論文 90-27-11-6

CMOS 회로의 Stuck-open 고장검출을 위한 로보스트 테스트 생성

(Robust Test Generation for Stuck-Open Faults in CMOS Circuits)

鄭 峻 模* 林 寅 七*

(Jun Mo Jung and In Chil Lim)

要 約

본 논문에서는 CMOS 회로의 stuck-open 고장 검출을 위한 로보스트(robust)테스트 생성방법을 제안한다. CMOS 회로에 대한 입력 벡터들간의 비트(bit)위치와 해밍중(Hamming weight)의 관계를 고려하여 초기화 패턴과 테스트 패턴을 구함으로써 stuck-open 고장검출을 위한 테스트 생성 시간을 감소시킬 수 있으며, 고장검출을 어렵게하게 하는 입력변이지연(input transition skew)의 문제를 해결하고, 테스트 시이퀀스의 수를 최소화시킨다. 또한 회로에 인가할 초기화 패턴과 테스트 패턴간의 해밍거리(hamming distance)를 고려하여 테스트 시이퀀스를 배열하므로써 테스트 시이퀀스의 수를 감소시킨다.

Abstract

In this paper robust test generation for stuck-open faults in CMOS circuits is proposed. By obtaining initialization patterns and test patterns using the relationship of bit position and Hamming weight among input vectors for CMOS circuit test generation time for stuck-open faults can be reduced, and the problem of input transition skew which make fault detection difficult is solved,and the number of test sequences are minimized.Also the number of test sequences is reduced by arranging test sequences using Hamming distance between initialization patterns and test patterns for circuit.

I. 서 론

최근 VLSI 기술의 발전으로 회로의 집적도가 크게 증가하여 이에 따라 회로의 테스트가 더욱 중요한 문제로 대두되고 있다.

현재 회로 구성에 많이 사용되고 있는 CMOS 소자는 출력단의 부하용량(load capacitance)때문에 전하저장 기능을 가지며 이 부하용량은 CMOS 회로에 stuck-open 고장이 일어날 때 회로의 출력값을 전상태 값(previous value)으로 유지하게 하므로, 다음 입력이 인가되더라도 회로의 출력은 전상태값을 갖게 된다. 그러므로 stuck-open 고장을 검출하기 위해서는 한개의 테스트 패턴으로는 고장검출이 곤란하며, 초기화 패턴과 테스트 패턴이 연속적으로 배

*正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)
接受日字 : 1990年 4月 14日

열되는 테스트 시이퀀스가 필요하게 된다.[1-2]

CMOS 회로에서는 stuck-open 고장에 대한 테스트 시이퀀스로써 입출력선에서 일어나는 stuck-at 고장검출이 가능하므로, 현재 CMOS를 테스트하는 데는 stuck-open 고장의 검출이 가장 중요한 관심의 대상이 되고 있다.

임의의 테스트 시이퀀스로 회로를 테스트할 경우 입력변이 지연에 의하여 고장검출이 안되는 문제가 발생할 수 있는데, 이와같은 입력변이지연의 영향을 받지않는 테스트 시이퀀스를 생성하는 것을 로보스트 테스트 생성이라고 한다.[3,7]

