

論文 90-27-3-18

VLSI 논리설계 최적화를 위한 Redundancy 조사 가속화에 관한 연구

(On the Acceleration of Redundancy Identification
for VLSI Logic Optimization)

李 省 奉*, 鄭 正 和*

(Seong Bong Lee and Jong Wha Chong)

要 約

본 논문에서 게이트레벨 회로의 논리 최적화를 위한 논리적 redundancy 조사를 가속화하는 새로운 방법을 제안한다.

게이트레벨 회로의 redundancy 조사문제는 테스트패턴 생성문제와 마찬가지로 유한상태 탐색문제로서, 그 실행시간이 탐색의 크기에 의존한다. 본 논문에서는 효율적인 탐색을 위해, '동적 head line' 과 'mandatory 할당' 방법을 제안한다. 동적 head line은 redundancy 조사과정에서 동적으로 변경되어, 탐색에서의 backtracking 수를 감소시키며, mandatory 할당은 불필요한 할당을 피할 수 있어 탐색의 크기를 줄인다. 특히 이들 방법은 기존의 테스트패턴 생성문제에서 사용한 방법과는 달리, 회로 최적화에 따른 회로의 변경에 영향을 받지 않고 사용된다.

또한, 이들 방법을 기존의 redundancy 조사시스템에 실현하여, 그 유효성을 보인다.

Abstract

In this paper, new methods are proposed which speed up the logical redundancy identification for the gate-level logic optimization.

Redundancy indentification, as well as deterministic test pattern generation, can be viewed as a finite space search problem, of which execution time depends on the size of the search space. For the purpose of efficient search, we propose dynamic head line and mandatory assignment. Dynamic head lines are changed dynamically in the process of the redundancy identification. Mandatory assignment can avoid unnecessary assignment. They can reduce the search size efficiently. Especially they can be used even though the circuit is modified in the optimization procedure, that is different from the test pattern generation methods.

Some experimental results are presented indicating that the proposed methods are faster than existing methods.

I. 서 론

*正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)
接受日字: 1989年 6月 13日

게이트레벨 논리회로에서의 redundancy는 최종 칩 면적의 증대를 초래할 뿐만 아니라, 설계된 회로의 테스트에 많은 어려움을 주게 된다.^[1] 따라서, 설계의

최적화를 위해 이러한 redundancy를 효율적으로 찾아내는 방법이 요구된다.

Multi-레벨 회로에서의 redundancy 조사(Redundancy Identification: 이하 RI) 방법으로는, redundant한 회로 패턴을 패턴매칭에 의해 찾아내는 'local transformation'^[2,3]이 제안되었다. 그러나, 이 방법으로 찾아낼 수 있는 redundancy는 2-레벨의 회로패턴으로 제한되어 있다는 문제점이 있다. 일반적인 RI 방법으로는 테스트패턴생성(Automatic Test Pattern Generation: 이하 ATPG) 방법^[4,5]을 이용하는 방법^[7]과 이와 유사한 방법^[1]이 제안되었으나, 이 방법은 회로전체를 고려해야 하기 때문에 실행시간이 local transformation에 비해 대단히 크다. 따라서, 이를 효율적으로 줄이는 방법이 요구된다.

그런데, 게이트레벨 회로의 RI 문제는 ATPG 문제와 마찬가지로 유한상태(finite-state) 탐색문제^[6]로 볼 수 있으므로, 실행시간의 감소를 위해 ATPG 문제에서의 방법^[5]을 사용할 수 있다. 그러나, ATPG 문제에서는 주어진 회로가 ATPG 과정에서 변하지 않지만, RI 문제에서는 회로최적화에 따라 회로가 변경되기 때문에 ATPG 방법을 직접 적용할 수 없다.

본 논문에서는, 회로최적화에 따른 회로의 변경에 영향을 받지않고, RI 실행시간을 단축할 수 있는 새로운 방법 - '동적(dynamic) head line'과 'mandatory 할당(assignment)'을 제안한다.

II. Redundancy 조사

1. RI 문제와 ATPG 문제^[7]

회로내의 어떤 신호선을 0 (또는 1)로 대치하여도, 전체회로의 출력이 care 조건의 모든 입력에 대해 변경이전과 동일한 값을 갖는다면, 그 신호선은 0-redundant (또는 1-redundant)하다고 한다. 회로내에서 이러한 redundant한 신호선을 찾는 RI 문제와 기존의 ATPG 문제와의 관계를 다음 정리로 요약할 수 있다.

