

PLA에서의 고장검출 및 고장진단에 관한 연구

(A Study on Fault Detection and Diagnosis in PLA)

李 殷 高*, 金 熙 碩**, 林 濟 鏗**

(Lee, Eun-Seol Kim, Hi-Seok and Lim, Chae-Tak)

要 約

PLA는 fusing에 의하여 programming이 되므로 일반적인 조합논리에서는 존재하지 않는 fusing에 의한 고장이 발생하게 된다.

本論文에서는 이와 같은 고장의 검출과 고장진단을 위하여 test 입력을 규칙적으로 구하는 방법과 회로가 redundant 상태이거나 irredundant 상태이거나 관계없이 고장검출이 가능하도록 회로를 구성하는 방법을 제안하였다.

Abstract

Because PLA'S are programmed by fusing, faults by fusing which do not exist in conventional random logic exist in them.

In this paper two methods are proposed. One is a regular test pattern generation method to detect and diagnose such faults and the other is a method of designing a network by which faults are detected regardless of whether PLA'S are redundant or not.

I. 序 論

PLA(programmable logic array)는 two-level AND-OR(혹은 NOR-NOR)行列의 배열로 多入力, 多出力에 적합한 논리 배열이다. 또한 PLA는 memory와 흡사한 규칙적인 구조로 되어 있기 때문에 LSI 혹은 VLSI의 제조에 적합하다.

PLA는 H.Fleisher와 L. I. Maissel^[1]에 의하여 제안되었으며, fusing에 의하여 programming이 되므로 일반적인 random combinational logic에서는 발생하지 않는 fusing에 의한 고장 상태가 존재하게 된다. 특히 사용자가 fusing을 할 수 있도록 고안된

것이 field programmable logic array(FPLA)이다.

이처럼 PLA에서 발생하는 fusing에 의한 고장을 검출하기 위하여 다음과 같은 방법들이 제안되었다. J. E. Smith^[2]는 sharp 연산자(#)의 연산을 이용하여 test 입력을 구하는 방법을 제안하였으며, V. K. Agarwal^[3]는 SAE(stuck-at-Equivalence) 회로를 이용하여 multiple fault 검출의 효율성을 제안하였다. S. J. Hong과 D. L. Ostapko^[4]는 PLA의 입력 decoder 부분이 2-입력 decoder로 구성된 경우를 기본으로 하며 test 입력을 구하는 방법을 제안하였으며, 특히^[5]에서는^[4]에서 처리하지 못한 redundant 상태의 고장을 검출하는 방법이 제안되었다.

本 논문에서는 앞의 논문에서는 처리하지 못한 고장이 발생한 위치 판별과^{[2],[3],[4]}에서 처리 못한 redundant 상태의 고장을 검출하는 방법을 제안하였으며, minimal test 입력을 구하기 위하여 sharp 연산자(#)를 이용한 연산과 Cap(\cap)의 연산을 사용하였다. Shift register와 2-입력 equivalence(=exclusive-NOR) Gate를 사용하여 보수 test 입력을 구

* 準會員, ** 正會員

漢陽大學校 工科大學 電子工學科

(Dept. of Electronics Engineering, Han Yang Univ.)

接受日字: 1981年 2月 16日

하는 방법을 제안하였다.

또한 PLA의 회로가 redundant 상태이거나 ir redundant 상태이거나에 관계없이 고장검출이 가능하도록 control bit를 사용하므로써 모든 single fault의 검출이 가능하게 되었으며, 몇 가지의 경우에는 single fault가 발생하거나 multiple fault가 발생하여도 고장의 종류 및 고장이 발생한 위치의 판별도 가능하게 되었다.

2. PLA의 구조 및 특성

PLA는 일반적으로 입력 decoder, AND-OR 배열, 출력 inverter 부분으로 나타낼 수 있다.

