

미지구조 기능 블록에 대한 경로 지연고장 검사입력 생성기

A Path Delay Fault Test Pattern Generator for Unknown-Structure Functional Blocks

任容兌* · 姜成昊**
(Yong-Tae Yim · Sung-Ho Kang)



社 團
法 人

大 韓 電 氣 學 會

THE KOREAN INSTITUTE OF ELECTRICAL ENGINEERS

미지구조 기능 블록에 대한 경로 지연고장 검사입력 생성기

論文

46~2~18

A Path Delay Fault Test Pattern Generator for Unknown-Structure Functional Blocks

任容兌* · 姜成昊**
(Yong-Tae Yim · Sung-Ho Kang)

Abstract - Most of the available delay fault test pattern generators consider circuits which consist of only discrete logic gates whose structures are known. In this research, a new path delay fault test pattern generation algorithm for the circuits including any unknown-structure functional blocks are devised. Based on the new algorithm, a path delay fault test pattern generator is developed. The experimental results show that the new path delay fault test pattern generator is efficient and accurate.

Key Words : Path delay fault model, Robust test, Delay testing, Delay test pattern generator

1. 서론

논리 회로가 정확히 동작하기 위해서는 회로 내의 경로에 대한 지연시간이 정해진 시간을 초과하지 않아야 한다. 만일 제조과정에서 발생하는 결함이나 임의의 변수에 의해 회로의 지연이 정해진 시간을 초과하게 된다면 지연고장(delay fault)이 일어났다고 한다. 지연고장을 검사하기 위해서는 보통 두 패턴의 검사입력이 필요하다. 첫번째 시간 프레임에 입력 벡터가 주어지고 이 벡터에 의해 회로는 초기화된다. 회로가 안정된 상태를 유지할 때 두번째 시간 프레임에 입력 벡터를 회로에 가하고 원하는 동작 주파수에서 출력을 샘플링함으로써 회로의 지연고장을 검사한다. 이 때 두번째 입력 벡터는 첫번째 입력 벡터와는 반대값을 가지게 하여 출력에서 특정한 천이가 일어나게 함으로써 지연고장이 일어났는지를 검사하게 된다.

일반적으로 지연고장의 모델은 게이트 지연고장 모델(gate delay fault model)[1, 2]과 경로 지연고장 모델(path delay fault model)[3, 4, 5]의 두 가지가 있다. 게이트 지연고장 모델은 지연이 회로 내의 특정 게이트에 국소화되는 것을 말한다. 경로 지연고장 모델은 회로 내의 여러 게이트들의 작은 지연들이 모여 회로 전체의 지연고장을 발생시키는 경우이다. 게이트의 수는 경로의 수 보다 적으므로 게이트 지연고장 모델을 사용할 경우 적은 양의 메모리로 빠른 시간 내에 검사입력을 생성할 수 있다. 또한 국소화된 지연고장을 다루므로 지연고장을 정량적으로 나타낼 수 있다. 하지만 회로 전체에 분포되어 누적되는 작은 지연고장들에 대해서는 아무런 효과가 없다. 즉 회로에서 각 게이트들의 지연은 규정된 시간을 초과하지 않지만 회로의 출력에서는 실제 지연고장이 발생하는 경우는 게이트 지연고장 모델로는 검출할 수가 없다. 따라서 이를 해결하

기 위해 경로 지연고장 모델을 이용한다. 그러나 경로 지연고장 모델은 게이트의 입력과 분기선(fanout)의 수에 따라서 경로의 수가 무한히 증가하므로 경우에 따라서는 모든 경로를 다 검사하지 못 할 수도 있다. 따라서 일반적으로 작은 회로에 대해서는 모든 경로에 대해 검사를 하지만 큰 회로에 대해서는 임계경로(critical path)들을 생성해서 검사하게 된다. 이러한 단점에도 불구하고 국소화된 지연고장뿐만 아니라 분포된 지연고장들도 다룰 수 있기 때문에 현재 경로 지연고장 모델이 많이 사용된다. 본 논문에서도 경로 지연고장 모델을 사용하여 기능 블록에 대한 지연고장 검사입력의 생성을 다룬다.

지연고장의 검사입력 유형은 일반적으로 HFR(hazard-free robust)[6], ROB(robust)[7, 8, 9], SNR(strong-non robust)[10], WNR(weak-non robust) 검사입력[10]의 4가지로 나눌 수 있다. HFR 검사입력은 경로 상의 신호에 동적 해저드(dynamic hazard)가 없으며 회로의 나머지 부분에서의 지연과 무관하게 지연고장을 검출할 수 있는 두 검사입력 벡터를 말한다. ROB 검사입력은 회로의 나머지 부분에서의 지연과 무관하게 지연고장을 검출할 수 있지만 신호에 동적 해저드가 있을 수 있다. ROB 검사입력의 조건을 만족하지 못할 때 이를 NR(non robust) 검사입력[7, 8, 9]이라고 하는데 이는 다시 SNR 검사입력과 WNR 검사입력의 두 종류로 나눌 수 있다. 경로 상의 전이(transition)가 경로 상에 있는 각각의 소자들에 도달하기 전에 모든 경로의 입력들이 최종값으로 안정된다면 경로 지연고장을 검출할 수 있는 두 패턴의 테스트를 WNR 테스트라고 한다. SNR 테스트는 경로의 입력에 정적 해저드가 없다면 ROB 테스트가 되고, 그렇지 않다면 WNR 테스트가 된다.