[정리 1] 회로내의 어떤 신호선이 0 (1) -redundant 하면, 그 신호선에 대한 stuck-at-0 (1) fault에 대한 테스트 패턴이 존재하지 않는다. 또한 그 역도 성립한다. (즉 RI 문제는 ATPG 문제와 동일하다)

그런데, ATPG 문제가 유한상태 탐색 문제^[6]이기 때문에, RI 문제 또한 유한 상태 탐색문제로 볼 수 있다. 따라서, 실행시간의 단축을 위해서는 탐색에 있어서의 backtracking 수를 줄이는 것이 매우 중요하다. 그러나, ATPG 문제에서는 주어진 회로가 ATPG 과정에서 불변이지만, RI 문제에서는 회로 최적

화에 따라 회로가 변경되기 때문에, RI 실행시간 단축 방법은 회로의 변화에 무관하게 사용할 수 있어야 한다.

2. ATPG를 이용한 RI 방법

ATPG를 이용한 일반적인 RI 방법은 사용하는 ATPG 알고리즘에 따라 다르다. 본 논문에서의 RI 방법은 PODEM 알고리즘^[8]을 변경하여 사용하기 때문에, 이 알고리즘을 기초로 하여 기술하면 다음과 같은 procedure로 나타낼 수 있다.

```

Redundancy 조사(조사 신호선, 조사할 redundancy)
{
  A : /*backtrace 과정*/
  할당할 주입력과 값(0/1)을 선택;
  B : /*implication 과정*/
  할당된 값에 따른 회로변화를 추적;
  /* redundancy 조사 */
  IF (주출력의 값이 DV/DB 인가)
    RETURN (non-redundant);
  IF (조사를 계속할 필요가 있는가)
    GO TO A;
  /* backtracking 과정 */
  IF (할당된 PI에 대해 미시도한 값이 있는가)
    미시도한(untried) 값을 할당;
    GO TO B;
}
RETURN
}

```

이러한 RI 과정은 decision tree로 나타낼 수 있다. 예를 들어, 그림 1의 회로에서 신호선 s5의 1-redundant 여부를 판정한다고 하자. 이를 decision tree로 나타내면 그림 2와 같다. 여기서 점선으로 나타낸 것은 backtracking 과정을 나타내고, 직선은 backtrace 과정과 implication 과정을 의미한다.

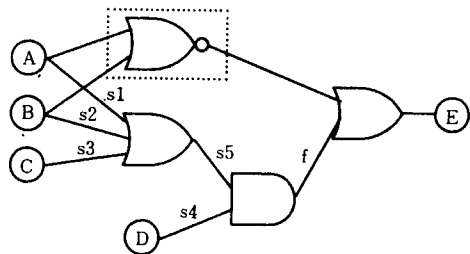


그림 1. 예제 회로 (1)
Fig. 1. Illustrative circuit (1).

만약 조사한 신호선이 redundant 한 경우에는 그림 2에서 처럼 decision tree의 root에까지 backtracking이 발생하며, non-redundant 한 경우에는 root가 아닌 중간 node에서 중단된다.

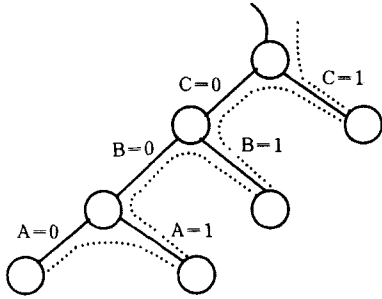


그림 2. Decision tree의 예
Fig. 2. Example of decision tree.

Ⅲ. Redundancy 조사 가속화방법

1. 동적 head line

기존 ATPG 시스템 FAN^[5]에서 제안한 'head line'은 tree 형태로 구성된 부분회로(subcircuit)의 root 신호선을 의미한다. 이와 같은 tree 형태의 회로를 fanout-free 회로라 한다. 이러한 회로는 다음과 같은 특징이 있다.

[정리 2] Fanout-free 부분회로에서는 root 신호선을 1/0값으로 할당하는 leaf 신호선의 0/1패턴이 반드시 존재하며, 이 부분회로내의 모든 신호선은 반드시 root 신호선을 통해 회로의 다른 부분에 영향을 준다.

정리 2는 RI의 backtrace 과정을 주입력 신호선이 아닌 head line에서 중지하여도 조사신호선의 redundant 여부를 판정할 수 있다는 것을 의미한다.

