트 이벤트가 주출력까지 전달되어야 한다.

테스트를 고려한 상태할당 기술을 바탕으로 전력 소비도 최소화 할 수 있는 상태할당 기술을 소개하는 본 논문은 다음과 같이 구성되어 있다. II 절에서는 상태할당을 통한 순차회로 구조에서 상태변수들간의 의존성을 비교하고, III 절에서는 테스트를 고려한 최적의 상태할당 기술을 기술한다. IV 절에서는 테스트 및 저전력을 고려한 상태할당 방법을 소개하며, V 절에서는 벤치마크 회로에 대한 실험결과를 보여주고, 마지막 장에서는 결론을 기술한다.

II. 순차회로에 대한 상태할당 방법

순차회로에 대한 논리 합성에 있어서 주된 관심사 중의 하나는 유한상태기를 이용하여 추출한 상태천이표에서 각 상태들에 이진 코드 값을 할당하는 것이다. 이 경우, 그 상태들과 출력 변수의 기능적인 의존도 뿐만 아니라 보다 적은 수의 메모리 소자들만으로 회로가 이루어 질 수 있도록 할당을 하여야 한다. 그러나 실제로 상태할당이 상이하게 되어지는 모든 경우의 수는 상태 수와 비교하여 지수 함수적으로 많아지는데, 예를 들어 n 개의 상태들로 구성된 유한상태기는 $n!$ 개의 다른 상태할당을 가진다. 따라서 그 중에서 가장 최적으로 할당된 상태들을 구하는 것은 NP-complete 문제이다.

다음에는, 본 논문에서 연구된 부분스캔을 고려한 최적화된 상태할당의 연구 동기를 예를 통하여 설명한

PS/ X1X2	NS			
	00	01	11	10
a	e	c	d	e
b	g	a	b	g
c	a	c	h	e
d	c	a	f	g
e	e	c	d	e
f	g	a	b	g
g	a	c	h	e
h	c	a	f	g

(a)

α		β	
y1y2y3	y1y2y3	y1y2y3	y1y2y3
a : 0 0 0	a : 0 0 0	a : 0 0 0	a : 0 0 0
b : 1 0 0	b : 1 1 0	b : 1 1 0	b : 1 1 0
c : 0 1 0	c : 0 1 0	c : 0 1 0	c : 0 1 0
d : 0 1 1	d : 1 0 0	d : 1 0 0	d : 1 0 0
e : 1 1 1	e : 0 0 1	e : 0 0 1	e : 0 0 1
f : 1 0 1	f : 1 1 1	f : 1 1 1	f : 1 1 1
g : 1 1 0	g : 0 1 1	g : 0 1 1	g : 0 1 1
h : 0 0 1	h : 1 0 1	h : 1 0 1	h : 1 0 1

(b)

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

다. 우선 그림 1의 (a)와 같은 상태전이표로 표시되는 유한상태기를 임의의 상태할당을 거쳐 구현하는 경우 상태 변수들간의 의존성을 비교해본다. 그림 1의 (a)에 있는 상태전이표로부터 (b)의 상태할당에 의하여 회로를 구현하면 유한상태기내의 3개의 플립플롭 Y1(y1), Y2(y2), Y3(y3) (Y는 다음상태, y는 현재상태) 들간에는 “상호간” 완전한 의존성이 생기게 된다 (그림 2). 또한^[16]에서는 스캔-그래프에서 서로 다른 단순 사이클들이 얼마나 존재하는 가를 알 수 있다. 그림 2에 의해서 구현되는 회로에서 이 스캔-그래프를 구성하면 2개의 단순 사이클이 존재함을 알 수 있다. 이와 같은 경우에는 3개의 플립플롭 가운데 최소한 하나의 플립플롭을 스캔 가능하게 하여야 부분 스캔에 의한 테스트가 이루어 질 수 있다. 반면, 그림 3과 같이 회로가 구현된 경우에는 유한상태 기내의 3개의 플립플롭 Y1(y1), Y2(y2), Y3(y3) 들간에는 “한 방향으로의” 의존성(dependency)만이 생기게 된다. 이는 부분 스캔의 입장에서는 매우 바람직한 구조이다. 왜냐하면 단순 사이클이 존재하지 않음을 알 수 있으며 이는 회로내의 플립플롭을 전혀 스캔 가능하게 하지 않아도 부분 스캔에 의한 테스트가 가능할 수 있기 때문이다. 또한, 표 1과 같이 모든 현재상태에서 다음상태로 천이 되는 비트별 변화의 차이에서도 상태할당β가 상태할당 α보다 회로 면적 증가도 최소화하므로 부분 스캔 테스트에 효율적이라 할 수 있다.