Jain[5]은 CMOS 회로에 대하여 논리모델(logic model)을 실현하고 이 논리모델에 대하여 D알고리듬을 적용하는 테스트 생성방법을 제안하였지만 이 방법은 테스트 생성시간이 많이 소요되기 때문에 대규모 회로인 경우에 불리하며, 이 방법에 의해 생성된 테스트 시이퀀스로 회로를 테스트 할 경우 로보스트 테스트가 되지 않는다. Bate[4]는 CMOS 회로에 대하여 오일러 사이클(eulerian cycle)을 이용함으로써 빠른 시간내에 테스트 생성하는 방법을 제안하였지만, 이 방법은 입력변수의 수가 n일때 테스트 시이퀀스의 수가 $n2^n$로 되어 입력변수가 많아질 경우 테스트 시간이 크게 증가하는 단점이 있다. Sherlekar[3]는 로보스트한 테스트 생성방법을 제안하였지만, 이 방법은 각 stuck-open 고장마다 카르노도(karnaughmap)를 사용해야 하므로 테스트 생성시간이 많이 필요한 다중패턴 테스트(multipattern test)방식이 된다. 또한 이 방법은 입력변수가 많을 경우 카르노도의 사용이 곤란하게 되므로 테스트 생성에 제약을 받게 된다. 현재 CMOS 회로에 대한 보다 효율적인 테스트 생성방법이 요구되고 있으며 입력변이지연에 영향을 받지 않는 로보스트 테스트 시이퀀스를 구하는 것이 중요하다. 또한 테스트 시간을 줄이기 위하여 가장 적은 수의 테스트 시이퀀스를 생성하는 것이 필요하다.

본 논문에서는 CMOS 회로의 stuck-open 고장검출을 위한 로보스트 테스트 생성방법을 제안한다. CMOS 회로 출력에 대한 진리표에서 입력 벡터들간의 비트 위치와 해밍중의 관계를 고려하여 초기화 패턴과 테스트 패턴을 구함으로써 stuck-open고장을 위한 테스트 생성시간을 감소시킬 수 있으며, 고장검출을 어렵게 하게 하는 입력변이지연의 문제를 해결하고, 테스트 시이퀀스의 수를 최소화시킨다. 그리고 초기화패턴과 테스트 패턴간의 해밍거리를 고려하여 테스트 시이퀀스를 배열하므로써 테스트 시이퀀스의 수를 감소시킨다.

Ⅱ. 입력변이지연을 고려한 로보스트 테스트 생성

임의의 테스트 시이퀀스로 회로를 테스트할 경우 입력변이지연에 의하여 고장검출이 안되는 문제가 발생할 수 있다. 그러므로 회로에 대한 테스트 생성을 할 때 이러한 문제를 방지하기 위하여 로보스트한 테스트 시이퀀스를 생성해야한다.

[정의 1] 함수의 출력값이 '1'로 나오게 하는 입력 벡터를 α라 하고, '0'으로 나오게하는 입력 벡터를 β라 한다. 또한 입력 벡터의 i번째를 각각 α_i, β_i라 하고, i번째 임의의 입력 벡터를 T_i라한다.

그림 1의 b회로에 대하여 F_0의 보수값 f_0는 AND/OR 게이트(gate)만으로 구성되어 있다. $f_0=(x+z)(yw'+x')$출력에 대해 테스트 생성을 하기 위하여 초기화 패턴 $\beta=(0010)$와 테스트 패턴 $\alpha=(0111)$을 인가할 경우, β는 PMOS 회로망을 활성화 하는 입력 벡터가 되며 α는 NMOS 회로망을 활성화하는 입력 벡터가 된다.

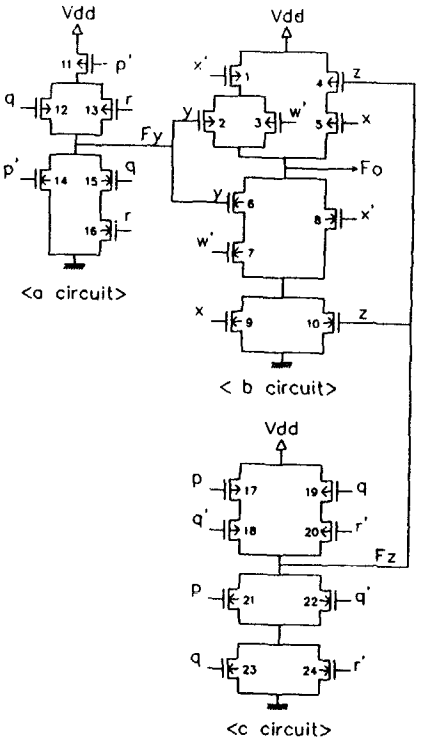


그림 1. $F_0=\{(x+p'q+q'r)(pq'w'+pr'w'+x')\}'=f_0'$ 인 CMOS 회로

Fig. 1. A CMOS circuit of $F_0=\{(x+p'q+q'r)(pq'w'+pr'w'+x')\}'=f_0'$.

