

多重故障 테스트가 가능한 PLA 設計

(A Testable PLA's Design for Multiple Faults)

李 在 旻*, 金 殷 成*, 林 寅 七*

(Jae-Min Lee, Eun-Sung Kim and In-Chil Lim)

要 約

본 논문에서는 다중고장 검출이 가능하고 부가회로가 적은 새로운 PLA 설계방식을 제안하였다. 부가 회로로는 하나의 쉬프트레지스터 열과 2비트 디코더에 제어입력을 사용하였다. 제어입력을 사용하므로써 비트선들을 효과적으로 제어할 수 있다. 종래의 논문들에서 구체적으로 다루지 않았던 단락고장과 서로 다른 모델 간의 다중고장을 다루었으며 고장의 등가관계 및 지배고장(Dominant Faults)을 정의하여 다중고장을 해석하였다. 또한 본 설계방식을 Folded PLA에도 적용하였다.

Abstract

This paper proposes a testable design method of PLA's with low overhead and high fault coverage for multiple faults. Only a shift register and control input of 2-bit decoder are used for extra hardware. By using a control input, the bit lines are controlled effectively. As the fault model, bridging faults and multiple faults of different fault models are particularly considered. "Fault equivalence relation" and "dominant faults" are defined to be used for detection of multiple faults. Also, an easily testable folded PLA by this method is described.

I. 序 論

최근 LSI/VLSI 기술의 발전으로 회로의 집적도가 증가함에 따라 설계시간이 짧고, 오류가 적으며 검사가 용이한 설계방식이 요망되고 있다. Array logic은 이러한 요구에 부응하여 LSI/VLSI설계에 널리 이용되고 있으며, 특히 PLA(Programmable Logic Array)는 규칙적이고 유동적인 구조를 가지므로 회로변경이 쉽고, 설계자동화가 용이하게 이루어져 그 사용이 날로 증가하고 있다.^{1)~4)}

한편 PLA의 규모가 커지고 집적도가 높아짐에 따라 상대적으로 신뢰도 문제가 또한 크게 대두되고 있으며, 이러한 이유로 최근 PLA의 테스트에 대한 연구가 활발히 진행되고 있다.^{1)~3)}

지금까지 발표된 테스트가 용이한 화로설계 방식들은 부가회로를 사용하지 않고 PLA의 규칙성을 이용한, 테스트 생성이 함수 종속적인 설계방식과 부가회로를 사용한 테스트 생성이 함수 독립적인 설계방식 등 두가지로 나눌 수 있는데 전자는 테스트 대상이 되는 PLA의 함수에 따라 테스트를 생성함으로써 부가 하드웨어가 필요없게 된다.

그러나 테스트 생성을 위한 알고리즘이 복잡하고 고장검출률이 그다지 높지 못한 단점이 있다. 후자는 테스트 집합이 함수 독립적이므로 테스트 생성을 위한 노력이 적게 들고 고장검출률은 높지만, 부가 하드웨어량이 많고 이 부가하드웨어 사용으로 정상 동작시간 지연이 생기는 문제점을 갖고 있다.

본 논문에서는 위 두 가지 방법 사이의 trade-off 관계를 고려하여 부가회로가 적고 다중고장에 대해 높은 검출률을 갖는 테스트가 용이한 PLA의 설계방식을 제안한다. 제안한 설계방식으로 stuck-at 고장,

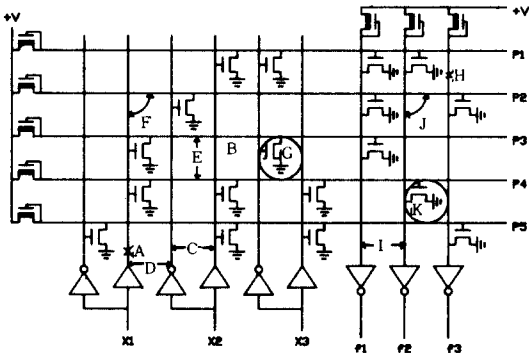
*正會員, 漢陽大學校 電子工學科
(Dept. of Electronic Eng., Han Yang Univ.)
接受日次: 1986年 3月 10日

접점고장은 물론 단락고장도 검출 가능하며 동일 모델의 다중고장과 서로 다른 모델의 다중고장 검출이 가능하다. 특히 PLA의 칩 이용률을 크게 높일 수 있는 2비트 decoded PLA⁽¹⁾를 설계할 때 2비트 디코우더에 제어입력을 달아 비트선(bit line)을 효과적으로 제어할 수 있게 한다. 또 고장 간의 등가관계 및 지배고장을 정의하여 다중고장을 해석하였으며 본 방식을 Folded PLA에도 적용하였다.

II. PLA의 고장 모델

초기의 PLA테스트에 관한 논문들에선 오직 stuck-at 고장만을 취급하였으나 점차 점점고장 및 단락고장도 고려하게 되었다. 그러나 단락고장 및 다중고장의 테스트에 대해서는 아직 충분한 연구가 이루어지지 못하고 있다. 본 논문에서 고려하고 있는 고장은 stuck-at고장, 점점고장은 물론 모든 가능한 단락고장(인접 신호선 및 인접하지 않은 신호선들의 단락고장)과 다중고장이다.

