

論文94-31A-4-13

## k-bounded 회로에서의 효과적인 결함검출 방법

## (An effective fault detection method for k-bounded circuits)

李 貴 相 \*

(Guee Sang Lee)

## 要 約

Fujiwara에 의해 제안된 k-bounded 회로는 부분회로들이 나무구조를 이루고 한 부분회로의 입력의 수가 k개로 제한된 논리회로이다. Fujiwara는 이러한 회로에서 회선수 m에 대하여 고착형 단결함이  $O(16^k m)$ 시간에 검출될 수 있음을 보였으나 이는 실제회로에 적용하기에는 매우 비효율적임을 알 수 있다. 본 논문에서는 k-bounded 회로에서 고착형 단결함이 Table lookup과 입력분할에 의하여  $O(2^k m)$ 시간에 검출될 수 있음을 보인다. 또한 고착형 다중결함은  $O(4^k m)$ 시간에 검출됨을 보인다.

## Abstract

k-bounded circuits are combinational circuits proposed by Fujiwara whose subcircuits are partitioned in a way that they form a tree and have some restricted k inputs respectively. Fujiwara proposed a  $O(16^k m)$  fault detection algorithm for k-bounded circuits where m is the number of signal lines in the circuit. This algorithm is very ineffective to be applied to real circuits, even for small values of k. In this paper, it is shown that a single stuck-at fault in k-bounded circuits can be detected in  $O(2^k m)$  time, and multiple stuck-at faults are detected in  $O(4^k m)$  time by using table lookup and input partitioning.

## 1. 서론

조합형 논리회로에서의 결함검출은 그동안 많은 연구가 진행되었으나, 이는 NP-complete인 문제로써 효과적인 알고리즘 개발의 어려움을 안고 있는 문제이다. 이러한 어려움을 극복하기 위해서 설계단계에서부터 결함검출문제의 해결을 고려하는 DFT

(Design for Testability)에 관한 연구가 집중적으로 진행되어 왔다. 특히 최근에는 VLSI회로의 설계 자동화의 한 단계인 논리합성에서 회로의 면적을 줄이고 결함검출을 용이하게 하기 위하여 100% 결함검출가능한 회로의 생성을 위한 연구가 많이 수행되어 왔다. 그러나 이는 결함검출의 어려움을 근본적으로 해결해 주지는 않는다. 즉, 결함검출이 불가능한 선들을 제거하여도 결함검출벡터발생의 문제는 여전히 NP-complete이기 때문이다. 이의 해결을 위해서 논리합성시 결함검출셋트를 동시에 발생시키는 연구도 진행되고 있으나, 소규모회로에 대한 이론적인 단계를

\* 正會員, 全南大學校 電算學科  
(Dept. of Computer Science, Chonnam Nat'l Univ.)  
接受日字 : 1993年 4月 17日

벗어나지 못하고 있다. 또 다른 해결방안은 polynomial time에 결함검출셋트를 발생시킬 수 있는 회로를 합성하는 것이다. 이러한 회로들을 P-testable 회로라고 하자. 대표적인 P-testable 회로로써는 이단계회로(two-level circuits)와 나무회로(tree circuits)를 들 수 있으나 회로의 구조가 극히 제한되어 일반적으로 응용되는 데에는 어려움을 안고 있다. 최근에 제안된 P-testable 회로로는 Fujiwara의 k-bounded 회로와 이를 확장한 Chakradhar의 (K,k)-회로를 들 수 있다. k-bounded 회로는 다음과 같이 정의할 수 있다.

(정의) 조합논리회로 C는 다음과 같은 조건들을 만족할 때 k-bounded 회로라고 불린다.<sup>1</sup>

1. 회로 C는 임의의 부분회로의 집합  $S = \{C_1, C_2, \dots, C_t\}$ 로 분할되고 각 부분회로는 최고 k개의 입력을 갖는다.
2. 각 부분회로를 노드(node)로 하는 그래프 G는 사이클을 갖지 않는다.

(K-k)-tree<sup>12)</sup>는 k-bounded 회로의 개념을 확장한 것이나 본 논문에서는 k-bounded 회로에서의 결함검출만을 고려한다.

