

論文93-30B-4-10

시스템 診斷을 위한 失藏 MUX의 檢査패턴 生成 알고리즘 (The Test Pattern Generation Algorithm of Embedded MUX for the System Diagnosis.)

李康鉉*, 金容得**

(Kang Hyeon Rhee, Yong Deak Kim)

要約

본 논문에서는 디지털 시스템의 오진단을 방지하고자, 고장이 존재하는 실장 MUX의 검사패턴을 생성하는 알고리즘을 제안한다. 시스템이 다수의 기능블럭으로 분할이 될 때, 실장 MUX에 고장이 존재하면 관측하고 싶은 기능블럭을 진단할 수 없게 된다. 제안된 검사패턴 생성 알고리즘은 2단 및 다단 레벨로 설계된 MUX 검사에 모두 적용할 수 있다. 고장 검출률(fault coverage)이 100%가 되어 별도의 고장 시뮬레이션이 필요 없으며, 검사패턴이 규칙성 있고 용이하게 2d로 생성되는 알고리즘이었다. 그리고 검사비용의 감소는 기존의 segmentation 검사방식에 비해서 85%임을 확인하였다.

Abstract

In this paper, we propose the test pattern generation algorithm of the embedded faulty MUX for the prevention of misdiagnosis of digital systems. When the system is partitioned with a large number of functional blocks, if the faults are existed in a embedded MUX then it can not diagnose the wanted observation of functional block. The proposed test pattern generation algorithm can apply the MUXs that designed 2-level and multi-level both. Fault coverage becomes 100% and so it is no necessary of the additional fault simulation and the proposed algorithm that have the regular and easily generated 2d test patterns. And we confirmed that the reduction of test cost becomes 85%, compared with the conventional segmentation testing scheme.

1. 서론

전자산업의 다양화한 발전에 따라 디지털 시스템은 다기능화, 다목적화 그리고 대규모화로 변화되고 있다. 이러한 시스템의 설계는 CAD 툴(tool)에 의한

설계자동화의 기술 도입으로 칩(chip)의 집적도가 매우 높은 one-chip의 LSI/VLSI화로 이루어진다. 이에 따라 이들 시스템의 생산 이후에 진단이 심각한 문제점으로 대두되고 있다.^[1] 설계된 시스템의 논리적 기능은 100% 진단될 수 있어야 하고 redundancy가 없어야 된다. 시스템의 진단비용은 생산원가와 직접적으로 관련되므로, 이제는 시스템 진단에 대한 중요성이 기술의 차원이 아닌 경영의 차원에서 다루어지면 안되게 되었다. 대두된 진단문제의 평가는 시스템에 대한 진단방법과 검사패턴(test pattern) 생성, 그리고 검사응답의 분석으로 이루어진다.

*正會員, 朝鮮大學校 電子工學科
(Dept. of Elec. Eng., Chosun Univ.)

**正會員, 亞洲大學校 電子工學科
(Dept. of Elec. Eng., Ajou Univ.)

接受日字: 1992年 3月 18日

이러한 시스템 진단을 위한 연구는 그림 1과 같이 시스템을 다수의 기능 블럭으로 분할하고, 각 블럭에 대한 정상동작을 진단하는데 블럭의 신호선은 블럭의 PI, PO와 이웃 블럭과의 데이터를 수수하는 BI (boundary input), BO(boundary output)의 4종류가 된다.^[2,3] 이들 신호선을 외부에서

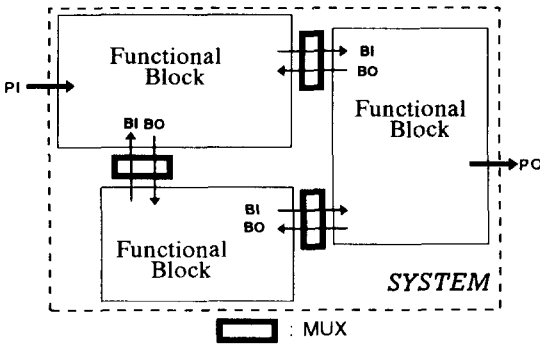


그림 1. 기능 블럭의 시스템 분할

Fig. 1. Partitioned system with functional block.

