論文 82-19-4-5

컴퓨터를 이용한 PLA의 故障 檢出에 관한 硏究

(A Study on the Computer Aided Fault Detection in PLAs)

林濟鐸*, 李斗秀*, 金熙碩*, 李殷烏**

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

要 約

PLA는 多入力-多出力에 적합하지만 入力變數의 증가와 積項의 증가에 따라 종래의 방법으로는 테스트 入力의 생성이 곤란하게 된다.

본 논문에서는 이와 같은 테스트 入力을 구하기 위하여 sharp 연산과 cap 연산을 동시에 처리하는 效果的인 컴퓨터 알고리듬을 구성하고 이것을 프로그램에 의해서 실증하였다.

Abstract

It is a time-consuming work to generate test inputs of a PLA as inputs and product terms are increasing. In this paper we design a computer algorithm which is efficient for processing sharp and cap operator simultaneously. An assembly language program is coded and run successfully.

I. 序 論

일반적인 組合論理回路를 2-level AND-OR 게이트의 구조로 치환하여 프로그램을 하므로서 원하는 기능을 가지도록 배열한 것이 PLA(programmable logic array)이며 多入力-多出力의 論理函數를 실현하는데 적합하다.

PLA에서는 fusing을 하므로써 사용자가 원하는 出力을 얻게 되는데 使用者가 fusing을 直接할 수 있도록 고안된 것이 FPLA(field programmable logic array)이다.

Fleisher와 Maissel[1]에 依하여 제안된 PLA는 기억장치와 흡사한 규칙적인 구조로 되어 있으므로 LSI/VLSI의 設計에 그 응용도가 높다.

PLA는 fusing에 의하여 프로그래밍을 하므로 일반적인 組合論理回路에서는 존재하지 않는 fusing에 의한 故障이 발생하게 된다. 이 故障을 檢出하기 위하여 Smith[2]는 sharp연산(#)을 사용하여 테스트 入力을 구하였으며 Agarwal[3]은 SAE(stuck-at-equivalence) 回路로 치환하여 multiple fault 檢出에 대하여 논하였다. Hong과 Ostapko[4]는 PLA의 入力 decoder 部分이 2入力 decoder로 된 경우를 기본으로 하여 테스트 入力을 구하는 방법을 제안하였다.

저자의 論文[5]에서는 위의 論文[2],[3],[4]에서 해결하지 못한 고장 위치 파빙과 redundant 상태의 고장을 검출하는 방법을 제안하였으며, sharp 연산과 cap연산을 사용하여 고장 상태를 고려하지 않고 직접 PLA의 상태표로 부터 테스트 入力을 求하는 방법을 제안하였다.

그러나 [5]에서 제안한 방법으로 테스트 入力을 구하는 경우 PLA의 入力 變數와 積項이 증가하게 되면 테스트 入力을 求하기가 어렵게 된다.

本 論文에서는 이와 같은 문제점을 해결하기 위해 [5]에서 제안된 고장 검출의 방법을 改善해 컴퓨터

*正會員, **準會員, 漢陽大學校 工科大學 電子工學科
(Dept. of Elec. Eng., Hangyang Univ.)

接受日字 : 1982年 4月 13日

를 사용하여 테스트 入力을 효율적으로 求할 수 있는 알고리듬을 제안하였다. 그리고 PLA의 fusing 되는 부분에서 발생하는 고장을 分析, 分類하였으며, 特히 bridge 형태의 고장으로 치환할 수 있는 고장에 對하여도 論하였다. 그러므로 sharp 연산과 cap연산을 동시에 처리하여 PLA의 상태표로 부터 직접 테스트 入力을 구하는 것이 가능하게 되었으며 이를 어셈블리 언어로 프로그램하여 실행하므로써 효과적인 결과를 얻은 것이다.

