論文91-28A-12-10

조합회로에 대한 계층 구조적 테스트 패턴 생성 알고리듬의 비용 모델

(A Cost Model of Hierarchical Automatic Test Pattern Generation Algorithms for Combinational Logic Circuits)

閔 炯 福*

(Hyoung Bok Min)

要 約

본 논문에서는 테스트 패턴을 자동적으로 생성시켜주는 알고리듬의 비용 모델을 제시하였다. 조합회로에 대한 게이트 레벨 및 계층 구조를 이용한 테스트 패턴 생성 알고리듬의 비용을 모델링함으로써, G가 회로내의 게이트 갯수라 할 때, 게이트 레벨 알고리듬은 그 비용이 G^2으로 증가하는 반면, 계층 구조를 이용한 테스트 패턴 생성 알고리듬은 몇가지 조건하에서 Glog G로 증가함을 이 비용 모델을 이용하여 보였다. 이 비용 모델은 특정한 테크닉이 왜 테스트 패턴 생성 시스템의 성능을 개선시켜 주는지를 설명할 수 있게하기 때문에 테스트 패턴 생성 알고리듬을 분석하는 도구로써 유용하며, 실제로 계층 구조를 이용한 테스트 패턴 생성 알고리듬의 성능이 게이트 레벨 테스트 패턴 생성 알고리듬보다 우수하다는 것을 설명할 수 있었다.

Abstract

A cost model of test generation is presented in this paper. The cost of flat gate-level and hierarchical modular level test generation for combinational logic circuits are modeled. The model shows that the cost of hierarchical test generation grows as GlogG under some assuptions, while the cost of gate-level test generation grows G^2, where G is the number of gates in a circuit under test. The cost model derived in this paper is used to explain why some test generation techniques are faster and why hierarchical test generators should be faster than flat test generators on large circuits.

I. 서 론

디지탈 회로를 구현하는 반도체 칩의 복잡도가 매년 기하 급수적으로 증가함에 따라 이들 칩을 테스트하기가 날로 어려워지고 있다. 이러한 어려움을 극복하기 위하여, 지난 20여년 동안 자동 테스트 패턴 생성(automatic test pattern generation, ATPG) 알고리듬을 포함하여 테스트 기법에 관한 연구가 활발히 이루어져 왔다. 랜덤 패턴을 이용하여 테스트할 경우 테스트할 수 없는 결함들이 존재하게 되므로, 알고리듬을 이용한 자동 테스트 패턴 생성 과정이 불가피하게 되었다. 이러한 ATPG 시스템을 사용할 때에는 그 계산 과정이 컴퓨터 시간을 소모하기 마

*正會員, 成均館大學校 電氣工學科
(Dept. of Electrical Eng., Sungkyunkwan Univ.)
接受日字 : 1991年 7月 22日

련인데, 만약 그 시간이 현실적으로 너무 길다면 그러한 ATPG 시스템은 사용 불가 판정을 받게 될 것이다. ATPG에 걸리는 컴퓨터 시간 (비용, Cost) 은 여러가지 요소의 영향을 받는다. 회로내의 게이트 갯수, 회로의 구조, ATPG 알고리듬 등이 중요한 요소들이다. 회로의 크기가 증가함에 따라, 기존 알고리듬은 그 성능 부족으로 비 현실적인 알고리듬이 되어 버리므로, 더욱 성능이 개선된 새로운 알고리듬을 개발하여야 한다. 그러나 이러한 개발된 알고리듬이 현실적으로 사용할 수 있을 만큼 우수한 것인지를 알기 위해서는 그 알고리듬의 구현에 노력을 투자하기 이전에 그 알고리듬의 성능에 대한 모델링 (비용 모델)을 해 볼 필요가 있다.

이러한 필요에 부응하는 노력이 일부 이루어져 온다. 스캔 기법[1]을 이용하는 대형 디지탈 회로에 대한 ATPG 비용을 예측하기 위한 모델이 Goel[2]에 의하여 개발되었다. 그 비용 모델은 모델링 기법의 기초를 제공하였으나, 백트랙 (backtrack)이 필요없는 극히 비현실적인 ATPG 시스템에 대하여 모델링하였다. 그 후, 대부분의 ATPG 시스템이 미리 정해진 일정한 양의 백트랙을 행한 이후에는 테스트 생성을 포기한다는 점을 이용하여 현실화한 비용 모델이 개발되었다.[3]

최근 게이트 레벨 ATPG가 가지는 성능 개선의 한계를 극복하기 위하여 회로의 계층 구조(hierarchical description)를 이용하여 성능을 개선하려는 시도가 있으며,[4,5,6,7] 이러한 시도는 부분적으로 성공을 거두고 있다. 본 논문에서는 [3]에서 개발된 비용 모델을 기초로 하여, [7]에서 기술된 계층구조적인 ATPG (hierarchical automatic test pattern generation, HATPG) 시스템의 비용 모델을 추출함으로써 HATPG가 성능 개선을 이룰 수 있음을 보이고, 동시에 [3]에서 개발된 비용 모델을 ATPG 시스템의 성능 분석을 위하여 사용할 수 있음을 보이고자 한다.