다단계회로 중에서 P-testable한 회로는 위에서 제시한 두가지 회로 외에도 나무(tree)회로등 몇가지가 제시되었으나, 실질적인 논리합성방법은 거의 제안되지 못하였다. 그러나 최근에 제안된 논리합성도구인 FACTOR<sup>15)</sup>에서는 Fujiwara가 제시한 k-bounded 회로의 실제적인 구현방법을 보인다. FACTOR는 각 단계마다 입력을 회귀적으로 분할하여 세계의 부분회로를 만들어 내다. 이 때 FACTOR는 부분회로간의 연결선의 개수를 최소화시키도록 최적의 입력분할을 찾아내는데, FACTOR의 논리합성 결과는 mis<sup>1)</sup> 등과 비교할 때 상당히 좋은 것으로 나타났다. 또한 설계면적의 최소화외에도 지연시간의 감소, 결함검출능력의 향상<sup>5,6)</sup> 등의 일반적인 논리합성목표를 자연스럽게 해결해 주고 있다.

이처럼 P-testable 회로의 구현이 가능해짐에 따라 이러한 회로에서의 실제적인 결함검출벡터의 발생이 효과적으로 시행될 수 있는 방법의 개발이 필요하게 되었다. Fujiwara<sup>1)</sup>는 k-bounded 회로에서 단결함을 검출하는  $O(16^k m)$  시간의 알고리즘을 제시하였다. 이때 k는 한 부분회로의 입력의 개수, m은 회로의 연결선의 개수를 가리킨다. 이 알고리즘은 결함검출벡터의 발생이 m에 대해 선형(linear) 시간에 해결됨을 보이기 위한 것이었으므로 실제로 적용되기에는 매우 비효율적이다. 이는 k가 4 또는 5의 매우 작은 수일때만 고려하더라도 바로 알 수 있다. 본 논문

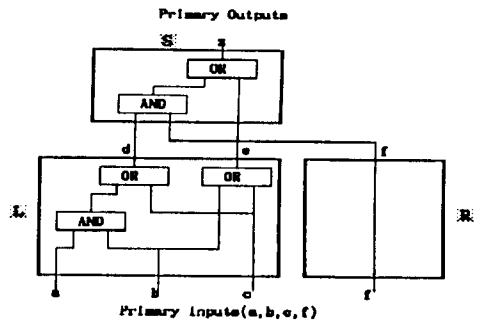
에서는 k-bounded 회로에서의  $O(2^k m)$ 인 결함검출벡터 발생 알고리즘을 제시한다. 이외에도, 효과적인 결함검출벡터 발생 방법은 D-algorithm 등 여러가지 heuristic도 생각할 수 있으나, 여기서는 m에 대한 polynomial time 알고리즘만 고려하기로 한다.

II. 단결함검출 벡터 발생 알고리즘

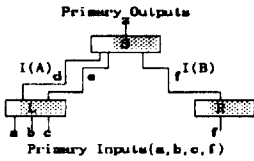
k-bounded 회로에서 어떤 단결함이 존재한다고 하자. 이 단결함을 검출하기 위해서는 결함효과 발생(fault excitation)과 결함효과 전파(fault propagation)의 두 과정이 필요하다. 여기서는 이 두과정의 수행을 위해 각 부분회로에서 그 부분회로에 해당하는 진리표를 작성한다. 다음 절에서는 각 부분회로의 진리표 작성에 대하여 기술하고 이어서 이를 이용한 결함효과 발생과 결함효과 전파의 알고리즘을 제시한다. 전체적인 단결함 검출 벡터 알고리즘은 다음과 같다.

```

TestGeneration(S, F)
/* S : k-bounded circuit */
/* f : 단결함(single stuck-at fault) */
{
    SetVector(S); /* 진리표 작성 및 */
                  /* PI Vector 설정 */
    Partition Vector (S);
                  /* 각부분회로의 Input Partition */
    FaultExcitation(f);
                  /* Fault Excitation을 위한 PI */
                  /* Vector를 진리표에서 찾아냄 */
    FaultPropagation(f);
                  /* 결함효과를 주출력선(PO)까지 전파 */
                  /* 하면서 Test Vector를 형성해 나감 */
}
    
```