표 1. 모든 State 천이에 대한 비트 변화의 수
Table 1. The number of bit change for all state transitions.

	a-e	a-c	a-d	h-a	h-f	h-g	합계
상태할당α	3	1	2	1	1	3	52
상태할당β	1	1	1	2	1	2	48

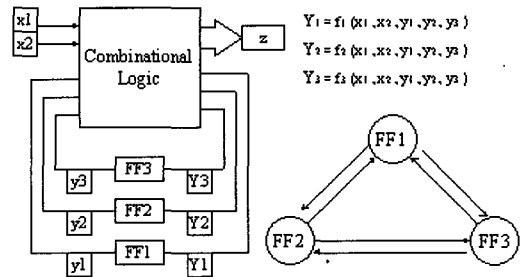


그림 2. 상태할당 α에 의한 회로구조 및 스캔 그래프

Fig. 2. Circuit structure and Scan Graph by State Assignment α.

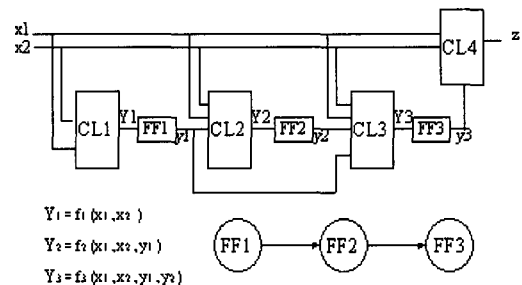


그림 3. 상태할당 β에 의한 회로구조 및 스캔 그래프

Fig. 3. Circuit structure and Scan Graph by State Assignment β.

III. 테스트를 고려한 최적화된 상태할당 기술 소개

1. 테스트를 고려한 상태할당

먼저, 최적화된 상태할당 기술을 위한 용어를 정의한다

정의 1] S : FSM 상태천이도의 심벌상태

정의 2] 분할(partition) : 교집합이 S인 연결되지 않은 부분집합

정의 3] 블록(block) : 각 부분집합

정의 4] m - 블록분할 : 부분집합의 수가 m 인 경우

정의 5] i&s : 상태 s에서 입력 i에 의하여 천이 되는 다음상태

정의 6] 분할 쌍(partition pair) : (p1,p2) 는 각각의 입력i에 대하여 p1의 같은 블록에 상태 s1과 s2가 같이 있으면 i&s1, i&s2가 p2에서 같은 블록에 있는 분할의 순서쌍(ordered pair)

정의 7] 의존성 : 다음상태 변수들의 값은 남아 있는 변수의 값에 의존하는 경우

논문^[19]의 예제를 인용하여 용어를 설명한다. 그림 5 M의 상태전이표로부터 다음과 같은 두 개의 분할 $p' = (a, d : b : c, e : f)$ 와 $p'' = (a, e : b, d, c, f)$ 를 보자. p' 은 4-블록 분할이고 p'' 는 2-블록 분할이다. 또한 순서쌍 (p', p'') 은 분할쌍이다. 그러면 분할과 상태할당과의 관계는 어떠한지를 그림 4의 예제를 통해 알아본다.

PS/ X1X2	NS				y1y2y3	
	00	01	11	10		
a	a	c	d	f	a	000
b	c	b	f	e	b	011
c	a	b	f	d	c	101
d	e	f	b	c	d	001
e	e	d	c	b	e	100
f	d	f	b	a	f	111

그림 4. 상태전이표 M 및 상태할당
Fig. 4. State Transition Table M and State Assignment.

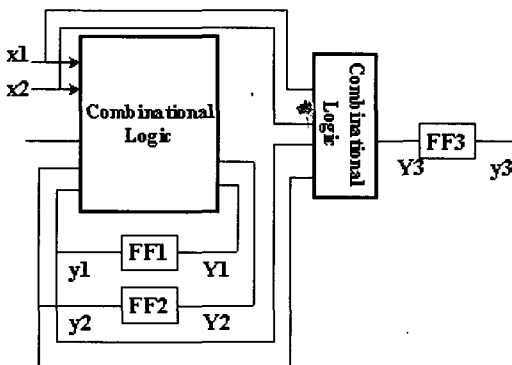


그림 5. 분할쌍에 의한 회로 구조
Fig. 5. Circuit Structure by Partition Pairs.

상태할당으로부터 각각의 상태변수 하나는 해당 2-블록 분할($y1$ 은 $p1 = (a, b, d : c, e, f)$, $y2$ 는 $p2 = (a, c, d, e : b, f)$ 그리고 $y3$ 은 $p3 = (a, e : b, d, c, f)$)에 대응시킬 수 있음을 알 수 있다. 뿐만 아니라 두 개 이상의 상태변수들로부터는 해당 m -블록 분할($y1$ 과 $y2$ 로부터 $p12 = (a, d : b : c, e : f)$ 의 4-블록 분할)을 대응시킬 수 있다. 즉, 상태할당과 블록 분할은 구별 없이 같이 생각할 수 있는 것인데, 블록 분할로부터 순차회로내의 메모리 소자들간의 상호 의존도를 유추해 낼 수 있다. 이를 그림 5에서 살펴보면 다음상태 변수 $Y3$ 은 현재상태 변수 $y1$ 과 $y2$ 및 입력 변수에 의존적임을