이 논문은 다음과 같은 구조로 이루어져 있다. 제 II 절에서는 [3]에서 개발되고 검증된 ATPG의 일반적인 비용 모델을 기술한다. 이는 새로운 내용은 아니지만, 본 논문을 완성된 형태로 만들고, 독자의 편의를 위해서다. 제 III 절에서는 제 II 절에 기술된 모델을 이용하여 계층 구조적인 ATPC 시스템의 비용 모델을 추출한다. 제 IV 절에서는 제 III 절에서 추출된 비용 모델의 의미를 해석하여 계층 구조적 ATPG가 성능 개선을 이룰 수 있는 이유를 설명할 것이다. 제 V 절에서는 이 비용 모델에서 보인 바와 같이 계층 구조를 이용한 ATPG가 우수함을 보인 실험 결과를 제시하고, 제 VI 절에서 결론을 내리고자 한다.

II. 일반적인 비용 모델 (Generalized Cost Model)

비 용 모델을 추출함에 있어서 우리는 ATPG시스템이 경로 확성화 (path senitization) 방법을 이용하는 알고리듬[8,9,10,11,12]을 채택하고 있다고 가정한다. 또한, confilct를 감지하기 위한 heuristic은 ATPG 시스템마다 서로 다르고, 이러한 heuristic이 없어도 테스트 패턴 생성이 가능하므로 heuristic 수행에 드는 비용 (시간)은 직접 모델링하지 않았다. 이러한 heuristrc의 예는 PODEM[9]의 x-path checking, FAN[10]의 unique sensitzation 및 multiple backtrace, TOPS[11]의 dominator, SOCRATES[12]의 dynamic unique sensization 및 learning 등을 들 수 있다.

ATPG가 목표로 하는 결함 (target faults)의 갯수를 f라 하자. 이러한 결함의 갯수는 다음과 같이 표시될 수 있다.

$$f = \sum_{n=0}^{2^m-1} fn \qquad (1)$$

여기서 f_n이란 n개의 백트랙 (backtracking)을 사용하여 테스트 패턴이 생성되거나 테스트 불가능함이 증명될 수 있는 결함의 갯수를 의미하며, m은 알고리듬의 search space의 차원 (dimension)의 수를 의미한다. 예를 들면, PODEM의 경우 m은 회로의 입력단자 (primary inputs)의 수를 의미한다.

목표 결함들 중에서 테스트 패턴을 생성시키기 쉬운 결함에 대하여 테스트 패턴을 생성하는데 드는 평균 비용 (시간)을 p라 하자. 여기서 테스트 패턴을 생성시키기 쉽다는 말은 백트랙을 전혀 사용하지 않고 테스트 패턴을 생성시킬 수 있다는 의미로 사용된다. 따라서 p라는 비용은 목표 결함 하나에 대한 implication, backtrace등과 같이 ATPG에 사용되는 기본 동작을 위한 비용을 의미한다.

백트랙을 한번 수행하는데 드는 비용은 αp로 모델링하였다. 경로 확성화 방식의 ATPG에서는 경로를 선택할 때마다 경로의 선택을 위한 결정 과정 (decision process)이 필요하며, 그 선택이 틀린 것으로 밝혀질 때에는 백트랙을 수행하고, 이러한 모든 결정에는 그 결정을 검증하기 위한 비용이 수반된다. 그 비용은 바른 결정이든 잘못된 결정이든 평균적으로 같은 크기의 비용을 수반하므로, α는 쉬운 결함에 대하여 테스트 패턴을 생성하는데 필요한 바른 결정의 갯수의 역이 된다. 이와 같이 백트랙 비용을 모델링할 때, n개의 백트랙을 수행하는데 드는 비용은 $n\alpha p$로 나타낼 수 있다.

이제까지 제시된 파라미터들을 사용하여 ATPC비용, C_t를 다음과 같이 나타낼 수 있다.

$$C_t = f_0 p + f_1(p+\alpha p) + f_2(p+2\alpha p) + \cdots + f_{2^n-1}$$

$$(1+2^n-1)\alpha)p \quad = p\sum_{n=0}^{2^m-1} f_n(1+n\alpha) \qquad (2)$$

식(2)는 ATPG비용은 백트랙이 필요없는 경우, 백트랙이 1번, 2번, 3번, …필요한 경우의 비용의 합이라는 극히 단순한 사실을 식으로 표현한 것이다.