테스트시, 위와같은 초기화 패턴과 테스트 패턴을 회로에 인가하였을 때 다음과 같은 입력변이지연이 일어날 수 있다.

A : (0100)→(0101)→(0111)
B : (0100)→(0110)→(0111)

입력이 A와 같이 변할 때 f_0에 대한 출력값은 $\beta \rightarrow \beta \rightarrow \alpha$로 되어 중간 초기화 패턴에 대한 출력값이 초음 초기화 패턴에 대한 출력값과 같아지므로 A와 같은 입력 변이지연이 일어날 때는 고장검출이 가능하다. 그러나 B와 같이 f_0의 출력값이 $\beta \rightarrow \alpha \rightarrow \alpha$로 변하게 되면 중간 초기화 패턴에 대한 출력값이 처음 초기화 패턴에 대한 출력값과 다르게 되므로 고장검출을 할 수 없게 된다.

[정의 2] 입력 벡터 T_i가 '1'을 가지는 비트 위치마다 입력 벡터 T_j가 '1'을 가지고 있으며, T_j의 해밍중이 T_i의 해밍중 보다 같거나 크면 T_j는 T_i를 포함(cover)한다 라고 하며 $T_j \geq T_i$로 표시한다.

예를 들어 T_i=1010, T_j=1011 일 때, T_j는 T_i를 포함한다고 하며, $T_j \geq T_i$ 또는 1011≥1010로 표기한다. 또한 T_i=1001, T_j=0101일 때는 서로 포함하지 않게 된다.

어떤 함수의 입력 변수가 $X_1, X_2, \cdots, X_n$이고, f(X, Xi)=X_i'를 만족시키는 X가 존재할 때, 즉. $X_1, \cdots, X_{i-1}$ $X_{i+1}, \cdots X_n$의 값들에 의해 위의 식이 성립할 때, 그 함수를 이루는 데는 변수 X_i'가 필요하게 된다.

그림 1의 회로를 테스트하기 위해서는 먼저 회로 b에 대한 테스트를 한다. 함수에 필요한 변수들에 의하여 이루어진 진리표를 확장 진리표(extended truth table)라 하는데, 회로 출력 f_0에 대한 확장진리표를 구하는 데 필요한 변수는 w', x, x', y, z가 된다. F_0의 변수들에 대해 확장 진리표를 구하지 않고 f_0에 대한 변수들에 의해 구하는 이유는 f_0에 필요한 변수에 대해 테스트 시이퀀스를 구하면 보수값을 취하지 않고 직접 CMOS회로에 인가시킬 수 있기 때문이다. 회로 출력에 필요한 변수들에 의해 확장 진리표를 구하면 표 1과 같다.

회로 b에서 함수 f_0에 대한 테스트 패턴 β는 β중에서 임의의 테스트 패턴 β_i, β_j가 $\beta_i \geq \beta_j$를 만족하면 β_i를 테스트 패턴으로 선택한다.

AND/OR 게이트 만으로 이루어진 회로에서는 어떤 입력 벡터 T_i, T_j가 있을때, $T_i \geq T_j$가 성립하면 $F(T_i) \geq F(T_j)$가 성립한다. 또한 이러한 회로에 s-a-1 또는 s-a-1이 발생하더라도 이와같은 성질[6]은 성립한다.