알 수 있다. 그리고 메모리소자 FF3과 FF2, FF3과 FF1간에는 이러한 상호회환이 없다. 이와 같이 상태할당(분할)을 통하여 구현된 모든 순차회로의 테스트를 용이하게 하는 요인은 분할쌍의 결과이다. 이러한 분할을 이용한 상태 할당은 회로의 면적최소화면에서도 바람직한 결과를 낼 수 있다는 것이 알려져 있다^[16,20]. 본 논문에서는 임의의 상태 할당(분할)을 가능한 한 분할쌍을 연속적으로 이용한다는 면에서는^[18] 과 같으나 분할이 반드시 2-블록 분할이 되도록 하지 않고 m -블록 분할까지 모두 허용한다는 것이^[18]과 다르다.

이와 같은 차이점으로부터 부분 스캔을 위한 상태할당에서 본 논문에서의 방법이^[17]에서의 방법보다 더욱 일반화되어진 방법이라고 할 수 있다. 즉, ^[17]에서의 방법은 상태 변수 하나들간의 의존성만을 고려하여 상태 할당을 수행하는 반면에, 본 방법은 상태 변수 그룹(그룹에는 상태변수가 1개 이상 있을 수 있음)들간의 의존성을 고려하여 상태할당을 수행하므로 더욱 일반화되어지고, 따라서 부분 스캔의 측면에서 상태 변수들간의 의존성을 더욱 전역적(global)으로 고려하여 최소화할 수 있다. 이를 좀 더 구체적인 상황으로서 설명하면 다음과 같다. 임의의 유한상태기 M을 2-블록 분할에 의하여 부분 스캔을 고려한 분할쌍을 구하고 상태변수를 구한 경우를 $Y1, Y2, Y3, Y4, Y5, Y6$ 라고 했을 때 이들 Yi ($i = 1, 2, \dots$ 또는 6)와 Yj ($i = 1, 2, \dots$ 또는 6)간에는 상호 의존성이 존재하게 되어 부분 스캔의 측면에서는 바람직하지 않게 되는 반면에, 4-블록 분할에 의하여 부분 스캔을 고려한 분할쌍을 구하고 상태변수를 구한 경우를 Ya, Yb, Yc 라고 했을 때 이들 간에는 일방 의존성만이 존재하게 됨으로 부분 스캔의 측면에서 바람직하게 될 수 있다. 여기에서 바람직하다는 의미는 첫째로는 적은 수의 플립플롭을 스캔 가능하게 만듦으로서 오버헤드를 최소화할 수 있다는 것과, 둘째로는 좀 더 적은 수의 스캔 가능 플립플롭만을 사용하는 것이 가능하지는 않더라도 이와 같은 상태할당을 통한 상태변수들간의 의존성 정보를 이용하여 효과적으로(예로 좀 더 빠른 시간 안에) 스캔이 필요한 플립플롭들을 선택할 수 있다는 것이다.

2. 상태변수들의 의존성을 줄이는 방법

본 장에서는 상태 변수들의 의존성을 줄이는 방법을 설명하는데 필요한 정의는 다음과 같다.

정의 8] 닫혀진 분할(Closed partition) : P 분할되어

- 진 m-블록 내에 있는 상태들에 대해 다음 상태들이 분할된 n-블록 내에 존재한 경우
- 정의 9] $\pi(0)$: 닫힌 분할 중 product을 통해 하나의 블록 내에 상태 하나만이 존재하는 경우
- 정의 10] y_1, y_2, \dots, y_k : 상태들에 할당될 변수들
- 정의 11] a, b, c, \dots : 상태(state)들
- 정의 12] predecessor partition ($v(ab)$) : 한 블록(ab), 나머지 상태들은 분리된 블록을 포함한 분할
- 정의 13] m-partition = $m(v(ab))$: $r(ab)$ 에 의해 다음상태들간의 product를 통해 유도되어진 다음상태들을 포함하는 가장 작은 분할
- 정의 14] M-partition = $M(v)$: $m(ab)$ 에 의해 분할된 블록들에 포함되어지는 v들의 합(sum)을 결정하는 가장 큰 분할
- 정의 15] Mm pairs : $(M(v_1), m(v_1))$ 는 각각의 입력에 대하여 $M(v_1)$ 의 같은 블록에 상태 s_1 과 s_2 가 같이 있으면 $i \& s_1, i \& s_2$ 가 $m(v_1)$ 에서 같은 블록에 있는 분할의 순서쌍(ordered pair)

• 방법 1 : 우선 임의의 상태들 a, b에서 predecessor partition($v(ab)$)으로 m-partition($m(v(ab))$)계산한다. 같은 방식으로 $m(v(ab)), m(v(ac)), \dots, m(v(c))$ 를 구한다. 다음은 이들 $m(v(ab))$ 의 모든 가능한 sum을 구하는 M-partition을 구한다. 나머지 역시 같은 방식으로 남아있는 M-partition을 구한다. 이렇게 하면 여러 개의 Mm pairs를 구할 수 있다.