백트랙을 테스트 패턴이 발생될 때까지 계속하면 비현실적으로 긴 시간을 소요하므로, 모든 ATPG 시스템에서는 백트랙의 수가 일정한 한도를 넘어가면 테스트 패턴 생성을 포기한다. 이러한 백트랙 숫자의 한도를 l이라 하면, 백트랙이 l을 넘는 결함에 대해서는 백트랙을 l번만 수행하므로 식(2)는 다음과 같이 쓸 수 있다.

$$C_t = p\{\sum_{n=0}^{l} f_n(1+n\alpha) + f_l^*(1+l\alpha)\}, \qquad (3)$$

$$f_l^* = \sum_{n=l+1}^{2^m-1} f_n, \qquad (4)$$

여기서 f_l^*은 테스트 패턴 생성에 l이 넘는 백트랙이 필요한 결함의 숫자이며, 이러한 결함에 대해서는 ATPG를 포기하므로 l회의 백트랙만을 수행하고 끝이 나며, 이러한 결함을 aborted fault라고 부른다. 식(3)은 선형 함수이므로 배분 법칙에 따라 나누어 쓸 수 있다. 즉,

$$C_t = p\{\sum_{n=0}^{l} f_n + f_l^* + \alpha\sum_{n=0}^{l} nf_n + \alpha l f_l^*\} \qquad (5)$$

또한 이 식의 제 1항과 제2항을 결합시키면 모든 결함의 갯수, f와 같으므로 다음 식이 성립한다.

$$C_t = p\{f + \alpha l\ f_l^* + \alpha\sum_{n=0}^{l} nf_n\} \qquad (6)$$

식(6)에 나타난 세개의 항들 중에서 두번째 항, $\alpha l f_l^*$은 ATPG가 포기된 목표 결함에 대한 백트랙에 쓰인 비용이며, 세번째 항은 테스트 패턴 생성에 성공한 결함의 백트랙에 쓰인 비용이다. 이와 같은 테스트 패턴 생성에 성공한 결함의 백트랙 숫자의 평균치를 b_f라 하고 식(6)을 다시 쓰면 식의 의미가 더욱 확실하다. b_f는 그 정의에 의하면 다음과 같다.

$$bf = \frac{1}{f}\sum_{n=0}^{l} nf_n \qquad (7)$$

또한, f_l^*은 ATPG에 실패한 결함의 갯수이며, ATPG의 성공도를 나타내기 위하여 일반적으로 사용되는 개념인 Fault Coverage를 도입할 수 있다. Fault Coverage(FC)는 다음과 같이 정의된다.

$$FC = \frac{f - f_l^*}{f} \qquad (8)$$

식(7)과 식(8)을 식(6)에 결합시키면 다음 식을 얻는다.

$$C_t = pf\{1 + \alpha l(1-FC) + \alpha b_f \qquad (9)$$

식(9)가 일반적인 ATPG 비용 모델이다. 이 식에서 FC와 b_f는 ATPG에 사용된 휴리스틱 (heuristics)이 얼마나 효율적이었는지를 반영하는 파라미터이며, α는 필요한 백트랙의 수를 반영한다. 첫번째 항목, pf는 Goel[2]이 모델링했던 백트랙이 필요 없는 테스트 패턴 생성 시스템의 비용을 나타낸다. 중괄호에 싸인 부분은 백트랙 비용을 나타낸다. 중괄호에 싸인 부분의 두번째 항은 ATPG에 실패한 결함에 대한 백트랙 비용을 나타내고 세번째 항은 ATPG에 성공한 결함에 대한 백트랙 비용을 나타낸다.

Ⅲ. 계층 구조적 ATPG 시스템의 비용 모델

계층 구조적 ATPG 시스템의 비용 모델은 제Ⅱ절에서 기술된 모델에 근거하여 추출할 수 있다. 계층 구조적 ATPG 시스템에 대해서는 [7]에 자세히 기술되어 있으므로 본 논문에서는 그 개요만을 간단히 언급하고자 한다.

디지탈 회로의 크기가 급격히 증가함에 따라, 디지탈 회로를 설계할 때, 계층 구조를 이용하는 top-down 설계 기법이 일반화 되었다. 즉 디지탈 회로를 고유의 기능을 갖는 모듈 (module)로 갈라주고, 이 모듈들을 다시 의미있는 더 작은 모듈로 갈라서 최종적으로는 게이트 혹은 트랜지스터 레벨까지 내려가 설계되는 기법을 사용하지 않을 수 없게 되었다. 계층 구조적 ATPG시스템은 이러한 설계 기법에서 파생되는 각 모듈의 기능을 이용하여, ATPG동작을 게이트나 트랜지스터에 대하여 행하는 대신 모듈에 대하여 동작시킴으로써 성능을 급격히 향상시킨다.

회로의 계층 구조를 이용한 ATPG(HATPG) 시스템의 비용 모델을 추출하기 위하여, 그 디지탈 회로가 그림1에 보인바와 같이 k-ary tree 구조를 갖는 것으로 가정한다. 그림1은 3계층 4-ary tree를 보이고 있으며, leaf 노드는 게이트를 나타내고, root 노드는 회로 전체를 나타낸다. 중간 노드들은 회로 내의 고유한 기능을 갖는 모듈들을 의미한다. 물론 대부분의 회로가 이와 같은 완전한 트리구조(complete tree)가 아니지만, 이와 같이 가정함으로써 문제를 단순화하여 추출된 비용 모델의 의미를 보다 쉽

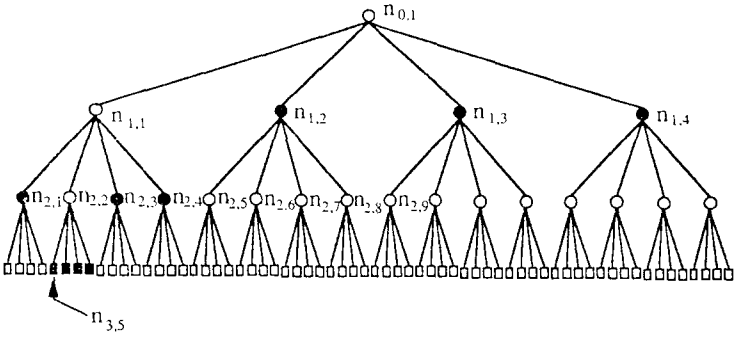

그림 1. 비용 계산에 사용된 회로의 계층 구조 모델
Fig. 1. Circuit hierarchy used to compute ATPG cost.