(a) 회로 예



(b) 회로구성 요약도

입력	출력	PI vector
000	0	a'b'c'f'
001	1	a'b'c'f
010	0	a'b'c'f'
011	1	a'b'c'f
100	0	ab'c'f'
101	1	ab'c'f
110	1	ab'c'f
111	1	ab'c'f

(c) 부분회로 S의 진리표

입력	출력	PI vector
000	d	a'b'c'u'
001	0	a'b'c'u'
010	0	a'b'c'u'
011	1	a'b'c'u'
100	0	ab'c'u'
101	1	ab'c'u'
110	1	ab'c'u'
111	1	ab'c'u'

(d) 부분회로 L의 진리표

f	f(out)	PI vector
0	0	f'
1	1	f

(e) 부분회로 R의 진리표

그림 1. 부분회로와 진리표의 작성

Fig. 1. A subcircuit and its table.

1. 부분회로의 진리표 작성

하나의 부분회로의 입력이 k개이므로 각 진리표의 항의 개수는  $2^k$ 개가 될 것이다. 다만 이 진리표에는 그 부분회로의 입력과 출력이 갖는 값 외에 그러한 입출력 값을 제공하는 전체회로의 주입력벡터(Primary Input vector 또는 PI vector)를 갖고 있어야 한다. 다음의 그림 1은 주입력변수가 a, b, c, f라고 가정할 때, 회로 예와 임의의 각 부분회로에 대한 진리표의 예를 AND와 OR 게이트만을 이용하여 나타낸 것이다. 이 진리표의 작성은 가장 낮은 층(level)에서 시작한다. 그림 1에서 부분회로 L과 R의 진리표가 완성된 후에 부분회로 S의 진리표가 완성된다. 이 때 주의를 기울여야 할 부분은 PI vector를 결정하는 부분이다. 부분회로 L과 R의 진리표 작성시에는 PI vector가 입력변수들의 값에 의해서 결정된다. 그러나, 부분회로 S의 PI vector는 L과 R의 PI vector들을 이용하여 결정한다. 예를 들어 S의 첫번째 입력인 000 (=def)는 그림 1(d), (e)의 de=00일 때와 f=0 일 때의 PI vector들에 의하여 a'b'c'f'으로 결정된다. 이 과정은 여러가지 방법으로 구현될 수 있다. 예를 들면, 그림 1(b)에서의 첫번째 입력  $i = i_1, i_2 (=000(def))$ 로서  $i_1=00(de), i_2=0(f)$ 이다. 이와 같은  $i$ 에 대해서 이를 발생시키는 PI vector  $v$ 는 다음과 같이 구해진다.

$$v = v_1 \cdot v_2$$

위의 식에서  $v_1$ 은 L의 출력표에서 de=00 일 때의 PI vector로서 여기서는 a'b'c'이고,  $v_2$ 은 R의 출력표에서 f=0 일 때의 PI vector로서 여기서는 f'이다. 따라서  $v = v_1 \cdot v_2 = a'b'c'd'$ . 또는 abcd=0000이다. 나머지 S의 입력에 대해서도 똑같이 실행한다.

위의 결과에 따라 하나의 소정리를 도출해 낼 수

있는데, 그것은 다음과 같다.

소정리 1) Procedure SetVector에 의한 진리표의 작성은 회로 전체에 대해서  $O(2^k m)$ 시간이 걸린다. m은 회로내의 게이트 입출력선의 수이다.

(증명) 진리표의 작성은 주출력선에 연결된 부분회로 부터 시작하여 회귀적으로 SetVector()를 호출함으로써 실행된다. 가장 낮은 층의 진리표가 완성되면 그윗 층의 진리표는 이미 완성된 진리표들을 참조하여 만들어질 수 있다. 각 진리표의 작성(출력표 작성 포함)은  $2^k$ 개의 입력을 고려해야 하므로  $O(2^k)$ 번 각 부분회로의 출력값을 계산하여야 한다. 그러므로 회로전체에 대해서는 (각 선의 값을 매번 계산하여야 하므로)  $O(2^k m)$ 시간이 걸린다.