관측하기 위해선 각 블럭의 경계에 MUX를 실장한다. 이때, MUX 회로에 stuck-at fault(이하 s-a 고장)가 존재하게 되면 원치 않은 다른 기능블럭을 진단하는 오류가 발생할 뿐 아니라 관측하고 싶은 기능블럭을 진단할 수 없게 되는 문제점이 발생된다.

본 논문에서는 시스템의 진단 전에 고장이 존재하는 실장 MUX를 미리 검사하여 오진단을 방지하고자, PLD 및 FPGA의 게이트 레벨에서의 MUX의 검사패턴을 생성하는 알고리즘을 제안한다. 제안된 알고리즘은 기존의 부울 미분식^[12,13]의 가능한 모든 검사패턴을 생성해야 하는 실행시간이 많이 걸리는 단점과 경로활성화^[14]에서 경로탐색의 어려움의 단점이 개선되어 의사 코드(pseudo code)로서 검사패턴의 생성 알고리즘의 구조화 표현이 가능하게 되었다. 그리고 생성된 검사패턴을 2단 및 다단 레벨의 AND-OR MUX에 적용하여 고장적용 및 고장 검출율의 효용성 확인과 그 결과들을 검토한 후에 결론을 맺는다.

II. 이론적 배경

MUX의 입력선은 그림 2와 같이 출력에 정보를 보내는 데이터 Di와 이를 선택 결정하는 어드레스 선 Sj로 이루어 진다.

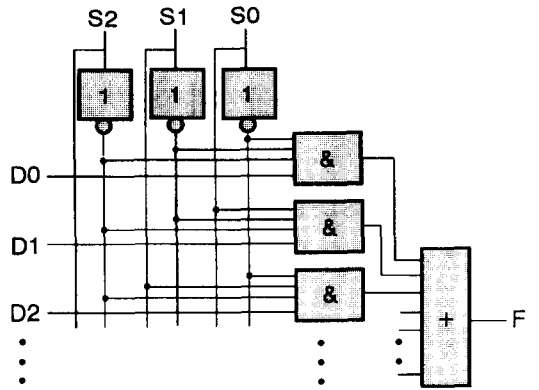


그림 2. MUX의 입력선 구조

Fig. 2. Inputs structure of MUX.

여기서

- d : 데이터 입력선의 수
- a : 어드레스 입력선의 수
- n : 전체입력 수 (n=a+d)
- Di : 데이터 입력 (0<i<=d)
- Sj : 어드레스 입력 (0<j<=a)

이다.

MUX의 입력 고장은 데이터 선과 어드레스 선에 단일(single) s-a 고장이 존재하게 되면, 데이터 선의 고장은 경로 활성화 기법으로 검사패턴을 생성하여 출력에서 관측할 수 있지만, 어드레스 선에 고장이 존재하면 데이터 선의 선택에 오류가 발생하여 원하는 입력 Di를 출력에 전달하지 못한다. 그림 2의 MUX에서 S0, S1이 00일때, 출력에는 D0가 나타나는데 S0에 s-a-1이 존재하면 D1이 나타난다. S0, S1이 11일 때, S0에 s-a-0이 존재하면 D3대신 D2가 출력에 나타난다.

이러한 s-a 고장을 검출하는 검사패턴 생성의 체계적인 방법은 고장이 출력까지 전파될 수 있도록 고장이 존재하는 노드부터 출력으로 경로를 활성화시켜야 한다.^[4,6] 이를 위하여 Roth는 복수의 경로를 동시에 활성화하는 D-알고리즘^[7,8]을 제안하였다. 논리소자를 통과하는 경로가 활성화되기 위해서는, 그 소자의 진리표 특성에 따른 특이커버를 정의하여 일치조작을 실행한다. 표 1은 각 논리 게이트의 특이커버이다.

특이커버의 정의는 적 계열의 논리 게이트 입력은 1을 할당하고, 합 계열의 논리 게이트 입력은 0을 할당한다. 이는 회로의 입력에서 출력으로 순추적(forward tracing)을 실행하는 과정이다.

