# 스위치 레벨 결함 모델을 사용한 결함시뮬레이터 구현

연 윤 모† · 민 형 복††


## 요 약

VLSI 회로에서 스위치 레벨 결함 모델은 stuck-at 결함만 사용하는데 한계가 있다. 따라서 본 연구는 스위치 레벨 결함 모델인 트랜지스터 stuck-open과 stuck-closed 결함을 다룰 수 있는 결함 시뮬레이터를 구현한다. 스위치 레벨 회로는 이론적으로 신호 흐름이 양방향으로 전달되지만 실제로 대부분의 신호 흐름은 약 95%정도가 단방향으로 설정되어 평가되는 것으로 나타내고 있다. 본 연구에서는 스위치 레벨 회로를 단방향 그래프 모델로 변환시켜 해석한다. 스위치 레벨 회로는 EDIF 컴파일러에 의해 입력되고 두개의 단방향으로 재구성된 자료 구조를 만든다. 스위치 레벨 회로는 신호 흐름 경로가 도입되는 지배적 경로 기법이 제시된다. 지배적 경로는 경로를 판단하여 최종 출력 상태값을 결정하는 논리 시뮬레이션을 수행한다. 스위치 레벨 결함 시뮬레이션은 노드들로 연결되는 경로 상에 임의 트랜지스터의 stuck-open, stuck-closed 결함을 주입시키고, 트랜지스터 저항값을 적용한 노드 세기의 계산에 의한 지배적 경로를 평가한다. 이 때 최초 입력은 two pattern vector를 인가하여 정상회로의 최종 출력 상태값과 결함회로의 출력 상태값을 비교하여 결함 검색하며, 그 결함 검색의 정확성을 보인다.


# An Implementation of the Fault Simulator for Switch Level Faults

Yun-Mo Yeon † · Hyoung-Bok Min ††


## ABSTRACT

This paper describes an implementation of fault simulator that can handle switch level fault models such as transistor stuck-open and stuck-closed faults as well as stuck-at faults. It overcomes the limitations when only stuck-at faults are used in VLSI circuits. Signal flow of a transistor switch is bidirectional in its nature, but most of signal flows in a switch level circuits, about 95%, are in one direction. This fault simulator focuses on the way which changes a switch level circuit into a graph model with two directed edges. Two paths from Vdd to ground and from ground to Vdd are the two directions. Logic simulation is performed along dominant signal flows. The switch level fault simulation estimates the dominant path by injecting switch-level faults, and two pattern vectors are used for fault simulation. Experimental results are shown to demonstrate correctness of the fault simulator.


## 1. 서 론

반도체의 급속한 발전으로 VLSI 칩은 회로가 커지고 복잡해지고 있다. 방대하고 복잡한 VLSI 칩은 설계하는 비용과 시간이 많이 소요되지만 개발 툴에 의해 해결되고 있다. 또한 생산하기 전에 발생 가능한 결함을 찾아내는 것이 중요한 과제이다. 결함 검색을 위한 도구로서 여러 기법의 결함 시뮬레이터 툴이 개발되어 왔다. 이와 같은 결함 시뮬레이터는 테스트

† 정 회 원:청주전문대학 전자계산과 부교수
†† 정 회 원:성균관대학교 전기공학과 부교수
논문접수:1996년 2월 10일, 심사완료:1996년 12월 28일

**김 용 길**

1969년 7월 26일생
1995년 2월 아주대학교 전자공학과 졸업(학사)
1997년 2월 아주대학교 전기전자공학부 졸업(석사)
1997년~현재 대우 고등기술연구원 연구원
주관심분야: VLSI 설계, 멀티미디어 시스템

**임 강 빈**

1969년 4월 27일생
1992년 2월 아주대학교 전자공학과 졸업(학사)
1994년 2월 아주대학교 전자공학과 졸업(석사)
1994년 3월~현재 아주대학교 전기전자공학부 박사과정
주관심분야: 컴퓨터 구조, VLSI 설계, 영상 부호화, 멀티미디어 시스템 설계 등

**김 용 득**

1946년 1월 30일생
1971년 연세대학교 전자공학과 졸업
1973년 연세대학교 대학원(공학석사)
1978년 연세대학교 대학원(공학박사)
1973년~1974년 불란서 ESE 연구원
1978년~1980년 미국 Stanford대학교 연구교수 재직
1978년~현재 아주대학교 전기전자공학부 교수
주관심분야: 디지틀 시스템 하드웨어 응용, ISDN 응용 및 접속방안

