

# 부분 스캔을 고려한 최적화된 상태 할당 기술 개발

조상욱(曹尙煜), \*양세양(梁世陽), 박성주(朴成柱)  
한양대학교 전자계산학과  
\*부산대학교 컴퓨터공학과

## Development of Optimized State Assignment Technique for Partial Scan Designs

Sangwook Cho, \*Saeyang Yang, Sungju Park  
Department of Computer Science & Engineering, Hanyang University  
\*Department of Computer Engineering, Pusan National University  
E-mail : swcho@cse.hanyang.ac.kr, \*syyang@hyowon.cc.pusan.ac.kr, parksj@cse.hanyang.ac.kr

### Abstract

The state assignment for a finite state machine greatly affects the delay, area, and testabilities of the sequential circuits. In order to minimize the dependencies among state variables, therefore possibly to reduce the length and number of feedback cycles, a new state assignment technique based on m-block partition is introduced in this paper. After the completion of state assignment and logic synthesis, partial scan design is performed to choose minimal number of scan flip-flops. Experiment shows drastic improvement in testabilities while preserving low area and delay overhead.

### I. 서론

칩의 집적도에 따라 설계점검 및 칩제작후의 기능점검 등은 더욱 더 어려운 문제로 부각되고 있다. 이러한 테스트 문제를 해결하기 위하여 체계적인 테스트설계 기술이 널리 개발되고 있다. 주문형 반도체를 테스트하는 방법으로서 많이 사용되는 스캔 방식은 회로내의 플립플롭이나 래치를 제어, 관찰하는 것을 용이하도록 회로 구조를 설계함으로써 테스트를 효과적으로 수행 할 수 있게 한다[1]. 그러나 회로내의 모든 메모리 소자들을 스캔 가능하게 하는 완전스캔(full scan) 방식은 성능 저하 및 칩 크기 증가와 같은 문제점을 내포하고 있다. 이를 해결하는 효율적인 방법으로서 회로내의 모든 메모리 소자들 대신 최소 부분만을 스캔 가능하게 하는 부분 스캔(partial scan) 방식이 널리 사용되고 있다[2]. 이와 같은 배경 하에서 테스트 기술을 고려하여 회로의 면적 최소화와 함께 최적화 대상으로서 상태 할당 기술을 연구하였다. 상태 할당 방법에는 One-hot, random, Jedi와 같은 방법이 개발되었다[3-4]. One-hot은 n 상태들로 구성되어진 유한상태기는 정확하게 n 플립플롭을 사용해서 할당되어지는 것이다. 이는 가장 많은 플립플롭을 필요로 하지만 다음 상태와 output logic을 줄이는 면에 있어서 효율적이다. Random은 logN 개의 플립플롭에 임의로 상태를 할당한다. 그리고 Jedi는 역시 logN 개의 플립플롭에 상

태를 할당하며 멀티 레벨을 위한 symbol encoding에 목표를 두고 있다. 이 방법은 심볼의 쌍들간에 무게를 부여하는 휴리스틱 방법을 적용하고 있다. 이는 또한 문자 절약면에서 효율적인 방법이다. 이러한 상태 할당기술은 구현되는 동기 순차회로의 면적 및 속도에 큰 영향을 미친다[5-9]. 상태 할당이 구현되는 회로의 면적뿐만 아니라 회로의 테스트 가능성에 지대한 영향을 미칠 수 있다는 연구가 발표되고 있다.[6,10]. 2-블록 분할(2-block partition)을 이용하여 순차테스트를 용이하게 하면서 면적최소화도 가능하게 하는 상태 할당 방법을 제안하였다[11]. 본 논문에서는 [11]에서의 방법을 일반화시켜 m-블록 분할을 이용한 상태 할당을 통하여 테스트가능도 및 면적최소화를 효과적으로 상대할 수 있는 방법과 이로부터 몇 가지 흥미 있는 이론적 결론들을 도출하고자 한다. 본 논문은 다음과 같이 구성되어 있다. 2절에서는 순차회로에 대한 상태 할당 방법을 살펴보고 3절에서는 부분 스캔을 고려한 최적의 상태 할당 기술 방법을 살펴보고 4절에서는 상태변수들의 의존성을 줄이는 방법을 소개한다. 5절에서는 상태 할당 기법의 벤치마크 회로에 대한 실험결과를 보이고 마지막 절에서는 결론을 기술한다.