상기 방법은 메모리 소자들간의 상호 의존성을 줄일 뿐만 아니라 상태할당을 위한 선택 범위 또한 줄여준다. 그러나 구한 Mm 분할쌍 모두를 적용하는 것 또한 비효율적인 단점이 있다. 이러한 단점을 다음과 같은 방법으로 보완한다.

• 방법 2 : 구한 Mm 분할쌍들 중에서 closed partition이면서 $\pi(0)$ 인 것을 산출해낸다. 위의 조건을 만족한 분할쌍에 상태할당을 통하여 메모리 소자들간에 의존성이 감소되는 것을 확인 할 수 있다.

다음은 그림 4의 상태천이표 M을 이용하여 본 논문에서 적용한 방법으로 구한 예를 보여주고 있다.

$$\begin{aligned}
 & p1 = \{a, b, c; d, e, f\} \\
 & P2 = \{a, e; b, f; c, d\} \\
 & p1 \cdot p2 = \{a; b; c; d; e; f\} = \Pi(0) \\
 & \begin{array}{l}
 y_1 y_2 y_3 \\
 a: 000 \\
 b: 011 \\
 c: 010 \\
 d: 110 \\
 e: 100 \\
 f: 111
 \end{array}
 \end{aligned}$$

logical equation of assignment

$$\begin{aligned}
 Y_1 &= f_1(x_1, y_1) \\
 Y_2 &= f_2(x_1, x_2, y_3) \\
 Y_3 &= f_3(x_1, x_2, y_2)
 \end{aligned}$$

그림 6. Mm pairs에서 $\pi(0)$ 를 만족하는 예
Fig. 6. Example satisfying with $\pi(0)$ in Mm pairs.

$$\begin{aligned}
 & p1 = \{a, c, f; b; d, e\} \\
 & P2 = \{a, b, c, d, f; e\} \\
 & p1 \cdot p2 = \{a, c, f; b; d; e\} \neq \Pi(0) \\
 & \begin{array}{l}
 y_1 y_2 y_3 \\
 a: 000 \\
 b: 110 \\
 c: 000 \\
 d: 100 \\
 e: 101 \\
 f: 000
 \end{array}
 \end{aligned}$$

logical equation of assignment

$$\begin{aligned}
 Y_1 &= f_1(x_1, y_1) \\
 Y_2 &= f_2(x_1, x_2, y_1, y_2, y_3) \\
 Y_3 &= f_3(x_1, x_2, y_1, y_2, y_3)
 \end{aligned}$$

그림 7. Mm pairs에서 $\pi(0)$ 를 만족하지 못하는 예
Fig. 7. Example not satisfying with $\pi(0)$ in Mm pairs.

그림 6에서는 상태변수 Y_1 과 Y_2 는 상호 제한루프가 줄어들어 부분 스캔에 의해 테스트가 용이한 반면에 그림 7에서는 상대적으로 메모리 소자들간의 상호 제한루프가 형성되므로 회로를 부분스캔이나 논 스캔에 의하여 테스트하는 것이 용이하지 못할 것이다. 또한 그림 8은 메모리 소자들간의 상호 의존성을 최소화하는 상태할당 알고리즘을 보여주고 있다

1. encoding 할 n state를 위한 $k = \lceil \log n \rceil$ state 변수 사용
2. while (states > 2) {
 상태천이표에서 M, m 구함
 IF (M, m = (분할쌍 AND Closed partition))
 Mm pairs
 Mm pairs 중에서 $\pi(0)$ 를 구함
 M 분할내에서 상태변수 y_r 이 선택 ($1 \leq r < k$)
 state 변수에 state value assign
 m 분할 내에서 상태 변수 y_{k-r} 이 선택
 state 변수에 state value assign
 }
3. Block partition 시 partition 간에 partition pair 가 아닌 경우 2 의 과정을 반복

그림 8. 상태할당 알고리즘
Fig. 8. State Assignment Algorithm.

IV. 테스트 및 저전력을 고려한 상태할당 방법

1. 저전력을 고려한 상태할당 기술

일반적으로 CMOS 회로의 평균 전력소비 모델은 다음과 같다^[21,22].

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

P_{ave} 는 소모되는 평균 전력을 나타내며, f 는 주파수, V_{dd} 는 공급전압이고, C_i 는 회로의 부하 캐피타이며, n_g 는 회로의 게이트의 수이고, P_i 는 스위칭 확률이다. 전력소비를 줄이기 위해서 평균 전력소비를 결정짓는 파라미터 중 V_{dd} 나 f 는 설계과정에서 결정되므로 상수로 취급하고, 자원공유를 최소화하여 스위칭 변화를 감소시키는 방법으로 저전력을 구현한다. 식 (1)에서 신호의 천이 과정에서 발생하는 스위칭 변화를 줄이는 것이 소비전력을 감소하는데 중요한 요소임을 알 수 있다.