표 1. b회로에 대한 확장진리표
Table 1. Expanded truth table for b circuit.

w'	x	x'	y	z	f_0	F_0
1	0	1	0	0	0	1
1	0	1	0	1	1	0
1	0	1	1	0	0	1
1	0	1	1	1	1	0
1	1	0	0	0	0	1
1	1	0	0	1	0	1
1	1	0	1	0	1	0
1	1	0	1	1	1	0
0	0	1	0	0	0	1
0	0	1	0	1	1	0
0	0	1	1	0	0	1
0	0	1	1	1	1	0
0	1	0	0	0	0	1
0	1	0	0	1	0	1
0	1	0	1	0	0	1
0	1	0	1	1	0	1

예를들어 β에 해당하는 테스트 패턴이 β_i=(01011), β_j=(01001)이 있을 때, $\beta_i \geq \beta_j$의 관계가 성립한다. 이 관계는 위에서 설명한 AND/OR 회로의 성질때문에 β_i에 의해 검출되지 않는 고장은 β_j에 의해 검출할 수 없다는 것을 의미한다. 그러므로 β_j를 제외한 β_i만을 테스트 패턴으로 사용한다. f_0에 대한 β는 결국 PMOS 회로망을 활성화시키는 입력 벡터가 되는데, 포함관계가 성립하는 β중에서 해밍중이 최대인 입력 벡터는 단일 경로를 활성화하므로 테스트 패턴으로 선택된다.

또한 같은 방법으로 테스트 패턴 α는 α에 해당하는 입력벡터중 임의의 패턴 α_i, α_j가 $\alpha_i \geq \alpha_j$를 만족하면 α_i는 테스트 패턴으로 사용하지 않고 α_j만을 테스트 패턴으로 사용한다.

b회로의 f_0에 대하여 테스트 패턴중 포함관계에 의해 제거되지 않는 테스트 패턴의 부류가 3가지가 된다. 각 테스트 패턴들과 비교한 결과 제거되지 않는 테스트 패턴 즉, 서로 포함관계가 없는 β는 10110, 11001, 01011이 되고, α는 11010, 00101이 된다.

CMOS 회로의 stuck-open 고장을 검출하기 위한 테스트 시이퀀스는 테스트 패턴과 초기화 패턴으로 이루어진다.

테스트 패턴 β에 대한 초기화 패턴은 서로 포함관계가 있는 α중에서 해밍중이 가장 큰 입력벡터들인 10111, 11011로 구해지며, 표 2에 나타냈다.

그 이유는 초기화 패턴 α는 NMOS회로망에서 최대

표 2. (a) 테스트 패턴 (b) 초기화 패턴 (c) 고장 검출
Table 2. (a) Test patterns, (b) Initialization patterns, (c) Fault detection.

	w'	x	x'	y	z	f
a	1	0	1	1	0	0
b	1	1	0	0	1	0
c	0	1	0	1	1	0
d	0	0	1	0	1	1
e	1	1	0	1	0	1

(a)

	w'	x	x'	y	z	f
g	1	0	1	1	1	1
h	1	1	0	1	1	1
i	0	0	1	0	0	0
j	0	1	0	0	0	0

(b)

	검출되는 고장	
a	4,5	PMOS
b	1,2	PMOS
c	1,3	PMOS
d	8,10	NMOS
e	6,7,9	NMOS

(c)

의 경로를 활성화하고 서로 포함관계가 있는 α중에서 해밍중이 가장 크며, α에 대한 테스트 패턴도 서로 포함관계가 있는 β중에서 해밍중이 가장 큰 β이기 때문에 테스트시, 테스트 패턴β로 옮겨가기 직전까지 변수들간에 입력변이 지연이 일어나도 그 함수에 대한 출력은 초기값 α에 대한 출력값을 유지하기 때문이다.

같은 이유로 테스트 패턴 α에 대한 초기화 패턴은 서로 포함관계가 없는 β중에서 해밍중이 가장 작은 입력 벡터인 00100, 01000로 구해진다.

b회로에 입력되는 a회로를 테스트 할 때는 a회로의 함수가 $Fy(p, q, r) = (p' + qr)' = fy'$ 이므로 fy를 이루는데 필요한 변수는 p', q, r이 된다. 그러므로 확장 진리표는 b회로에 대한 방법과 같이 구해지며, a회로에 대한 테스트 패턴과 초기화 패턴은 표 3에 나타낸 것과 같이 구해진다.