Ⅱ. PLA의 一般的인 特性

PLA는 그림 1과 같은 구조로 되어 있으며, 入力 decoder에는 보수상태(0)와 정상상태(1)의 2 가지 상태가 存在한다. 그림 1에서 dot는 fusing을 하지 않은 상태를 나타내며, 한 入力 變數에 對하여 2 가지 상태

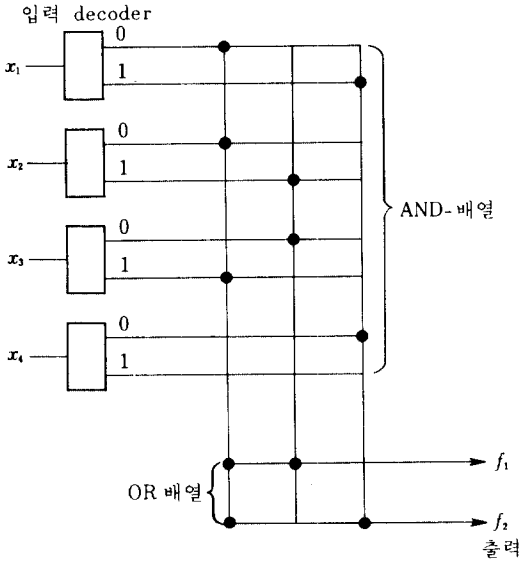


그림1. PLA의 기본 구조
Fig. 1. Basic structure of PLA.

표 1. PLA의 상태표
Table 1. A cubical notation of PLA.

x_1	x_2	x_3	x_4	f_1	f_2
0	0	1	X	1	1
X	1	0	X	1	0
1	X	X	0	0	1

를 모두 fusing하면 don't care상태 (X)가 되므로 PLA가 정상적으로 動作하려면 적어도 1個 이상이 fusing 되어야만 한다.

표 1은 그림 1의 상태를 나타내며 이것을 sum-of-product의 형태로 표시하면

$f_1 = \bar{x}_1 \bar{x}_2 x_3 + x_2 \bar{x}_3$

$f_2 = \bar{x}_1 \bar{x}_2 x_3 + x_1 \bar{x}_4$ 가 된다.

Ⅲ. 故障 상태의 分析

PLA의 入力에서 發生하는 單一 故障은 6 가지의 경우가 있으며 出力에서는 2 가지로 區分된다. 표 2는 故障의 상태를 나타내며, 9와 10은 fusing 되는 部分이 바뀌었으므로 回路를 살펴보면 二重 故障이 된다.

7과 8은 OR배열에서 發生하는 故障으로 出力에서 積項에 영향을 준다. 特히 5와 6의 고장은 그림 2 (a)와 같이 정상상태와 보수상태가 모두 連結되어 있으므로 x에 어떠 값을 입가해도 出力은 항상 0이 되므로 그림2 (b)의 bridge 형 故障으로 치환할 수 있다.

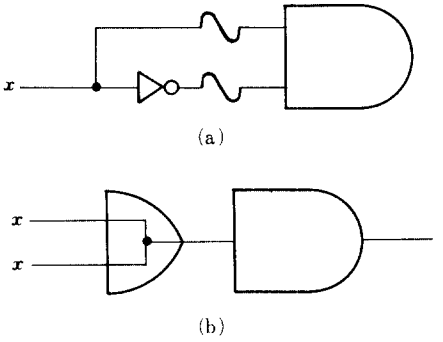


그림2. (a) 5와 6의 고장 (b) Bridge형 고장
Fig. 2. (a) Fault of 5 and 6. (b) Bridge type fault.

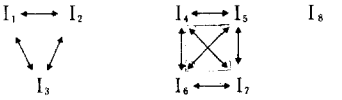
Ⅳ. 테스트 入力 생성 알고리듬

우선 sharp 연산과 cap의 연산을 이용하여 테스트 入力을 생성하는 방법에 對하여 論하기로 한다.

1) 各 入力들 사이의 관계를 조사하기 위하여 sharp(#) 연산을 한다.

2) Cap의 관계가 있는 入力들 間에 cap의 연산을 하여 테스트 入力을 求한다.

入力의 數가 8個인 경우 (1)의 연산을 한 結果가 다음과 같을 때



(단 "$\longleftrightarrow$"는 cap의 관계가 있는 入力을 나타냄.)

표 2. 고장의 분류

Table 2. Classification of faults.