게 이해할 수 있게 할 것이다. 각 게이트는 f_g개의 결함을 가지고 있다고 가정하며, 모든 결함은 백트랙 없이 테스트 패턴을 생성할 수 있다고 가정한다. 백트랙에 관한 문제는 추후 세 II 절에서 보인 모델과 결합시킴으로써 해결할 것이다.

그림1에서 까만색으로 채워진 노드들은 목표 결함이 $n_{3,5}$에 있을 경우 계층 구조적 ATPG 시스템에서 테스트 패턴 생성에 사용되는 노드들을 나타낸다. HATPG에서는 목표 결함이 있는 노드 (게이트)와 그 노드의 이웃 노드(siblings) 및 목표 결함이 있는 노드의 상위 노드들 (ancestors)의 이웃 노드들에 대하여 ATPG 동작을 하게 된다. 이는 빠진 것이 없는 완전한 회로 모델에 대하여 ATPG 동작을 할 수 있게 하면서도 게이트 레벨의 결함 모델을 사용할 수 있게 하고 가장 적은 갯수의 회로 모델에 대하여 ATPG 동작을 함으로써 고성능 시스템을 구현할 수 있게 한다.

그림1에 보인 바와 같이 회로의 i번째 계층의 j번째 노드를 $n_{i,j}$로 나타내고 그 모듈(노드)에 대하여 ATPG 동작에 걸리는 비용 (시간)을 $s_{i,j}$로 나타내자. 이때 그림1에서 표시된 바와 같은 목표 결함 $n_{3,5}$에 대하여 ATPG에 드는 비용 $C_{3,5}$는 다음과 같이 구할 수 있다.

$$C_{3,5}=(s_{1,2}+s_{1,3}+s_{1,4})+(s_{2,1}+s_{2,3}+s_{2,4})+(s_{3,5}+s_{3,6}+s_{3,7}+s_{3,8}) \tag{10}$$

모든 게이트가 f_g개의 결함을 가지고 있다고 가정했으므로, 게이트 $n_{3,5}$의 결함에 대하여 테스트 패턴을 생성하기 위한 비용은 $f_gC_{3,5}$로 표시되며, $C_{3,1}$, $C_{3,2}$, …, $C_{3,64}$에 대해서도 비슷한 식을 유도할 수 있고, 이들 비용의 합계가 3계층을 갖는 회로에서의 ATPG 비용, C_3가 된다. 즉,

$$C_3=f_g(4^2(4-1)\sum_{j=1}^{4}s_{1,j}+4^1(4-1)\sum_{j=1}^{4^2}s_{2,j}+4\sum_{j=1}^{4^3}s_{3,j}) \tag{11}$$

이 식은 회로의 계층구조가 L계층이고, 모든 모듈이 k개의 모듈을 그 내부에 갖는 (L-level k-ary tree) 회로 구조에 대하여 일반화 될 수 있다. 즉, 이러한 회로에 대한 ATPG 비용은 다음과 같다.

$$\begin{aligned}C_L&=f_g(k^{L-1}(k-1)\sum_{j=1}^{k}s_{1,j}+k^{L-2}(k-1)\sum_{j=1}^{k^2}s_{2,j}+\cdots+k\sum_{j=1}^{k^L}s_{L,j})\\&=f_g(k-1)\sum_{i=1}^{L}\left(k^{L-i}\sum_{j=1}^{k^i}s_{i,j}\right)+f_g\sum_{j=1}^{k^L}s_{L,j}\end{aligned} \tag{12}$$

이 식을 보다 단순화하기 위하여, 각 모듈에 대한 ATPG 비용의 평균치의 개념을 도입한다. 즉, i번째 계층에 있는 모듈에 대한 ATPG 비용의 평균치를 $\hat{s}_i$로 표시하면, i번째 계층에는 모두 k^i개의 모듈이 존재하므로 $\hat{s}_i$는 다음과 같이 나타낼 수 있다.

$$\hat{s}_i=\frac{1}{k^i}\sum_{j=1}^{k^i}s_{i,j} \tag{13}$$

식12에 표현된 계층 구조적 ATPG의 비용은 식(13)을 이용하여 다시 쓸 수 있으며 그 식은 다음과 같이 표시된다.

$$C_L=f_g(k-1)\sum_{i=1}^{L}(k^i\hat{s}_i)+f_g\sum_{j=1}^{k^L}s_{L,j} \tag{14}$$

회로 내의 게이트 갯수(tree의 leaf 노드의 갯수)를 G라 하면, $G=k^L$으로 표현된다. 또한 게이트 하나하나에 대한 ATPG 비용은 거의 같다고 할 수 있으므