표 1. 논리소자의 특이커버

Table 1. Singular cover of Logic gates.
(X=don't care)

AND		NAND		OR		NOR		EXOR	
入力	出力	入力	出力	入力	出力	入力	出力	入力	出力
1 1	1	0 X	1	1 X	1	0 0	1	X X'	1
0 X	0	X 0	1	X 1	1	1 X	0	X X	0
X 0	0	1 1	0	0 0	0	X 1	0		

게이트 레벨에서의 고장은 경로의 논리상태가 "1", "0"으로 고정되어 발생된다. 회로 내부 노드의 정상 신호가 1일때, 0을 발생시키는 고장신호는 s-a-0로서 D로 표현되며, 정상신호가 0일때, 1을 발생시키는 고장신호는 s-a-1로서 D'로 표현된다. 게이트의 출력에 이러한 고장이 발생하면 표 2와 같이 입력에 일치조작의 논리값 할당이 이루어진다 이는 출력에서 입력으로 역추적(back tracing)을 실행하는 과정이다.

표 2. 논리 게이트의 일치조작

Table 2. Consistency operation of Logic gates.

出力	入力檢査 패턴				
	AND	NAND	OR	NOR	EXOR
0	0 X	1 1	0 0	1 X	0 0
0	X 0	1 1	0 0	X 1	1 1
1	1 1	0 X	1 X	0 0	0 1
1	1 1	X 0	X 1	0 0	1 0

그리고 MUX와 같이 출력이 거의 모든 입력에 의존하는 다입력-단일출력으로 구성된 회로에서 실행하는 검사방법은 회로를 구조적으로 분할하여 검사하는 segmentation 검사이다. [9-11] 그림 3은 segmentation 검사를 위한 회로분할이다.

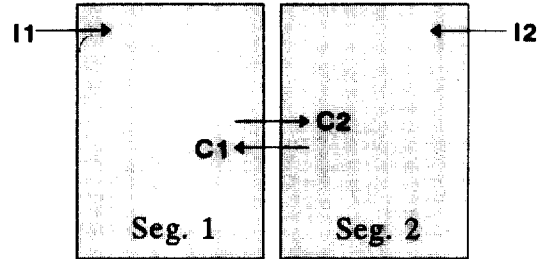


그림 3. 회로의 Segmentation 검사
Fig. 3. Segmentation test of circuit.

분할회로의 Seg. i에 대한 exhaustive 검사패턴의 수는 입력의 지수로 나타나는

$$\text{Test patterns(Seg. p)} = 2^{(I_p + C_p)} \quad (1)$$

이다.

여기서

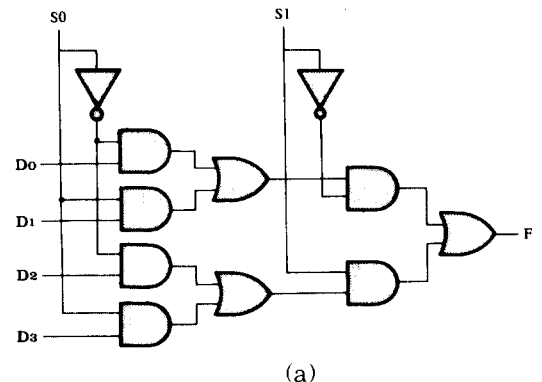
- m : 회로분할 수 ($1 < p <= m$)
- I_p : Seg. p 회로의 주입력 수
- C_p : 이웃하는 Seg. p로 부터 분할선(cutline)을 통하여 들어오는 보조 입력선

이다.

그러므로 회로의 검사비용은 식(2)와 같이 검사패턴의 전체 수로 정해진다.

$$\text{COST(test patterns)} = \sum_{\text{all Seg. p}} 2^{(m \text{ Seg. p})} = \sum_{p=1}^m 2^{(I_p + C_p)} \quad (2)$$

그림 4(a)의 4 데이터 비트 AND-OR MUX를 검사하기 위하여 (b)와 같이 분할하고, 분할된 I_p 와 C_p 는 표 3과 같다.



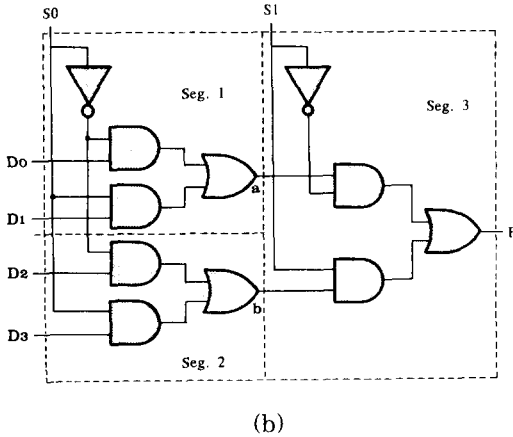


그림 4. (a) 4:1 멀티플렉서 (b) 회로의 분할
Fig. 4. (a) 4:1 Multiplexor, (b) Circuit partitioning.