다음 그림 1은 NMOS PLA에서 발생 가능한 고장 형태를 도시한 것이다.



A, B, H : stuck-at고장
 C, D, E, F, I, J : 단락고장
 G, K : 점점고장

그림 1. PLA의 고장 모델
 Fig. 1. Fault models of PLA.

III. 테스트가 용이한 PLA의 구성

PLA는 다양한 기술로 실현될 수 있는데 최근 논리 회로의 고밀도 설계가 가능한 MOS기술이 PLA 설계에 널리 사용되고 있다. 그림 2에 제안한 회로는 1개의 슈프트레지스터 열을 추가하여 각 적항선을 개별적으로 선택하고, 선택한 적항선을 활성화시킬 수 있는 입력벡터를 인가해 줌으로써 활성화된 경로상의 고장이 회로 내 프로그램 상태의 영향을 받지 않고

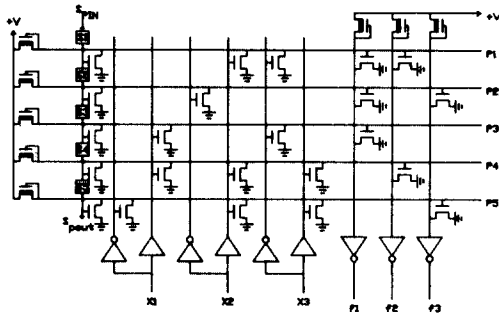


그림 2. 제안한 기본 testable PLA
 Fig. 2. The proposed basic testable PLA.

출력에 전달되어 테스트가 용이하도록 설계한 NOR-NOR논리 MOS PLA회로이고 그림 3은 그에 대한 특성 matrix(personality matrix)를 나타낸다.

$$\begin{array}{l}
 R_1 \\
 R_2 \\
 R_3 \\
 R_4 \\
 R_5
 \end{array}
 \left|
 \begin{array}{cccccc}
 2 & 0 & 1 & 1 & 1 & 0 \\
 2 & 1 & 2 & 1 & 0 & 1 \\
 0 & 2 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 \\
 1 & 0 & 0 & 0 & 0 & 1
 \end{array}
 \right.$$

	AND array	OR array
1	connection to uncomplemented input	connection to output
0	connection to complemented input	no connection to output
2	no connection	not used

그림 3. 그림 2의 PLA에 대한 특성 matrix
 Fig. 3. Personality matrix for the PLA in fig2.

IV. 테스트 생성 알고리즘

1. 테스트 생성 원리

부가된 슈프트레지스터를 이용, 선택하고자 하는 적항선을 1로 세트(set)한다.

이때 선택한 적항선을 활성화시키기 위해 AND array의 프로그램 상태에 따라 필요한 입력벡터를 인가한다. 이렇게 하므로써 그림 4와 같이 선택된 적항선을 통해 고장이 외부출력으로 전달되도록 만든다.

2. 테스트 알고리즘

테스트가 용이하도록 설계된 PLA를 테스트하기

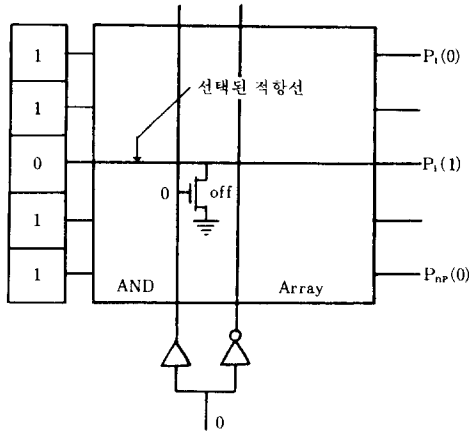


그림 4. 테스트 생성 원리
Fig. 4. Principle of test generation.

위한 테스트 집합의 생성 절차는 다음과 같다.

[테스트 집합 생성 절차 1]

(절차 1) 입력된 특성 matrix의 내용을 읽는다.

(절차 2) 슈프트레지스터를 전부 1로 세트하여 모든 출력값이 0가 되게한다.

(테스트 패턴 T_0)

(절차 3) 특성 matrix의 행 R_i 에 대하여 테스트 패턴 T_1 를 생성한다.

(절차 4) 절차 3을 $i=1$ 부터 $i=n_p$ (적항선의 갯수)까지 반복 수행한다. 테스트패턴 T_i 를 생성하는 알고리즘은 다음과 같다.

알고리즘 1 :

단계 1) 선택한 P_i 의 값을 1로 하는 입력벡터를 구한다. 즉 이 입력벡터는 P_i 에 연결된 소자들을 off시키는 입력으로서 R_{ai} 가 되고, 이 때의 정상출력은 R_{ai} 가 된다(테스트 입력 t_1).

(i) R_{ai} 의 내용중 don't care에 대한 입력값은 임의로 줄 수 있으므로 편의상 좌측에 인접한 값으로한다.

(ii) 최초의 R_{ai} 의 내용이 don't care이면 우측에 인접한 값을 그 때의 입력값으로 한다.