2. 결함효과 발생(fault excitation)

결함효과와 발생은 단결함이 일어난 선을 단결함의 고착값(stuck-at value)으로 고정시키고 그 부분회로의 모든 입력값에 대하여 출력값을 계산하여 정상적인 출력과 비교하면 된다. 각 부분회로에 해당하는 진리표가 작성되어 있으므로 단결함이 일어난 부분회로의 진리표에 의해 결함효과를 발생시킬 수 있는 입력값을 찾아낼 수 있다. 이는 최악의 경우  $2^k$ 개의 입력을 검사하여야 한다.

3. 결함효과와 전파(fault propagation)

결함효과와 발생에 의하여 단결함이 발생한 부분회로의 출력선에는 두가지의 값이 나타난다. 즉, 결함이 없을 때의 정상치와 결함이 있을때의 결함치이다. 이제 남은 문제는 이 결함효과가 어떻게 주출력선(Primary output line)에 전파될 수 있는가 하는 것이다. 결함효과가 전파될 수 있는 지는 단결함이 일어난 곳부터 주출력선까지의 부분회로들에서 각각 결정 될 수 있지만 결함효과와 전파도중에 결함효과를 전파시킬 수 있는 주입력벡터의 값이 존재하지 않는 경우에는 이전의 부분회로, 즉 더 낮은 층의 부분회로로 환원되어 다른 입력값을 고려하여야 한다. 이 경우 폭발적인 환원(backtracking)이 일어날 수 있다. 이를 해결하기 위해서 여기서는 어떤 부분회로에 정상치와 결함치의 두가지 입력이 주어질 때 이 결함효과가 주출력선까지 전달할 수 있는지를  $O(2^k m)$ 시간에 결정할 수 있음을 보인다.

어떤 부분회로의 입력들중, 결함전달경로(fault propagation path)에 있는 부분회로에 연결된 입력을 I(A), 다른 입력들을 I(B)라 하자. (그림 1(b)의 S의 경우 I(A)=(d, e), I(B)=(f)이다.) I(A)와 I(B)가 갖는 입력벡터의 값들을 각각 IV(A), IV(B)라

하자. (그림 1의 경우,  $IV(A)=\{00, 01, 10, 11\}$ 이다) 여기서는  $IV(A)$ 와  $IV(B)$ 를 주출력선에 생성시키는 출력값에 따라 분할한다. 만일 같은 분할집합(partitioned set)에 속하면 이들은 주출력선에 항상 같은 값을 생성시킨다. 그림 2에 제시된 예제를 보자. 여기서 L과 S의 진리표에 포함될 PI vector는 그림 1에 나타나 있으므로 여기서는 생략한다.

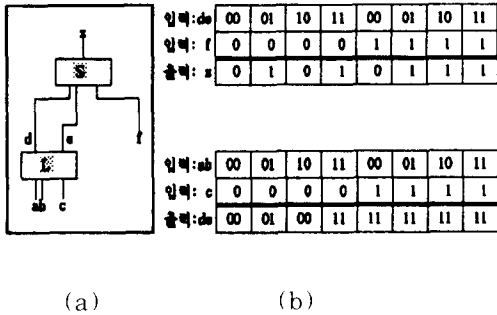


그림 2. 입력셋트 분할의 예  
Fig. 2. An example of input value partitioning.

그림 2(b)에서 부분회로 S의 왼쪽 입력값들(d,e)은 입력 f=0 일 때 출력 z의 값에 따라 다음과 같이 두개의 집합으로 분할할 수 있다.

$$Set_S(1)=\{00, 10\} \text{ /* if } f=0, z=0 \text{ */}$$

$$Set_S(2)=\{01, 11\} \text{ /* if } f=0, z=1 \text{ */}$$

$Set_S(1)$ 과  $Set_S(2)$ 에서  $f=1$  일 때에 출력값 z는 각각 0과 1 이다. 또한  $f=0$  일 때에  $Set_S(1)$ 은 다시 다음과 같이 분할될 수 있다.