PLA의 상태를 나타내는 표에서 입력 부분에 존재하는 "1"의 값은 normal한 상태를 "0"의 값은 보

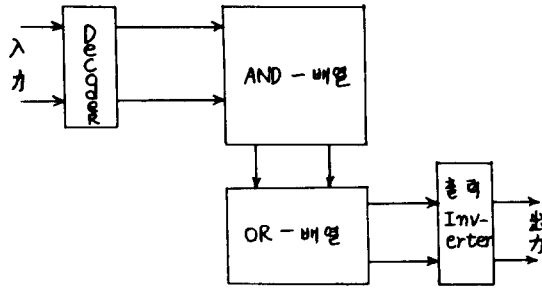


그림 1. PLA의 기본구조
Fig. 1. Basic structure of a PLA.

수상태를 나타내며, ×는 Don't care 상태를 나타낸다. 즉 PLA의 상태를 나타내는 표가 표1과 같을 때 이것을 sum-of-product의 형태로 표시하면

$$\begin{cases} f_1 = x_1 \bar{x}_3 \bar{x}_4 + x_1 x_3 \bar{x}_4 \\ f_2 = x_2 x_4 \bar{x}_5 \\ f_3 = x_1 \bar{x}_3 \bar{x}_4 + x_2 x_4 \bar{x}_5 + \bar{x}_1 \bar{x}_2 \end{cases} \text{가 된다.}$$

표 1. PLA의 상태를 나타내는 표
Table 1. A cubical notation of a PLA.

x_1	x_2	x_3	x_4	x_5	f_1	f_2	f_3
1	×	0	0	×	1	0	1
×	1	×	1	0	0	1	1
1	×	1	0	×	1	0	0
0	0	×	×	×	0	0	1

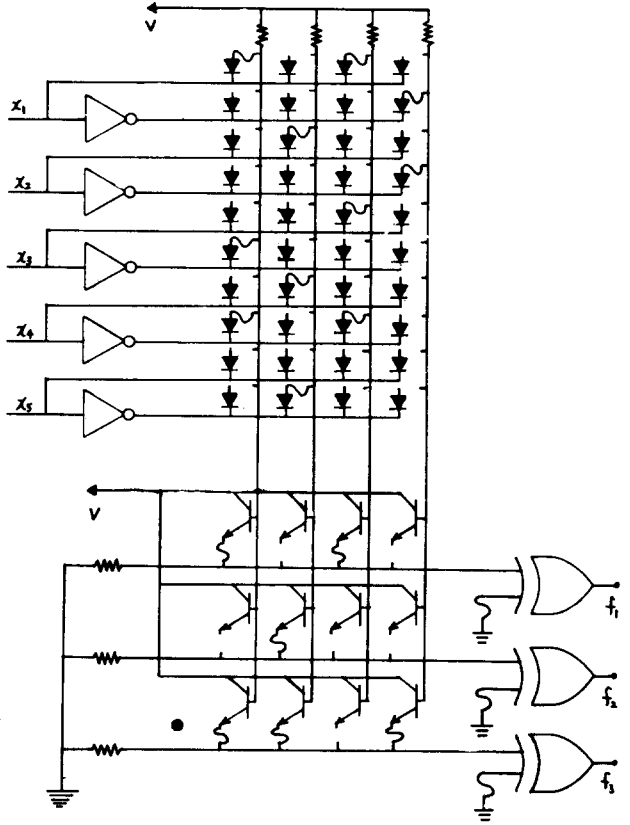


그림 2. FPLA의 내부회로
Fig. 2. A network of a FPLA.

표1을 FPLA의 회로로 나타내면 그림2^[6]와 같다. 여기서 AND 배열은 diode로, OR 배열은 transistor로 되어 있으며, 출력의 inverter 부분은 exclusive-OR로 되어 있으므로 fusing을 하게 되면 \bar{f}_n 의 출력을 얻을 수 있다.

3. PLA의 Fusing부분에서 발생하는 고장의 종류

PLA의 fusing되는 부분에서 발생할 수 있는 고장의 종류는 다음과 같이 분리할 수 있다. 1~6까지는 입력부분에서 발생할 수 있는 single fault의 경우를 7과 8은 출력부분에서 발생할 수 있는 single fault의 경우를 나타낸다. 특히 5와 6의 상태는 AND형 IBF(input bridging fault)^[7]로 치환할 수 있으며, 5의 상태는 x_n 이 \bar{x}_n 에 의하여, 6의 상태

표 2. 고장의 분석

Table 2. Fault analysis.