(단계 2) 테스트 입력 t_1 중 R_{ai} 의 don't care에 대한 입력만을 그 보수값으로 바꾼다. 이 때의 정상출력값은 R_{oi} 가 된다(테스트 입력 t_2). don't care가 없는 경우 t_2 는 필요없다.

(단계 3) 테스트 입력 t_2 중 don't care가 아닌 R_{ai} 의 입력에 대해 하나씩 그 보수값을 취하여 준다. 이 때의 정상출력값은 0가 된다(테스트 t_3). 즉 t_3 의 입력 패턴은 t_2 와 1개 비트만이 다르게 된다.

(i) row R_{ai} 의 배열이 120인 경우에는 don't care의 우측값 0을 그 보수값1로 취해줄 때, 이

don't care에 대해서도 그와 같은 값을 취해 준다.

(ii) 마찬가지로 021의 경우도 don't care 우측 값을 보수값으로 취할 때 이 don't care에 대해서도 같은 값을 취해준다

(iii) 020나 121인 경우에도 don't care의 값은 바꾸어 주지 않는다.

여기서

P_i 는 선택하여 활성화 되는 적항선

R_i 는 특성 matrix의 i 번째 row의 내용

R_{ai} 는 R_i 중 AND array 부분의 내용

R_{oi} 는 R_i 중 OR array 부분의 내용을 가르킨다.

테스트 절차에 들어가기 전에 먼저 부가회로인 슈프트레지스터를 테스트한다.

슈프트레지스터를 테스트하기 위해 S_{pin} 에 n_p 개의 0 값과 1 값을 각각 인가하고 S_{pout} 을 조사한다. 제안한 방식으로 설계된 PLA내 존재하는 고장 형태 A에서 K까지의 모든 단일고장의 집합을 F_s 라 할 때 다음 정리가 성립한다.

(정리 1) 생성된 테스트 패턴 T_0 와 T_1 는 집합 F_s 에 속하는 고장을 검출할 수 있다.

(증명) 테스트 패턴 T_0 는 정상 동작 시 적항선 및 외부출력을 0로 만드는 패턴이므로 외부출력을 1로 만드는 고장인 형태 H중 stuck-at-0 고장, 형태 B의 stuck-at-1 고장을 검출할 수 있다. 테스트 패턴 T_1 중 t_1 은 선택한 적항선을 1로 만드는 패턴이므로 이 적항선을 0으로 만드는 고장인 형태 A의 stuck-at-1 고장, 형태 B의 stuck-at-0 고장, 형태 E의 고장, 형태 G의 extra device고장, 형태 H의 stuck-at-1 고장, 형태 F, I, J 및 K의 고장을 검출할 수 있다. t_2 는 t_1 에서 don't care에 대한 입력을 보수값으로 바꾸어 준 패턴이므로 형태 G의 extra device 고장을 검출할 수 있다. t_3 는 R_{ai} 가 don't care가 아닌 입력 비트선을 1로 만드는 패턴이므로 형태 A의 stuck-at-0 고장, 형태 C의 고장, 형태 D의 고장, 형태 G의 missing device 고장을 검출할 수 있다. (증명끝)

그림 3의 특성 matrix로부터 적항선 P_i 및 P_j 에 대한 테스트 패턴을 구한 예를 표 1에 나타낸다.

1비트 디코더를 갖는 PLA내 존재하는 모든 단일고장(F_s)을 검출하기 위한 테스트 길이(테스트 패턴의 수)는 $T_L = 1 + C + 2n_p - D$ 가 된다. 여기서 슈프트레지스터의 테스트는 제외된 것이며, 식에서 C는 And array의 점점수, n_p 는 적항선의 수, D는 don't care가 없는 적항선의 수를 나타낸다.

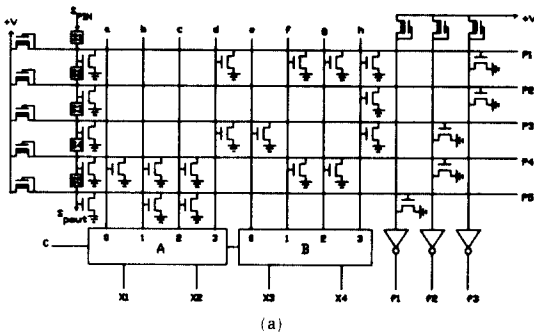
V. 고장 검출이 용이한 2비트 decoded PLA의 구성

표 1. 입력화일 및 적항선 P₁과 P₃에 대한 테스트 패턴

Table 1. Input file and test pattern for P₁, P₃.