	정 상	고 장	
	x	x	
1	1	X	x→X(Dont care)
2	0	X	$\bar{x}$→X(Dont' care)
3	X	1	X→x
4	X	0	X→$\bar{x}$
5	1		⟶x→ x ·$\bar{x}$(B)
6	0		⟶$\bar{x}$ →$\bar{x}$ ·x(B)
7	f_n : 1	f_n : 0	積項이 fusing됨
8	f_n : 0	f_n : 1	積項이 fusing안됨
9	1	0	x→$\bar{x}$
10	0	1	$\bar{x}$→x

테스트 入力은 $T_1=I_1\cap I_2\cap I_3$, $T_2=I_4\cap I_5\cap I_6\cap I_7$, $T_3=I_8$의 3個가 된다. 特히 I_8은 어느 入力과도 cap의 관계가 없으므로 I_8 自信이 테스트 入力이 된다.

入力의 數(=積項의 數)가 작고, 入力 變數가 작을 경우에는 (1)과 (2)의 方法으로 故障 상태를 고려하지 않고 간단히 求할 수 있지만, 積項 의 數가 증가하고, 入力 變數가 증가하면 위 方法으로 테스트 入力을 求하기 어렵다. 그러므로 이와 같은 문제점을 해결하기 위하여 그림 3 의 알고리듬을 제안하여 컴퓨터로 처리할 수 있도록 하였다. Sharp 연산은 多入力의 변수를 最小化 하기 위하여 使用하는데 그 性質은 다음과 같다.

표 3. Sharp 연산표

Table 3. Coordinate sharp product.

a # b	b: 0	1	X
0	Z	ϕ	Z
a 1	ϕ	Z	Z
X	1	0	Z

$\begin{cases} a = 0\ 1 X\ 0 \\ b = X\ 1\ 0\ 0 \end{cases}$ 인 경우,

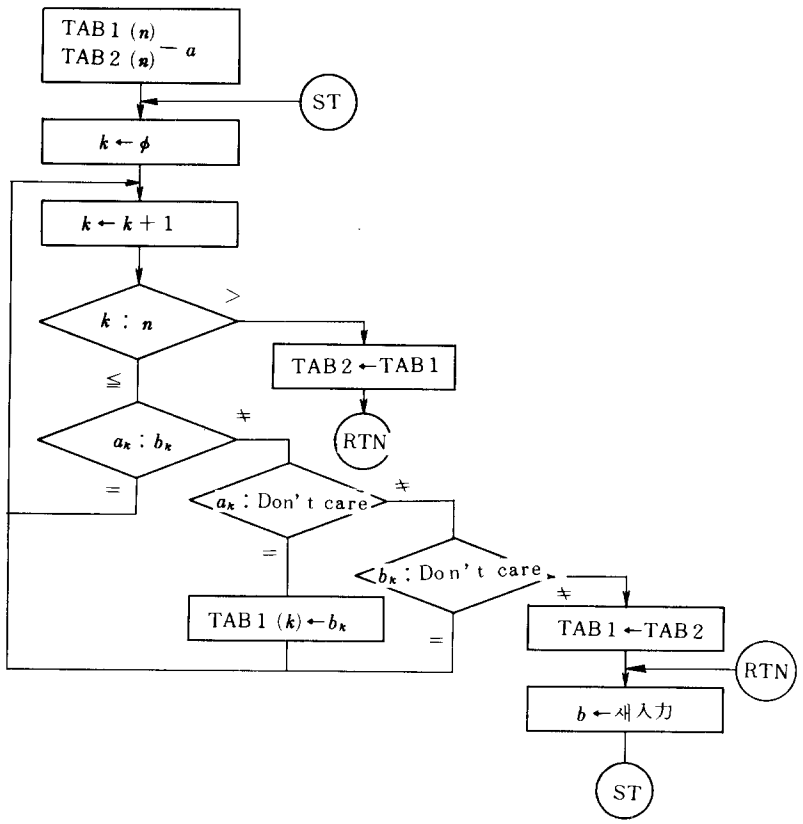


그림3. 테스트 入力 생성 알고리듬

Fig. 3. Algorithm for test input generation.

a#b는 a에서 b와 공통되는 요소를 제외한 요소를 나타낸다.

```
  0 1 X 0
#X 1 0 0
  Z Z 1 Z
```

표3에 依해 연산한 結果에 ϕ가 하나도 存在하지 않는다면 a와 b는 cap의 관계가 있지만 ϕ가 하나라도 存在한다면 cap의 관계가 없으므로 a가 主項(prime implicant)이 된다. 여기에서는 ϕ가 하나도 存在하지 않으므로 cap의 관계가 있다.