표 3. 그림 4의 각 Si의 Ii와 Ci
Table 3. Ii and Ci elements of Si in Fig. 4.

	I _p	C _p
Seg. 1	D0, D1, S0	---
Seg. 2	D2, D3, S0	---
Seg. 3	S1	a, b

III. 실장 MUX의 검사패턴 생성 알고리즘 제안

1. 데이터 선의 고장 검출

그림 2의 MUX 입력에 대한 데이터 선에 고장이 존재할 때, 이를 검출하기 위해서 다음 정리를 세운다.

[정리 1] MUX의 데이터 선 선택은 어드레스 값에 따라 이루어진다.

[증명 1] 8비트 MUX의 출력함수

$$F = D0 \cdot S2' \cdot S1' \cdot S0' + D1 \cdot S2' \cdot S1' \cdot S0 + \dots + D7 \cdot S2 \cdot S1 \cdot S0 \quad (4)$$

에서 D1에 대한 부울미분(boolean difference)^[12]을 실행하면

$$\frac{dF}{dD1} = S2' \cdot S1' \cdot S0 \quad (5)$$

이다. Q.E.D.

그러므로 MUX의 출력은 어드레스 값에 따라 하나의 데이터 선만을 선택한다. 증명 1로부터 MUX는 하나 이상의 데이터 선을 검사할 수 없다. 그러므로 고장이 존재하는 데이터 선을 출력으로 경로를 활성화시키면 고장을 검출할 수 있다.

[정의 1] 데이터 선 Di의 s-a-0은 특이커버로부터 Di=1로, s-a-1은 Di=0로 지정되므로 고유의 검사패턴을 2개 갖게되어 총, 2d의 검사패턴이 요구된다.

이 알고리즘을 의사 코드로 표현하면 그림 5와 같다.

```

1:  begin
2:      for j=0 to (2d-1)
3:          if j <= (d-1)
4:              AT (Sa-1, ..., S1, S0) = j
5:                  if k=j
6:                      Dk=1
7:                  else
8:                      Dk=X /* X=don't care */
9:              else
10:                 AT (Sa-1, ..., S1, S0) = (j-d)
11:                     if k=(j-d)
12:                         Dk=0
13:                     else
14:                         Dk=X
15:                 end if
16:             next j
17:         end
    
```

그림 5. 제안된 실장 MUX의 데이터 선 검사패턴 생성 알고리즘

Fig. 5. The proposed test pattern generation algorithm of the embedded MUX's data lines.

2. 어드레스 선의 고장 검출

MUX의 설계는 2단 및 다단 레벨로 설계할 수 있는데, 이들 모두 어드레스 선에 고장이 존재하면 선택할 데이터 선 외의 다른 데이터 선이 선택된다. 어드레스 선의 고장은 그림 6과 같이 어드레스 선의 INV의 입력과 출력의 2부분에 존재한다. 그리고 어드레스선의 발생 가능한 단일 s-a-1, s-a-0 고장은 표 4와 같다. 이를 기반으로 다음의 정리를 세울 수 있다.

[정리 2] 그림 6의 어드레스 선의 S0에서 s-a-0 고장은 MUX의 출력 F에 활성화 되지 않는다.

[증명 2] 이는 AND 게이트의 데이터 선에 대한

s-a-0와 동일하므로 그림 5의 알고리즘에 의해 생성된 패턴 외의 다른 검사패턴을 요구하지 않는다. Q.E.D.

표 4. 어드레스 선의 발생 가능한 단일 s-a 고장
Table 4. Single s-a faults of address line.