		S1	S2	S3	S4	S5	X1	X2	X3	F1	F2	F3
# 3 3 3 5	1	1	1	1	1	1	-	-	-	0	0	0
	0	1	1	1	1	1	0	0	1	1	1	0
	0	1	1	1	1	1	1	0	1	1	1	0
	0	1	1	1	1	1	1	1	1	0	0	0
	0	1	1	1	1	1	1	0	0	0	0	0
2 0 1	1 1 0											
2 1 2	1 0 1											
0 2 1	1 0 0											
0 0 0	0 1 0											
1 0 0	0 0 1											

그림 5에 제안한 (a)의 회로는 (b)의 논리함수로 표현되는 carry look ahead 회로를 그림 6과 같은 제어입력을 갖는 2비트 디코우더를 사용하여 2비트 decoded PLA를 구성한 것이다. 그림 7은 2비트 decoded PLA에서 부가회로의 기능 및 테스트 생성 원리를 나타낸 것이다.



$$\begin{aligned}
 f_1 &= x_1x_2' + x_1'x_2 \\
 f_2 &= x_3'x_4x_1x_2 + x_3x_4x_2' + x_3'x_4x_1' + x_3x_4x_1x_2' + x_3x_4x_1x_2 \\
 f_3 &= x_3x_4 + x_1x_2x_4 + x_1x_2x_3
 \end{aligned}$$

그림 5. 테스트가 용이한 2비트 decoded PLA (carry look ahead 회로)

Fig. 5. A testable 2bit decoded PLA (Carry Look Ahead circuit).

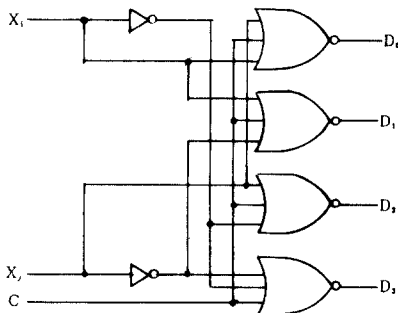


그림 6. 제어입력을 갖는 2비트 디코우더
Fig. 6. 2bit decoder with control input.

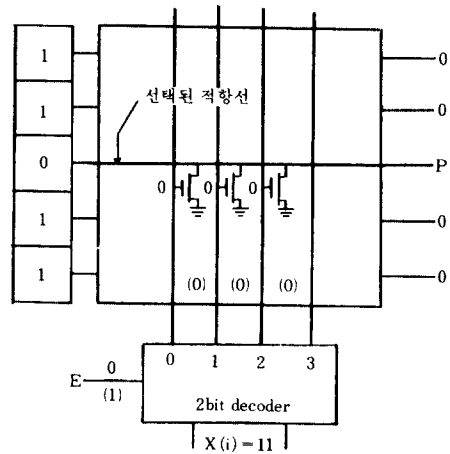


그림 7. 부가회로의 기능
Fig. 7. Function of extra hardware.

2비트 decoded PLA의 테스트 생성절차는 다음과 같다.

[테스트 생성절차 2]

테스트 생성절차 1과 동일하며 절차 3의 T₁ 생성알고리즘은 다음과 같다.

[알고리즘 2]

(단계 1) 모든 쉬프트레지스터 S(i) = 1로, 제어입력 C = 1로 주어 쉬프트레지스터의 출력과 연결된 소자를 테스트한다. 이 때 정상출력은 0가 된다(테스트 입력 t₁).

(단계 2) i = 1부터 n_p까지 차례로 S(i) = 0로 하고 고장이 활성화 될 수 있도록 입력벡터 *(i)를 인가한다. 여기서 디코우더의 입력벡터 *(i)를 인가하는 방법은 다음과 같다.

쉬프트레지스터에 가까운 디코우더부터 그 입력을 00, 01, 10, 11의 순서로 인가하고 나머지 디코우더의 입력벡터는 디코우더의 출력과 적항선이 교차하는 And array의 프로그램된 소자를 off 시킬 수 있는 입력을 인가한다(테스트 입력 t₂). 이 때 선택된 적항선이 입력벡터에 의해 0가 되었을 때 정상출력은 0가 되고 1이 되었을 때는 R_{0i}가 된다.

제어입력을 갖는 2비트 디코우더를 사용하면 PLA의 프로그램 상태에 따라 디코우더의 입력값을 일일이 구하지 않고, 제어입력값을 1로 세트하므로써 쉬프트레지스터에 의해서 선택된 적항선을 활성화 시킬 수 있다. 그림 5에서 제어입력 C = 1이고 S₁ = 0, S₂ ~ S₅ = 1이면 적항선 P₁이 1로 활성화되어 적항선 P₁의 stuck-at-0 고장, 비트선 e와 적항선 P₁의 단락 고장등이 동시에 검출된다. 만약 C = 0이고 S₁ = 0,

$S_2 \sim S_3 = 1$ 이면 적항선 P_1 의 stuck-at-0 고장은 입력 벡터 $X_1 X_2 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$, $X_3 X_4 = (0 \ 0)$ 를 인가하므로써 검출 가능하다. 그런데 비트선 e와 적항선 P_1 의 단락고장을 검출하기 위해서는 비트선을 모두 0으로 만들어야 하는데 $X_3 X_4 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{pmatrix}$ 중 어느 입력벡터로도 가능하지 않다.

(정리 2) 생성된 테스트 패턴 T_1 는 제안한 2비트 decoded PLA내 존재하는 고장들의 집합(F_2)에 속하는 모든 단일 고장을 검출할 수 있다.