로, 게이트 하나에 대한 ATPG 비용을 s라 하면, 식(14)는 다음과 같이 쓸 수 있다.

$$C_L = f_g(k-1)G\sum_{i=1}^{L}\hat{s}_i + f_g G s \qquad (15)$$

트리 구조내의 모든 모듈과 게이트들 각각에 대한 ATPG비용의 평균치를 ŝ으로 표시하면 ŝ은 다음과 같이 나타낼 수 있다.

$$\hat{s} = \frac{1}{L}\sum_{i=1}^{L}\hat{s}_i = \frac{1}{L}\sum_{i=1}^{L}\left(\frac{1}{k^i}\sum_{j=1}^{k^i} s_{i,j}\right) \qquad (16)$$

식(15)와 식(16)을 결합시키면 그 의미가 보다 확실한 다음과 같은 식을 얻을 수 있다.

$$\begin{aligned} C_L &= f_g G\{(k-1)\hat{s}\ L + s\} \\ &= f_g G\{(k-1)\hat{s}\ log_k(G) + s\} \end{aligned} \qquad (17)$$

이 식이 백트랙이 사용되지 않을 경우에 대한 계층 구조적 ATPG 시스템의 비용 모델이다. 일반적인 비용 모델에서의 백트랙 비용에 대해서는 제 II 절에서 기술하였으므로, 이를 이용하여 식(17)과 결합 시킴으로써 백트랙까지 고려한 비용 모델, C_h을 얻을 수 있다.

$$\begin{aligned} C_h &= C_L\{1+\alpha b f + \alpha l(1-FC)\} \\ &= f_g G\{(k-1)s\ log_k(G) + s\}\{1+\alpha b_f + \alpha l(1-FC)\} \end{aligned} \qquad (18)$$

이 식이 계층 구조적 ATPG시스템의 비용 모델이며, 편의상 각 파라미터의 의미를 다시 요약해 보기로 한다.

f_g = 게이트 하나에 들어있는 결함 갯수의 평균치
G = 회로내의 게이트 갯수
k = 회로의 계층 구조에서 하나의 모듈내에 있는 모듈의 갯수
s = 회로내의 모든 모듈에 대한 ATPG비용의 평균치
s = 게이트 하나에 대한 ATPG 비용
α = 백트랙 한번에 드는 상대 비용 비율
b_f = 결함 하나당 백트랙 회수의 평균치
l = 백트랙 횟수의 제한치
FC = Fault Coverage

Ⅳ. 계층 구조를 이용한 ATPG의 비용 모델 해석

제 III 절에서는 계층 구조를 이용한 ATPG의 비용 모델을 추출하였다. 이제 그 비용 모델의 의미를 해석해 보고자 한다. 백트랙 (backtrack)에 관한 부분, 식 (18)의 두번째 중괄호에 싸인 부분)은 이미 제 II 절에서 다루었고 계층 구조를 이용한 ATPG 시스템에서도 그 논의가 유효하므로, 계층 구조를 이용한 ATPG 시스템 고유의 특성(식(17)으로 표시되는 부분)에 대해서만 논의하기로 한다.

식(17)에 표현된 계층 구조를 이용한 ATPG의 비용 모델을 해석하기 위하여 두가지의 극단적인 경우에 대하여 고려해 보기로 한다. 그 하나는 가장 낙관적인 해석이다. 즉, 계층 구조를 이용한 ATPG 시스템이 가장 고성능을 낼 수 있는 조건을 제시하고, 그 경우의 성능을 알아 본다. 다른 하나는 가장 비관적인 해석이다. 즉, 계층 구조를 이용한 ATPG 시스템의 성능이 어떤 경우에 가장 떨어지며 그 경우의 성능은 어떠한가 살펴 본다.

1. 낙관적 해석

디지탈 조합회로의 계층 구조(hierarchy) 내에 있는 게이트를 포함한 모든 모듈에 대한 ATPG동작이 모든 목표 결함에 대한 테스트 패턴 생성 과정을 통하여 평균적으로 같은 시간 (비용)에 이루어진다고 가정해 보자. 즉, 회로내의 모든 모듈이 매우 간단한 기능을 가져서 모듈에 대한 처리 시간이 모듈의 크기에 무관한 경우이며, 이는 가장 낙관적인 가정이다. 이때 모든 모듈의 처리 시간을 s_h라 하면 다음 식이 성립한다.

$$s_{i,j} = s = s_h \qquad (19)$$

이 식을 식(16)과 식(17)에 대입하면 다음 식이 얻어진다.

$$C_{h,l} = f_g\ G\ s_h\{(k-1)log_k(G)+1\} \qquad (20)$$

이 식에서 (f_g G)는 테스트 패턴 생성을 위한 회로 내의 모든 목표 결함의 갯수를 나타낸다. 따라서 ATPG 비용이 목표 결함의 수에 비례함은 상식과 일치한다. 또한 이러한 목표 결함의 수는 테스트 패턴의 질을 유지하기 위하여 게이트 레벨 결함 모델을 이용하는 한 변화할 수 없다. 이미 언급한 바와 같이, 계층 구조를 이용한 ATPG 시스템에서는 테스트 패턴의 질을 유지하면서도 패턴 생성의 비용을 낮추는 것이 그 목표이다.