순차 회로의 상태할당을 위해 상태천이표에서 입력에 따라 천이되는 관계를 확률적으로 계산한다. 확률 모델로서 사용하는 Markov Chain은 순차 특성에 있어서 동적 변화의 상관 관계를 가진다. 시스템에 관한 정보와 입력에 대한 확률이 주어지면 상태천이도에 대한 천이확률을 계산할 수 있다. 입력에 대한 확률은 상태천이도에서 얻을 수 있으며 천이확률은 그림 9와 같이 상태의 입력에 의존한다.

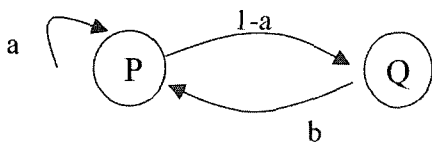


그림 9. m-block의 Markov Chain
Fig. 9. Markov Chain of the m-block.

그림 9는 기본적인 Markov Chain 형태를 3장에서 언급한 m-block으로 변환하여 나타낸 그림이다. 현재 상태에서 다음 상태로 천이되는 조건은 단지 현재 상태에 의해서만 결정되는 것을 나타낸다. 이를 블록사이에서도 확장 할 수 있다. 그림 10은 상태천이표를 바탕으로 작성된 상태천이도이다. 현재 상태에서 다음 상태로 천이조건 확률은 다음과 같은 식(2)로 정의한다.

$$p_{i,j} = \text{Prob}(Ns_j | Ps_i) \quad (2)$$

예를 들어 s2에서 s3으로 천이 조건 확률은 4가지의

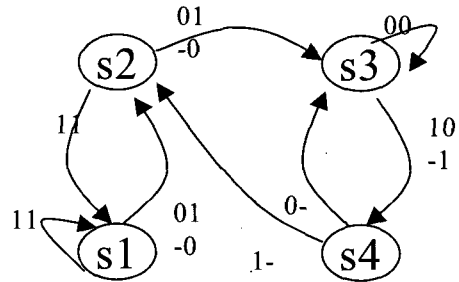


그림 10. 상태천이도(STD)
Fig. 10. State Transition Diagram.

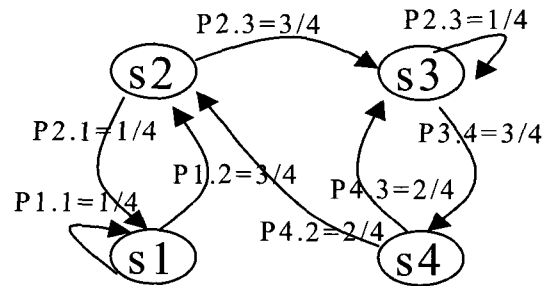


그림 11. 천이 조건 확률로 표시된 상태천이도
Fig. 11. Conditional Transition Probability of STD.

입력 값 중에 01, 00, 10 일 경우에만 천이가 일어나므로 3/4이다. 모든 상태사이에서 일어나는 천이 조건 확률은 그림 11에 나타나 있다. 그러나 천이 조건 확률은 입력 값의 변화만을 고려한 것이므로 스위칭 변화를 정확하게 나타낼 수 없다. 따라서 정확한 스위칭 변화를 계산하기 위해서는 현재 상태를 고려한 전체 천이 조건 확률을 다음과 같은 식(3)으로 나타낼 수 있다.

$$Gp_{i,j} = p_{ij} \cdot Ps_i \quad (3)$$

여기서 Ps_i 는 회로가 현재 상태 i 에 존재할 현재 상태확률이다. 그래서 현재상태확률을 구하기 위해서는 Markov Chain의 성질을 이용하여 구할 수 있다. 천이조건확률 $p_{i,j}$ 를 다음과 같은 matrix B로 나타낸다. Ps_i 를 구하기 위해서는 다음과 같은 관계를 이용하여 얻을 수 있다.

$$P = B^T \cdot P$$

$$\sum_{i=0}^{\text{상태수}} Ps_i = 1$$

그림12는 식(4)와 식(5)로부터 각각의 현재 상태확률을 구한 과정과, 현재 상태와 전체 천이확률을 표

시한 그림이다. 실제로 상태간에 상태값의 변화가 적게 상태를 할당함으로써 전력소모를 줄일 수 있다^[20]. 만약 현재상태 s_i 에서 다음상태 s_j 로 전체 천이 확률이 가장 높다고 가정하면 상태 s_i 와 s_j 간에 가장 빈번하게 스위칭 변화가 일어난다는 것을 알 수 있다. 이들 상태사이에 상태값의 변화를 최소화한다.