Head line은 RI 알고리즘의 전처리(preprocessing) 단계에서 구성되어, 실제 RI 과정에서는 변하지 않는다. 그러나, 그림 3의 예제 회로에서 head line s2 = 0 할당되었다고 하자. 이 때 implication 과정에 의해, 신호선 s2=s5=s10=s17=0, s16=1로 할당된다. Fanout branch 신호선 s9와 s13은 각각 신호선 s5=0와 s10=0의 영향 때문에 신호선 s19=1(0)로 할당되는데 어떠한 영향도 줄 수 없다. 따라서 이미 값이 할당된 신호선 s2, s5, s10, s16, s17와 fanout branch 신호선 s9, s13을 제거한 점선 부분의 회로는 fanout-free 회로가 된다. 이 때, 신호선 s19는

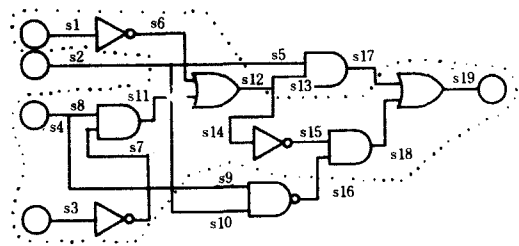


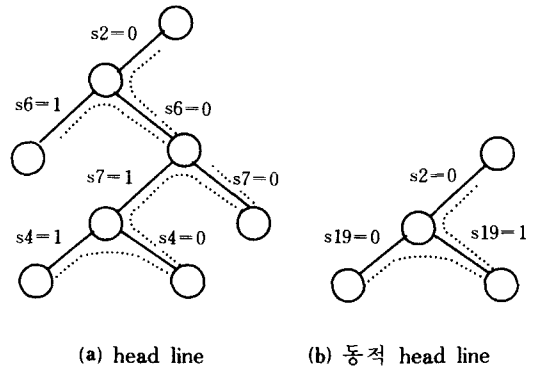
그림 3. 예제 회로(2)
Fig. 3. Illustrative circuit (2).

이 부분회로의 root 신호선임을 알 수 있다. 즉 신호선 s19는 신호선 s2=0인 경우에는 head line이 될 수 있음을 알 수 있다.

[정의 1] 한 게이트 G의 입력신호선 s에 의해 게이트의 출력값이 결정되면, 이 입력신호선 s는 다른 입력신호선을 'block' 한다고 한다.

[정의 2] RI 과정에서, 이미 값이 할당된 신호선과 block 된 신호선을 제거한 회로에서 생성되는 head line을 '동적 head line'이라고 한다.

Backtrace 과정을 head line에서 중지시킴으로서 backtracking 수와 backtracking 사이의 실행시간을 감소시킬 수 있는 것과 마찬가지로, 위에서 제안한 동적 head line에서 backtrace 과정을 중지시킴으로서, backtracking 수와 backtracking 사이의 실행시간을 감소시킬 수 있다. 그림 4는 그림 3의 회로에 대한 decision tree를 나타낸 것으로, (a)는 기존의 head line을 이용한 경우이며, (b)는 동적 head line을 이용한 경우이다.



(a) head line (b) 동적 head line

그림 4. 그림 3에 대한 decision tree
Fig. 4. Decision tree of fig. 3.

동적 head line은 RI의 implication 과정에서 생성되며, backtracking에서 다시 제거된다. 표 1은 2 입력 NAND 게이트에 대한 확장된 implication 표를 나타낸 것으로, 동적 head line을 처리할 수 있다. 이 표에서 H는 동적 head line을 나타낸다. 또한 어떤 신호선은 B값을 가질 수 있는데, 이는 block된 신호선을 나타낸다.

표 1. 확장된 implication (표 2 입력 NAND 게이트)

Table 1. Extended implication table (2 input NAND gate).

	X	0	1	DV	DB	H
X	X	1	X	X	X	X
0	1	1	0	1	1	1
1	X	1	1	DB	DV	H
DV	X	1	DB	DB	1	X(DB)
DB	X	1	DV	1	DV	X(DV)
H	X	1	H	X(DB)	X(DV)	H

또한, 이 표에서 () 안의 값은 D-frontier^[4]에 속하는 게이트의 수가 1일때 사용된다. 그림 5는 그림 3에서 head line s7=DV 일 때, head line s2=0으로 할당되어 발생하는 implication 과정을 나타낸 것으로 () 안 값을 사용한 예를 보인 것이다. 이때 fanout branch 신호선 s8은 신호선 s9=B로 할당됨에 따라 H값으로 할당되는 것을 알 수 있다. 이와같은 특수한 경우의 implication 과정 역시, backtracking 수를 줄여, redundancy 여부를 판정하는 시간을 줄이는데 도움을 준다. 또한, B(block) 값의 도입으로 기존의 PODEM 알고리즘에서의 X-path 조사 과정^[6]을 생략할 수 있다는 장점도 있다.