(증명) 테스트 패턴 T_0 는 정상동작 시 적항선 및 외부출력을 0로 만드는 패턴이므로 외부출력을 1로 만드는 고장인 형태 H중 stuck-at-0 고장, 형태 B의 stuck-at-1 고장, 형태 J의 고장을 검출할 수 있다. 또 테스트 패턴 T_1 중 t_1 은 모든 적항선을 0로 만드는 테스트 패턴이므로 적항선을 1로 만드는 고장인 쉬프트레지스터에 연결된 소자의 missing 고장을 검출할 수 있다. 테스트 패턴 t_2 는 적항선을 하나씩 선택한후 비트선을 순서적으로 0 또는 1로 만드는 패턴이므로 나머지 고장들을 검출할 수 있다(증명 끝).

2비트 decoded PLA내 존재하는 형태 A에서 K까지의 단일고장들을 검출하기 위한 테스트 길이는 $T_L = 2n_p \cdot n_i + 1 + n_p - R$ 가 된다. 여기서 쉬프트레지스터의 테스트는 제외된 것이며, 식에서 n_p 는 적항선의 수, n_i 는 입력변수의 수, R은 디코우더의 출력선과 적항선들이 교차하는 각 디코우더 영역들 중, 프로그램되지 않은 최소 점접수들의 합이다.

그림 6에서 n_p 가 5, n_i 가 4, R이 $9(1+3+2+1+1+2)$ 이고 $T_L=37$ 이 된다. PLA의 프로그램률이 낮을수록 R이 커지고 상대적으로 T_L 은 짧아진다. 표 2는 그림 5의 회로에 대한 입력화일 및 적항선 P_1 및 P_2 에 대한 테스트 패턴을 나타낸 것이다.

VI. 다중고장의 검출

지금까지 발표된 기존 논문들에서는 단락고장 및 서로 다른 모델에 대한 다중고장을 충분히 고려하지 못하고 있는데^{1,4,6,7,11} 고장검출률을 높이기 위해 실제로 고려해야할 다중고장은 단락고장을 포함한 동일 종류 및 서로 다른 종류의 단일고장이 복수개 발생하는 다중고장이다.

(정의 1) 서로 다른 모델의 고장이 다중으로 발생했을 때 이중 한 고장이 다른 고장에 관계없이 외부출력값을 단독으로 결정할 때 이 고장은 다른 고장에 대해 지배적(dominant)이라 한다.

활성화된 경로상의 고장은 활성화되지 않은 경로상의 고장에 대해 지배적이고, 활성화된 경로상의 다중

표 2. 그림 5의 회로에 대한 입력화일 및 적항선 P_1, P_2 에 대한 테스트 패턴

Table 2. Input file & test pattern for the product line P_1, P_2 of the circuit in fig.6.

```

:-----:
:          main inputs & decoder inputs          :
: S1 S2 S3 S4 S5   x1 x2 x3 x4   C   f1 f2 f3   :
: 1 1 1 1 1   - - - -   1 0 0 0   :
:-----:
:           5           :
:           3           :
:           2           :
222 21 21 11 00 1   :
222 22 22 21 00 1   :
222 21 12 21 01 0   :
114 2 21 12 01 0   :
2 1 1 2 22 22 1 0 0  :
:-----:
: 0 1 1 1 1   - - - -   1 0 0 1   :
: 0 1 1 1 1   0 0 0 0 0 0 0 0 0 1   :
: 0 1 1 1 1   0 1 0 0 0 0 0 0 0 1   :
: 0 1 1 1 1   1 0 0 0 0 0 0 0 0 1   :
: 0 1 1 1 1   1 1 0 0 0 0 0 0 0 0   :
: 0 1 1 1 1   0 0 0 0 1 0 0 0 0 0   :
:-----:
:           1 0 0 0 0   :
: 0 1 1 1 1   0 0 0 1 0 0 0 0 0 0   :
:           0 1 0 0 0   :
: 0 1 1 1 1   1 0 0 0 1 1 0 0 0 0   :
:           0 1 0 0 0   :
:-----:
: 1 0 1 1 1   - - - -   1 0 0 1   :
: 1 0 1 1 1   0 0 0 0 0 0 0 0 0 1   :
:           0 1 0 0 0   :
: 1 0 1 1 1   0 1 0 0 0 0 0 0 0 1   :
:           0 1 0 0 0   :
: 1 0 1 1 1   1 0 0 0 0 0 0 0 0 1   :
:           0 1 0 0 0   :
: 1 0 1 1 1   1 1 0 0 0 0 0 0 0 1   :
:           0 1 0 0 0   :
: 1 0 1 1 1   0 0 1 1 1 0 0 0 0 0   :
:           0 1 0 0 0   :
:           1 0 0 0 0   :
:           1 1 0 0 0   :
:-----:

```

고장은 그 가운데 외부출력에 가장 가깝게 영향을 줄 수 있는 고장이 다른 고장들에 대해 지배적이다. 다음 예 1은 지배고장을 이용, 다중고장이 검출됨을 보여준다.

(예 1) 그림 8에서 P_1 과 b의 교차점의 extra device 고장 및 P_1 과 P_2 의 단락고장, 그리고 P_1 의 stuck-at-1 고장이 동시에 발생했을 때 이 중 지배고장은 P_1 의 stuck-at-1 고장이므로 테스트 패턴

$$\begin{matrix} S_1 & S_2 & S_3 & S_4 & X_1 & X_2 & X_3 & X_4 & C \\ 1 & 1 & 1 & 1 & - & - & - & - & 1 \text{로서} \end{matrix}$$

이 다중고장을 검출할 수 있다.