식(20)의 중괄호 안에 들어 있는 표현, $\{(k-1)\log_k(G)+1\}$는 ATPG에 사용되는 모듈의 갯수를 의미한다. 그림1에서 까만색으로 채워져 있는 노드들이 목표 결함이 $n_{3,5}$에 존재할 경우 테스트 패턴 생성에 이용되는 모듈들을 나타낸다. 그 갯수는 단지 몇단계의 과정을 거치면 구해낼 수 있다.

k값은 하나의 모듈안에 들어 있는 계층 구조상 하

나 아래 계층의 모듈 갯수를 의미한다. 이 갯수는 모듈의 기능에 달려 있을 뿐, 모듈의 크기와는 무관하며, 회로 설계시 설계자가 관리할 수 있는 크기로 제한된다. 따라서 이 갯수 보다는 또하나의 변수, G의 의미가 더욱 중요하다. 즉 ATPG에 사용되는 모듈의 갯수는 log(G)에 비례한다는 점이다. 회로 크기(G)가 매우 클 때, G의 변화에 대한 log(G)의 변화는 극히 작다.

식(20)에서 ATPG의 비용은 Glog(G)에 비례하므로, 회로 크기에 따른 ATPG비용의 증가율은 게이트 레벨 ATPG의 증가율(G^2에 비례, [3]참조)에 비하여 극히 작다. 따라서, 계층 구조를 이용하는 ATPG 시스템은 게이트 레벨의 평면적인 구조의 회로 모델을 사용하는 ATPG 시스템에 비하여 훨씬 고성능이다. 이를 달리 표현하면 계층구조를 이용하는 ATPG 시스템이 테스트 패턴을 생성시킬 수 있는 회로 크기의 한계가 크다는 것을 의미한다.

2. 비관적인 해석

ATPG를 수행함에 있어서 회로내의 각 모듈의 처리 시간이 그 모듈내에 있는 게이트 수에 비례한다고 가정해 보자. 이는 모듈 기능이 매우 복잡하여, 게이트 대신에 모듈을 ATPG에 사용하는데 따를 이득이 없는 경우이다. 이러한 경우에 대한 비용 모델을 추출하면 다음과 같다.

회로 내의 모듈의 처리 시간, $s_{i,j}$가 그 모듈 내의 게이트 수에 비례하면, i번째 계층에 있는 모듈은 그 내부에 k^{L-1}개의 게이트를 가지므로, 다음과 같은 식이 성립한다.

$$s_{i,j}=k^{L-i}\ s \tag{21}$$

이 식을 식(16)의 $s_{i,j}$에 대입하여 그 결과를 식(17)에 대입하면 다음과 같은 비용 모델이 추출된다.

$$C_{h,2}=f_g\ s\ G^2 \tag{22}$$

이 식에서는 ATPG 비용이 회로 크기의 제곱에 비례함을 보이고 있으며, 이 비용 모델은 [3]에 나타난 게이트 레벨 ATPG의 비용 모델과 같다. 즉, 계층구조를 이용하는 ATPG 시스템에서 모듈을 처리하는 비용이 그 모듈내의 회로 크기에 비례하면, 게이트 레벨 ATPG 비용과 같아지므로, 계층구조를 이용하는 ATPG 시스템의 이득이 없어진다.

3. 현실적인 해석

위에서 언급한 두가지 경우에는 계층 구조적 ATPG 시스템의 비용에 대한 극단적으로 낙관적인, 또한 극단적으로 비관적인 경우를 보이고 있다. 몇가지 연구에서 보였듯이,[4,5,6,7] 모듈의 의미있는 기능을 가지는 한, 모듈을 처리하는 시간은 그 내부의 게이트를 처리하는 시간보다는 훨씬 짧은 것이 보통이며, 그 시간이 회로 크기에 비례하는 것도 아니다. 따라서 실제의 계층 구조적 ATPG 시스템에서 비용 모델은 식(20)과 식(22)를 각각 lower bound 및 upper bound로 하는 중간이 될 것이다. 정확한 위치는 ATPG에 사용되는 모듈의 기능이 얼마나 간단한가 하는 정도에 달린 것이므로 일률적으로 규정할 수는 없다. 계층 구조적 ATPG 시스템의 하나인 FANHAT시스템[7]의 실험 결과는 이러한 비용 모델을 뒷받침하고 있다.