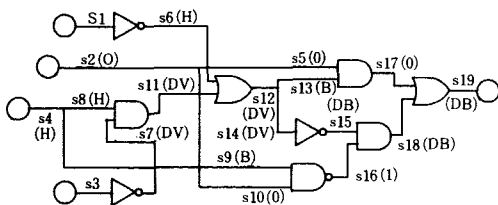


그림 5. 그림 3에 대한 implication 과정 (신호선 s7=DV, s2=0)
Fig. 5. Implication of fig. 3 (signal s7=DV, s2=0)

3. Mandatory 할당

만약 RI 과정에서 backtracking이 발생하면, 미시도한(untried) 값을 다시 할당하게 된다. 그러나, 이러한 미시도한 값을 할당할 경우에는, 다시 backtracking이 발생하는 경우가 많다. 특히 한 신호선이 redundant한 경우에는 decision tree의 root에까지 backtracking이 발생하기 때문에, backtracking을 발생하는 할당을 회피하는 것이 backtracking 수를 줄일 수 있으며, 또한 불필요한 implication 과정을 피할 수 있다.

예를들어 그림 1의 회로에서 신호선 s5가 1-redundant함을 조사할 경우, 조사 과정을 decision tree로 나타내면 그림 2와 같다. 주입력 C=B=A=0로 할당하면 backtracking이 발생한다. 이 때, 미시도한 값 1을 A에 할당하게 된다. 이 경우는 신호선 s5=1로 할당하기 때문에 반드시 backtracking이 발생하게 된다. 마찬가지로 B=C=1로 할당하는 경우도 backtracking이 발생한다. 이와 같이 미시도한 값을 할당하면 반드시 backtracking이 발생한 경우, 미시도한 값에 대한 고려를 강제적으로 제외시키는 것을 'mandatory 할당'라 한다. 즉 mandatory 할당에 의해 할당된 값에 대해서는, 미시도한 값에 대한 할당과 implication 과정을 생략하고, decision tree의 상위 node로 직접 backtracking 한다. 그림 6은 그림 1의 회로에 대해 mandatory 할당을 사용한 경우에 대한 decision tree를 나타낸 것이다.

Mandatory 할당은 RI의 backtrace 과정에서 수행된다. Mandatory 할당이 발생하는 경우는, 첫째, 미시도한 값이 할당되는 경우, 조사신호선에 조사할 redundancy(0/1)와 동일한 값을 갖게 되는 경우에 발생하며, 둘째, D-frontier에 존재하는 게이트수가 1일때 이 게이트의 입력신호선에 어떤 값을 할당하는

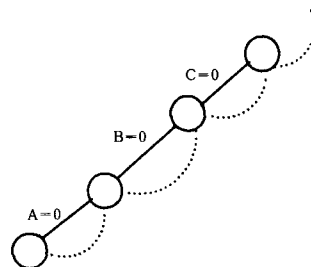


그림 6. 그림 1에 대한 mandatory assignment for fig. 1

Fig. 6. Effect of mandatory assignment for fig. 1.

경우에 발생한다.

RI의 backtrace 과정에서 mandatory 할당을 처리하기 위해, mandatory flag을 사용한다. 이 flag는 위의 2 경우에 ON 되어, backtrace 과정에서 ON/OFF 된다. Backtrace 과정의 완료시 이 flag가 ON으로 유지되면, 이때의 할당은 mandatory 할당이 된다. Backtrace 과정에서 첫째, 현재 고려 중인 게이트가 NAND/OR(NOR/AND)이고, backtrace시의 논리값이 0(1)인 경우, 둘째, 이 게이트의 X값을 갖는 입력 신호선이 1인 경우에 이 flag는 ON 상태를 유지한다.

IV. 실험 및 고찰

본 논문에서 제안한 방법의 유효성을 입증하기 위해 여러 예제회로에 대하여 프로그램 실험을 행하였다. 본 방법은 IBM-AT(DOS)상에서 C언어로 실험하였다.

표 2는 실험결과를 보인 것이다. 회로의 size는 게이트수(신호선수)의 형태로 나타냈다. Level은 회로의 최대 level수를 나타내고, R신호선#은 redundant하다고 판정된 신호선의 수를 나타낸다. 표 2에서 경우 1은 본 논문에서 제안하는 방법을 사용한 경우이며, 경우 2는 기존의 PODEM 알고리즘에 기존의 head line 방법을 추가시킨 경우로서, 이는 동적 head line의 효과를 정확히 실험하기 위해서이다.

표 2. 실험 결과
Table 2. Exprimnt result.