그림 13은 그림 12을 정수화하여 무게 인자(weight)상태천이도로 변형하였다. 이는 자신의 상태로 천이되는 것을 제외하고 두 상태 간에 천이는 하나로 나타내었다.

$$B = \begin{bmatrix} \frac{1}{4} & \frac{3}{4} & 0 & 0 \\ \frac{1}{4} & 0 & \frac{3}{4} & 0 \\ 0 & 0 & \frac{1}{4} & \frac{3}{4} \\ 0 & \frac{2}{4} & \frac{2}{4} & 0 \end{bmatrix} \begin{aligned} P_1 &= \frac{1}{4} P_1 + \frac{1}{4} P_2 \\ P_2 &= \frac{3}{4} P_1 + \frac{2}{4} P_4 \\ P_3 &= \frac{3}{4} P_2 + \frac{1}{4} P_3 + \frac{2}{4} \\ P_4 &= \frac{3}{4} P_3 \\ P_1 + P_2 + P_3 + P_4 &= 1 \end{aligned}$$

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

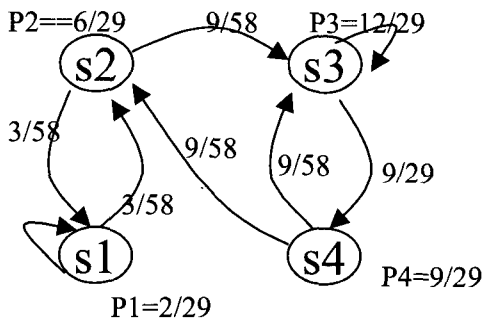


그림 12. 전체 천이 확률로 표시된 상태천이도
Fig. 12. Global Transition Probability of STD.

여기서 s_3 과 s_4 간에 무게인자가 27로 가장 높게 나왔다. 그러므로 두 상태 간에 상태값의 변화는 최소화하여서 각 상태간에 스위칭 확률을 작게 가지도록 상태 할당을 한다. 차례로 s_4 와 s_2 , s_3 와 s_2 사이도 상태 값의 변화를 줄이도록 상태할당을 한다.

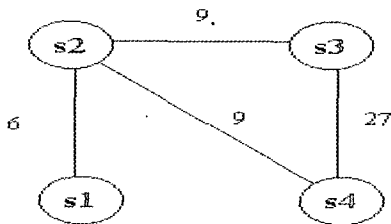


그림 13. 무게인자(weight)로 표시된 상태천이도
Fig. 13. Weight of STD.

2. 테스트 및 저전력을 고려한 상태할당 기술

메모리 소자간에 상호 의존성을 줄여서 상태할당을 한 m-블록 분할 방법에 상기의 상태천이 확률을 적용하여 소비전력을 감소시키는 상태할당 기술을 소개해 본다. 3장에서 제시한 m-블록 분할 알고리즘을 이용한다. 각 블록에 대한 블록천이도를 만든다. 블록천이도를 통해 블록천이 조건확률과 현재 블록에서 다음 블록의 천이되는 전체 천이확률을 계산한다. 이에 대한 전제로서 상태 천이 조건확률과 전체 천이확률이 계산되어 있어야한다. 이를 바탕으로 무게인자로 표시된 블록천이도를 구성한다. 최대 무게인자로 연결된 상태들을 포함하는 블록 분할을 선택한다. 이와 같이 선택한 블록 분할에 상태할당을 수행한다. 상태할당시 무게인자의 비율에 따라 상태들간에 각 상태변수에 대응되는 플립플롭의 변화 횟수가 최소화하도록 상태할당한다. 이것은 현재 블록에서 다음 블록으로 천이되는 무게 인자에 따라 블록들간에 Hamming distance를 최소화하는 것이다. 같은 블록에 있는 상태들의 상태 값은 블록 분할에 따라 같은 값을 가지고 있고 현재 블록내에 있는 상태에서 천이되는 다음 블록에 포함되어 있는 상태로의 무게인자를 고려한 것이므로 플립플롭간에 의존성을 유지하면서 스위칭의 변화를 최소화 하기 때문에 전력 소비를 최소화 할 수 있다.

예를 들어 4.1절에서 사용한 예에 따라 아래와 같이 3 분할로 구성할 수 있다.

$$(r_1, r'_1) = (\{\overline{S_1 S_2 S_4} : \overline{S_3}\}, \{\overline{S_1 S_2 S_3} : \overline{S_4}\})$$

$$(r_2, r'_2) = (\{\overline{S_1} : \overline{S_2 S_3 S_4}\}, \{\overline{S_1} : \overline{S_2 S_3 S_4}\})$$

$$(r_3, r'_3) = (\{\overline{S_1 S_3} : \overline{S_2 S_4}\}, \{\overline{S_1 S_4} : \overline{S_2 S_3}\})$$

분할(r_3, r'_3)이 선택되어 블록 분할 알고리즘에 따라 상태할당을 수행한다. 상태할당 후 s_3 과 s_4 는 각각 00, 11로 상태값의 차이는 2가 된다. 그러므로 그림 13에서 상태 s_3 과 s_4 간에 무게인자가 27로 가장 높게 나왔으므로 두 상태간에 상태값은 각각 00, 10으로 최소화하여 각 상태간에 스위칭 확률을 작게 가지도록 상태할당을 한다. 이와 같이 상태변수간에 상호 의존성을 줄여서 상태할당을 한 m-블록 방법과 무게인자를 바탕으로 테스트 및 전력 소모량을 최적화 할 수 있다.