### II. 순차회로에 대한 상태 할당 방법

순차회로에 대한 논리 합성에 있어서 주된 관심사 중의 하나는 유한상태기를 이용하여 추출한 상태전이표에서 각 상태들에 값을 할당하는 것이다. 그러나 상태들과 output 변수의 기능적인 의존도 뿐만 아니라 보다 작은 메모리 소자들만으로 회로가 되도록 할당을 하여야 하는데 실제로 상태 할당이 되어지는 경우의 수는 다양하게 나타남을 알 수 있다. 예를 들어 4개의 상태들로 구성된 유한상태기는 4! 개의 다른 인코딩을 가진다. 그 중에서 가장 최적으로 값이 할당된 상태들을 구하는 것은 NP-complete 문제이다. 그림 1에 있는 상태전이표로부터 그림 2의 상태 할당  $\alpha$ 에 의하여 회로를 구현하면 구현된 유한상태기내의 3개의 플립플롭 Y1(y1), Y2(y2), Y3(y3) (Y는 다음상태, y는 현재상태) 들 간에는 "상호간" 완전한 의존성이 생기게 된다. (그림 3). 또한 스캔-그래프에서 서로 다른 단순 사이클들이 얼마나 존재하는 지를 조사하면 알 수 있다[12]. 상태 할당 방법  $\alpha$ 에 의하여 구현되는 회로에서 이 스캔-그래프를 구성하면 2

부분 스캔을 고려한 최적화된 상태 할당 기술 개발

개의 단순 사이클이 존재함을 알 수 있다. 이와 같은 경우에는 3개의 플립플롭 가운데 최소한 하나의 플립플롭을 스캔 가능하게 하여야 부분 스캔에 의한 테스트가 이루어질 수 있다.

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

그림 1 상태천이표

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

그림 2 상태 할당  $\alpha$ ,  $\beta$

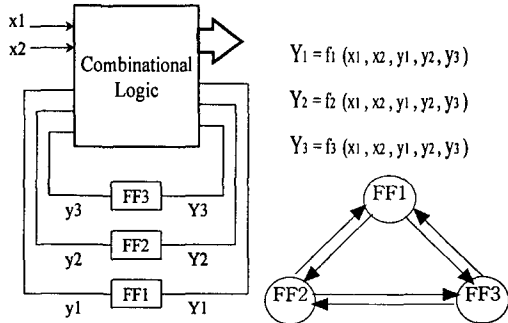
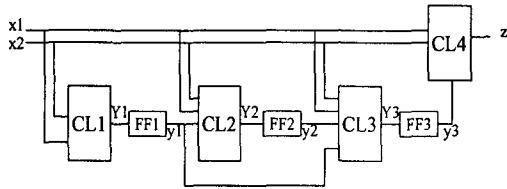


그림 3 상태 할당  $\alpha$ 에 의한 회로구조 및 스캔 그래프



$Y_1 = f_1(x_1, x_2)$

$Y_2 = f_2(x_1, x_2, y_1)$

$Y_3 = f_3(x_1, x_2, y_1, y_2)$



그림 4 상태 할당  $\beta$ 에 의한 회로구조 및 스캔 그래프

반면, 상태 할당을 방법  $\beta$ 와 같이 한다면 구현된 유한상태 기계의 3개의 플립플롭  $Y_1(y_1)$ ,  $Y_2(y_2)$ ,  $Y_3(y_3)$  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(그림 4).

표 1 현재상태에서 다음상태로의 비트별 변화

	a-e	a-c	a-d	.....	h-a	h-f	h-g	합계
상태 할당 $\alpha$	3	1	2	.....	1	1	3	52
상태 할당 $\beta$	1	1	1	.....	2	1	2	48

표 1과 같이 현재상태에서 다음상태로의 비트별 변화의 차이에서도 상태 할당  $\beta$ 가 6비트 더 적게 나타남을 알 수 있다. 따라서 상태 할당  $\beta$ 가 상태 할당  $\alpha$ 보다 부분 스캔 시에 회로 면적 증가를 억제시킴을 알 수 있다.

III. 부분스캔을 고려한 최적화된 상태 할당 기술 소개

우선 최적화된 상태 할당 기술을 위한 용어를 살펴보면 다음과 같다

- 정의 1] 분할(partition) : 교집합이 S인 연결되지 않은 부분집합
- 정의 2] m-블록분할 : 부분집합의 수가 m 인 경우
- 정의 3] i&s : 상태 s에서 입력 i에 의하여 천이 되는 다음상태
- 정의 4] 분할 쌍(partition pair) : (p1,p2) 는 각각의 입력에 대하여 p1의 같은 블록에 상태 s1과 s2가 같이 있으면 i&s1, i&s2가 p2에서 같은 블록에 있는 분할의 순서쌍(ordered pair)
- 정의 5] 의존성 : 다음상태 변수들의 값은 남아 있는 변수의 값에 의존하는 경우

예를 통하여 정의를 설명한다. 본 예제는 [13]에서 인용하였다. 그림 5 M의 상태천이표로부터 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 상태 할당과의 관계는 어떠한 지를 알아보자.