또한 c회로를 테스트 할 때는 회로의 함수가 $Fz(p, q, r) = \{(p+q')(q+r')\}' = fz'$이므로 fz를 이루는데 필요한 변수를 구하면 p, q, q', r'가 된다.

그러므로 확장 진리표는 b회로에 대한 방법과 같이 구하면 되고 테스트 패턴과 초기화 패턴은 표4와 같이 구해진다. 초기화 패턴과 테스트 패턴을 구하는 알고리듬은 아래와 같다.

[단계 1] 회로 함수에 필요한 입력 변수를 결정하여 확장 진리표를 구한다.

표 3. (a) 테스트 패턴 (b) 초기화 패턴 (c) 고장 검출(a 회로)
Table 3. (a) Test patterns, (b) Initialization patterns, (c) Fault detection(a circuit).

	p'	q	r	fy
k	0	0	1	0
l	0	1	0	0
m	1	0	0	1
n	0	1	1	1

(a)

	p'	q	r	fy
o	1	1	1	1
p	0	0	0	0

(b)

	검출되는 고장	
k	11, 12	PMOS
l	11, 13	PMOS
m	14	NMOS
n	15, 16	NMOS

(c)

표 4. (a) 테스트 패턴 (b) 초기화 패턴 (c) 고장검출(c 회로)
Table 4. (a) Test patterns, (b) Initialization patterns, (c) Fault detection(c circuit).

	p	q	q'	r'	fz
g	0	1	0	1	0
r	1	0	1	0	0
s	0	0	1	1	1
t	1	1	0	0	1

(a)

	p	q	q'	r'	fz
u	1	0	1	1	1
v	1	1	0	1	1
w	0	0	1	0	0
x	0	1	0	0	0

(b)

	검출되는 고장(s-op)	
q	17, 18	PMOS
r	19, 20	PMOS
s	22, 24	NMOS
t	21, 23	NMOS

(c)

[단계 2] β중 $\beta_i \geq \beta_j$ 관계에 있는 β_i 가운데 해밍중이 가장 큰 β_i를 테스트 패턴으로 선택한다.

[단계 3] α중 $\alpha_i \geq \alpha_j$ 관계에 있는 α_j 가운데 해밍중이 가장 작은 α_j를 테스트 패턴으로 선택한다.

[단계 4] 테스트 패턴을 제외한 β중 $\beta_i \geq \beta_j$ 관계에 있는 β_j가운데 해밍중이 가장 작은 β_j를 α에 대한 초기화 패턴으로 선택한다.

[단계 5] 테스트 패턴을 제외한 α중 $\alpha_i \geq \alpha_j$관계에 있는 α_i가운데 해밍중이 가장 큰 α_i를 β에 대한 초기화 패턴으로 한다.

회로b를 테스트 할 때 주입력(primary input)에 해당하지 않는 입력y와 z는 p, q, r에 의해 구동이 된다. b회로에 입력되는 초기화 패턴과 테스트 패턴은 위에서 구해진 b회로 자체의 입력에 의한것 뿐만 아니라 p, q, r에 의한 입력변이지연에 의한 문제도 방지되어야 한다. 여기서 y, z에 대한 입력을 살펴 보면 y, z의 변화를(초기화 패턴→테스트 패턴)의 관계로 표현할 때, y, z는 b회로를 테스트 할 때(00→10), (00→01), (11→10), (11→01), (11→11)로 변해야 함을 알 수 있다.

b회로에 대한 입력 y, z는 Fy, Fz의 값이므로 p, q, r에 대한 Fy, Fz의 출력중 입력변이지연에 의한 문제를 방지하는 테스트 시이퀀스를 구해야 한다. p, q, r에 대한 Fy, Fz의 값을 살펴보면 표 5와 같다.