본 논문에서 제안한 상태할당 방법을 평가하기 위해 Minimum Transition Function(MTF)를 정의하였다.

$$MTF = \sum_{i=1}^{\text{상태수}} Weight_i * HammingDistance(Ps_i, Ns_i)(6)$$

식(6)은 각 상태의 연결성에 따른 전체 상태값의 변화에 대한 합이다. 이는 상태할당 후 스위칭 정도를 비교 평가를 할 수 있다. 위의 결과에서 단지 상태변수 시간에 의존성만을 고려한 경우 MTF는 84이지만 스위칭 정도와 더불어 고려했을 때 MTF는 60으로 줄어든다. 이와 같이 MTF를 최소화하는 것은 전력 소비를 감소하는 것과 동일하다.

V. 실험결과

순차회로의 정보가 담겨있는 kiss 파일에 대하여 SIS 도구를 사용하여 본 연구의 알고리즘을 적용하여 상태할당을 한 후 전력 소비량을 측정하고 순차회로용 테스트 생성기와 고장 시뮬레이터를 통하여 성능평가를 한다. 상위정보를 포함한 게이트 수준에서의 테스트 및 저전력을 고려한 설계는 순차회로에 최적의 상태할당을 하는 알고리즘을 개발하여 이를 ISCAS 및 MCNC benchmark 회로에 적용하고 Berkeley 대학에서 개발한 논리 합성기인 SIS, 논리 합성 후 나온 blif 파일을 bench 파일로 바꾸는 스크립트 bliftobench, Illinois 대학에서 만든 순차회로용 테스트패턴 생성기인 HITEC을 실험하는 Tool로 사용하여 성능평가를 하였다.

표 2에서 보는 바와 같이 본 연구에서 제안하는 알고리즘을 상태할당에 이용하는 경우 기존의 상태할당 방법(One-hot, Random, Jedi, 2-block)에 비해 테스트 가능도가 현저히 향상되는 것을 알 수 있다. bbsse 회로는 다른 상태할당 방법에 비해 고장점검도가 높아졌으며, keyb 회로의 Mmpairs는 one-hot에 비해 0.74% 낮지만 jedi, random, 2-block방식에 비해 각각 5.38%, 1.84%, 3.22% 증진되었다. 상대적으로 모든 회로에 대해서 대체적으로 m-block 방식이 상태 의존성을 최소화하여 테스트 관점에서 효과적임을 보여준다.

표 2. 상태할당을 통한 고장점검도
Table 2. Fault Coverage using State Assignment.

Circuit	Ns/Nb	고장점검도(단위:%)					
		Jedi	Random	2-block	m-block	New-m block	One-hot
bbsse	16/4	98.24	90.66	98.12	98.85	98.26	97.80
mark1	16/4	98.10	94.47	98.85	98.85	98.83	97.12
keyb	19/5	91.50	95.04	93.66	96.88	93.94	97.62
s832	25/5	97.56	98.88	45.61	97.92	98.70	86.43
tbk	32/5	96.97	98.59	98.98	98.98	98.49	97.38

표 3. 상태할당을 통한 전력비교표
Table 3. Comparison of Power Consumption upon Different State Assignments.

Circuit	Ns/Nb	전력(단위 : μW)					
		Jedi	Random	2-block	m-block	New-m block	One-hot
bbsse	16/4	551.0	527.5	568.1	572.9	526.07	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	685.7	1362.2
s832	25/5	1161.4	1381.1	1181.0	1189.1	1176.2	2147.0
tbk	32/5	711.9	721.1	717.8	717.8	678.5	1051.4

표 3은 전력을 관측한 결과이며 역시 SIS Tool을 사용하였다. keyb인 경우 Mm pairs 보다 고장 점검도 면에서는 2.94% 감소 했지만 전력 소비면에서는 향상된 결과를 볼 수 있다. 대체적으로 영역 면에서 최적화된 jedi 방법에 근접 또는 좀더 나은 결과를 나타내었고 속도 면에서는 효율적인 속도를 나타내는 one hot방법에 대해 기대되는 만큼 높은 결과를 나타내었고 기존의 m-block 보다 나은 결과를 나타낸다. 결국 본 논문의 목적은 고장점검도 증진 및 저전력에 있었으며 실험을 통하여 다른 방법에 비해서 사용된 최적의 상태 할당보다 개선되었음을 확인하였다.

VI. 결론

본 논문에서는 m-블록 분할을 이용한 상태할당(또는 분할)을 통하여 테스트가능도 증가 및 저전력을 효과적으로 동시에 달성 할 수 있는 방법을 제안하였다. 고장시뮬레이션을 수행하여 다른 상태할당 기법에 비해 높거나 유사한 고장점검도를 나타냄을 알 수 있었고, 저전력 관점에서 볼 때도 효과적임을 알 수 있었다. 향후 계획으로는 scan을 기반으로 한 테스트가능도와 저전력 관계를 연구할 것이다.