회로	size	level	R 신호선#	경우1	경우 2
ADR	95(158)	14	1	10.5	12.9
CNT8	129(221)	8	1	1.9	2.2
DEMO	135(238)	19	0	120.9	190.1
SCH	108(183)	19	7	29.3	40.0
CCT	201(305)	29	40	981.5	1778.8

게이트수(신호선수) 실행시간 sec.

표 2의 실험 결과를 고찰하면, 본 논문에서 제안하는 방법(경우 1)이 실행시간의 단축을 기할 수 있다는 것을 알 수 있다. 특히 본 방법은 회로의 level수가 크면 클수록 그 단축효과가 커짐을 알 수 있다. 본 논문에서 제안하는 방법은 기존의 ATPG 알고리즘 FAN^[5] 등에서도 사용할 수 있기 때문에, 이들 알고리즘에 본 방법을 추가시키면, 이들 알고리즘의 실행속도도 향상시킬 수 있을 것으로 판단된다.

본 논문에서의 RI 방법은 논리회로 최적화를 목적으로 하고 있다. ATPG 문제에서는 입력회로가 ATPG 과정에서 불변이지만, 논리회로 최적화에서의 회로는 redundant한 신호선의 발생에 따라 계속 변경되기 때문에, FAN 등에서 제안하는 unique sensitization 등의 방법은 직접 사용할 수 없다. 따라서, ATPG에서 제안하는 여러 방법의 효율적인 관리에 대한 연구가 계속 되어야 할 것이다.

V. 결 론

본 논문에서는 게이트레벨 회로에서의 논리적 redundancy 조사를 가속화하는 새로운 방법을 제안했다.

게이트레벨 회로의 redundancy 조사문제는 유한상태 탐색문제로서, 그 실행시간이 탐색의 크기에 의존한다. 본 논문에서는 효율적인 탐색을 위해, 동적 head line과 mandatory 할당방법을 제안했다. 특히 이들 방법은 기존의 ATPG 문제에서 사용한 방법과는 달리, 회로최적화에 따른 회로의 변경에 영향을 받지않고 사용된다. 또한, 본 논문에서 제안하는 방법을 PODEM 알고리즘을 이용한 redundancy 조사 시스템에 실현하여, 제안하는 방법의 유효성을 보였다. 앞으로의 연구과제로는, 기존의 ATPG에서 사용되는 방법의 효율적인 관리에 대한 연구가 계속되어야 할 것이다.

參 考 文 獻

- [1] Daniel Brand, "Redundancy and don't cares in logic synthesis," *Trans. on Comp.*, vol. C-32, no. 10, Oct., 1983, pp. 947-952.
- [2] J.A. Darringer, et al., "LSS: A system for production logic synthesis," *IBM J. Res. Develop.* vol. 28, no. 5, 1984, pp. 537-544.
- [3] W.W. Cohen, et al., "A rule-based expert system for optimizing combinational logic," *IEEE Design & Test of Comp.*, Aug., 1985, pp. 22-32.
- [4] J.P. Roth, et al., "Programmed algorithms to compute tests to detect and distinguish between failures in logic circuits," *IEEE Trans. on Electronic Computers*, vol. EC-16, no. 5, Oct., 1967, pp. 71-83.
- [5] Fujiwara, H., and T. Shimono, "On the acceleration of test generation algorithms," *IEEE Trans. Comput.*, vol. C-32, pp. 1137-1144.

[6] P. Goel, "An implicit enumeration algorithm to generate tests for combinational logic circuits," *IEEE Trans. on Comp.*, vol. C-30, March, 1981, pp. 215-222.

[7] 이성봉, 정정화, "VLSI 테스트 이론을 이용한 Global Redundancy 조사", 대한 전자공학회 논문지, 제26권 제 4 호, pp.138-144.

著 者 紹 介



李 省 奉 (正會員)

1962年 1月 7日生. 1984年 한양대학교 전자공학과 졸업. 1986年 2月 한양대학교 대학원 전자공학과 졸업. 공학석사학위 취득. 1986年 3月~현재 한양대학교 대학원 전자공학과 박사과정 재학

중. 주관심분야는 VLSI CAD 특히 논리설계 및 검증 등임.



鄭 正 和 (正會員)

1950年 3月 10日生. 1975年 한양대학교 전자공학과 졸업. 1981年 3月 일본 와세다대학 박사학위 취득. 일본 NEC(주) 중앙연구소 연구원, KIET 위촉 연구원 University of California, Berkeley

교환교수. 1981年~현재 한양대학교 전자공학과 교수. 주관심분야는 VLSI CAD 특히 Layout 및 HDL 등임.