V. 실험결과

이제까지 회로의 계층구조를 ATPG에 이용하는 경우의 비용 모델을 추출하고, 그 의미를 해석해 보았다. 여기서는 실제의 계층 구조적 ATPG시스템에 대한 실험결과를 제시함으로써 비용 모델이 제시하는 것처럼 계층 구조적 ATPG 시스템이 더 우수함을 보이고자 한다. FAN 알고리듬에 기반을 둔 계층 구조적 ATPG하나가 본 저자에 의하여 발표된 바 있다.[7] 그 논문에서는 계층구조적 ATPG시스템의 알고리듬, 휴리스틱 등의 내용과 실험 결과만을 기술하였고, 당시에는 왜 계층 구조적 ATPG시스템이 더 우수한가를 설명할 이론적 기반이 없어서 발표할 수 없었다. 이제 본 논문에서는 그 이론을 제시하였으며, 실험 결과를 본 절에 제시하고자 한다. 실험 내용은 이미 발표된 바[7]와 같으나, 본 논문에 부가하여 논문 체제를 갖추고, 독자의 편의를 기하고자 한다.

실험은 본 저자에 의해 발표된 바와 같이,[7] FANHAT(fanout oriented hierarchical automatic test generation system)을 이용하였다. 이 ATPG 시스템은 계층구조적으로 기술된 회로를 입력으로 읽어 들여서 출력으로 테스트 패턴을 생성시킨다. FANHAT은 회로 모듈에 사용될 수 있는 휴리스틱을 부가하면 HATPG로 작동하며, 모듈 휴리스틱을 떼어내면 flat(게이트 레벨) ATPG로 작동되도록 설계되어 있으므로, 두가지 방법의 비교를 위하여 똑같이 FANHAT을 사용하였다. 게이트 레벨로 사용될 경우에는 FAN 알고리듬[10]으로 작동된다. FANHAT은 C언어로 작성되었으며, 실험은 UNIX 운영체제를 사용하는 DIGITAL Micro Vax II GPX를 이용하였다. 실험은 3개의 회로에 대하여 수행하였다. 비교적 작은 크기의 회로로는 3-line-to-8-line decoder(27gates),

4-bit ALU(74LS181, 65 gates)를 사용하였고, 크기가 큰 회로는 24-bit-by-24-bit fast multiplier(3903 gates)를 사용하였다.

표1, 2는 각각 3-to-8 디코더와 74LS181에 대한 ATPG에 걸린 CPU시간을 표시하였다. 3-to-8 디코더는 1-to-2 디코더 1개와 2-to-4 디코더 2개로 구성되었으며, 2-to-4 디코더는 1-to-2 디코더 3개로 구성되었다. 이러한 각각의 모듈에 휴리스틱을 적용하였을 경우와 휴리스틱 없이 게이트 레벨로만 ATPG를 하였을 경우 결과를 보였다. 74LS181의 경우 입력측에 있는 4개의 함수 선택 모듈 (Function selectors)과 출력측의 Lookahead Carry Generator 모듈 및 몇개의 기타 게이트로 이루어져 있다. 이러한 모듈에 휴리스틱을 적용하여 계층 구조적으로 ATPG를 수행한 결과와 게이트 레벨만을 적용한 결과를 표2에 보였다. 이와 같이 작은 회로의 경우 계층 구조를 사용함에 따라 생기는 성능개선은 미미하다.

표3에는 보다 큰 회로에 대한 실험 결과를 보였다. 실험회로는 24비트 숫자를 곱할 수 있는 곱셈기로서 게이트 레벨에서는 3901개의 게이트를 포함하고 있다. 이 곱셈기는 4비트 곱셈기, 4비트 덧셈기 등으로 이루어져 있다. 이와 같은 내부 모듈의 휴리스틱을 이용하는 경우 (계층 구조적 ATPG)와 휴리스틱 없이 게이트만을 이용하여 ATPG를 수행하는 경우의 결과를 보이고 있다. 표1, 2에서의 회로들의 경우, 회로가 간단하여, CPU시간을 제외한 다른 ATPG결과가 같으나, 표3에 보인 곱셈기의 경우 CPU 시간

표 1. 3-8 디코더 회로에 대한 테스트 패턴 생성시간(CPU time:단위=초)
Table 1. Test generation time for 3-to-8 decoder CPU time in seconds.

Gate Level (No Heuristics)	1-to-2 decoder Heuristics	2-to-4 decoder Heuristics	Both Heuristics
3.60	2.23	2.54	2.12

표 2. 74LS181 회로에 대한 테스트 패턴 생성시간(CPU time:단위=초)
Table 2. Test generation time for 74LS181 CPU time in seconds.

Gate Level (No Heuristics)	Function selector Heuristics	Lookahead Carry Heuristics	Both Heuristics
46.7	37.0	41.2	33.9

표 3. 24×24 Fast multiplier회로에 대한 테스트 패턴 생성 결과
Table 3. Test generation result for 24×24 fast multiplier.

ATPG Parameters	Gate Level (No Heuristics)	Multiplier Heuristics	Adder and Multipler Heuristics
CPU Time in seconds	12761	6932	1421
Backtracks	10272	10459	8929
Redundant Faults	258	258	258
Aborted Faults	66	64	65

이외의 다른 ATPG요소들의 값이 다르므로 이들을 보였다. 그 요소들은 CPU시간, 백트랙의 갯수, 테스트 패턴이 존재하지 않는 결함(Redundant faults)의 갯수, 테스트 패턴 생성을 실패한 결함(Aborted faults)의 갯수 들이다. 이 자료에서 백트랙의 갯수는 계층구조를 이용할 때 현저히 줄어들며, ATPG에 걸리는 시간은 획기적으로 줄어 든다는 것을 알 수 있다. 특히, 표1과 2에서 보였던 작은 회로에서는 성능개선이 미미하지만, 표3에서 보인 대형회로에서는 그 성능 차이가 훨씬 크며, 이 사실은 앞 절에서 보인 비용 모델로 설명할 수 있다.