칩의 설계에서는 특정한 기능을 수행하기 위해 기능 블록을 사용한다. 기능 블록에 사용되는 대표적인 셀인 표준셀(standard cell)은 일반적으로 그 구조가 알려져 있으나 경우에 따라서는 내부구조를 모르는 표준셀 라이브러리를 사용할 수도 있다. 더구나 표준셀의 사용으로도 칩면적과 전력 소모의 증가를 해결할 수 없을 때는 트랜지스터 레벨에서 직접 설계하게 된다. 만약 설계된 기능 블록의 내부 구조를 알 수 있다면 기

* 正會員 : 延世大 大學院 電氣工學科 碩士課程

** 正會員 : 延世大 工大 電氣工學科 助教授 · 工博

接受日字 : 1996年 8月 5日

最終完了 : 1996年 12月 16日

존의 검사입력 생성 기법으로도 검사입력의 생성이 가능하다. 하지만 그 내부 구조를 알 수 없을 때는 기존의 방법만으로는 실제 검사입력의 생성이 가능한 경로에 대해서도 검사입력을 생성하지 못할 수 있다. 따라서 본 논문에서는 이러한 기능 블록에 대해, 그 내부를 모를 경우 기능 블록의 부울함수만을 사용하여 검사입력을 생성하는 경우를 다룬다. 이 때의 기능 블록을 일반적인 의미의 기능 블록과 구분하기 위해서 미지구조 기능 블록(unknown-structure functional block:USFB)이라고 부르기로 하고, 약어로 미지블록이라 표기하였다. 미지블록에서는 회로의 시간특성을 분석할 때 미지블록의 내부 경로는 고려하지 않는다. 이는 미지블록을 하나의 단위로 취급함을 의미한다. 따라서 미지블록의 시간특성은 내부 경로에 관계없이 입력과 출력에서만 시간특성을 의미한다. 이러한 특성을 만족시키기 위해서 본 논문에서는 미지블록에 대한 검사입력의 유형과 입력 제한 조건을 재정의하고 이에 대한 이론적 근거를 제시한다. 또 이를 구현한 알고리즘과 이에 기초한 지연고장 검사입력 생성기를 제안한다.

2. 미지구조 기능 블록에 대한 경로 지연고장 검사입력의 유형 및 논리값

미지블록의 경로 지연고장에 대한 검사입력을 생성할 때는 내부 경로에서의 지연이 모두 무시할 수 있는 일정한 범위 내에 있어야 한다는 가정이 필요하다. 내부 경로에서의 지연이 경로마다 다를 경우, 검사입력을 생성하기 위해서는 그 미지블록의 내부 경로를 알아야 한다. 하지만 미지블록은 부울함수만이 정보로 제공되므로 이러한 경우는 논의에서 제외된다. 다중 출력의 미지블록은 각각의 출력에 대한 가능한 모든 경로들을 독립적으로 다룰 수 있으므로 단일출력의 미지블록을 이용하여 모델링할 수 있다. 따라서 본 논문에서는 단일출력의 미지블록만을 고려한다.

미지블록은 부울방정식으로 표현되므로 출력으로 논리값 0 또는 논리값 1을 가진다. 따라서 미지블록의 입력은 논리값 0, 논리값 1, 또는 논리값 X가 된다. 논리값 0이 두 시간 프레임 동안 해저드 없는 안정된 값을 유지한다면 S0라고 표시한다. 논리값 1에 대해서도 마찬가지로 S1으로 표시한다. 만약 설계된 기능 블록이 삼상 소자(tri-state device)를 포함한다면 그림 1과 같이 부울함수만이 사용되는 순수한 미지블록과 삼상 버퍼(tri-state buffer)를 이용해서 모델링할 수 있다.

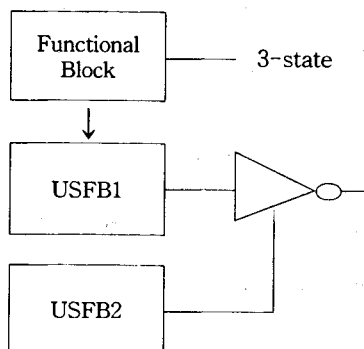


그림 1 삼상 기능 블록에 대한 모델링

Fig. 1 Modeling for 3-state functional blocks

지연고장을 검사하기 위해서는 두 패턴의 검사입력 벡터가 필요하다. 첫번째 벡터에 의해 신호가 갖는 논리값을 신호의 초기값(initial value)이라고 정의하고, 두번째 벡터에 의해 신호가 갖는 논리값을 신호의 최종값(final value)이라고 정의한다. 초기값과 최종값은 출력에서 지연고장을 검출할 수 있도록 천이(transition)가 일어나야 하므로 서로 반대값을 가져야 한다. 특정한 천이가 일어나는 미지블록의 입력과 출력을 미지블록의 경로라고 명명하자. 이 때 미지블록의 경로에 대한 입력을 경로 입력, 그 외의 입력들을 경로외(off-path) 입력이라고 명명한다. 미지블록에 대한 ROB 검사입력은 미지블록의 경로에서 발생하는 천이에 필요한 초기값과 최종값을 만족시키는 두 검사입력 벡터로 정의되며, 내부 경로나 경로의 입력의 지연에 관계없이 경로 입력이 초기값에서 변하기 전에는 미지블록의 출력이 변하지 않는다. HFR 검사입력은 ROB 검사입력 중에서 미지블록의 출력에 동적 해저드를 유발하지 않는 검사입력으로 정의되며, 경로 입력에서 발생하는 천이가 미지블록의 모든 내부 경로를 따라서 진행해갈 수 있을 때 지연고장을 검출할 수 있다.