$$Set_S(1)=\{00\} \text{ /* if } f=0, z=0; \text{ if } f=1, z=0: \text{ */}$$

$$Set_S(2)=\{01, 1\} \text{ /* if } f=0, z=1; \text{ if } f=1, z=1: \text{ */}$$

$$Set_S(3)=\{0\} \text{ /* if } f=0, z=0; \text{ if } f=1, z=1: \text{ */}$$

하나의 분할 집합  $Set_S(i)$ 에 있는 입력들은 항상 같은 출력값을 생성시킨다. 가령,  $Set_S(2)$ 의 입력값들 01과 11은 f의 값이 다르게 입력되더라도, 즉 f의 값이 0이나 1에 관계없이 항상 같은 z값을 생성함을 알 수 있다. 여기서 알 수 있듯이 하나의 분할집합 속에 있는 입력들이 결함치와 정상치가 될 때 그 결함효과는 주출력선으로 전달될 수 없다. 예를 들어  $Set_S(2)$ 에 속한 두 입력값 01과 11이 각각 부분회로 S에 전달된 정상치와 결함치라 하면 f에 어떠한 값이 입력되

더라도 이 부분회로의 출력값은 서로 같은 값을 갖는다. 따라서 그 결함효과는 S의 출력에 전달될 수 없다. 이제 L의 a,b 입력값의 분할을 고려해 보자. 먼저 c=0인 경우를 고려하면 출력인 d,e 값에 따라 다음과 같이 분할될 수 있다.

$$Set_L(1)=\{00, 10\} \text{ /* de=00 */}$$

$$Set_L(2)=\{01\} \text{ /* de=01 */} \text{ /* } Set_L(2)\text{에 같이 포함.}$$

$$Set_L(3)=\{11\} \text{ /* de=11 */}$$

여기서,  $Set_L(1)$ 에 있는 00과 10은 c=0일때 출력값 d,e 값이 00으로 동일한 값을 산출하기 때문에 같은 분할 집합에 포함시킬 수 있다. 따라서, a,b에 주어지는 어떤 결함치와 정상치도 서로 다른 분할 집합에 포함되어 있으면, 출력 d,e에 서로 다른 값이 L의 출력에 전달될 수 있다. 그렇지만, 이 결함효과가 S의 출력 z까지 전달되는지를 고려한다면, L의 분할은 S의 분할까지도 고려 하여야 한다. S의 분할시 01과 11은 어떠한 경우에도 같은 z값을 생성(즉, f=0 이나 f=1인 경우 모두 동일한 z값(=1)을 생성함.)하기 때문에 01과 11은 다음과 같이 같은 분할 집합에 포함시켜야 한다.

$$Set_L(1)=\{00, 10\}$$

$$Set_L(2)=\{01, 11\} \text{ /* } Set_L(2)=\{01\} \cup Set_L(3)=\{11\} \text{ */}$$

이 분할은 c=1인 경우를 고려해도 결과는 마찬가지이다. 그러면 회로내의 입력 a의 s-a-1(stuck-at-1)결함을 고려해 보자. 이 결함은 부분회로 L의 입력에 D'0(정상치 00, 결함치 10) 값을 생성시킬 수 있다. 여기에서 00과 10는 분할집합  $Set_L(1)$ 에 같이 포함되어 있으므로 c에 어떤 다른 값이 입력되더라도 항상 주출력선 z에 같은 결과치(=0)를 만들어 내므로 vector ab=00으로 결함이 검출될 수 없다. 입력 a의 s-a-1 결함에 대해 vector D'1(정상치 01, 결함치 11)을 고려하여도 마찬가지이다. 반면에, 입력 b의 s-a-1의 결함효과가 0D'(정상치 00, 결함치 01)을 고려하면 이들은  $Set_L(2)$ 와  $Set_L(1)$ 의 서로 다른 분할집합에 속해 있으므로 이 결함은 주출력선 z까지 전달될 수 있다. 이 결함효과를 전달하기 위해서는 먼저 c의 값이 0인 경우를 고려해 본다. 그러면 d,e에 0D'(정상치 00, 결함치 01)이 전달되므로 f값은 0으로 하면 주출력선 z에 D'(정상치 z=0, 결함치 z=1) 값이 전달 된다. 따라서 입력 b의 s-a-1 결함의 결함검출벡터(test vector)는 a'b'c'f' 즉, abcf=0000이다.