	00	01	11	10
00		b a	b	
01				
11				
10		a		

K-Map에서 보는 바와 같이 0100가 공통이므로 a#b는 0110가 된다. 그림3의 알고리듬에서는 sharp 연산을 수행하기 위해 入力들의 各 變數를 비교하여 양쪽 變數에 don't care 상태가 存在하지 않으면서 다른 경우(1)과 (0, 0과 1)에는 다음 入力을 선택하는 방법을 使用하였다. Cap의 연산을 수행하기 위해 各 변수가 같을 경우에는 다음 단계로 넘어가고 X; a(X→don't care, a∈0,1)인 경우 X←a로 치환하며 a;X인 경우에는 다음 단계로 넘어가는 방식을 使用하였다. 그러므로 入力 변수가 n個이고 積項이 m個인 경우 테스트 入力의 數 T는 $1 \leq T \leq m$이 된다.

T=1은 모든 入力이 cap의 관계가 있는 경우이며, T=m은 모든 入力이 cap의 관계가 없는 경우가 된다.

I_1= 0 1X 0 1X 1 0
I_2=X 1 0 0X 1XX
I_3=XX 0X 1 1 1 0

인 경우 이와 같은 알고리듬을 利用하여 求한 테스트 入力은 0 1 0 0 1 1 1 0가 된다. 테스트 入力을 求한 후에는 [5]에서 제안한 방법을 使用하여 고장 검출을 하게 된다. 그림4는 入力 변수가 16個, 積項이 48個인 경우 테스트 入力을 産出하기 위한 順序圖(flow-chart)를 나타내며 부록은 이것을 어셈블리 언어로 coding한 프로그램을 나타낸다.

Univac-9ϕ3ϕ을 사용하여 테스트 入力을 생성한 結果 CPU time이 約 1분 20초 정도가 소모되었으며, 入力변수가 16個, 積項이 96個인 경우에도 약 1분 21초가 소모되었다. 여기서 PLA의 入力은 random하게 인가하였으며, CPU time은 매번 入力을 바꾸게 하여 積項이 48個인 경우 6번의 결과를 平均한 값을 반올림 하였으며 積項이 96個인 경우도 6번의 결과를 평균한 값을 반올림하여 求한 값이다.

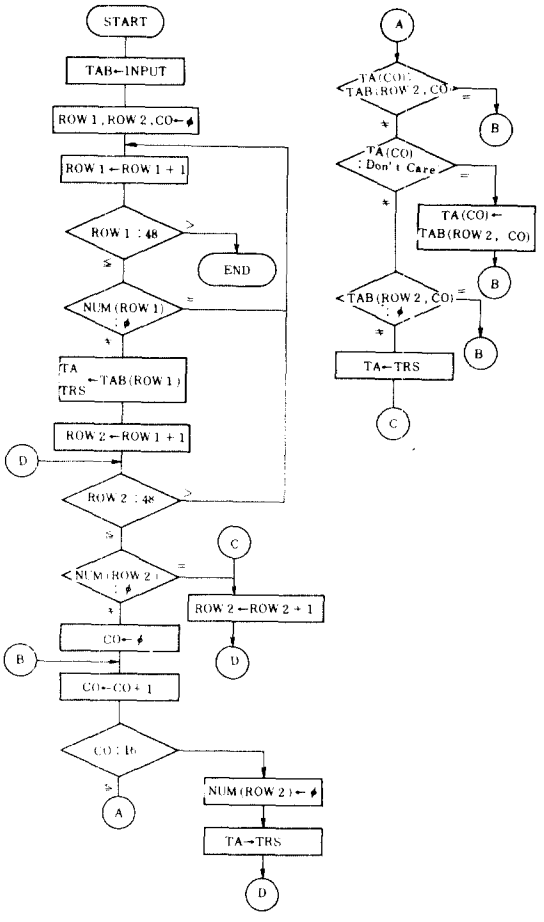


그림4. 테스트 入力 생성을 위한 플로우차아트
Fig. 4. Flow-chart for test input generation.