표 5. Fy, Fz에 대한 진리표
Table 5. Truth table for Fy, Fz.

p	q	r	Fy	Fz
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	0	0

그림 2는 Fy, Fz의 상태에 따른 입력 벡터 p, q, r의 변화를 나타낸 것이다. 예를 들어 Fy, Fz가 (00) 상태에서 (10) 상태로 변할 때, p, q, r은 초기화 패턴 (000)과 테스트 패턴 (100)을 선택하거나, 초기화 패턴 (111)과 테스트 패턴 (110) 선택하면 된다. 그 이유는 p, q, r에 대한 테스트 패턴과 초기화 패턴사이의 해밍거리가 '1' 차이가 나는 입력 벡터를 선택하면 y, z에 대한 입력변이지연 문제는 해결할 수 있기 때문이다. 한편 Fy의 출력값이 b회로를 통과해야 하기 때문에 Fy를 활성화하는 b의 입력은 Boolean difference에 의해 구해진다. 즉, $dfo/dy = x\,w' = 1$ 이므로

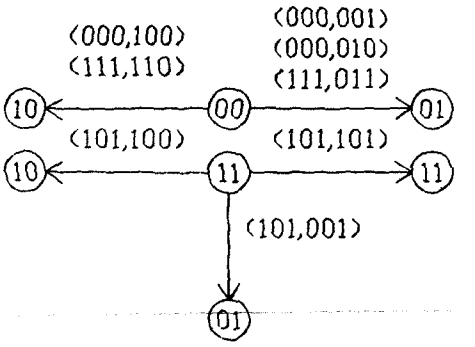


그림 2. Fy, Fz의 상태 다이어그램
Fig. 2. State diagram of Fy, Fz.

Fy의 출력값을 최종출력 F_o로 전달하는 데는 x=1, w=0가 필요하다. 또한 Fz의 출력값이 b회로를 통과해야 하기 때문에 Fz를 활성화 하는 b의 입력은 $dfo/dz = x' = 1$이므로 x=0가 필요하게된다.

Ⅲ. 테스트 시이퀀스의 배열

a회로를 테스트할 때는 초기화 패턴이 $f_y=1$, $f_y=0$에대해 각각 한개씩만이 존재함으로 초기화 패턴에 따른 각 보수의 출력값에 해당하는 테스트 패턴이 필요하고 테스트 시이퀀스는 p→m→ p→n→o→ k→o→l과 같이 된다. c회로를 테스트할 때는 초기화 패턴이 $f_z=1$, $f_z=0$에 대하여 각각 두개씩 존재하므로 테스트 패턴과 해밍거리가 최소인 초기화 패턴을 선택해야 한다. 그러므로 테스트 시이퀀스는 v→q→u→ r→ w→ s→ x→t와 같다. 그리고 b회로를 테스트할 때는 초기화 패턴이 $f_o=1$, $f_o=0$에 대해 각각 두개씩 존재하므로 테스트 패턴과 해밍거리가 최소인 초기화 패턴을 선택하여야 한다. 그러므로 테스트 시이퀀스는 g→ a→ h→ b→ h→ c→ i→ d→ j→ e와 같다.

한편 a회로와 같이 테스트 패턴β와 α사이에 해밍거리가 '1' 차이인 경우 α를 β의 초기화 패턴으로 겸용을 하여 사용해도 로보스트 테스트가 된다. 이와같은 경우는 회로의 규모가 커질수록 많아져서 테스트 시이퀀스의 수를 감소시킬 수 있다. 그러므로 p→ m→ p→ n→ k→ o→l와 같이 배열할 수 있으므로 테스트 시이퀀스가 감소하게 된다.