결함전달경로에 있는 부분회로의 개수가 임의의  $n$  개일 경우는 단순히 위의 2개의 부분회로의 경우를 확장하면 된다. 즉, 가장 상위 층의 부분회로부터 시작하여 분할을 시행한다. 분할이 끝나면 다음의 낮은 층의 부분회로로 내려가면서 분할을 시행하되 윗층의 분할결과를 고려한다. 모든 부분회로에 대하여 입력 분할이 끝나면 결함효과의 전달은 다음과 같이 수행될 수 있다. 어떤 단결함이 부분회로  $L$ 에서 발생하여  $S(1), \dots, S(n)$ 으로 결함효과가 전달되어야 한다고 가정해보자. 이때  $L$ 에서 결함효과를 발생시키되  $L$ 의 출력값( $S$ 의 입력값) 즉, 정상치와 결함치가  $S$ 에서 같은 분할셋트에 속하지 않도록 적절한  $L$ 의 입력값을 결정한다. 이러한 과정을  $S(n)$ 까지 반복하면 된다.

정리 1)  $k$ -bounded 회로에서 임의의 단결함이 일어났다고 할 때, 이의 결함검출 벡터(test vector)는  $O(2^k m)$ 시간에 발생시킬 수 있다.  $m$ 은 회로내의 게이트 입력선수의 수이다.

(증명) 먼저 주어진 회로에 대하여 각 부분회로의 진리표를 작성하고 입력값들을 분할한다. 이제 결함이 일어난 부분회로를  $C$ 라 하자. 먼저  $C$ 의 진리표로부터  $C$ 의 출력에 결함치와 정상치가 서로 다르도록(또한 서로 다른 분할 집합에 속하도록) 입력값을 선택한다. 결정된 입력값, 결함치, 정상치로서  $C$ 의 상위 부분회로들에 대해서도 같은 방법으로 입력값을 선택해 나간다. 이는 결함전달경로에 있는 부분회로들( $M$ 개라 하자.)와 진리표를 거쳐야 하므로  $O(2^k M)$ 시간이 걸린다. 만일 결정된 입력값에 대한 PI vector가 정의되지 않으면 이를 '기각한다. 위의 과정을 반복할 때에 결함효과를 주출력선에 전달할 수 있는 PI vector를 찾을 수 있다. 이에 걸리는 시간은 (진리표 작성)+(입력값들의 분할)+(결함효과를 전달할 입력벡터를 찾아나가는 시간) $=O(2^k M)+O(2^k M)+O(2^k M)+O(2^k m)$ 이다. (결함효과의 발생시간은 한 부분회로에 국한되므로 무시함.)

### Ⅲ. 다중결함의 검출

다중결함의 검출은 단결함 검출시의 각 부분회로의 진리표를 작성할 때에 각 입력선들이 0.1의 2진값 대신에 0.1,D,D'의 네가지 값을 갖는다고 가정하고 진리표를 작성한다. 나머지의 과정은 정확히 단결함 검출시와 같다.

일련의 과정을 하위레벨의 부분회로부터 시행하되 가장 상위레벨의 진리표가 작성되면 출력값이 D이거나

나 D'인 항목의 주 입력벡터(PI vector)가 주어진 결함(fault)을 검출할 수 있는 벡터(vector)이다. 진리표의 작성은 2진값 0.1 대신에 0.1,D,D'의 네가지 값을 사용하므로  $O(4^k m)$ 시간이 걸린다. 증명은 생략한다.

### Ⅳ. Fujiwara 알고리즘과의 비교

Fujiwara에 의해 제안된  $k$ -bounded회로 알고리즘은 Section III.의 다중결함에 대한 test vector를 찾기위해 각 부분회로 입출력선이  $\{0,1,D,D'\}$ 의 값을 갖도록 테이블을 작성하는 것과 유사하다. 단 한 가지 다른점은 본 논문의 방법들은 테이블을 기반으로 한 접근방법인 반면 Fujiwara의 알고리즘은 그래프를 형성해 나간다는 점이 다르다. Fujiwara의 알고리즘을 요약하면 다음과 같다.

(1) 그래프 GT의 형성

- ① 각 부분회로의 입력에  $\{0,1,D,D'\}$ 값을 가하여 가능한 모든 입력값(node)을 고려하여 이에 대한 출력값을 계산.
- ② 인접한 부분회로의 노드들 간에 일치하는 것을 edge로 연결.
- ③ edge로 연결되지 않은 노드 제거.

(2) test vector 발생

그래프 GT에서 트리를 찾는다. 단 이 트리는 모든 주입력선, 주출력선에 연결되어야 한다. 이때 주입력선들의 값이 테스트 벡터이다.