	정상	고장	
	x_n	x	
1	1	×	$x_n \rightarrow \times$
2	0	×	$\bar{x}_n \rightarrow \times$
3	×	1	$x_n (\times) \rightarrow x_n$
4	×	0	$x_n (\times) \rightarrow \bar{x}_n$
5	1	→	$x_n \rightarrow x_n \cdot \bar{x}_n (B)$
6	0	→	$\bar{x}_n \rightarrow \bar{x}_n \cdot x_n (B)$
7	$f_n : 1$	$f_n : 0$	출력 부분의 고장
8	$f_n : 0$	$f_n : 1$	출력 부분의 고장
9	1	0	$x_n \rightarrow \bar{x}_n$
10	0	1	$\bar{x}_n \rightarrow x_n$

는 \bar{x}_n 이 x_n 에 의하여 고장이 발생하였음을 나타낸다. 7의 상태는 f_n 에 존재하는 product 항이 끊어진 상태를, 8의 상태는 f_n 에 존재하지 않는 새로운 product 항이 새로이 연결되었음을 나타낸다. 9와 10의 고장상태는 double fault 인 경우를 나타낸다.

4. Test 입력 산출 방법

PLA의 상태를 나타내는 표에서 i 행에 존재하는 입력(=product 항)을 $I_i = a_{i1}, a_{i2} \dots a_{in}$ 으로 표시하며, i 행에 존재하는 출력을 $Y_i = b_{i1}, b_{i2} \dots b_{ip}$ 로 표시하기로 한다.

(단 $a_{in} \in \{0, 1, \times\}$, $b_{ip} \in \{0, 1\}$)

* Test 입력을 구하는 방법*

(i) 서로 다른 2개의 입력에 대하여 sharp 연산자(♯)를 이용한 연산을 하여 각 입력 사이에 cap(\cap)의 관계가 있는가를 조사한다. 단 cap의 관계가 있는 입력을 “ \leftrightarrow ”의 기호로 표시한다.

(ii) “ \leftrightarrow ”로 표시된 입력(=cap의 관계가 있는 입력)들 사이에 cap의 연산을 하여 구한 결과를 minimal test 입력으로 한다. ($I_i \leftrightarrow I_j : I_i \cap I_j = M_{ij}^0$ 으로 표기한다.)

(iii) Minimal test 입력의 각 bit를 하나씩 보수를 취하여 구한 결과를 보수 test 입력으로 한다. 보수 test 입력은 M_{ij}^k 로 표기하며, minimal test 입력 M_{ij}^0 에서 k번째 bit를 보수로 하여 구해진 test입력을 나타낸다. 특히, (ii)의 경우에서 cap의 관계가 있는 입력들이 여러개가 존재하는 경우에는 어떤 입력들 사이에 cap의 연산을 하는가에 대하여 논하기로 한다.

(a) $k = 2n$ 인 경우. (단, k 는 입력의 수)

$$I_1 \leftrightarrow I_2 \leftrightarrow I_3 \leftrightarrow \dots \leftrightarrow I_{k-1} \leftrightarrow I_k$$

$$* \text{ Minimal test 입력 : } M_{12}^0 = I_1 \cap I_2 \dots M_{k-1}^0 = I_{k-1} \cap I_k$$

$$* \text{ Minimal test 입력의 개수 : } \frac{k}{2}$$

(b) $k = 2n - 1$ 인 경우

$$I_1 \leftrightarrow I_2 \leftrightarrow I_3 \leftrightarrow \dots \leftrightarrow I_{k-2} \leftrightarrow I_{k-1} \leftrightarrow I_k$$

$$* \text{ Minimal test 입력 : } M_{12}^0 = I_1 \cap I_2 \dots M_{k-1}^0 = I_{k-1} \cap I_k$$

$$* \text{ Minimal test 입력의 개수 : } \frac{k-1}{2} + 1$$

(c) 3개 이상의 입력이 동시에 cap의 관계가 있으면 3개 이상의 입력에 대하여 동시에 cap의 연산을 하여 구한 결과를 minimal test 입력으로 한다.