S ₂ S ₁ S ₀	stuck-at-1			stuck-at-0		
	S ₂	S ₁	S ₀	S ₂	S ₁	S ₀
0 0 0	0	0	1	0	1	0
	0	1	0	1	0	0
	1	0	0	—	—	—
0 0 1	0	1	1	0	0	0
	1	0	1	—	—	—
0 1 0	0	1	1	0	0	0
	1	1	0	—	—	—
0 1 1	—	—	—	0	1	0
	—	—	—	0	0	1
1 0 0	1	0	1	—	—	—
	1	1	0	0	0	0
1 0 1	1	1	1	1	0	0
	—	—	—	0	0	1
1 1 0	1	1	1	1	0	0
	—	—	—	0	1	0
1 1 1	—	—	—	1	1	0
	—	—	—	1	0	1
	—	—	—	0	1	1

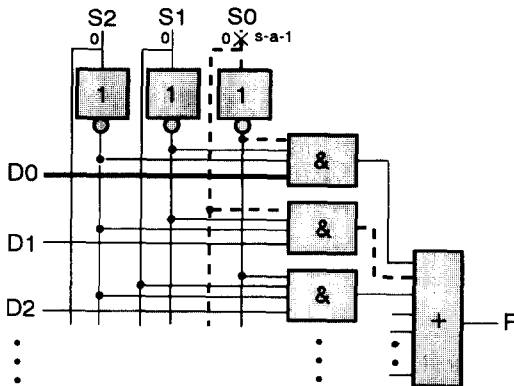


그림 6. MUX의 어드레스 선 고장.
Fig. 6. Stuck-at fault of MUX's address lines.

[정리 3] 그림 6의 S0에서 s-a-1은 경로가 연결

된 모든 AND 게이트를 통하여 출력 F에 활성화된다.

[증명 3] 특이커버에 의하여 S0의 고장 부분은 0을 할당하고 일치조작에 의하여 D0에 1을 할당한다. 그리고 게이트의 enable을 위해 다른 어드레스선은 특이커버에 의해 1을 할당한다. 그리고 어드레스의 s-a-1에 의해 오류를 선택된 다른 게이트를 disable하기 위해 그 데이터 입력은 0을 할당한다.

Q.E.D.

그러므로 증명 3으로 부터 S0 부분의 s-a-1은 이의 어드레스와 1큐브(cube) 관계가 있는 어드레스의 게이트가 오류 선택된다.

```

1: begin
2:   for j=0 to (2d-1)
3:     if j <= (d-1)
4:       AT (Sa-1, ..., S1, S0) = j
5:       if k=j
6:         Dk=1
7:       else
8:         Dk=x
9:     else
10:      AT (Sa-1, ..., S1, S0) = (j-d)
11:      if k=(j-d)
12:        Dk=0
13:      else if cube(k, j)=1
14:        Dk=0
15:      else Dk=x
16:    end if
17:  next j
18: end
    
```

그림 7. 제안된 실장 MUX의 어드레스 선 검사패턴 생성 알고리즘

Fig. 7. The proposed test pattern generation algorithm of the embedded MUX's address lines.

[정리 4] 그림 7로 부터 그림 6의 S0의 s-a 고장이 검출된다.

[증명 4] 특이커버와 일치조작에 의해 증명 2.3의 과정이 동일한 특성을 갖는다.

Q.E.D.

[정의 3] 그림 5와 7의 알고리즘에서, 3행부터 12행까지 동일하고 14행과 15행이 동일하므로 데이터 선의 검사패턴은 어드레스 선의 검사패턴에 모두 포함된다.

그러므로 정의 2.3으로 부터 MUX의 검사패턴은 그림 7의 어드레스 선에 대한 s-a 검사패턴의 생성으로 일축된다.

IV. 실행결과 및 검토

본 논문에서 제안한 디지털 시스템 진단의 실장 MUX에 대한 데이터 및 어드레스 선의 s-a 고장검출을 위한 검사패턴 생성 알고리즘을 2단 및 다단 레벨의 MUX를 모델로 하여 실행하였다.

8:1 MUX에 대하여 제안된 알고리즘을 실행한 결과, 생성된 검사패턴은 표 5와 같고, 이를 2단 및 다단 레벨의 MUX에 적용하여 논리 시물레이션을 수행하여 동일한 결과가 그림 8과 같이 나타났다.

표 5. 8:1 MUX의 검사패턴
Table 5. The generated test patterns of 8:1 MUX.