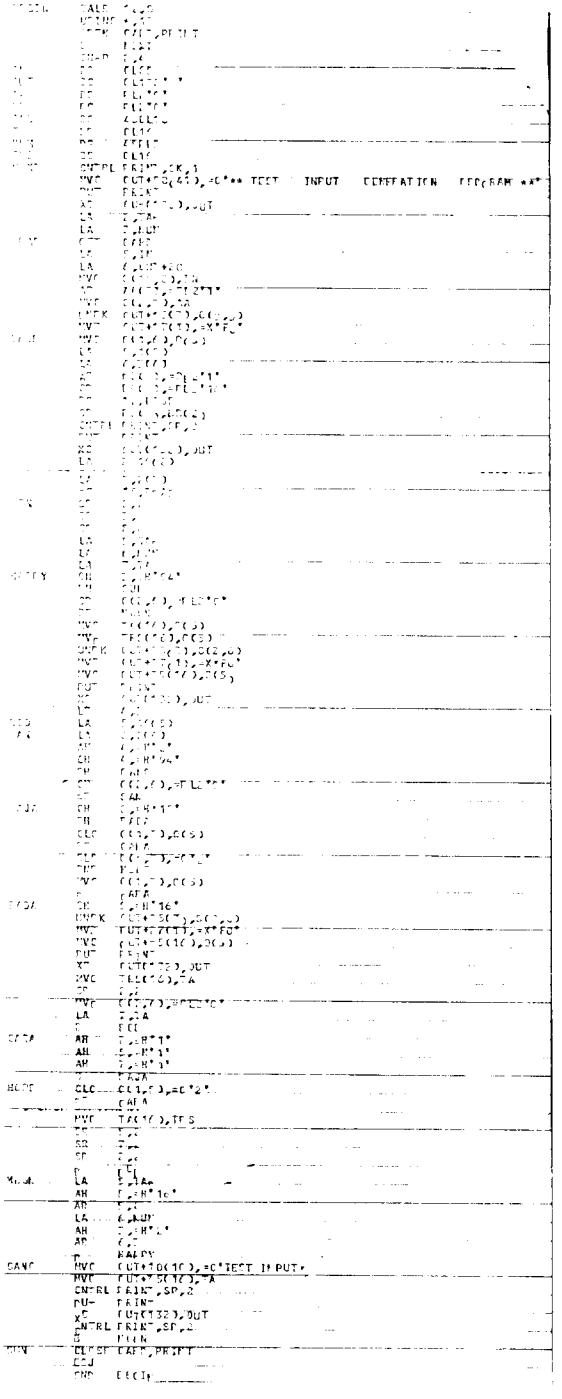
V. 結 論

본 논문에서는 테스트 入力을 구하는데 있어서 故障狀態를 고려하지 않고 PLA의 狀態表로 부터 직접 구하도록 한 방법[5]를 제안하였다.

컴퓨터를 이용한 본 故障 檢出法에 의하면 入力變數의 수와 積項의 수가 증가하여도 效果的으로 테스트 入力을 구할 수 있으며 또한 sharp연산과 cap 연산을 동시에 처리하므로 계산 시간이 단축된다.

따라서 입력 변수나 積項이 증가해도 비교 명령문의 변수만 바꾸어 주면 되므로 테스트 입력을 구하기가 간편해진다.

附 錄



謝 意

본 연구는 한국산학협동재단의 재성지원으로 이루어진 것이며 동재단에 깊은 감사를 드리는 바입니다.

參 考 文 獻

[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, pp. 845-853, November 1979.

[3] V. K. Agarwal, "Multiple fault detection in PLA's", *IEEE. Trans. on Computers*, vol. C-29, no. 6, pp. 518-522, June 1980.

[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, pp. 617-626, September 1979.

[5] 이우신, 김희석, 임재탁, "PLA에서의 고장 검출 및 고장 진단에 관한 연구", 전자공학회지, pp. 15-20. June 1981.

[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, pp. 720-727, July 1974.

[8] Douglas Lewin, *Computer-Aided Design of Digital Systems*. Crane, Russak & Company, 1977.