이와 같이 Fujiwara 알고리즘은 각 노드를 그래프 형식으로 연결하여 상하부분회로의 값이 일치하는 노드를 찾기 위해 이들을 서로 비교하므로 최악의 경우에  $O(4^k m) * O(4^k m) = O(16^k m)$  시간 만큼의 탐색(search)과정이 필요하다. 이에 반해, 본 논문에서는 각 레벨마다 D 또는 D'이 나타나는 항의 PI vector를 진리표 테이블에 기록하는 방법을 이용하므로 탐색과정이나 상하 부분회로간의 매칭(matching)과정이 없이 쉽게 결함검출벡터를 찾을 수 있으며, 다중결함검출시 최악의 경우에  $O(4^k m)$ 시간 만큼의 탐색과정만 필요하다. 뿐만아니라 Section II.에서 제시하였듯이 단결함인 경우에는  $O(2^k m)$ 시간에 결함 검출이 가능하다.

### Ⅴ. 실험 결과

본 논문에서 제시한 알고리즘과 Fujiwara의 알고리즘을 비교한 실험결과는 다음의 표 1과 표 2와 같다. 표 1은 GLUE(Gate LangUagE) 표현으로 이

루어진 실험을 위한 회로들을 나타낸다. 여기서 k는 한 부분회로의 입력선의 총 수이다.

표 1 실험을 위해 사용된 회로들  
Tab. 1 Example circuits for experiment.

회로명	주입력수	주출력수	부분회로사이의 연결선 수	k
a2	2	1	2	4
a4	4	1	2	4
a8	8	1	2	4
b2	2	1	3	6
b4	4	1	3	6
b8	8	1	3	6
c2	2	1	4	8
c4	4	1	4	8
c8	8	1	4	8
d2	2	1	5	10
d4	4	1	5	10
d8	8	1	5	10

표 2 결함 검출 벡터 발생 결과  
Tab. 2 Results of fault detection vector generation.

회로명	결함 수	본 논문의 알고리즘					Fujiwara 알고리즘				
		테이블 갯수	테이블 크기	테스트 패턴 수	테스트 벡터의 수	CPU 시간 (초)	사용한 노드회 수	노드 갯수	드레인 갯수	테스트 벡터의 수	CPU 시간 (초)
a2	24	7	32	2.5	8	0.2	304	228	4	1.6	
a4	48	15	80	5.8	17	0.4	864	7283	7	18.6	
a8	92	31	176	13	35	0.5	1984	40135	17	95.6	
b2	32	7	80	5.5	8	0.2	4144	4626	6	13.7	
b4	56	15	224	14	16	0.4	12384	190000	11	3974.8	
b8	112	31	512	32	42	1.0	-	-	-	*	
c2	40	7	272	16	6	0.3	65584	92908	4	236.2	
c4	64	15	800	49	30	0.6	-	-	-	*	
c8	128	31	1856	119	27	1.7	-	-	-	*	
d2	48	7	1040	62	9	1.8	-	-	-	*	
d4	80	15	3104	263	21	2.5	-	-	-	*	
d8	160	31	7232	679	37	6.2	-	-	-	*	

\* : out of memory  
※ : 단위는 10<sup>3</sup>

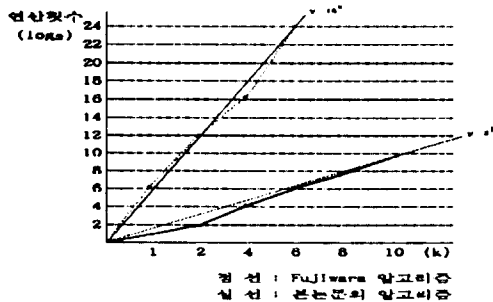


그림 3. 본 논문의 알고리즘과 Fujiwara 알고리즘의 비교

Fig. 3. Comparison between proposed algorithm and Fujiwara algorithm.

결함 검출 알고리즘은 C언어로 구현되어 NeXT station에서 실행되었다. 그 결과는 표 2에 주어져 있다. 여기서의 입력회로는 GLUE형식으로 주어지고 출력은 그 회로에 대한 결함 검출 세트이다. 이 실험에서 나타난 것과 같이 Fujiwara 알고리즘은 현실적으로 적용하기 어렵다는 것을 알 수 있다.