참고 문헌

[1] M. Abramovici et al., "Digital Systems Testing and Testable Design", Computer Science Press, 1994.
[2] K. T. Cheng and V. D. Agrawal, "A Partial Scan Method for Sequential Circuits with Feedback", IEEE Trans. on Computers,

- Vol. 39, No. 4, pp. 544-548, April 1990.
- [3] E. Goldberg et al., "Theory and Algorithms for Hypercube Embedding", IEEE Trans on CAD., Vol. 17, pp. 472-488, June 1998.
- [4] Saeyang Yang and Maciej 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.
- [5] D. B. Armstrong, "A Programmed Algorithm for Assigning Internal Codes to Sequential Machines", IRE Trans. on Computers, Vol. EC-11, pp. 466-472, Aug. 1962.
- [6] G. De Micheli, "Symbolic Design of Combinational Sequential Logic Circuits Implemented by Two-level Logic Macros", IEEE TCAD, Vol. CAD-5, pp. 597-616, Oct. 1986.1.
- [7] S. Devadas et al., "MUSTANG: State assignment of finite state machines targeting multi-level logic implementations", IEEE TCAD. Vol. 7, pp. 1290-1300, Dec. 1988.
- [8] T. Villa et al., "Synthesis of FSMs: Logic Optimization. New York: Kluwer Academic", 1997.
- [9] K. T. Cheng, and V. D. Agrawal, "Design of Sequential Machines for Efficient Test Generation," in Proc. of ICCAD, pp. 358-361, 1989.
- [10] Surti P, Chao L.F, Tyagi A, "Low power FSM design using Huffman-style encoding", European Design and Test Conference", ED&TC Proceedings, pp. 521-525, 1997.
- [11] G.D. Hachtel, M. Hermida, A. Pardo, M. Pon-cino, F. Somenzi, "Re-Encoding Sequential Cir-cuits to Reduce Power Dissipation", proc. IEEE/ACM Intl. Conf. on CAD, pp. 70-73, 1994.
- [12] 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.
- [13] C.-Y. Tsui, M. Pedram, C.-A. Chen, A.M. De-spain, "Low Power State Assignment Targeting Two- and Multi-level Logic Implementations", proc. IEEE/ACM Intl. Conf. on CAD, pp.82-87, 1994.
- [14] V. Veeramachaneni, A. Tyagi, S. Rajgopal, "Re-encoding for Low Power State Assignment of FSMs", proc. IEEE Intl. Symposium on Low Power Design, pp. 173-178, April 1995.
- [15] Srikanth Rao M. and S. K. Nandy, "Power Minimization Using Control Generated Clocks", proc. Design Automatin Conference, pp. 794-799, June 2000.
- [16] R. K. Brayton, G. D. Hatchel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli, Logic Minimization Algorithms for VLSI Synthesis, Norwell, MA: Kluwer Academic, 1984.
- [17] 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.
- [18] K. T. Cheng, and V. D. Agrawal, "Design of Sequential Machines for Efficient Test Generation," in Proc. of ICCAD, pp. 358-361, 1989.
- [19] Z. Kohavi, Switching and Finite Automata Theory, McGraw-Hill, 1978.
- [20] T. Villa et al., Synthesis of FSMs: Logic Optimization. New York: Kluwer Academic, 1997.
- [21] L. Benini and G. De. Micheli, "State assignment for loww power dissipation", IEEE Journal of Solid-State Circuits, vol. 30. March 1995.
- [22] 구경희 and 조경록, "상태전이확률을 이용한 비동기 회로의 저전력 상태할당 알고리즘", 전자공학회 논문지 1999년 4월 제 36 권 C 편 제 4 호

저 자 소 개



박 성 주(정회원)

한양대학교 전자공학과 학사(1983), 금성사 소프트웨어개발(1983-1986). Univ. of Massachusetts 전기 및 컴퓨터공학과 박사(1992), IBM Microelectronics 연구스텝(1992-1994), 한양대학교 전자컴퓨터공학부 부교수(1995-현재). 관심분야는 테스트 합성, Built-In Self Test, Scan Design, ATPG, ASIC 설계, 고속 신호처리 시스템 설계, 그래프이론 등



이 현 빈(정회원)

한양대학교 전자컴퓨터공학 학사(2001), 한양대학교 컴퓨터공학과 석사(2003), 한양대학교 컴퓨터 공학과 박사 과정(2003-현재), 관심분야는 SoC 테스팅, ASIC 설계, 네트워크 시스템 설계



조 상 욱(정회원)

한양대학교 전자계산학과 학사(1997), 한양대학교 전자계산학과 석사(1999), 한양대학교 컴퓨터공학과 박사과정(2000-현재), 관심분야는 테스트 합성, Scan Design, VLSI 시스템 & 테스팅, ASIC 설계 등