$$I_1 \leftrightarrow I_2 \leftrightarrow I_4 \leftrightarrow I_5$$



$$* \text{ Minimal test 입력 : } M_{123}^0 = I_1 \cap I_2 \cap I_3$$

$$M_{45}^0 = I_4 \cap I_5$$

(d) Cap의 관계가 전혀 없는 입력이 존재하는 경우에는 그 입력 자신을 minimal test 입력으로 한다.

$$I_1 \leftrightarrow I_2 \quad I_3$$

$$* \text{ Minimal test 입력 : } M_{12}^0 = I_1 \cap I_2$$

$$M_3^0 = I_3$$

이와 같은 방법에 의하여 구한 minimal test 입력에 Don't care 상태가 존재하면 $\times \rightarrow 1$ 혹은 $\times \rightarrow 0$ 의 2가지 경우중 한 경우를 임의로 선택하여 minimal test 입력을 결정한다.

그러므로 minimal test 입력의 개수는 입력변수에 의하여 결정되는 것이 아니라 입력의 수 (= product 항의 개수)에 의하여 결정된다. 입력변수가 1개, product 항이 ℓ 개, 출력이 p개인 PLA에서 minimal test 입력의 개수 k 는 다음과 같다. $1 \leq k \leq \ell$

즉, k 가 1인 경우는 product 항 ℓ 개가 모두 cap의 관계가 있는 경우이며, k 가 ℓ 인 경우는 product 항 ℓ 개가 전혀 cap의 관계가 없는 경우를 나타낸다. 이와 같이 2^n 개의 입력이 k 개의 test 입력으로 감소하는 결과를 얻을 수 있다.

5. Control Bit의 설정 방법과 보수 Test 입력의 산출을 위한 회로의 구성 방법

PLA의 회로가 redundant 상태가 되거나, 혹은 고장진단을 가능하게 하기 위하여 control bit를 추가로 설정하였으며, 그림 4와 같은 방법으로 회로내에 구성하면 된다. Control bit에 의하여 어떤 입력들이 고장검출이 되는지는 다음과 같다.

표 3. Control bit 과 입력과의 관계

Table 3. Relation between control bits and inputs (product terms).

control bit						고장검출
C_1	C_2	C_3	C_t		정상동작
1	1	1	1		
1	0	0	0	I_1	
0	1	0	0	I_2	
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
0	0	0	1	I_t	

보수 test 입력은 minimal test 입력으로 고장검출이 불가능한 경우를 가능하게 하며 고장이 발생한 위치 판별을 위하여 필요하다. 이와 같은 보수 test 입력이 minimal test 입력을 인가한 상태에서 자동적으로 구하여 지도록 다음과 같이 shift register 와 2-입력 equivalence gate 를 사용하는 방법을 제안하였다.

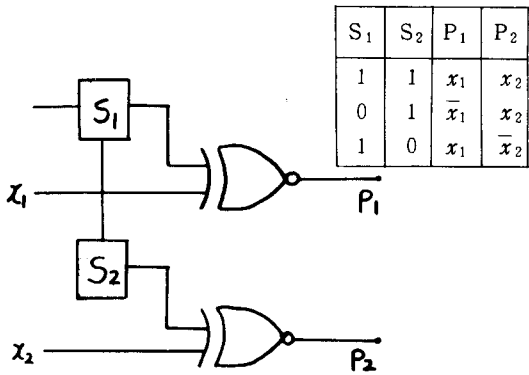


그림 3. 보수 test 입력의 산출 방법
Fig. 3. A method of complement test input generation.

그러므로 표 4를 앞의 모든 방법을 회로내에 구성하면 그림 4와 같다.