또한, 다음 그림 3은 본 논문에서 제시한 알고리즘과 Fujiwara 알고리즘을 한 부분회로의 입력선의 총 수인 k와 연산 횟수에 대해서 비교 분석한 것으로서, Fujiwara 알고리즘에 비해 본 논문의 알고리즘이 좋은 결과가 나타나는 것을 볼 수 있다.

VI. 결론 및 토의

본 논문에서는 k-bounded 회로에서의 결함검출 방법을 고려했다. 단결함인 경우  $O(2^k m)$  시간에 결함을 검출할 수 있음을 보였다. 이는 Fujiwara에 의해 제시된  $O(16^k m)$  단결함검출 알고리즘보다 개선된 것이다. 이 때 m은 회로내의 연결선들의 갯수이다. Fujiwara의 알고리즘은 각각의 부분회로에 대하여 위의 다중결함검출의 경우와 같이 {0.1.D, D'}의 4가지 값을 고려하여 위아래의 두 부분간의 값들을 하나하나 대조하여 서로 일치하는 것을 선택함으로써 test를 찾기 위한 그래프를 형성하여 나가므로  $O(4^k \cdot 4^k = 16^k)m$ 의 시간을 요한다. 여기서는 {0.1}의 두가지 값만을 고려하여 부분회로의 진리표를 작성하고 이 진리표만을 이용하여 결함효과를 전파할 수 있는 방법을 제시하였다.

본 논문의 알고리즘을 응용하기 위한 예제 회로들은 GLUE(Gate LangUagE)형식으로 이루어진 회로들을 이용하였다. 실험결과는 Section V.에 제시되었으며, 그 결과 Fujiwara의 알고리즘을 현실적으로 매우 적용하기 어려운 것으로 나타났다. 반면, 본 논문에서 제시된 알고리즘은 주어진 예제회로에 대해서 매우 효과적으로 동작하는 것으로 나타났으며, 본 논문의 알고리즘은  $O(2^k m)$ 시간에, 그리고 Fujiwara 알고리즘은  $O(16^k m)$ 시간에 처리됨을 볼 수 있었다.

앞으로는 k-bounded 회로의 stuck-open 또는 지연결함 모델(delay fault model)에 대해서도 결함검출의 효과적인 방법이 개발되어야 할 것으로 보인다.

參考文獻

[1] H.Fujiwara, "Computational Complexity of Controllability/Observability problems

- for combinational Circuits." *IEEE Transactions on Computers*, vol. c-39, pp.762-767, June 1990.
- [2] S.T. Chaktadhar, V.D.Agrawal and M. L.Bushnell. "Polynomial time solvable fault detection problem." *Fault-tolerant Computing Symposium*, pp.56-91, 1990.
- [3] T.T. Hwang, R.M. Owens and M.J. Irwin. "Exploiting Communication Complexity for Multi-level Logic Synthesis." *IEEE Transactions on CAD*, pp.1017-1027, Oct. 1990.
- [4] R.K. Brayton et al., "MIS: A Multi-level Logic Optimization System." *IEEE Transactions on CAD*, pp.1062-1081, Nov. 1987.
- [5] G.Lee, M.J.Irwin and R.M.Owens. "Test Generation in Circuits Constructed by Input Decomposition." Int'l Conf. on Computer Design, Cambridge, Mass., pp.107-111, Oct. 1990.
- [6] G.S. Lee, M. Hwang, M.J. Irwin, R. M. Owens. "Testability of a Class of Multi-level Reed Muller Circuits." *IFIP WG 10.5 Workshop on Application of the Reed Muller Expansion in Circuit Design*, Sep. 1993.

---

 著者紹介
 

---



李貴相(正會員)

1958年 2月生. 1980年 2月 서울대학교 전기공학과 졸업. 1982年 2月 서울대학교 컴퓨터공학과 졸업(석사). 1991年 8月 Pennsylvania State University(Ph. D). 1982年 2月 ~ 1983年 3月 금성통신연구소 연구원. 1983年 4月 ~ 현재 전남대학교 전산학과 조교수. 주관심 분야는 VLSI 설계자동화, 테스트링, 논리합성, 뉴럴 네트워크 등임.