**정 기 현**

1958년 10월 21일생
1990년 퍼듀대학교(미) 공학박사
1991년~1992년 현대반도체연구소 연구원
1992년~현재 아주대학교 전기전자공학부 교수
주관심분야: 컴퓨터 구조, VLSI 설계, 영상 서비스, 멀티미디어 시스템 등

패턴의 생성 및 디지털 회로의 faulty behavior를 알아보기 위하여 자주 사용된다. 결함 시뮬레이션은 지난 20년간 많은 연구가 축적되었으며 테스팅 관련 상용 툴 중에서도 가장 보편화된 것이라 할 수 있다. 그러나 상용 결함 시뮬레이터는 여전히 stuck-at 결함을 근간으로 하고 있으나 CMOS 회로 중심인 현재의 VLSI 설계 기반에서 stuck-at 결함만은 한계가 있다.

스위치 레벨 결함 시뮬레이션에 대한 여러 기법을 살펴보면 R. L. Wadsack과 S. M. Reddy, V. D. Agrawal, S. K. Jain은 고전적인 stuck-at 결함과 비고전적인 stuck-open, stuck-on을 고려한 결함 시뮬레이션(fault simulation)을 소개한다[1, 2]. M. D. Schuster와 R. E. Bryant는 concurrent fault simulation기법으로 MOS 회로를 그래프 모델로 각 노드들의 stuck-at 결함, stuck-open, stuck-closed 및 저항적인 open, shorts등의 결함 모델을 고려하고 있다[3]. 또한, J. P. Shen, W. Maly와 F. J. Ferguson의 MOS 회로에 대한 결함 해석에서는 line stuck-at 결함이 28%, stuck-open, stuck-closed 결함이 15%, floating line 결함이 21%, bridging 결함이 30% 존재한다고 말하고 있다[4]. Chi-yuan Lo, H. N. Nham과 A. K. Bose의 MOTIS simulation은 fault modelling과 fault collapsing을 수행하는 알고리즘으로 fault list 조정과 primitive gate evaluation의 견지에서 논의되었다[5].

여러 기법의 제시에서 살펴본 것처럼 고전적인 stuck-at 결함의 한계를 극복하기 위해 스위치 레벨 결함 모델이 제안되고 이를 위한 논리 시뮬레이션, 결함 시뮬레이션이 연구되었다. 이러한 연구 결과에 힘입어 스위치 레벨 결함에 대하여 많은 것을 알게 되었으나 그 결과가 상용 도구에 잘 적용되지 않는 것은 스위치 레벨 툴이 게이트 레벨에서 보다 훨씬 많은 회로 모델을 취급하는 데다가 알고리즘 또한 복잡하므로 CPU time이 엄청나게 많이 걸린다는 점과 시장성 문제 때문이라고 여겨진다.

스위치 레벨 논리 시뮬레이션은 그래프 이론을 기초한 Bryant와 Adler의 연구와 Hayes의 lattice이론을 기초한 방법등 다양한 모델링 접근을 시도하고 있다[6, 7, 8, 9]. 그래프 이론을 기초한 연구에서 Bryant는 global 알고리즘을 이용하고, Dan Alder는 distributed 알고리즘을 이용하여 모델링 기법을 제안하였다. VLSI 회로에서 트랜지스터는 이론적으로 양방향 신호 흐름을 가진다. 그러나 대부분의 신호 흐름은 단방향으로 평가되며 약 5% 정도만이 양방향 신호 흐름을 갖는다[10].

본 연구에서는 단방향 및 양방향 CMOS 회로를 처리 대상으로 한다. 또한 제시된 그래프 이론과 새롭게 제시하는 CMOS 회로 네트워크의 대칭성을 기초하여[11, 13, 14] 스위치 레벨 논리 시뮬레이션을 구현하고, 스위치 레벨 결함 시뮬레이터의 전 처리기로 사용한다. 대칭적 CMOS 회로는 최대 흐름 알고리즘을 변형한 지배적 경로(dominant path) 알고리즘으로 최종 출력의 정상 상태를 계산한다. 또한 CMOS 회로의 입력은 본 연구에서 구현한 EDIF(Electronic Design Interchange Format) 컴파일러로 받아들여 데이터 구조를 구성하고, 이 데이터 구조의 방향성을 양방향에서 단방향으로 재구성한 연결리스트를 사용한 논리 시뮬레이션을 구현한다. 이 논리 시뮬레이션을 토대로 비고전적인 stuck-open과 stuck-closed 결함 모델을 고려한 스위치 레벨 결함 시뮬레이터를 구현하고 그에 따른 실행 결과를 검토한다.