표 4. PLA의 상태를 나타내는 표

Table 4. A cubical notation of a PLA.

x_1	x_2	x_3	x_4	x_5	f_1	f_2	f_3
0	×	0	1	1	0	1	1
×	1	0	0	×	1	0	1
×	×	0	1	1	1	0	0
1	1	0	×	×	0	1	1

Minimal test 입력은 다음과 같이 구할 수 있다.
(방법 1) (Coordinate sharp product)

$$\begin{array}{l}
 I_1 \# I_2 \quad I_1 \# I_3 \quad I_1 \# I_4 \quad b_i \\
 0 \times 0 1 1 \quad 0 \times 0 1 1 \quad 0 \times 0 1 1 \quad a_i \# b_i \quad 0 1 \times \\
 \# \times 1 0 0 \times \quad \# \times \times 0 1 1 \quad \# 1 1 0 \times \times \quad a_i 1 \quad 0 Z \phi Z \\
 Z 0 Z \phi Z \quad Z Z Z Z Z \quad \phi 0 Z Z Z \quad \times 1 0 Z \\
 \therefore I_1 \leftrightarrow I_3 \\
 I_2 \# I_3 \quad I_2 \# I_4 \\
 \times 1 0 0 \times \quad \times 1 0 0 \times \\
 \# \times \times 0 1 1 \quad \# 1 1 0 \times \times \quad \therefore I_2 \leftrightarrow I_4 \\
 Z Z Z \phi 0 \quad 0 Z Z Z Z \\
 I_3 \# I_4 \\
 \times \times 0 1 1 \\
 \# 1 1 0 \times \times \quad \therefore I_3 \leftrightarrow I_4 \\
 0 0 Z Z Z
 \end{array}$$

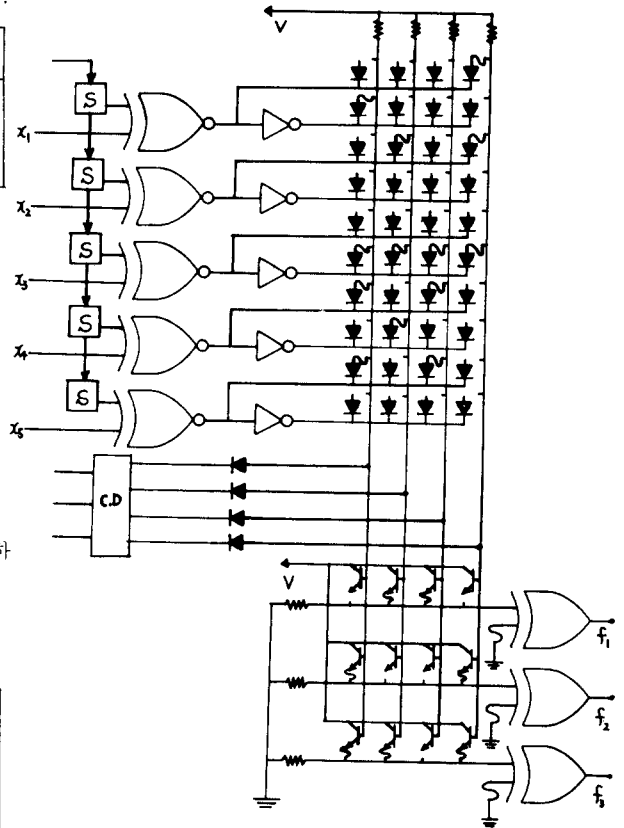


그림 4. 새로 제안된 회로
Fig. 4. Proposed network.

그러므로 $I_1 \leftrightarrow I_3 \leftrightarrow I_2$

Minimal test 입력은 2개가 된다.

$$M_{13}^0 = I_1 \cap I_3 \quad 0 \times \times 1 1$$

$$\frac{\cap \times \times 0 1 1}{0 \times 0 1 1} \rightarrow 00011 (\times \text{를 } 0 \text{로 치환})$$

$$M_{24}^0 = I_2 \cap I_4 \quad \times 1 0 0 \times$$

$$\frac{\cap 1 1 0 \times \times}{1 1 0 0 \times} \rightarrow 11000 (\times \text{를 } 0 \text{로 치환})$$

6. Single Fault와 Multiple Fault의 고장검출 및 고장진단

앞에서의 방법을 사용하여 single fault인 경우에 대하여는 표 5와 같으며, multiple fault인 경우에 대하여 표 6과 같이 나타낼 수 있다.

표 5. Single fault에 관한 고장진단

Table 5. Single fault diagnosis.