결국, CMOS 회로에 대한 테스트 패턴과 초기화 패턴간의 비트 위치와 해밍중의 비교만으로 간편하게 테스트 시이퀀스를 구할 수 있으며, 위의 알고리듬에 의한 테스트 시이퀀스는 회로의 입력변이지연

에 의한 문제를 막을 수 있고, 전체 테스트 시이퀀스의 수를 줄일 수 있게 되어 효과적인 테스트를 이룰 수 있다.

최종적인 초기화 패턴과 테스트 패턴은 표 6에 나타냈다.

표 6. 테스트 시이퀀스
Table 6. Test sequences.

w	x	p	q	r
0	0	1	0	1
0	0	1	0	0
0	1	1	0	1
0	1	0	0	1
0	1	1	0	1
1	1	1	0	1
1	0	1	1	1
1	0	0	1	1
1	1	1	1	1
0	1	1	1	0
0	1	1	0	0
0	1	0	0	0
0	1	1	0	0
0	1	1	1	1
0	1	1	0	1
0	1	0	1	1
0	1	1	1	0
-	0	1	1	0
-	0	0	1	0
-	0	1	0	0
-	0	1	0	1
-	0	0	0	1
-	0	0	0	0
-	0	0	1	1
-	0	1	1	1

Ⅳ. 결과 및 고찰

표 7에서 A는 본 논문의 회로에 대해 적용한 결과이고, B는 참고 논문[8]에서 사용된 회로에 대해 적용한 결과이다.

테스트 생성 알고리듬은 NP-complete한 문제이므로 알고리듬간의 비교는 일반적으로 테스트 생성 방법과 테스트 패턴의 수에 대한 비교와 검토로써 이루어진다.[1]-[8]

Sherlekar가 제안한 테스트 생성 방법은 한 개의 고장마다 모든 변수에 대한 진리표를 구해야하고 다중패턴 테스트가 필요하다는 단점이 있는 반면 본

표 7. 적용 결과
Table 7. Result of application.

	No. of test sequences A	B	Robust 테스트
Sherlekar	48	27	YES
Bate	161	65	YES
Jain	32	18	NO
본 논문	25	11	YES

논문에서는 그 단의 모든 고장에 대하여 한 개의 진리표만으로도 그 단의 모든 고장을 검출할 수 있는 테스트 패턴을 생성하였고, 한 개의 고장에 대하여 다중패턴 테스트가 아닌 한 개의 테스트 패턴과 한 개의 초기화 패턴으로 구성된 테스트 시이퀀스를 사용하였다.

Bate가 제안한 테스트 생성 방법은 테스트 패턴이 $n2^n$에 비례하여 필요하게 되므로 입력변수 n의 갯수가 커질수록 급격하게 테스트 패턴이 증가된다. 그러므로 표 7에서 보는 바와 같이 본 논문에 의한 테스트 패턴의 수는 n이 커짐에 따라 더욱 유리하게 됨을 알 수 있다.

Jain이 제안한 테스트 생성방법은 테스트 생성을 위하여 CMOS회로에 대하여 논리모델을 실현하고 이 논리모델에 대하여 D알고리듬을 적용하여 테스트 생성을 하기 때문에 많은 시간이 소요되는 반면에, 본 논문에서는 회로에 대하여 모델링이 전혀 필요없고 복잡한 D알고리듬을 적용하지 않으며 각 단의 회로 출력만 주어지면 테스트 패턴의 비트 위치와 해밍중의 관계만 고려하여 테스트 패턴을 생성하게 되므로 테스트 생성을 빠르게 할 수 있었다. 또한 회로의 고장 검출을 불가능하게 하는 입력변이 지연의 문제를 해결할 수 있었다. 본 논문의 테스트 생성방법을 제시한 예에 적용할 경우 기존의 테스트 생성방법으로 구한 테스트 시이퀀스의 수에 비해 15.6%에서 83.2%의 감소를 보였다.