본 연구의 구성은 2장에서 CMOS 회로 네트워크 모델 및 회로 입력과 논리 시뮬레이션에 대해 서술하였고, 3장에서는 스위치 레벨 회로 결함 모델과 결함 시뮬레이션의 구현을 제시하였으며, 4장은 제시된 스위치 레벨 시뮬레이션을 수행하였고 결과 검토를 기술하였다.

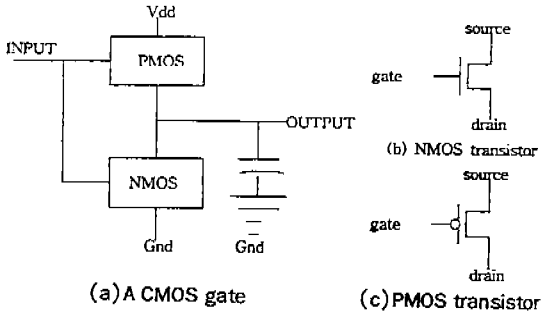
## 2. 스위치 레벨 회로 네트워크

### 2.1 CMOS회로 네트워크 모델

CMOS 회로의 트랜지스터는 PMOS와 NMOS쌍으로 이루어져 있으며 PMOS는 음 신호 동작을 하고, NMOS는 양 신호 동작을 하는 서로 보수의 특성으로 구성된다. 일반적인 CMOS 게이트는 대칭적인 네트워크로 이루어졌다 블록 다이어그램은 (그림 1(a))과 같고, 이 CMOS 회로의 블록 다이어그램에 대한 기본적인 트랜지스터의 표현은 (그림 1(b), (c))에 나타내었다.

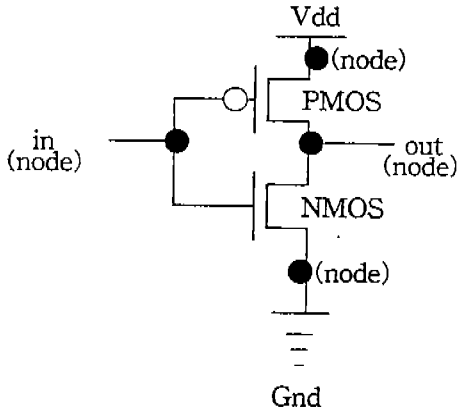
(그림 1)의 기본 개념을 토대로 기본 회로인 CMOS 인버터 회로의 다이어그램을 (그림 2(a))에 나타내었고, 그 회로를 그래프 모델로 (그림 2(b))에 나타내었다.

일반적으로 그래프 G(V, E)는 정점(vertices) V의

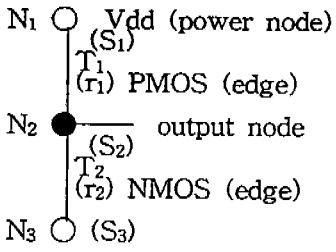



(a) A CMOS gate (c) PMOS transistor

(그림 1) CMOS 게이트와 NMOS, PMOS 트랜지스터
(Fig. 1) A CMOS gate and NMOS, PMOS transistor




(a) Switch level model of CMOS inverter




(b) A graph model for CMOS inverter

(그림 2) CMOS 인버터 회로의 다이어그램과 그래프 모델
(Fig. 2) A CMOS inverter diagram and graphic model

집합과 에지(edges)의 집합으로 구성하는 연결 함수로 정의된다. 따라서 (그림 2(a))의 CMOS 인버터 회로를 그래프 구조로 모델화하면 각 트랜지스터의 소오스와 드레인이 연결되는 점을 노드(정점)로 하고 각 트랜지스터는 에지(연결선)로 한다. 여기서 Vdd, Gnd는 소오스 노드로 지정한다. (그림 2(b))는 위의 CMOS 인버터 회로를 모델링 방법으로 구성한 그래프 구조이다.