F_i 를 각각 미지블록에 의해 구현되는 부울함수, 미지블록의 입력이라고 하자. 만약 시간 프레임 t 에서의 미지블록의 출력을 $F_i(i=k)$, 단 k 는 0 또는 1로 표시한다면, 미지블록에 대한 입력과 출력의 천이방향이 같을 때, 이 벡터를 비반전 전송 벡터(non-inverting propagation vector)라고 하고, 다음과 같이 나타내기로 한다.

$$BDNI(i) = \overline{F_0}(i=0) \cdot F_1(i=1)$$

또 이와는 반대로 입력과 출력의 천이방향이 다르다면 반전 전송 벡터(inverting propagation vector)라고 하고, 다음과 같이 나타낸다.

$$BDI(i) = \overline{F_0}(i=1) \cdot F_1(i=0)$$

미지블록의 경로에 대한 WNR 검사입력은 두번째 벡터가 미지블록의 경로 입력에 대해 전송 벡터가 되는 두 검사입력 벡터로 정의되고, SNR 검사입력은 WNR 검사입력 중에서, 같은 미지블록의 경로에 대해 첫번째 벡터에 의해 미지블록의 출력이 초기값을 가지는 벡터로 정의된다.

3. 입력 제한 조건

미지블록의 경로 입력이 초기값을 가질 때, 미지블록의 경로외 입력에 가해지는 벡터에 의해 미지블록의 출력이 초기값을 갖는다면, 이 벡터를 미지블록에 대한 초기화 벡터(initialization vector)라고 명명하자. 초기화 벡터의 한 경로외 입력이 논리값 X를 가지고, 다른 모든 경로외 입력들이 경로 입력의 천이를 만족하는 이진값(0 또는 1)을 유지할 때, 논리값 X를 갖는 경로외 입력에서 천이가 발생하더라도 이 천이가 미지블록의 출력에서 정적 해저드(static hazard)를 유발하지 않는다면, 이 벡터를 무해저드 초기화 벡터(hazard-free initialization vector)라고 한다. 이 때 경로외 입력이 이진값을 갖지 않고 논리값 X를 가져도 이 벡터가 무해저드 초

기화 벡터를 유지한다면, 이를 최소 무해저드 초기화 벡터(minimal hazard-free initialization vector)라고 부른다. V_1 을 미지블록의 한 경로에 대한 최소 무해저드 초기화 벡터라 하고, V_2 를 경로의 입력이 V_1 과 V_2 에서 같은 논리값을 갖는 전송 벡터라고 하자. 이 때 $\langle V_1, V_2 \rangle$ 가 V_1 에서 V_2 로 천이하는 동안 모든 경로의 입력들이 안정된 상태를 유지하는 벡터 쌍이라면 미지블록 경로에 대해 벡터 쌍 $\langle V_1, V_2 \rangle$ 는 같은 경로에 대해 ROB 검사입력이 되기 위한 필요충분 조건이 된다. 벡터 쌍 $\langle V_1, V_2 \rangle$ 가 ROB 검사입력이 되기 위한 충분조건인 이유는 벡터 쌍 $\langle V_1, V_2 \rangle$ 가 ROB 검사입력의 정의를 만족하기 때문이다. 그리고 만약 경로의 입력들이 안정된 상태를 유지하지 않는다면 경로 입력이 초기값을 유지하고 있는데도 V_1 에서 V_2 로 천이하는 동안 그 경로의 입력이 V_1 이 가졌던 값과는 반대의 값을 가질 수도 있다. V_1 은 최소 무해저드 초기화 벡터이므로, 결과적으로 최종값으로 바뀌는 미지블록의 출력이나 다른 경로의 입력에서의 천이가 미지블록의 출력에서 정적 해저드를 유발하게 된다. 만약 V_1 에서는 논리값 X 를 갖지만 V_2 에서는 논리값 0이나 논리값 1을 갖는 경로의 입력이 논리값 X 를 가진다면, V_2 는 더 이상 전송 벡터가 될 수 없으며, 결과적으로 V_2 를 경로의 입력으로 주고 경로 입력이 최종값을 가질 때 미지블록의 출력이 최종값으로 안정된 상태를 유지하지 못 할 수도 있다. 따라서 벡터 쌍 $\langle V_1, V_2 \rangle$ 는 ROB 검사입력이 되기 위한 필요조건이다.

미지블록에 대한 한 전송 벡터를 V 라고 하자. 이 때 V 가, 첫째로, 다른 모든 경로의 입력들이 V 를 만족하는 일정한 이진값을 갖고, 경로 입력이 초기값이나 최종값을 가질 때, V 에서 논리값 X 를 갖는 미지블록의 경로의 입력들이 미지블록의 출력에서 정적 해저드를 유발하지 않으며, 둘째로, 모든 경로의 입력들이 V 를 만족하는 일정한 이진값을 가질 때, 미지블록의 경로 입력에서 발생하는 초기값에서 최종값으로의 천이가 미지블록의 출력에서 동적 해저드를 유발하지 않으면, 이 벡터를 무해저드 전송 벡터(hazard-free propagation vector)라고 명명하자. 이 때 경로의 입력이 이진값을 갖지 않고 논리값 X 를 가져도 V 가 무해저드 전송 벡터를 유지한다면, V 를 최소 무해저드 전송 벡터(minimal hazard-free propagation vector)라고 명명한다. V 를 미지블록 경로에 대한 최소 무해저드 전송 벡터라고 하고, 두 시간 프레임 동안 같은 이진값(S_0 나 S_1)을 유지하는 무해저드 전송 벡터로 이루어진 경로의 입력들의 벡터 쌍을 $\langle V_3, V_4 \rangle$ 라고 하자. 여기서 V_3, V_4 는 $V_3 = V_4$ 의 관계를 가진다. 이 때 미지블록의 경로에 대해 벡터 쌍 $\langle V_3, V_4 \rangle$ 는 그 경로에 대해 HFR 검사입력이 되기 위한 필요충분조건이다. 벡터 쌍 $\langle V_3, V_4 \rangle$ 가 HFR 검사입력이 되기 위한 충분조건인 이유는 벡터 쌍 $\langle V_3, V_4 \rangle$ 가 HFR 검사입력의 정의를 만족하기 때문이다. 그리고 만약 경로의 입력들이 S_0 나 S_1 을 갖지 않으면 V_3 에서 V_4 로의 천이 동안에 그 경로의 입력이 V_3 이 가졌던 값과 반대의 값을 갖게 될 수도 있다. 결국 경로의 입력의 천이가 미지블록의 출력에서 정적 해저드를 유발하거나 경로 입력의 천이가 미지블록의 출력에서 동적 해저드를 유발하게 될 것이다. 또는 미지블록의 출력이 더 이상 경로 입력에 대해 민감하지 않게 되고 경로 입력과 경로의 입력에서의 천이가 미지블록의 출력에서 해저드를 유발할 수 있다. 따라서 벡터 쌍 $\langle V_3, V_4 \rangle$ 는 HFR 검사입력이 되기 위한 필요조건이다.