Test 입력에 의한 출력	고 장
(A) $I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0$ $I_i[M_{ij}^s] \rightarrow Y_{im} : \begin{cases} S=k: 0 \rightarrow 1 \\ S=k: 0 \end{cases}$	9.10 $x_k \rightarrow \bar{x}_k, \bar{x}_k \rightarrow x_k$
(B) $I_i[M_{ij}^0] \rightarrow Y_{im} : 1$ $I_i[M_{ij}^s] \rightarrow Y_{im} : \begin{cases} S=k: 0 \rightarrow 1 \\ S=k: \text{정 상} \end{cases}$	1.2 $x_k \rightarrow \times, \bar{x}_k \rightarrow \times$
(C) $I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0$ $I_i[M_{ij}^s] \rightarrow Y_{im} : \begin{cases} S=k: 1 \\ S=k: 0 \end{cases}$	3.4 $\times \rightarrow x_k, \times \rightarrow \bar{x}_k$ $(\times \rightarrow \bar{e}_k)$
(D) $I_i[M_{ij}^0] \rightarrow Y_{im} : 1$ $I_i[M_{ij}^s] \rightarrow Y_{im} : \begin{cases} S=k: 1 \rightarrow 0 \\ S=k: \text{정 상} \end{cases}$	3.4 $\times \rightarrow \bar{x}_k, \times \rightarrow x_k$ $(\times \rightarrow e_k)$
(E) $I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0$ $I_i[M_{ij}^s] \rightarrow Y_{im} : 0$	5.6 $x_n \cdot x_n(B), x_n \cdot x_n(B)$
(F) $I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0$ $I_i[M_{ij}^s] \rightarrow Y_{im} : 0$	7. 입력이 끊어진 상태
(G) $I_i[M_{ij}^0] \rightarrow Y_{im} : 0 \rightarrow 1$ $I_i[M_{ij}^s] \rightarrow Y_{im} : \text{정 상}$	8. 입력이 새로 연결된 상태

표 6. Multiple fault에 관한 고장진단

Table 6. Multiple fault diagnosis.

Test 입력에 의한 출력	고 장
$I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0$	(A) 형
$I_i[M_{ij}^s] \rightarrow Y_{im} : 0$	검출 가능

Test 입력에 의한 출력	고 장
$I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 1$ $I_i[M_{ij}^s] \rightarrow Y_{im} : \begin{cases} S=k, r \dots : 0 \rightarrow 1 \\ S \neq k, r \dots : \text{정 상} \end{cases}$	(B) 형 위치 판별 가능
$I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0$ $I_i[M_{ij}^s] \rightarrow Y_{im} : 0$	(C) 형 검출 가능
$I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 1$ $I_i[M_{ij}^s] \rightarrow Y_{im} : \begin{cases} S=k, r \dots : 1 \rightarrow 0 \\ S \neq k, r \dots : \text{정 상} \end{cases}$	(D) 형 위치 판별 가능
$I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0$ $I_i[M_{ij}^s] \rightarrow Y_{im} : 0$	(E) 형 검출 가능
$I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0$ $I_i[M_{ij}^s] \rightarrow Y_{im} : 0$	(A), (C), (E), (F) 검출 가능
$I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 1$ $I_i[M_{ij}^s] \rightarrow Y_{im} : \begin{cases} S=p, q \dots : 0 \rightarrow 1 \\ S=1.2 \dots : 1 \rightarrow 0 \\ S \neq p, 1 \dots : \text{정 상} \end{cases}$	(B), (D) 형 위치 판별 가능
$I_i[M_{ij}^0] \rightarrow Y_{im} : 0 \rightarrow 1$ $I_i[M_{ij}^s] \rightarrow Y_{im} : \text{정 상}$ $I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0$ $I_i[M_{ij}^s] \rightarrow Y_{im} : 0$	(G) 형 위치 판별 가능 (F) 형 검출 가능