(그림 2(b))에서 트랜지스터의 신호 흐름 방향은 $N_1$과 $N_2$, $N_2$와 $N_1$의 노드 세기에 의해 결정된다. 트랜지스터의 게이트는 그래프 구조에서 에지의 연결됨(turn-ON)과 연결되지 않음(turn-OFF) 상태를 결정하고 신호 흐름의 세기를 감쇄시키는 저항 요소로 작용한다.

스위치 레벨 회로의 노드 세기(strength)는 크기 순서에 의해 경로가 결정된다. 각 노드 세기($S_1$, $S_2$, …, $S_n$)의 집합이 $S_1 \ll S_2 \ll S_3 \ll \cdots \ll S_n$으로 정렬된 순서에 의해 지배적 경로를 갖게 된다. 또한 신호 흐름 세기는 스위치 레벨에서 트랜지스터의 소오스와 드레인을 통과할 때 저항특성에 의해 감쇄작용을 한다. 따라서 각 트랜지스터의 저항 성분에 의해 다음 노드 세기는 감소하게 된다. 즉, 각 트랜지스터 저항세기는 ($r_1$, $r_2$, …, $r_m$)의 집합으로 나타내며 그 세기 $r_i$는 power node의 세기 Spower보다 매우 작은 정수값으로써 식으로 나타내면 다음과 같다.

$$r_i \ll S_{power} \quad (1)$$

스위치 회로의 노드 세기 계산 형태는 다음과 같이 두 가지로 나누어 생각할 수 있다.

첫째로, 직렬연결에서는 저항특성의 합인 $\Sigma r_i$로 나타내며 목적노드 세기의 계산을 식으로 나타내면 $S_i = S_j - \Sigma r_i$가 된다.

둘째로, 병렬연결에서는 여러 개의 경로 중에서 저항특성인 $r_i$가 최소인 경로를 선택하게 되며 목적노드 세기는 $S_i = S_j - \mathrm{Min}(r_i)$ 식으로 계산될 수 있다. 여기서 $\mathrm{Min}(r_i)$에 속한 에지가 지배적 경로로 결정된다.

### 2.2 CMOS 회로 네트워크의 지배적 경로

2.1절에서 다루는 스위치 레벨 네트워크 모델은 네트워크의 평가를 위해서 공식적인 기법을 제시한다. 스위치 네트워크의 신호 흐름은 그래프 구조에서 각각 노드와 에지로 나타냈으며, 그 그래프는 원시 노드로부터 각 노드로 통하여 목적 노드까지 향하는 최대 흐름 경로를 계산하며 그 경로를 지배적 경로라 한다. Dan Adler는 Ford의 최단 경로 기법 알고리즘을

적용하였다. 그러나 본 논문에서는 Ford의 기법을 개량한 최대 흐름 기법을 적용하였다. 이 기법은 스위치 레벨 회로에 적용하여 네트워크의 에지 용량을 부여한 지배적 경로를 설정하였다.

(그림 3)은 본 논문에 적용한 지배적 경로를 계산하기 위한 변형된 최대 흐름 경로 알고리즘을 나타내고 있다. CMOS 회로는 두 개의 분리된 PMOS, NMOS 네트워크로 구성되어 있다. 즉, 두 개의 네트워크는 그룹화 네트워크 형태를 갖고 있다. 여기서 라인 1의 D는 경로를 초기화하며, 라인 2의 n은 초기화로 소오스 노드인 Vdd를 설정한다. 라인 3은 소오스 노드에 연결된 다음 노드이다. 라인 4, 5는 모든 노드의 초기화로 LIST 테이블에 에지의 가중치와 노드 세기를 기록한다. 라인 6부터 라인 15까지는 지배적 경로인 소오스 노드로부터 목적지 노드까지 전체 노드에 대해 반복 계산을 진행하는 루틴이다. 라인 8, 9는 다음 노드가 그룹화된 블록일 때, 최소 가중치 r인 에지를 찾아 R에 저장하고, 라인 10은 그룹 블록이 아닐 때, 가중치 r인 에지를 R에 저장한다. 각 단계에서 LISTS(m)은 연결되는 에지 가중치에 의해 계산된 노드의 세기를 기록한다. 그리고 라인 12에서는 지배적 경로 D에 선택된 노드 LISTS(m)을 연결한다. 소오스 노드의 가중치는 초기값을 r(n)≡0으로 한다. 또한 다음절에서 다룰 스위치 결함 시뮬레이션에서 트랜지스터 stuck-open 결함 모델은 에지 가중치를 r(m)≡99인 아주 큰 값으로 설정하고, 트랜지스터 stuck-closed 결함 모델은 r(m)≡1인 아주 적은 값으로 설정하여 적용하였다.

```
# The maximum flow algorithm for dominant path.
 1)  D ← ∅
 2)  n ← source node
 3)  m ← next node connected source node
 4)  for every node k in the path from source to sink do
 5)          LISTR(k) ← r(k), LISTS(k) ← s(k)
 6)  while m ≠ sink do
 7)      begin
 8)          if all arcs to the next layer are block, then
 9)              R ← minimum value of r(m) in the block
10)          else R ← r(m)
11)          LISTS(m) ← ( s(n→m) - R )
12)          D ← D ∪ LISTS(m)
13)          n ← m
14)          m ← next node connected m
15)      end
```