Pattern No.	S2	S1	S0	D0	D1	D2	D3	D4	D5	D6	D7	F
0	0	0	0	1	X	X	X	X	X	X	X	1
1	0	0	1	X	1	X	X	X	X	X	X	1
2	0	1	0	X	X	1	X	X	X	X	X	1
3	0	1	1	X	X	X	1	X	X	X	X	1
4	1	0	0	X	X	X	X	1	X	X	X	1
5	1	0	1	X	X	X	X	X	1	X	X	1
6	1	1	0	X	X	X	X	X	X	1	X	1
7	1	1	1	X	X	X	X	X	X	X	1	1
8	0	0	0	0	1	1	X	1	X	X	X	0
9	0	0	1	1	0	X	1	X	1	X	X	0
10	0	1	0	1	X	0	1	X	X	1	X	0
11	0	1	1	X	1	1	0	X	X	X	1	0
12	1	0	0	1	X	X	X	0	1	1	X	0
13	1	0	1	X	1	X	X	1	0	X	1	0
14	1	1	0	X	X	1	X	1	X	0	1	0
15	1	1	1	X	X	X	1	X	1	1	0	0

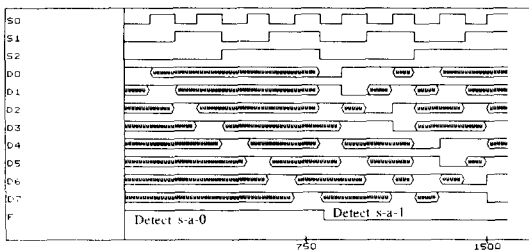


그림 8. 표 5에 의한 논리 시물레이션 결과
Fig. 8. Logic simulation result of Table 5.

표 5에 의하여 2단 및 다단 레벨 MUX의 각 노드에서의 단일 s-a 고장에 대하여 확인한 고장 검출을

은 두 회로 모두 100%이었다. 이 과정을 통하여 확인된 사항은, 2단 레벨 MUX는 표 5가 그대로 적용되었지만 다단 레벨 MUX의 경우는 표 6과 같이 1부분이 don't care가 되어도 고장검출에는 이상이 없음을 확인하였다.

그리고 제안된 검사패턴 생성 알고리즘과 기존 방식의 검사비용 비교는 표 7과 같다.

제안된 알고리즘의 검사비용은 exhaustive와 segmentation 검사에 비하여 각각 99%와 85%가 감소되었다. 그리고 표 5와 6을 비교해 보면 5는 어드레스 Sj값에 따라 검사패턴이 규칙성을 갖는 반면에 6은 이러한 특성이 없다. 이는 저단 레벨로 설계된 회로의 검사패턴 생성이 용이함을 알 수 있다.

표 6. 다단 레벨에서의 검사패턴
Table 6. Test patterns with multi-level.

Pattern No.	S2	S1	S0	D0	D1	D2	D3	D4	D5	D6	D7	F
0	0	0	0	1	X	X	X	X	X	X	X	1
1	0	0	1	X	1	X	X	X	X	X	X	1
2	0	1	0	X	X	1	X	X	X	X	X	1
3	0	1	1	X	X	X	1	X	X	X	X	1
4	1	0	0	X	X	X	X	1	X	X	X	1
5	1	0	1	X	X	X	X	X	1	X	X	1
6	1	1	0	X	X	X	X	X	X	1	X	1
7	1	1	1	X	X	X	X	X	X	X	1	1
8	0	0	0	0	1	1	X	1	X	X	X	0
9	0	0	1	1	0	X	1	X	1	X	X	0
10	0	1	0	1	X	0	1	X	X	1	X	0
11	0	1	1	X	1	1	0	X	X	X	1	0
12	1	0	0	1	X	X	X	0	1	1	X	0
13	1	0	1	X	1	X	X	1	0	X	1	0
14	1	1	0	X	X	1	X	1	X	0	1	0
15	1	1	1	X	X	X	1	X	1	1	0	0

표 7. 기존 알고리즘과 제안된 알고리즘의 비교결과

Table 7. Comparison results between the conventional and proposed algorithm.

Testing schemes	# of Test patterns	Remark
Exhaustive testing	2,048	2 ¹¹
Segmentation testing	112	Eq. (2)
Proposed algorithm	16	Table 4,5

V. 결론

본 논문에서는 디지털 시스템의 각 기능블럭의 경계에 실장되어 시스템 진단에 이용되는 MUX의 오

동작을 검출할 수 있는 검사패턴의 생성 알고리즘을 PLD와 FPGA로 구현되는 게이트 레벨 상에서 제안하였다.