(B), (D), (G) 형의 고장이 혼합된 multiple fault가 발생하여도 (G) 형의 고장은 출력이 정상상태에서 0인 곳에서 발생하므로 (B), (D), (G) 형의 고장은 검출 및 고장이 발생한 위치 판별이 가능하다. (B), (D), (F) 형의 고장이 혼합된 multiple fault가 발생하여도 같은 행에 존재하는 1의 상태를 갖는 다른 출력에서 (B), (D) 형의 고장을 검출할 수 있으므로 (B), (D), (F) 형의 고장검출 및 고장이 발생한 위치판별이 가능하다. 그리고 출력의 상태가 $\begin{cases} I_i[M_{ij}^0] \rightarrow Y_{im} : 1 \rightarrow 0 \\ I_i[M_{ij}^s] \rightarrow Y_{im} : 0 \end{cases}$ 인 경우에는 (A), (B), (C), (D), (E) 형이 혼합된 multiple fault이거나 (A), (C), (E), (G) 형이 혼합된 혹은 소고장 상태가 모두 혼합된 경우가 생기게 된다. 특히 (A)~(G)의 고장이 PLA의 각행에 한 가지씩만 존재하면 100%의 고장검출이 가능하다.

7. 결 론

PLA에서 고장검출을 하기 위하여 본 논문에서 제안한 방법을 이용하면 다음과 같은 결과를 얻게 된다.

(1) Test 입력을 구하는 방법이 규칙적이고 간단하다. 기존의 방법에서는 test 입력을 구하는데 있어서 입력

변수를 고장이 발생한 상태로 치환하여 구하였지만 본 논문에서는 PLA의 상태를 나타내는 표로부터 직접 구하는 방법, 즉 고장이 어떤 형태로 발생할 것인가에는 개의치 않고서 test 입력을 구할 수 있도록 하였다.

(2) test 입력 pattern이 감소된다. Minimal test 입력만을 구하면 되므로, 즉 PLA의 product 항의 갯수에 의하여 test 입력이 결정된다. ($2^n \rightarrow k$)

(3) PLA가 redundant 이거나 irredundant 이거나에 관계없이 고장검출이 가능하다. Control bit를 회로에 추가로 설정하였으므로 가능하게 되었다.

(4) 몇 가지의 경우에는 고장위치 판별 및 고장의 종류까지도 알아낼 수 있다. Single fault가 발생한 경우에는 100%의 고장검출이 가능하며, 출력의 각 행이 적어도 2개 이상의 1의 값을 갖는 경우에는 고장의 종류를 완전히 분리 시킬 수가 있으며 (A), (B), (C), (D), (F), (G)의 고장상태에서는 고장위치 판별도 가능하게 되었다.

1. H. Fleisher and L.I. Maissel, "An Introduction to Array Logic", IBM. J. Res. Develop., Vol. 19, pp. 98-109. Mar. 1975.
2. J. E. Smith, "Detection of Faults in Programmable Logic Arrays", IEEE. Trans. on Computers, Vol. C-28, No. 11, November, 1979, pp. 845-853.
3. V. K. Agarwal, "Multiple Fault Detection in PLA'S", IEEE. Trans. On Computers, Vol. C-29,

- No. 6, June, 1980, pp. 518-522.
4. D. L. Ostapko and S. J. Hong, "Fault Analysis and Test Generation for programmable Logic Arrays", IEEE. Trans. On Computers, Vol. C-28, No. 9, September, 1979, pp. 617-626.
5. S. J. Hong and D. L. Ostapko, "FITPLA: A Programmable Logic Array for Function Independent Testing", in 10th Int. Symp. Fault-Tolerant Comput. Kyoto, Japan, pp. 131-136, October, 1980.
6. "Signetics Bipolar & MOS Memory", Sunnyvale, CA: Signetics, 1977.
7. K. G. Y. Mei, "Bridging and Stuck-At Faults", IEEE. Trans. On Computers, Vol. C-23, No. 7, July, 1974, pp. 720-727.
8. M. Morris Mano, "Digital Logic and Computer Design", Englewood Cliffs, N. J.: Prentice-Hall, 1979.
9. Samuel C. LEE, "Modern Switching Theory and Digital Design", Englewood Cliffs, N. J.: Prentice-Hall, 1978.
10. Douglas Lewin, "Computer-Aided Design of Digital Systems", Crane, Russak & Compan 1977.
11. Yahiko kambayashi, "Logic Design of Programmable Logic Arrays", IEEE. Trans. On Computers, Vol. C-28, No. 9, September, 1979, pp. 609-617.