미지블록의 경로에 대한 최소 무해저드 초기화 벡터와 최소 무해저드 전송 벡터를 찾기 위해서는 미지블록을 트랜지스터 레벨에서 분석해야 한다. 하지만 이러한 작업은 힘들기 때문에 본 논문에서는 미지블록을 두 종류로 분류하였다. 만약 미지블록이 수작업으로도 검사입력을 생성할 수 있을 만큼 복잡하지 않거나 미지블록에 대한 정확한 정보가 주어진다면, 모든 미지블록의 경로에 대해서 최소 초기화 벡터와 최소 전송 벡터가 각각 최소 무해저드 초기화 벡터와 최소 무해저드 전송 벡터가 될 수 있는지를 검사할 수 있다. 위의 경우를 만족한다면 이를 무해저드 미지블록(hazard-free USFB)이라고 명명하였다. 그 외의 경우는 해저드 미지블록(hazard USFB)이라고 부르기로 한다. 미지블록을 트랜지스터 레벨에서 접근할 수 없거나 수작업으로 하기에 너무 복잡한 경우에도 역시 해저드 미지블록이 된다.

무해저드 미지블록에 대해서는 앞의 벡터 V_1 을 최소 초기화 벡터로, 벡터 V 를 최소 전송 벡터로 사용할 수 있다. 하지만 해저드 미지블록에 대해서는 최악의 경우 미지블록 경로 전체에 대해 무해저드 전송 벡터가 존재하지 않을 수도 있다. 왜냐하면, 모든 경로의 입력들이 S_0 나 S_1 이 됨에도 불구하고 경로 입력에서의 천이가 미지블록의 출력에 동적 해저드를 유발할 수도 있기 때문이다. 따라서 해저드 미지블록을 포함하는 경로에 대해서는 HFR 검사입력의 존재를 보장할 수가 없으므로, 그러한 검사입력의 생성을 시도하지는 않는다. 해저드 미지블록에 대한 무해저드 초기화 벡터는 경로의 입력으로 논리값 X 를 가질 수 없다. 그렇지 않을 경우, 경로 입력이 초기값을 유지할 때 경로의 입력에서의 상승천이나 하강천이가 미지블록의 출력에 동적 해저드를 유발할 수도 있기 때문이다. 따라서, ROB 검사입력을 생성할 경우, 해저드 미지블록에 대한 무해저드 초기화 벡터는 경로의 입력으로 이진값을 가져야 하고 최소 전송 벡터는 경로의 입력으로 S_0 나 S_1 을 가져야 한다.

미지블록의 구조에 대한 정보가 제공된다면, 미지블록의 경로 지연에 대한 검사입력이 존재하지 않더라도 내부 경로의 지연에 대한 검사입력은 존재할 수 있다. 해저드 미지블록의 경우, SNR 이나 WNR 검사입력이 존재하지 않는다면, 내부 경로에 대해서도 역시 존재하지 않는다. 그러나 ROB 검사입력에 대해서는 미지블록의 구조를 안다면 내부 경로에 대한 검사입력은 존재할 수도 있다. 하지만 무해저드 미지블록에 대해서는, 임의의 내부 경로 역시 모든 검사입력(HFR, ROB, SNR, WNR)에 대해서 검사가 불가능하다.

미지블록의 전송 벡터를 구하기 위해 본 논문에서는 F 와 $\bar{F}$ 의 SOP(sum-of-products) 함수를 이용하였다. SOP으로 나타낼 때 F 와 $\bar{F}$ 의 모든 항(cube)은 주 함축항(prime implicant)이 되어야 한다. 여기서 항은 SOP에서 서로 논리곱의 형태로 된 집합이고, 항 c 로 표현되는 임의의 벡터가 함수 F 를 1이 되게 한다면, c 를 F 의 함축항(implicant)이라고 정의한다. c 에 포함되는 F 의 함축항이 더 이상 없다면 c 를 주 함축항이라고 한다. F 의 주 함축항과 $\bar{F}$ 의 주 함축항을 조합해서 경로 천이에 대한 전송 벡터를 구성하기 때문에 모든 주 함축항이 벡터에 포함되어야 한다. 만약 모든 주 함축항이 각각의 벡터에 포함되지 않는다면 가능한 증감 조건(sensitizing condition)들을 전부다 나타낸다고 할 수 없다. 반전이 일어나지 않는 경우, $\bar{F}_0(i=0)$ 의 항과 $F_1(i=1)$ 의 항을 곱한 결과에서 주 함축항들의 합으로 전송 벡터

$$BDNI(i) = \overline{F_0}(i=0) \cdot F_1(i=1)$$

를 구성한다. 이와 유사한 방법으로 반전이 일어나는 경우는 전송 벡터

$$BDI(i) = F_0(i=0) \cdot \overline{F_1}(i=1)$$

를 구성할 수 있다. 이와 같이 구성된 전송 벡터에 대해, $BDNI(i)$ 나 $BDI(i)$ 의 항 집합들은 각각 미지블록 입력 i 에 대한 모든 최소 비반전 전송 벡터나 최소 반전 전송 벡터의 집합과 같다.

위의 내용은 해저드 미지블록에 대한 ROB, SNR, WNR 검사입력을 구하는데 사용되었다.