PS/ X1X2	NS				y1y2y3
	00	01	11	10	
a	a	c	d	f	a : 0 0 0
b	c	b	f	e	b : 0 1 1
c	a	b	f	d	c : 1 0 1
d	e	f	b	c	d : 0 0 1
e	e	d	c	b	e : 1 0 0
f	d	f	b	a	f : 1 1 1

그림 5 상태천이표 M 및 상태 할당

M을 그림 5와 같이 상태 할당 하였다고 하자. 각각의 상태 변수 하나는 해당 2-블록 분할( $y_1$ 은  $p_1 = (a, b, d : c, e, f)$ ,  $y_2$ 는  $p_2 = (a, c, d, e : b, f)$  그리고  $y_3$ 은  $p_3 = (a, e : b, d, c, f)$ )에 대응시킬 수 있음을 알 수 있다. 또한 두개 이상의 상태 변수들로부터는  $y_1$ 과  $y_2$ 로부터  $p_{12}=(a, d : b : c, e : f)$ 의 4-블록 분할을 대응시킬 수 있다. 즉, 상태 할당과 블록 분할은 구별 없이 같이 생각할 수 있는 것인데, 블록 분할로부터 순차회로내의 메모리 소자들간의 상호 의존성도 유추해 낼 수 있다. 이를 그림 6에서 살펴보면 다음상태 변수  $Y_3$ 은 현재상태 변수  $y_1$ 과  $y_2$  및 입력 변수에 의존적임을 알 수 있고 메모리소자 FF3와 FF2, FF3와 FF1간에는 상호 관계가 없다. 이와 같이 상태 할당 (또는 분할)을 통하여 구현된 모든 순차회로의 테스트를 용이하게 하는 요인은 분할 쌍의 결과이다. 이러한 분할을 이용한 상태 할당은 회로의 면적최소화면에서도 바람직한 결과를 낼 수 있다는 것이 알려져 있다[10,12]. 본 논문에서는 임의의 상태 할당(분할)을 가능한 한 분할쌍을 연속적으로 이용한다는 면에서는 [11]과 같으나 분할이 반드시 2-블록 분할이 되도록 하지 않고 m-블록 분할까지 모두 허용한다는 것이 [11]과 다르다.

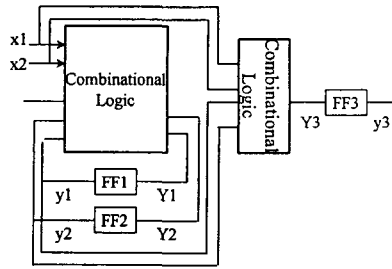


그림 6 분할쌍에 의한 회로 구조

#### IV. 상태변수들의 의존성을 줄이는 방법

상태 변수들의 의존성을 줄이는 방법을 설명하는데 필요한 정의를 우선 살펴보자

- 정의 6] Closed partition : P 분할되어진 m-블록 내에 있는 상태들에 대해 다음 상태들이 분할된 m-블록 내에 존재한 경우
- 정의 7]  $\pi(0)$  : closed partition 중 product를 통해 하나의 블록 내에 상태 하나만이 존재하는 경우
- 정의 8] predecessor partition ( $\gamma(ab)$ ) : 한 블록(ab), 나머지 상태들은 분리된 블록을 포함한 분할
- 정의 9] m-partition= $m(\gamma(ab))$  :  $r(ab)$ 에 의해 다음 상태들간의 product를 통해 유도되어진 다음 상태들을 포함하는 가장 작은 분할
- 정의 10] M-partition =  $M(\gamma)$  :  $m(ab)$ 에 의해 분할된 블록들에 포함되지 않는  $\gamma$ 들의 sum을 결정하는 가장 큰 분할
- 정의 13] Mm pairs :  $(M(\gamma_1), m(\gamma_1))$  는 각각의 입력에 대하여  $M(\gamma_1)$ 의 같은 블록에 상태  $s_1$ 과  $s_2$ 가 같이 있으면  $i \& s_1, i \& s_2$ 가  $m(\gamma_1)$ 에서 같은 블록에 있는 분할의 순서쌍(ordered pair)

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

상기 방법은 메모리 소자들간의 상호 의존성을 줄일 뿐만 아니라 상태 할당을 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 상태 할당을 통하여 메모리 소자들간의 의존성이 감소된 상태로 되는 것을 확인 할 수 있다. 다음은 그림 5의 상태천이표 M을 이용하여 본 논문에서 적용한 방법으로 구한 예를 보여주고 있다.