제안된 알고리즘은 2단 및 다단 레벨로 설계된 MUX 검사에 모두 적용됨을 확인하였으며, 고장 검출율도 100%가 되어 별도의 고장 시뮬레이션이 필요 없으며, 검사패턴이 규칙성 있고 용이하게 2d로 생성되는 알고리즘이었다. 그리고 검사비용은 회로의 PO가 PI에 모두 의존하는 segment 검사방식에 비해서도 85%가 감소됨이 확인되었다.

이상의 결론으로 시스템 진단을 위한 검사의 scheduling 전에 기능블럭의 경계에 실장된 MUX를 선처리 검사함으로써 MUX의 고장에 따른 기능블럭의 오류 선택 및 오류 검사를 사전 방지할 수 있으므로 이 분야에 사용이 널리 기대되며, DFT (design for testability) 분야의 LSSD를 위한 LFSR의 MUX와 시스템이 one-chip으로 구현화될 때, BIST(built-in self testing)의 MUX에도 제안된 알고리즘으로 생성된 검사패턴의 적용 가능성이 확인되었다.

參 考 文 獻

[1] 이강현, 김용득, "t-분포를 이용한 회로분할의 경계노드 탐색에 관한 연구," 대한 전자공학회 논문집, vol.27, no.9, 1990.9, pp.132-137.
 [2] Bozorgui-Nesvat and E. J. McCluskey, "Structured Design for Testability to Eliminate Test Pattern Generation," 10th Annual Symposium on Fault Tolerant Computing, pp. 158-163, Kyoto, Japan, Oct. 1 980.
 [3] CAD 기술특강, 대한전자공학회, 1991.
 [4] Goel P., "An implicit enumeration algorithm to generate tests for combinational logic circuit," *IEEE Trans. on Computers*, vol.C-30, no.11, Nov. 1981, pp. 215-222.
 [5] Bozorgui-Nesvat and E. J. McCluskey,

"Design for Autonomouse Test," *IEEE Trans. on Computers*, vol.C-33, no.6, pp. 541-546, June 198 4.
 [6] Fujiwara H. and T. Shimono, "On the acceleration of test generation algorithm," *IEEE Trans. on Computers*, vol.C-32, no.12, 1983, p p. 1137-1144.
 [7] Roth J. P., "Diagnosis of Automation Failures: A Calculus and a Method," *IBM J. Res. and Dev.* 10, 1966, pp. 278-281.
 [8] Benmehrez C. and J. F. McDonald, "Mearsured Performance of program med Implementation of the Subscripted D-Algorithm," *Proc. 20th Design Automation Conf.*, 1983, pp. 308-315.
 [9] I. Shperling and E. J. McCluskey, "Circuit Segmentation for Pseudo - exhaustive Testing," *CRC. TR.*, Stanford Univ., no.87-2, Feb. 1987.
 [10] E. J. McCluskey, "Verification testing - A Pseudo-exhaustive test technique," *IEEE Trans. on Computers*, vol.C-33, no.6, June 1984, pp. 541-546.
 [11] 이강현, 김용득 "Design for Testability를 위한 검사방식," 대한 전자공학회지, 설계자동화 특집, 1992.1.
 [12] Frederic F. Sellers, M. Y. Hsiao and L. W. Bearson, "Analyzing Errors with the Boolean Difference," *IEEE Trans. on Computers*, vol C-17, no.7, July 1968.
 [13] Marinos P. N., "Derivation of Minimal Complete Sets of Test Input Sequences using Boolean Difference," *IEEE, Trans. on Computers*, v ol.C-20, no.1, Jan. 1971, pp. 25-32.
 [14] Roth, J. P., Diagnosis of automata Failures: A calculus and a method., *IBM J. Res. Dev.* 10: pp.278-281.

著 者 紹 介

李 康 鉉 (正會員) 第 29 卷 B 編 第 5 號 參照)
 현재 조선대학교 전자공학과부교수

金 容 得 (正會員) 第 29 卷 B 編 第 5 號 參照)
 현재 아주대학교 전자공학과 교수