입력 a와 b가 각각 $\overline{F_0}(i=0)$ 와 $F_1(i=1)$ 의 항일 때, a, b의 합집합을 ab라고 하자. 이 때 입력 b에는 있지만 입력 a에는 없는 모든 신호들은 두 번째 시간 프레임 동안에만 V_2 에 필요한 논리값을 갖는 모든 경로의 입력들이다. 입력 a, b 쌍에 의해 생성되는 항이 더 이상 어느 항도 포함하지 않는다면 이를 주 항이라고 명명하자. ab가 주 항이라면 ab는 앞의 벡터 쌍 $\langle V_1, V_2 \rangle$ 를 나타낸다. V_1 에서 이진값을 갖는 모든 경로의 입력들은 V_1 에서 V_2 로 천이하는 동안에 그 값을 유지한다. 위와 같이 모든 입력 a, b 항들의 합으로 구성된 항 ab가 더 이상 어느 것도 포함하지 않으면 SOP 형태의 전송 벡터 $BDrr(i)$ 를 구성할 수 있다. 이런 방식으로 구성된 $BDrr(i)$ 의 모든 항은 주 항이 된다. 이 때, $BDd_d(i)$ 에서 d_i 와 d_o 가 각각 미지블록의 경로 입력과 미지블록의 출력에 대한 천이의 방향(하강천이에 대해서는 f, 상승천이에 대해서는 r)을 나타낸다면, $BDd_d(i)$ 의 항 집합은 앞의 $\langle V_1, V_2 \rangle$ 벡터 쌍의 모든 가능한 집합과 같다.

무해저드 미지블록의 ROB 검사입력 생성에 위의 사실을 이용했다.

4×1 MUX를 이용하여 앞에서 설명된 무해저드 미지블록에 대한 검사입력 생성 과정의 예를 들 수 있다. MUX의 입력을 각각 i_0, i_1, i_2, i_3 라고 하고, 선택입력을 s_0, s_1 라고 할 때, 4×1 MUX의 모든 주 합축항을 포함하는 부울방정식은

$$F = i_0 \overline{s_0} \overline{s_1} + i_1 \overline{s_0} s_1 + i_2 \overline{s_0} s_1 + i_3 s_0 s_1 \\ + i_0 i_1 \overline{s_1} + i_2 i_3 s_1 + i_0 i_3 s_1 + i_0 i_2 \overline{s_0} + i_1 i_3 s_0 + i_0 i_1 i_2 i_3$$

따라서,

$$\overline{F} = \overline{i_0} \overline{s_0} \overline{s_1} + \overline{i_1} \overline{s_0} s_1 + \overline{i_2} \overline{s_0} s_1 + \overline{i_3} s_0 s_1 + \overline{i_0} \overline{i_1} \overline{s_1} \\ + \overline{i_2} \overline{i_3} s_1 + \overline{i_0} \overline{i_2} \overline{s_0} + \overline{i_1} \overline{i_3} s_0 + \overline{i_0} \overline{i_1} \overline{i_2} \overline{i_3}$$

가 된다.

만약 한 경로 입력이 i_0 를 포함한다면 상승천이의 비반전 전송 벡터는

$$BDrr(i_0) = \overline{F_0}(0)F_1(1) \\ = (\overline{s_0} \overline{s_1} + \overline{i_1} \overline{s_1} + \overline{i_2} \overline{s_0} + \overline{i_1} \overline{i_2} \overline{i_3})(\overline{s_0} \overline{s_1} + i_1 s_1 + i_2 s_0 + i_1 i_2 i_3) \\ = \overline{s_0} \overline{s_1} + \overline{i_1} \overline{s_1} \overline{s_0} + \overline{i_2} \overline{s_0} \overline{s_1} + \overline{i_1} \overline{i_2} \overline{i_3} \overline{s_0} \overline{s_1}$$

가 된다. 따라서 이 경로에 대한 ROB 검사입력의 경로의 입력은

$$s_0 = S0, s_1 = S0 \text{ 또는} \\ i_1 = S0, s_1 = S0, s_0 = X0 \text{ 또는} \\ i_2 = S0, s_0 = S0, s_1 = X0 \text{ 또는} \\ i_3 = S0, i_2 = S0, i_3 = S0, s_0 = X0, s_1 = X0$$

가 된다. 같은 식으로 경로 입력이 s_0 를 포함할 때 하강천이 비반전 전송 벡터는

$$BDff(s_0) = F_0(1)\overline{F_1}(0) \\ = i_1 s_1 \overline{i_0} + i_3 s_1 \overline{i_2} + i_1 s_1 \overline{i_0} \overline{i_2} + i_3 s_1 \overline{i_0} \overline{i_2} + i_1 i_3 \overline{i_0} \overline{i_2}$$

따라서 이 경로에 대한 ROB 검사입력의 경로의 입력은

$$i_1 = S1, s_1 = S0, i_0 = X0 \text{ 또는} \\ i_3 = S1, s_1 = S1, i_2 = X0 \text{ 또는} \\ i_1 = S1, s_1 = S0, i_0 = X0, i_2 = X0 \text{ 또는} \\ i_3 = S1, i_1 = S1, i_0 = X0, i_2 = X0 \text{ 또는} \\ i_3 = S1, s_1 = S1, i_0 = X0, i_2 = X0$$

가 된다. 여기서 x는 x입력에 대한 신호가 두 번째 시간 프레임에만 존재함을 의미한다.

4. 알고리즘

미지블록에 대한 경로 지연고장 검사입력 생성 알고리즘을 그림 2에 나타내었다.