(그림 3) 지배적 경로 계산을 위한 최대 흐름 알고리즘
(Fig. 3) The maximum flow algorithm for dominant path

2.3 회로의 입력

다수 회사에서 설계한 VLSI 회로는 정보들 간에 호환성이 요구되며 호환성 해결을 위해서 표준형식을 지정하였다. 설계자료에 대한 교환 형식인 EDIF가 이와 같은 문제점을 해결하고 있다. EDIF는 전자회로 설계를 서로 간의 호환성을 유지하기 위해서 자료의 정보를 교환하는데 사용하는 표준형식으로써 계층적 구조를 갖는다. 즉, EDIF 파일은 한 개 또는 그 이상의 library가 있으며, library는 필요에 따라 정의되는 technology와 cell에 관한 정보를 갖고 있다.

EDIF 컴파일러는 EDIF 형태 파일을 해석하기 위해 EDIF 파서(parser)를 이용하여 구현하였다. 파서는 GNU의 bison을 이용하여 파서 생성기를 만든다. EDIF 문법을 기술한 edif. y는 bison에 의해 소오스 코드와 헤드 파일을 생성하고, getEDIF()는 각 스위치 노드에 대한 정보를 edif.gaterc에서 읽고, 스위치 회로에 대한 기술을 circuit.edif 입력파일에서 읽어 swnode_list[]를 생성한다. 스위치 레벨의 각 노드 및 에지의 테이블로 구성된 swnode_list[]는 필요한 정보를 갖는다. 또한 파서는 계층적 구조의 EDIF 정보들이 트리 형태로 표현되는데 이 구조의 최하위 cell들을 직접 연결하는 경로를 계산할 수 있다. EDIF 컴파일러에 의해 평평하게 만들어진 회로는 SWNODE 구조체에 저장된다.

결국 EDIF 형식으로 기술된 계층적 구조의 CMOS 회로가 평면적 구조로 변경되고 이것을 이용하여 논리 시뮬레이션 및 결함 시뮬레이션을 수행하게 된다.

또한 회로의 입력에 필요한 CMOS 회로의 자료 구조는 각 노드를 SWNODE로 나타내며, SWNODE의 자

```
typedef
struct swnodedef {
        char *name;
        int  swnode_num, swnodetype;
        int  num_fis, num_fos, num_fios;
        int  num_ffis, num_ffos;
        struct swnode_ogdef *fi_list, *fio_list;
        struct swnodedef **fo_list, **FIO_list, **fwd_list, **bwd_list;
        int  primarygatelevel, fwdgatelevel, bwdgatelevel;
        int  edge_flag, prior_state, current_state, good_state;
        int  real_R, resct_R, store_R,  s_d, s_0, s_1;
        int  tr_open_flag, tr_closed_flag;
} SWNODE, *SWNODE_PTR;
```

(그림 4) 스위치 노드의 자료 구조 형식
(Fig. 4) The data structure format for node of switch

료구조는 (그림 4)과 같다. (그림 4)는 입력단(fi_list)이나 입출력단(fio_list)에 원시 자료 구조를 갖도록 하였으며, 경로의 재구성을 위한 리스트로 fwd_list와 bwd_list의 자기 참조자료 구조를 갖도록 하였다.