$y_1 y_2 y_3$	$p1=\{a,b,c; d,e,f\}$
a: 000	$P2=\{a,e; b,f; c, d\}$
b: 011	$p1 \cdot p2 = \{a; b; c; d; e; f\} = \Pi(0)$
c: 010	
d: 110	logical equation of assignment
e: 100	$Y_1 = f_1(x_1, y_1)$
f: 111	$Y_2 = f_2(x_1, x_2, y_3)$
	$Y_3 = f_3(x_1, x_2, y_2)$

그림 7 Mm pairs에서  $\pi(0)$ 를 만족하는 예

$y_1 y_2 y_3$	$p1=\{a,c,f; b; d,e\}$
a: 000	$P2=\{a,b,c,d,f; e\}$
b: 110	$p1 \cdot p2 = \{a,c,f; b; d; e\} \neq \Pi(0)$
c: 000	
d: 100	logical equation of assignment
e: 101	$Y_1 = f_1(x_1, y_1)$
f: 000	$Y_2 = f_2(x_1, x_2, y_1, y_2, y_3)$
	$Y_3 = f_3(x_1, x_2, y_1, y_2, y_3)$

그림 8 Mm pairs에서  $\pi(0)$ 를 만족하지 못하는 예

그림 7에서는 Y1 과 Y2는 상호 제한루프가 줄어들어 부분 스캔에 의해 테스트가 용이한 반면에 그림 8과 같이 상대적으로 메모리 소자들간의 상호 제한루프가 형성되므로 회로를 부분스캔이나 논 스캔에 의하여 테스트하는 것이 용이하지 못할 것이다. 또한 그림 9은 메모리 소자들간의 상호 의존성을 최소화하는 상태 할당 알고리즘을 보여주고 있다

```

1. encoding 할 n state를 위한  $k = \lceil \log_2 n \rceil$  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 9 상태 할당 알고리즘

#### V. 실험결과

부분 스캔을 위한 행위적 유한상태기 합성을 위하여 연구된 알고리즘의 효능을 벤치마크 회로들로 실험을 통하여 검증하였다 (표 2). 실험 방식은 One-hot, Random, Jedi 상태 할당 알고리즘을 수행하고 논리합성 단계를 거친 합성 결과와 본 논문에서의 상태 할당 알고리즘을 수행하고 같은 논리합성 단계를 거친 합성 결과 각각에 대하여 순차적 ATPG를 수행시켜 fault coverage를 구하여 비교하였다. 본 논문의 실험은 SIS라는 Berkeley 대학에서 개발한 논리 합성기와 Illinois 대학에서 만든 순차회로용 테스트패턴 생성기 HITEC을 사용하여 수행하였다

표 2 상태 할당을 통한 고장점검도

Circuit	Ns/Nb	고장 점검도(단위:%)			
		Jedi	Random	Mm pairs	One-hot
Mark1	16/4	98.10	94.47	98.84	97.12
bbsse	16/4	98.24	90.66	98.85	97.80
s1	20/5	1.97	0.88	1.06	51.69
s832	25/5	97.56	98.88	97.92	86.43
styr	30/5	1.36	29.32	1.28	87.15
keyb	19/5	91.50	95.04	96.88	97.62
tbk	32/5	96.97	98.59	98.98	97.38
s1494	48/6	96.81	96.34	94.87	56.97
scf	121/7	95.76	44.59	96.72	-

표 2에서 보이는 바와 같이 연구된 알고리즘을 상태 할당에

이용하는 경우 기존의 상태 할당 방법(One-hot, Random, Jedi)에 비해 테스트 가능성이 매우 향상되는 것을 알 수 있다. scf회로의 one-hot 상태 할당 방법은 SIS tool에서 상태 할당이 되지 않은 경우이다. tbk회로는 다른 상태 할당 방법에 비해 고장점검도가 높아졌으며, keyb 회로는 one-hot에 비해 0.74% 낮지만 jedi, random 방식에 비해 각각 5.38%, 1.84% 증진되었다.

표 3 상태 할당을 통한 면적 및 속도비교 ( ) : gate수

Circuit	Ns/Nb	상태 할당을 통한 면적 및 속도			
		Jedi	Random	Mm	One-hot
mark1	16/4	76096(44)	86304(51)	75168(41)	132704(68)
		18.72	14.97	16.22	12.26
bbsse	16/4	115536(64)	111600(65)	111824(62)	174000(84)
		19.65	24.55	17.84	18.43
keyb	19/5	182352(99)	312736(162)	177248(97)	381408(188)
		30.22	36.28	30.44	19.36
s1	20/5	152192(81)	198592(108)	165184(93)	369808(191)
		26.77	41.38	23.75	15.60