일반적으로 검사입력을 찾는 과정에서 각 유형의 검사입력에 대해 먼저 경로의 입력의 제한 조건을 설정해야 한다. 경로 상

```

read boolean equations for USFBs
for all USFBs
    construct USFB type data structures
for all paths
    for a path
        if (prime) implicants for USFBs remain
            generate  $F$ s and  $\overline{F}$ s
            construct product terms
        if current USFB type is backward implicable
            do backward implication
        for all remaining (prime) implicants
            choose fanins and evaluate with equations
            if no conflict occurs then return TESTED
        else
            generate  $F$ s and  $\overline{F}$ s
            construct product terms
        return UNTESTED
  
```

그림 2 경로 지연고장 검사입력 생성 알고리즘

Fig. 2 Path delay fault test pattern generation algorithm

의 모든 소자들은 스택 형태의 지정 공간에 저장된다. 미지블록에 대해서는 경로의 입력의 제한 조건을 구하기 위해 부울방정식을 사용해야 하므로 매번 이를 계산할 경우 프로그램의 효율을 떨어뜨릴 수 있다. 따라서 미지블록에 대해서는 입력 제한 조건을 설정하지 않고 역방향 유추(backward implication)(이는 부분적인 경우이다)와 순방향 유추만을 수행한다. 해당 소자가 미지블록이 아니라면 그 유형을 검사하여 경로 입력의 천이에 적합한 경로의 입력 제한 조건을 설정하고 이를 지정 공간에 저장한다. 경로의 입력 제한 조건들은 모두 절대적이므로 설정 과정에서 충돌이 생긴다면 현재의 검사입력 유형으로는 이 경로에 대해 검사입력을 생성할 수 없다. 경로의 입력 제한 조건의 설정이 끝나면 역방향 유추(backward implication)가 수행된다. 이 과정에서 앞서 구성된 지정 공간이 검색된다. 지정 공간에서 차례로 선택하여 역방향 유추가 가능하다면 주 입력에 이를 때까지 반복해서 역방향 유추를 수행한다. 일반적으로 미지블록은 역방향 유추가 불가능하지만, SOP 형태의 항들이 하나의 신호만으로 구성된다면 역방향 유추가 가능하다. 그림 3에서처럼 두 가지 경우를 생각할 수 있는데, 미지블록의 출력이 0이고 SOP의 항들이 하나의 신호만으로 구성된다면 모든 항들이 0이 되어야 한다. 마찬가지로 미지블록의 역 SOP에서 항들이 하나의 신호만으로 구성되고 출력이 1이면 모든 항들은 1이 되어야 한다.

```

if(  $F = \bar{a} + b + \bar{c} = 0$  ) then
     $\bar{a} = 0, b = 0, \bar{c} = 0$  //유추가능
if(  $\bar{F} = \bar{a} + b + \bar{c}$  and  $F = 1$  ) then
     $F = a\bar{b}c = 1$  then
         $\bar{a} = 1, b = 1, \bar{c} = 1$  //유추가능
    
```

그림 3 미지블록의 역방향 유추 예

Fig. 3 Example of backward implication for USFBs

역방향 유추 과정에서 이미 설정된 값과 충돌된다면 이 유형의 검사입력으로는 현재의 경로에 대해 검사입력 생성이 불가능하다. 역방향 유추가 불가능하거나 부분적으로 가능하다면 스택 형태의 선택 공간에 저장하여 지정(justify) 과정에서 검색한다. 역방향 유추가 주 입력에 도달하면 나머지 주 입력에 임의의 값을 할당하고 순방향 유추를 수행한다. 이 과정에서 역방향 유추에 의해 이미 설정된 값과 충돌이 생기면 역추적(backtrack)이 이루어진다. 순방향 유추가 끝나면 남아 있는 소자들에 대해 지정이 수행된다. 일반적으로 미지블록은 선택할 수 있는 값이 다른 소자에 비해 많으므로 지정 과정에서는 남아 있는 미지블록의 SOP 함수를 생성한 다음 선택 공간에 저장하고 차례로 꺼내어 미지블록에 대한 지정이 가장 먼저 수행된다. 이 때 모든 주 함축항과 주 항을 모두 다 생성한다면 시간과 메모리의 부담이 커진다. 따라서 본 논문에서는 한정된 수의 주 함축항과 주 항만을 생성하여 경로의 입력 조건으로 시도하고, 실패했을 경우에만 남아있는 주 함축항과 주 항 중에서 고정된 수를 생성하는 방법을 택하였다. 또 탐색 공간을 줄이기 위해 선택 공간은 안정된 값이 가장 먼저 지정되고 그 다음으로 두번째 시간 프레임의 값이, 마지막으로 첫번째 시간

프레임의 값이 지정되도록 정렬해 놓는다. 선택된 소자의 출력이 필요한 값과 같으면 그 소자는 선택 공간에서 제거되고 남은 소자에 대해서 지정을 반복한다. 만약 소자의 출력값이 필요한 값과 충돌된다면 역추적을 통해 가장 최근에 할당된 주 입력을 선택 가능한 다른 값으로 할당하여 지정을 수행한다. 만약 더 이상의 선택 가능한 값이 남아있지 않다면 그 검사입력의 유형으로는 현재의 경로에 대한 검사입력의 생성이 불가능하게 된다.

그림 4의 회로를 통해 앞에서 제시된 알고리즘을 적용해 보자.