### 2.4 CMOS회로 논리 시뮬레이션

평면적으로 구조화된 SWNODE swnode_list[]의 초기형태는 양방향으로 연결된 리스트이다. 단방향 리스트는 양방향 리스트를 이용하여 Vdd에서 Gnd방향으로, Gnd에서 Vdd방향으로 연결된 리스트로 재구성한 경로를 만들었다. C언어로 구현된 단방향 연결리스트 구조의 재구성 프로그램은 (그림 5)과 같다.

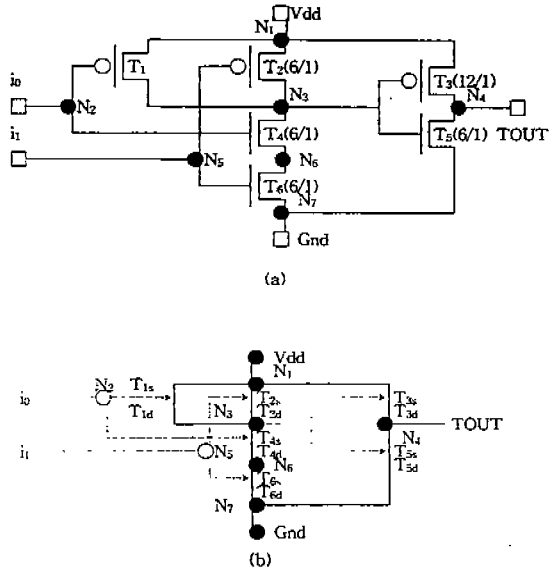
```
/* A restructure Program */
  SWNODE swnode_list[ ];
  for(swnode_number) {
     if(swnode_type == transistor) {
        forward_list[0]  = swnode_list[real].FIO_list[drain];
        backward_list[0] = swnode_list[real].FIO_list[source];
     }
     else {
        for(swnode_list[real].inout_pin_numbers) {
           next_list = swnode_list[real].FIO_list[inout];
           if(swnode_list[real] ==
                          (next_list->fio_list[source].swnode))
              forward_list[count1++] = next_list;
           else
              backward_list[count2++] = next_list;
        }
     }
  }
```

(그림 5) 단방향 연결리스트 구조의 재구성 프로그램
(Fig. 5) The restructure program of unidirectional linked list structure

트랜지스터는 이론적으로 양방향의 신호 흐름을 가지지만 실제로 대부분의 신호 흐름은 95%정도가 단방향으로 설정되어 평가되는 것으로 나타나고 있으므로 초기에 가정한 경로 Vdd→Gnd 혹은 Vdd←Gnd를 설정하여 모델링을 수행한다. 이와 같은 가정 하에 (그림 5)의 단방향 연결리스트 구조의 재구성 프로그램을 구현하여 EDIF 컴파일러에서 만들어진 평면적인 CMOS 회로 구조를 사용하도록 하였다. (그림 5)의 재구성 프로그램은 모든 노드에 대해서 각각의 노드에 연결된 양방향 경로를 소오스에서 드레인으로 향한 순방향 연결리스트(forward_list[])와 드레인에서 소오스로 향한 역방향 연결리스트(backward_list[])로 나누어 재구성하였다.

CMOS 2-입력 AND게이트 회로를 예로써 (그림 6 (a))에 나타내고, AND게이트 회로에 대한 그래프 모델을 (그림 6(b))에 나타내었고, Vdd와 Gnd는 소오스 노드이며 실제로 Vdd와 Gnd 세기를 그대로 $N_1$과 $N_7$에 각각 전달하는 소오스 노드로 처리된다. 트랜지스터의 도전율은 비율값(width/length)인 6/1로 나타내었으며, 저항치는 역수로 계산한다. 기호(•)는 노드를 의미하며 각각 $N_1$, $N_2$, …, $N_7$의 라벨(label)로 나타내었다.