Ⅵ. 결 론

본 논문에서는 계층 구조를 이용하는 테스트 패턴 자동 생성 시스템의 비용 모델을 추출하였다. 그 비용의 lower bound와 upper bound는 각각 회로내의 게이트 수를 G라 할 때 G log(G) 및 G^2에 비례하는 것으로 나타내어지며, 더우기 모듈의 처리 비용이 어느 모듈의 경우나 비슷할 정도로 모듈의 기능 함수가 간단한 경우에는 G log(G)에 비례할 수도 있음을 보였다. 이는 게이트 레벨의 평면적인 회로 모델을 이용하는 경우에 비하여 계층 구조를 이용하는 테스트 패턴 자동 생성 시스템의 성능이 일반적으로 훨씬 우수함을 나타낸다.

이 비용 모델은 본 논문에서 보인 바와 같이 계층 구조를 이용한 테스트 패턴 생성 알고리듬이 테스트 패턴 생성 시스템의 성능을 개선시켜 주는 이유를 설명할 수 있게 하기 때문에, 다른 특정한 테크닉이 테스트 패턴 생성의 성능을 개선할 경우 그 테스트 패턴 생성 알고리듬을 분석하는 도구로서 유용하게 사용할 수 있다. 또한, 이러한 비용 모델을 결함 시뮬레이션의 비용 모델과 결합시킨다면, 테스트 패턴

자동 생성에 필요한 시간을 미리 예측하게 하여 주어진 회로에 대한 테스트 패턴 생성이 현실적으로 가능한 시간 내에 가능할 것인지 판단할 수 있다.

參 考 文 獻

[1] E.B. Eichelberger, and T.W. Williams, "A Logic Design Structure for LSI Testing," *Proceedings of the 14th Design Automation Conference,* pp. 462-468, June 1977.

[2] P. Goel, "Test Generation Costs Analysis and Projections," *Proceedings of the 17th Design Automation Conference,* pp. 77-84, June 1980.

[3] H.B. Min, and W.A. Rogers, "Search Strategy Switching: An Alternative to Increased Backgracking," *Proceedings of the International Test Conference,* pp. 803-811, Aug. 1989.

[4] B. Krishnamurthy, "Hierarchical Test Generation: Can AI Help?" *Proceedings of the International Test Conference,* pp. 694-700, Sept. 1987.

[5] H.W. Daseking, R.I. Gardner, and G.B. Weil, "VISTA: VLSI CAD System," *IEEE Transactions on Computer-Aided Design,* vol. CAD-1, no. 1, pp. 36-52, Jan. 1982.

[6] B.T. Murray, and J.P. Hayes, "Hierarchical Test Generation Using Pre-Computed Tests for Modules," *Proceedings of the International Test Conference,* pp. 221-229, Sept. 1988.

[7] H.B. Min, W.A. Rogers, and H.A. Luh, "FANHAT: Fanout Oriented Hierarchical Automatic Test Generation System," *Proceedings of the International Conference on Computer Aided Design,* pp. 470-473, Nov. 1989.

[8] J.P. Roth, "Diagnosis of Automata Failures: a Calculus and a Method," *IBM Journal of Research and Development,* vol. 10, no. 4, pp. 278-291, July 1966.

[9] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Transactions on Computers,* vol. c-30, no. 3, pp. 215-222, March 1981.

[10] H. Fujiwara, and T. Shimono, "On the Acceleration of Test Generation Algorithms," *IEEE Transactions on Computers,* vol. c-32, no. 12, pp. 1137-1144, Dec. 1983.

[11] T. Kirkland and M.R. Mercer, "A Topological Search Algorihtm for ATPG," *Proceedings of the 24th Design Automation Conference,* pp. 502-508, June 1987.

[12] M.H. Schulz, and E. Auth, "Advanced Automatic Test Pattern Generation and Redundancy Identification Techniques," *Proceedings of the 18th Symposium on Fault Tolerant Computing,* pp. 30-35, June 1988.

감사의 글

이 연구는 1991년도 교육부 학술 연구 조성비의 지원을 받았으며 이에 감사를 드립니다. 또한 본 논문을 심사해 주시고 조언을 해주신 심사 위원들께도 감사드립니다.

著 者 紹 介

閔 炯 福(正會員)

1958年 2月 22日生. 1976年～1980年 서울대학교 전자공학과 졸업(공학사). 1980年～1982年 한국과학기술원 전기 및 전자공학과(공학석사). 1982年～1985年 금성통신(주) 연구소 주임연구원 1985年～1986年 미국 Columbia대학교 연구원. 1986年～1990年 The University of Texas at Austin 졸업(Ph. D.). 1991年～현재 성균관대학교 전기공학과 조교수