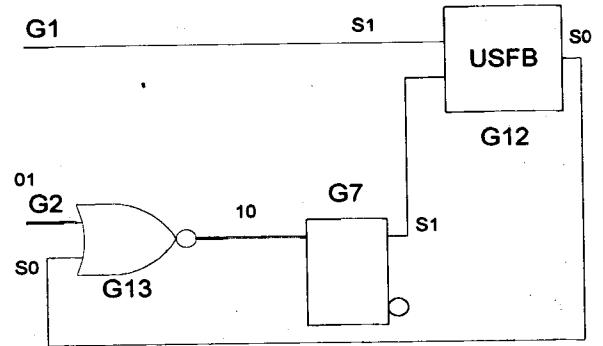


그림 4 미지블록을 포함한 회로에 대한 ROB 검사입력의 생성
Fig. 4 Robust test pattern generation for a circuit including an USFB

값은 선으로 표시된 부분이 검사입력의 생성 경로에 해당된다. 그리고 G12는 부울식이 $C = \bar{A} + B$ 인 미지블록이다. 경로 입력이 상승천이이고 G13의 게이트 유형이 NOR이므로 ROB 검사입력에 대한 경로의 입력의 제한조건을 구하면 S0이 된다. 따라서 경로의 입력에 해당하는 G12를 역방향 유추를 위해 지정 공간에 저장한다. 그 다음 역방향 유추를 수행하기 위해 G12를 지정 공간에서 꺼내어 역방향 유추가 가능한지를 조사한다. G12는 미지블록이고 역방향 유추가 불가능하므로 선택 공간에 저장되고 역방향 유추는 종결된다. 지정을 수행하기 위해 앞서 구성된 선택공간에서 G12를 꺼내어 지정해야 할 논리 값을 조사한다. G12는 S0을 가져야 하고 미지블록이므로 부울식을 통해 입력이 가져야 할 값을 계산한다. 본 논문에서 사용한 미지블록은 모두 해저드 미지블록이고 ROB 검사입력의 유형이므로 출력이 안정된 값을 가지기 위해서는 모든 입력이 안정된 값을 가져야 한다. 따라서 G1과 G7이 모두 S1이 되면 된다. 결국 이 경로에 대한 ROB 검사입력이 생성된다.

5. 결 과

앞에서 제시된 알고리즘은 약 22,000 줄 정도의 C 언어로 구현되었다. 일반적으로 검사입력의 생성은 표준 ISCAS 89 벤치 회로[11]를 이용하게 된다. 하지만 미지블록에 대한 표준 벤치 회로가 없기 때문에 ISCAS 89 벤치 회로 내의 소자들 중 일정 비율을 미지블록으로 바꾸어 실험하였다. 실험은 펜티엄 120MHz의 CPU, 16 Mbyte의 주 메모리를 가진 IBM PC에서 수행되었다. 검사 입력을 생성할 때 모든 경로를 다 고려하는 것은 고성능의 하드웨어를 요구하므로 본 실험에서는 정적 시간 분석을 바탕으로 가장 긴 경로 1000개를 선택하여 사

용하였다. 검사입력의 생성은 크게 세 단계를 거친다. 첫번째 단계에서 검사입력을 생성할 회로, 경로, 미지블록에 대한 방정식의 파싱이 이루어진다. 두번째 단계에서는 검사입력의 생성을 위한 데이터 구조를 형성하게 된다. 마지막 단계에서는 앞 단계에서 구성된 데이터 구조를 바탕으로 앞에서 제시된 알고리즘을 통해 실제 검사입력을 생성한다. 실험 결과를 표 1에 나타내었다. 표에서 ROB는 생성된 ROB 검사입력의 수를, SNR는 생성된 SNR 검사입력의 수를, WNR는 생성된 WNR 검사입력의 수를 나타낸다. 또 CPU는 검사입력의 생성에 소요된 시간을 초단위로 나타낸 것이다. 마지막으로 Mem은 사용된 메모리의 양을 Mbyte단위로 나타낸 것이다. 실험 결과를 분석해 보면 s713, s838, s1423 회로의 실험 결과에서 생성된 ROB 검사입력의 수가 상대적으로 적다. 이는 세 회로의 검사입력의 생성에 선택된 경로들이 실제로 ROB 검사입력을 생성하기가 어려운 이유도 있지만, 또 다른 이유는 실험에 사용된 미지블록들은 모두 해저드 미지블록이라는 사실에 있다. 보다 자세히 설명하면, 일반적으로 해저드 미지블록이 포함된 회로에서 생성된 ROB 검사입력의 수가 그렇지 않은 회로에서 생성된 수보다 적다. 그 이유는 회로에 해저드 미지블록이 포함되어 있다면 ROB 검사입력에 대해서는 그 미지블록의 경로의 입력들은 제한 조건으로서 두 시간 프레임 동안 정적 해저드가 없는 안정된 논리값을 가져야 하기 때문이다. 이는 일반적인 게이트만을 포함한 회로에서의 ROB 검사입력에 대한 경로의 입력의 제한 조건보다 더 엄격한 조건이다. 예를 들어 AND 게이트의 경로입력이 상승천이일 경우 ROB 검사입력이 되기 위해서는 경로의 입력은 두 시간 프레임 동안 X1의 값을 가져야 한다. 하지만 AND 게이트와 같은 기능을 하는, $f=xy$ 를 부울함수로 가지는 미지블록에 대해서는 경로입력이 상승천이일 경우 ROB 검사입력이 되기 위해서는 경로의 입력이 두 시간 프레임 동안 S1을 가져야 한다. 결국 미지블록의 개수가 많아질수록 ROB 검사입력이 되기 위해 경로의 입력이 가질 수 있는 값의 범위가 줄게 되므로 생성되는 ROB 검사입력의 수는 줄어들 수밖에 없다. 이는 미지블록의 경로의 입력 제한 조건의 정의에 따른 것으로 미지블록의 검사입력을 생성하기 위해서는 반드시 감수해야 하는 일반적인 현상이라 할 수 있다. 하지만 ROB 검사입력의 수가 줄어드는 만큼 SNR 검사입력이나 WNR 검사입력이 늘어나게 된다. 실험 결과의 비교 분석을 위해서는 미지블록을 다룰 수 있는 기존의 발표된 연구 결과가 있어야 하는데 현재까지는 미지블록에 대한 검사입력의 생성에 관해 발표된 연구결과가 없고 기존의 게이트 레벨 결과와도 경로가 다른 관계로 인해 제안된 검사입력 생성기의 성능을 정확히 비교 분석할 수가 없다. 그러나 실험 결과에서도 나타나듯 미지블록에 대한 부울연산 처리 시간의 증가에도 불구하고 전체 검사입력 생성시간은 실시간 내에 처리됨을 알 수 있다. 최근의 칩 생산에서는 양질의 칩을 만들기 위해 고장 검사(fault testing)에 점차 많은 비중을 두고 있으며 이에 따라 고장 검사에 소요되는 시간이 중요한 문제가 되고 있다. 따라서 고장 검사를 위한 검사입력 생성의 시간을 줄이는 것은 고장 검사 비용의 측면에서 매우 중요한 요소이며 제안된 검사입력 생성기는 이러한 점에서 우수한 성능을 보인다고 여겨진다. 그리고 제안된 검사입력 생성기는 기존의 검사입력 생성기들로는 완전한 검사입력의 생성이 어려운 미지블록에 대한 검사입력을 생성할 수 있으므로 VLSI 회로에 대한 검사입력 생성의 범위를