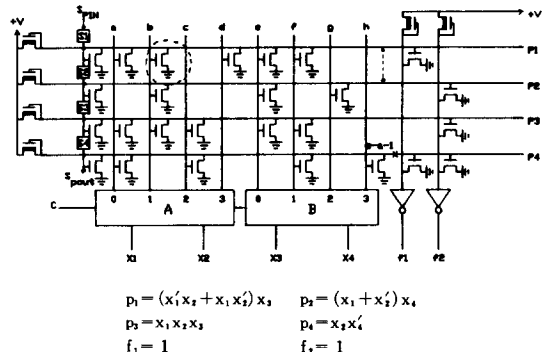


그림 8. 다중고장 발생 시 지배고장의 효과
Fig. 8. Effect of dominant fault for multiple faults.

Ⅶ. 고장의 등가관계

(정의 2) 제안한 테스트 방법에 의해 PLA를 테스트할 때 활성화된 경로상의 서로 다른 고장이 하나의 테스트 패턴에 의해 출력에 동일한 값으로 나타날 때 이러한 고장들은 서로 등가관계라고 한다. 다음 (1)의 a), b), c), d), e), f)는 서로 등가관계의 고장들이고 (2)의 a), b), c), d), e)도 서로 등가관계의 고장들이다.

- (1) a) 선택된 외부출력선의 stuck-at-1 고장
- b) 선택된 적항선의 stuck-at-1 고장
- c) 디코우더 출력이 1인 비트선과 적항선이 교차하는 지점의 missing device 고장
- d) 디코우더 출력이 1인 비트선과 0인 비트선간의 단락고장
- e) 디코우더 출력이 1인 비트선과 선택되지 않은 적항선과의 단락고장
- f) OR array내 선택된 출력선의 stuck-at-0 고장
- (2) a) 선택된 외부출력선의 stuck-at-0 고장
- b) 디코우더 출력이 1인 비트선과 적항선이 교차하는 곳의 extra device 고장
- c) 선택된 적항선과 다른 적항선과의 단락고장
- d) 디코우더 출력이 0이고 비트선과 적항선이 만나는 교차점에 프로그램 되어 있을 때 비트선의 stuck-at-1 고장
- e) 디코우더 출력이 0인 비트선과 쉬프트 레지스터에 의해 선택된 적항선의 단락고장

일반적으로 등가관계의 고장들이 동시에 발생하는 다중고장의 경우, 마스킹(masking)현상이 일어나 고장 검출이 불가능한 경우가 생길 수 있다.

예 2는 다중고장 발생 시 생기는 마스킹 조건 하에서도 제안한 테스트 방식에 의해 다중고장이 검출됨을 보여 주고 있다.

(예 2) 그림 9에서 P₁과 a의 교차점에 missing device 고장과 등가관계를 갖는 P₁의 stuck-at-0 고장이 동시에 발생했을 때, 테스트 패턴

$$\begin{matrix} S_1 & S_2 & S_3 & S_4 & X_1 & X_2 & X_3 & X_4 & C \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{matrix}$$

을 인가하면 missing device 고장이 stuck-at-0 고장에 의해 마스킹되어 외부출력에 전달되지 않는다. 그러나 지배고장인 P₁의 stuck-at-0 고장을 검출하기 위한 테스트 패턴

$$\begin{matrix} S_1 & S_2 & S_3 & S_4 & X_1 & X_2 & X_3 & X_4 & C \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{matrix}$$

또는 0 1 1 1 - - - 1

을 인가하므로써 이 다중고장을 검출할 수 있다.

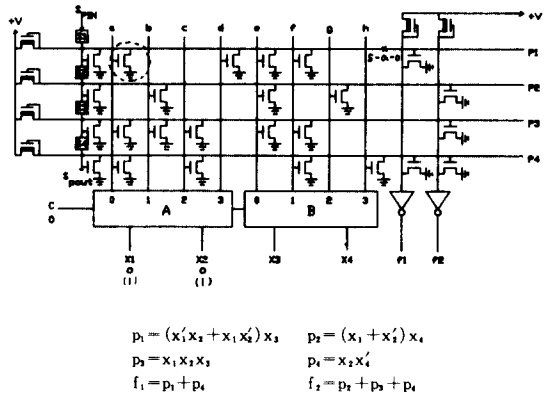


그림 9. 마스킹 조건 하에서의 지배고장의 효과
Fig. 9. Effect of dominant fault in masking condition.

제안된 방식으로 설계된 PLA내 발생하는 다중고장에 대하여 다음의 정리가 성립한다.

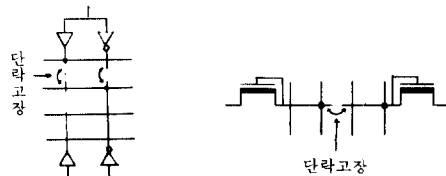
(정리 3) 단일고장 검출용 테스트 집합으로 지배고장이 존재하는, 형태 A에서 K까지의 고장들에 의한 어떠한 다중고장도 검출 가능하다.