Ⅴ. 결 론

본 논문에서는 CMOS 회로의 stuck-open 고장검출에 대한 로보스트 테스트 생성방법을 제안하였다. CMOS 회로 출력에 대한 진리표에서 입력 벡터들간의 비트위치와 해밍중의 관계를 고려하여 초기화 패턴과 테스트 패턴을 구하므로써 stuck-open 고장을 위한 테스트 생성시간을 감소시킬 수 있으며, 고장검

출을 어렵게 하게하는 입력변이지연의 문제를 해결하였고, 또한 테스트 시이퀀스의 수를 감소시켰다. 그리고 회로에 인가할 초기화 패턴과 테스트 패턴간의 해밍거리를 고려하여 테스트 시이퀀스를 배열함으로써 테스트 시이퀀스의 수를 최소화하였다.

이 방법은 입력변이지연의 문제를 해결함으로써 테스트시, 테스트 시이퀀스의 빠른 인가가 가능하게 되며, 테스트 시이퀀스를 최소화하므로 본 논문에서 제시한 예에 적용할 경우 기존의 테스트 생성방법에 비해 15.6%~83.2%의 테스트 시이퀀스의 감소를 보였다. 그러므로 현재 테스트 공정이 반도체 생산 공정중 큰 부분을 차지하고 있으므로 본 논문에서 제안한 방법을 이용할 경우, CMOS 회로에 대한 효과적인 테스트 생성과 테스트를 수행할 수 있을 것이다. 또한 본 논문에 의한 방법은 Domino CMOS와 같은 Dynamic CMOS회로에 적용이 가능하다.

參 考 文 獻

[1] S.M. Reddy and M.K. Reddy, "Testable Realizations for FET Stuck-open Faults in CMOS Combinational Logic Circuits," *IEEE Trans. Comput.* vol. C-35, no. 8, pp. 742-754, Aug. 1986.

[2] R.L. Wadsack, "Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits," BSTJ., vol. 57, pp. 1449-1474, May-June 1978.

[3] S.D. Sherlekar and P.S. Subramanan, "Conditionally Robust Two-Pattern Tests and CMOS Design for Testability," *IEEE Trans. Compt.*, vol. 7, no. 3, pp. 325-332, Mar. 1988.

[4] J.A. Bate and D. M. Miller, "Exhaustive Testing of Stuck-Open Faults in CMOS Combinational Circuits," *IEE Proc. vol.* 135, Pt. E, no. 1, pp. 10-17, Jan. 1988.

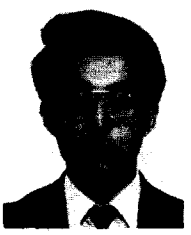
[5] S.K. Jain and V.D. Agrawal, "Modeling and Test Generation Algorithms for MOS Circuits," *IEEE Trans. Compt.* vol. C-34, no. 5, pp. 426-433, May 1985.

[6] R. Betancourt, "Derivation of Minimum Test sets for Unate Logic Circuits," *IEEE Trans. Comput.*, vol. C-20, pp. 1264-1269, Nov. 1971.

[7] G. Gupta and N.K. Jha '"A Universal Test Set for CMOS Circuits," *IEEE Trans. CAD.*, vol. 7, no. 5, pp. 59-597, May 1988.

[8] 정준모, 장원학, 임인칠, "CMOS 회로에 대한 테스트 생성방법," 대한전자공학회 추계종합학술대회 논문집, 제11권, 제 6 호, pp. 497-500, 1988년 11월.

著 者 紹 介



鄭 峻 模(正會員)
1962年 6月 28日生. 1985年 2月 한양대학교 전자공학과 졸업. 1987年 2月 한양대학교 대학원 전자공학과 석사학위 취득. 1987年 3月~현재 한양대학교 대학원 전자공학과 박사과정 재학중. 주관심분야는 VLSI Design, Logic Testing, Testable Design 및 Simulation.

林 寅 七 (正會員) 第25卷 第8號 參照
현재 한양대학교 전자공학과 교수