넓혔다는 점에서 매우 의미가 있다. 최근의 칩 설계 경향이 칩 면적과 전력 소모의 최소화를 위해 특정한 기능의 구현에 표준셀을 이용하거나 CMOS 레벨에서 직접 설계하여 사용하고 있다는 점을 고려할 때, 만약 이러한 블록들에 대한 부울함수만 주어진다면 제안된 검사입력 생성기를 통해서 검사입력의 생성이 가능하므로 보다 양질의 칩 생산을 이룰 수 있다. 이러한 점들은 제안된 검사입력 생성기의 산업·기술적 이용가치가 매우 높음을 시사한다.

표 1 검사입력 생성의 결과

Table 1 Test pattern generation results

회로	ROB	SNR	WNR	CPU(sec.)	Mem(Mbytes)
s713	0	14	108	17.16	11.8
s820	120	20	85	21.21	1.93
s832	120	20	85	21.09	1.84
s838	1	54	122	524.67	5.41
s1196	55	180	39	101.59	4.80
s1238	40	123	28	92.29	4.93
s1423	5	0	0	768.18	5.28
s1488	348	96	375	230.16	3.82
s1494	345	94	371	224.72	3.85
s5378	149	204	76	17350.15	13.09

6. 결 론

칩면적과 전력 소비의 문제를 해결하기 위해 표준셀이나 CMOS 회로로 직접 설계된 미지블록이 사용될 경우, 기존의 지연고장 검사입력 생성기로는 미지블록을 포함한 경로에 대해서 완전한 지연고장 검사입력의 생성이 불가능하기 때문에 미지블록을 다룰 수 있는 지연고장 검사입력 생성기를 설계하였다. 실험 결과에서도 알 수 있듯이 제안된 검사입력 생성기는 게이트 회로뿐만 아니라 미지블록을 포함하는 회로에 대해서도 효율적인 알고리즘을 바탕으로 빠른 시간 내에 검사입력을 생성할 수 있으므로 고집적회로의 지연고장 검사에 이용될 수 있다.

참 고 문 헌

- [1] Z. Barzilai and B. Rosen, "Comparison of AC Self-Testing Procedures", Proc. of International Test Conference, pp. 89-94, 1983.
- [2] J. Waicukauski, E. Lindbloom, B. Rosen, and V. Iyengar, "Transition Fault Simulation", IEEE Design and Test, pp. 32-38, April. 1987.
- [3] Gordon L. Smith, "Model for Delay Faults Based Upon Paths", Proc. of International Test Conference, pp. 342-349, 1985.
- [4] S. Reddy, C. Lin, and S. Patil, "An Automatic Test Pattern Generator for the Detection of Path Delay Faults", Proc. of International Conference on Computer Design, pp. 284-287, 1987.
- [5] M. Schulz, K. Fuchs, and F. Fink, "Advanced Automatic Test Pattern Generation Techniques for Path Delay

- Faults", Proc. of Fault Tolerant Computing Symp., pp. 44-51, 1989.
- [6] J. Savir and W. McAnney, "Random Pattern Testability of Delay Faults", Proc. of International Test Conference, pp. 263-273, 1986.
- [7] C. Lin and S. Reddy, "On Delay Fault Testing in Logic Circuits", IEEE Trans. on CAD, pp. 694-703, Sep. 1987.
- [8] M. Schulz, F. Fink, and K. Fuchs, "Parallel Pattern Fault Simulation of Path Delay Faults", Proc. of Design Automatic Conference, pp. 357-363, 1989.
- [9] F. Fink, K. Fuchs, and M. Schulz, "Robust and Non robust Path Delay Fault Simulation by Parallel Processing of Patterns", IEEE Trans. on Computers, pp. 1527-1536, Dec. 1992.
- [10] B. Underwood, S. Kang, and W. Law, "A Path-Delay Test Generator for Large VLSI Circuits", Proc. of International Conference on VLSI and CAD, pp. 368-371, 1993.
- [11] F. Brglez et al., "Combinational profiles of sequential benchmark circuits", Proc. of International Symp. Circuits and Systems, pp. 1929-1934, 1989.

저 자 소 개



강 성 호 (姜 成 昊)

1963년 4월 13일생. 1986년 서울대 공대 제어계측공학과 졸업. 1988년 The University of Texas at Austin 전기 및 컴퓨터공학과 졸업(석사). 1992년 The University of Texas Austin 전기 및 컴퓨터공학과 졸업(공학박).

1989년~1992년 Schlumberger Inc. Research Scientist. 1992년~1992년 The Univ. of Texas at Austin Post Doctoral Fellow. 1992년~1994년 Motorola Inc. Senior Staff Engineer. 1994년~현재 연세대 공대 전기공학과 조교수



임 용 태 (任 容 兌)

1970년 1월 1일생. 1995년 연세대 공대 전기공학과 졸업. 1995년~현재 연세대 대학원 전기공학과 석사과정