(증명) 정리 2에 의해서 임의의 다중고장이 발생할 때 지배고장에 대한 테스트 패턴을 인가하면 외부출력으로 고장값이 전달되어 고장 유무를 판단할 수 있다.

Ⅷ. 테스트가 용이한 Floded PLA의 설계

1. Floded PLA에서 고려해야 할 고장 모델

Floded PLA에서는 일반 PLA에서 대상으로 하는 고장 모델들 외에 그림10에 도시한 것 같은 folding에 의해 분리된 신호선들의 단락고장을 고려해야 한다.



(a) Column folding (b) Row folding

그림10. Floded PLA에서 추가로 고려해야 할 고장
Fig. 10. Special fault model in floded PLA.

2. 테스트가 용이한 설계 및 테스트집합 생성

그림11은, 테스트가 용이하도록 설계된 floded PLA를 나타낸 것이다. 일반 PLA의 테스트를 위한 설계 방식과 같이 적항선을 제어하기 위한 쉬프트레지스터열을 부가회로로 사용하고 있다. 그림11의 PLA에서

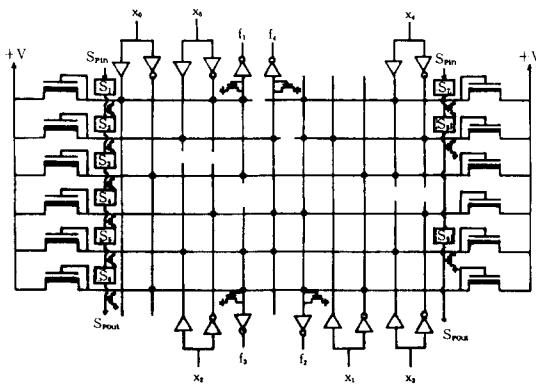


그림 11. 테스트가 용이한 Folded PLA의 설계
Fig. 11. A testable design of folded PLA.

고장들을 테스트하기 위해서 column folding의 경우 현재 테스트 입력패턴이 인가되는 입력선과 folding방이 되는 반대쪽 입력선에는 그 테스트 입력패턴의 보수값을 인가한다. row folding의 경우에는 테스트 모드(mode)시 하나의 적항선만이 1로 세트되고 folding된 선의 반대쪽 쉬프트레지스터에는 0이 세트되어 있으므로 일반 PLA를 테스트하는 테스트 집합으로 이 고장을 검출할 수 있다. 따라서 folded PLA의 테스트 집합은 III 절의 테스트 절차에서 테스트 집합 T_1 에 해당 신호선에 대한 테스트 입력신호의 보수값을 추가해 줌으로써 얻을 수 있다. 이 수정된

테스트 집합으로 folding된 신호선 간의 단락고장을 포함한 모든 고장(F_s)을 검출할 수 있다. 다음 표3은 그림11에서 적항선 P_i 를 선택하여 테스트 패턴을 구한 것이다.

표 3. 그림11의 Folded PLA에 입력파일 및 테스트 패턴

Table 3. Input file & test pattern for circuit in fig 11.

f	x0 x5	f1 f4	x4
2			
3	x2	f3 f2	x1 x3
2	0 1	1 0	. .
6	. 0	0 . 0	- -
0 1 7 0 1 2 0	1 1	0 0	. .
2 3 0 7 1 0 1	. 0	0 . 0	- -
1 2 4 0 0 1 3	0 0	0 0	. .
2 1 0 1 1 0 1	. 1	0 . 0	- -
2 0 7 1 0 2 0			
5 2 1 3 1 1 2			

IX. 종래 방식과의 비교

제한한 1비트 디코더를 갖는 PLA, 2비트 decoded PLA 및 folded PLA를 테스트하기 위한 테스트 생성 알고리즘을 및 PC/AT(XENIX) 상에서 C언어로 프로그래밍하여 임의의 PLA에 대한 특성matrix를 입력 파일로 주면 자동적으로 테스트가 생성되도록 하였다. 다음 표 4는 종래의 대표적인 PLA 설계방식^{15,6,11}

표 4. 제안된 방식과 종래방식과의 특징비교
Table 4. Comparison of proposed method and conventional method.

	HO-PLA	FKO-PLA	SKF-PLA	Proposed PLA
Decoder	2 bit	2 bit	1 bit	2 bit
SR 열의 수	1	1	2	1
부가 Logic	decoder 1개 AND OR Gate EOR Tree	AND OR Gate 2 개의 EOR Cascade	AND OR Gate	×
부가 입력	$\text{Log}(P+1)+2$	5	3	1
부가 출력	4	2	2	×
고장검출능력	단일고장	단일, 다중 고장일부	다중고장	단일, 다중고장
테스트길이	$2.5ni+2np+5$	$2ni+5np$	$np(2ni+1)+4ni+7$	1bit: $1+C+2np-D$ 2bit: $2ni \cdot np+1+np-R$
테스트특성	함수독립적	함수독립적	함수독립적 단, 출력은 종속적	함수종속적
고장모델	stuck-at 점 점 단락고장	stuck-at 점 점 고 장	stuck-at 점 점 고 장	stuck-at 점 점 단락고장

과 제안된 설계방식의 특징을 비교한 것이다.