s832	25/5	262624(149)	304384(169)	261696(146)	438994(201)
		25.28	20.70	22.13	14.84
styr	30/5	392080(211)	505296(274)	361456(200)	600880(315)
		73.73	68.29	42.94	19.66
tbk	32/5	163792(86)	263088(140)	169360(89)	255664(131)
		28.32	46.82	34.79	12.88
s1494	48/6	521536(264)	548448(284)	481632(261)	749360(366)
		58.52	73.99	55.02	18.21
scf	121/7	655168(364)	862576(465)	645424(358)	-
		51.97	45.62	45.87	-

표 3 는 면적 및 속도를 관측한 결과이다. 표에서 보는 바와 같이 면적면에서 Jedi 방법에 근접 또는 좀더 나은 결과를 나타내고, 속도면에서는 One-hot 방식보다는 낮지만 Jedi 보다는 나은 결과를 나타냄을 알 수 있다. 이는 이미 앞에서 언급되었듯이 제안된 상태 할당 알고리즘에 의하여 유한상태기의 상태변수간의 의존성을 최소화할 수 있기 때문이다. 표 4는 SIS에서 생성된 Blif 파일을 Bench 파일로 변환하여 구조분석 및 테스트 가능성도 통합에 의한 부분스캔 기술을 적용한 후 고장 점검도를 구했다. 실험결과 Jedi에 비해 나은 결과보다 본 논문이 제안한 방법에서도 향상된 결과가 있음을 알 수 있었다.

표 4 구조분석 및 테스트 가능성도 통합에 의한 실험 결과

Circuit	Ns/Nb	고장점검도(단위:%)					
		Jedi			Mm pairs		
		no scan	scan1	scan2	no scan	scan1	scan2
s832	25/5	97.56	99.18	99.84	97.92	99.20	99.20
styr	30/5	1.36	1.90	98.78	1.28	1.41	99.74
tbk	32/5	96.97	98.99	98.99	98.98	99.23	99.49
s1494	48/6	96.81	98.60	99.20	94.87	98.29	99.36
scf	121/7	95.76	80.55	99.55	96.72	99.15	99.60

### V. 결론 및 추후계획

본 논문에서는 유한상태기의 테스트를 부분스캔 방법으로 수행하는 전제하에서 이를 기존의 면적 최소화와 함께 최적화할 대상으로서 고려하는 상태 할당 기술을 연구하였다. 고장점검도를 증진시키는데 목표를 두고 있다. 즉 연속적인 m-블록 분할을 이용한 상태 할당(또는 분할)을 통하여 테스트 가능성도 증가 및 면적최소화를 효과적으로 동시에 달성

할 수 있는 방법을 제안하였다. 다른 상태 할당 기법에 비해 고장 점검도가 개선되었고 영역과 속도 관점에서도 효과적임을 알 수 있었다. 본 논문에서 개발한 기술이 고집적도의 주문형 반도체 테스트 비용 절감에 커다란 기여를 할 수 있으리라 확신한다.

### 참고문헌

- [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] Randy H. Katz, Contemporary Logic Design, University of California Benjamin Cummings/Addison Wesley Publishing Company, 1993
- [4] 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.
- [5] E. Goldberg et al., "Theory and Algorithms for Hypercube Embedding," IEEE Trans on CAD., Vol. 17, pp. 472-488, June 1998.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] T. Villa et al., Synthesis of FSMs: Logic Optimization. New York: Kluwer Academic, 1997.
- [11] K. T. Cheng, and V. D. Agrawal, "Design of Sequential Machines for Efficient Test Generation," in Proc. of ICCAD, pp. 358-361, 1989.
- [12] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. L. Sangiovanni-Vincentelli, "Logic Minimization Algorithms for VLSI Synthesis", Norwell, MA: Kluwer Academic, 1984.
- [13] Z. Kohavi, "Switching and Finite Automata Theory", McGraw-Hill, 1978..