X. 결 론

본 논문은 부가회로가 적고 단일고장 및 다중고장에 대한 고장검출률이 높은 테스트가 용이한 PLA의 설계방식을 제안하였다. 제안한 방식의 특징을 요약하면 다음과 같다.

- 1) 부가 하드웨어가 적다.
 - 2) 테스트 생성이 함수 독립적인 설계방식들에서 발생하는 exclusive-OR tree에 의한 시간지연이 없다.
 - 3) 기존 방법^{11-14, 16}에서 고려하지 않거나 부분적으로 검출 가능하던 인접 신호선 사이의 단락고장과 인접하지 않은 신호선 사이의 단락고장이 검출 가능하며 또한 동일 종류 및 서로 다른 종류의 고장이 복수개 발생하는 다중고장 거의 대부분이 검출 가능하게 되어 기존 방법보다 높은 고장검출률을 갖는다.
 - 4) 테스트 생성이 함수 종속적이나 특성 matrix로부터 손쉽게 테스트집합을 생성할 수 있다.
 - 5) 2비트 decoded PLA의 경우 PLA의 프로그램 밀도가 낮을수록 테스트 수가 줄어들어 실제 PLA에 적용 시 테스트 생성이 간단해 진다.
 - 6) Folded PLA에도 적용 가능하다.
- 이상의 특징이외에도 부가 입출력이 극히 적어 외부 단자로 인해 칩 사이즈가 크게 증가하는 것을 피할 수 있다.

종래의 방법에서 구체적으로 다루지 않던 단락고장의 테스트를 가능케 했으며 특히 인접하지 않은 신호선 간의 단락고장도 검출가능케 하였다. 그리고 고장의 등가관계 및 지배고장을 정의하고 이를 이용 다중고장을 해석하였다. PLA의 사용이 점점 증대되고 있는 현재 본 설계방식은 설계 및 테스트의 편의와 비용절감에 기여할 수 있을 것으로 기대된다.

참 考 文 献

[1] J.E. Smith, "Detection of Faults in Programmable Logic Arrays," *IEEE Trans. Comput.*, vol. C-28, pp. 845-853, Nov. 1979.

[2] D.L. Ostapko, S.J. Hong, "Fault Analysis and Test Generation for Programmable Logic Arrays (PLA's)," *IEEE Trans. Comput.*, vol. C-28, pp. 617-627, Sep. 1979.

[3] E.B. Eichelberger, E. Lindbloom, "A Heuristic Test-Pattern Generator for Programmable Logic Arrays," *IBM J. Res. Develop.*, vol. 24, pp. 15-22, Jan.

1980.

[4] H. Fujiwara, K. Kinoshita, "A Design of Programmable Logic Arrays with Universal Tests," *IEEE Trans. Comput.*, vol. C-30, pp. 823-828, Nov. 1981.

[5] S.J. Hong, D.L. Ostapko, *FITPLA: A Programmable Logic Array for Function Independent Testing*, 10th Int. Symp. F.T.C., pp. 131-136, June 1980.

[6] K.K. Saluja, K. Kinoshita, H. Fujiwara, "An Easily Testable Design of Programmable Logic Arrays for Multiple Faults," *IEEE Trans. Comput.*, vol C-32, pp. 1038-1046, Nov. 1983.

[7] J. Khakbaz, *A Testable PLA Design with Low Overhead and High Fault Coverage*. 13th Symp. F.T.C., pp. 426-429, 1983.

[8] C.W. Cha, *A Testing Strategy for PLAs*. 15th Design Automation Conf., pp. 328-334, June 1978.

[9] G.D. Hachtel, A.R. Newton, A.L. Sangiovanni-Vincentelli, "An Algorithm of Optimal PLA Folding," *IEEE Trans. CAD*, vol. CAD-1, pp. 63-76, April 1982.

[10] W. Grass, *A Depth-First Branch-and-Bound Algorithm for Optimal PLA Folding*, 19th Design Automation Conf., pp. 133-140, 1982.

[11] H. Fujiwara, K. Kinoshita, and H. Ozaki, *Universal Test Sets for Programmable Logic Arrays*, in Dig. 10th Int. Symp. Fault-Tolerant Computing, pp. 137-142, Oct. 1980.

[12] 金殷成, 林寅七, "附加回路가 簡單한 PLA의 Testable Design", 大韓電子工學會 夏季綜合學術大會 論文集, vol. 7, no. 1, pp. 260-263, 1984. 7.

[13] J. Khakbaz, S.B. Nesbat, *Minimizing Extra Hardware for Fully Testable PLA Design*, Proceedings of the IEEE ICCAD, SANTA CLARA, CA, Nov., pp. 102-104, 1985.

[14] 李在昉, 金殷成, 林寅七, "多重故障 檢出이 가능하고 附加回路가 적은 Testable PLA의 設計", 大韓電子工學會 秋季綜合學術大會 論文集, vol. 8 no. 2, pp. 551~554, 1985.

[15] TsuToMu SASAO, "Input Variable Assignment and Output Phase Optimization of PLA's," *IEEE Trans. on Comput.*, vol. C-33, pp.879-894, Oct. 1984. *