또한, (그림 6(b))에서 실제 그래프의 노드(Ni)는 점(•)이며 트랜지스터의 소오스($T_{is}$)와 드레인($T_{id}$)의 연결선이다. 그래프에서 에지는 실선인 (—)이며 트랜지스터에 해당된다. 여기서 점선 (┈)은 트랜지스터의 게이트 입력신호로 가상노드($N_2$, $N_5$)를 가지게 된다. CMOS회로의 경로 결정은 그룹으로 나누어 진행되도록 한다. 그룹화는 최초 입력부터 최종 출력까지의 수평적인 블록으로 나누어진다. 즉, 트랜지스터 게이트 입력은 단계적으로 인가될 수 있도록 나누며 나누어진 부분의 블록을 각각의 그룹으로 본다. 경로 결정은 그룹단위로 진행되며 진행순서는 최초 입력에서 시작하여 최종 출력으로 끝난다.




(그림 6) 2-입력 AND게이트 CMOS회로와 그래프 구조
(Fig. 6) Two-input AND gate of CMOS circuits and a structure of graph

예로서 2-입력 AND게이트 회로의 그룹화는 다음과 같다.

그룹1과 그룹2로 나누어질 때,

1) 그룹1($T_1$, $T_2$, $T_4$, $T_6$)에서 경로는 순방향과 역방향으로 각각 두개의 경로가 설정되며

순방향으로 Vdd→$N_1$→$T_1$→$N_3$→$T_4$→$N_6$→$T_6$→$N_7$→Gnd,

Vdd→$N_1$→$T_2$→$N_3$→$T_4$→$N_6$→$T_6$→$N_7$→Gnd와

역방향으로 Gnd→$N_7$→$T_6$→$N_6$→$T_4$→$N_3$→$T_1$→$N_1$→Vdd,

Gnd→$N_7$→$T_6$→$N_6$→$T_4$→$N_3$→$T_2$→$N_1$→Vdd가 있다.

2) 그룹2($T_3$, $T_5$)에서는 Vdd→$N_1$→$T_3$→$N_4$→$T_5$→$N_7$→Gnd의 순방향 경로와

Gnd→$N_7$→$T_5$→$N_4$→$T_3$→$N_1$→Vdd

의 역방향 경로가 설정된다.

노드 세기에 의한 경로의 결정은 트랜지스터의 저항치에 의해 계산되는 각 노드 값이 순차적인 값으로 Vdd부터 Gnd로 향하거나 Gnd부터 Vdd로 향하는 지배적인 경로로 결정된다. 여기서 두 가지의 연결상태를 고려하게 되며, 첫째로 직렬연결 부분의 경우는 순차적 지배경로로 결정하였으며, 둘째로 병렬연결 부분의 경우는 여러 개의 병렬 경로 중에 노드 세기가 가장 큰 경로를 선택하는 선택적 지배경로로 결정하였다.

전 처리 과정으로 만들어진 위의 결과에 각 트랜지스터의 게이트 상태 값(0, 1, X)을 event로 동작시켜 노드간의 에지 연결이나 삭제를 고려한 후 정확한 회로의 논리적 모델링을 실시한다. 여기서 게이트의 상태 값이 "0"(low)인 경우 스위치는 turn-OFF로 동작하고, "1"(high)인 경우 스위치가 turn-ON으로 동작하며, X인 경우는 unknown으로 해석하였다.

어느 한 노드 세기를 세 개의 값으로 정의한다. 초기의 노드 세기를 S_def, 논리 '1' 세기인1-strength를 S_1로 나타내며 순방향 경로의 계산으로, 논리 '0' 세기인 0-strength를 S_0으로 나타내었으며 역방향 경로의 계산에 의한 노드의 세기값이다. 세 개의 노드 세기에 관계된 노드의 논리값을 다음과 같은 관계식에 의해 결정하였다.

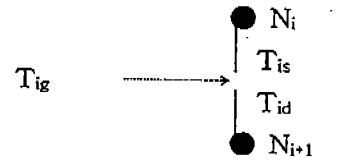
$$\text{노드의 상태} \begin{cases} \text{L, if } S\_0 > 0 \text{ and } S\_1 < S\_def \\ \text{H, if } S\_1 > 0 \text{ and } S\_0 < S\_def \\ \text{X, otherwise.} \end{cases} \quad (2)$$

## 3. 스위치 레벨 회로 결함 시뮬레이션

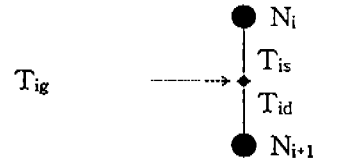
### 3.1 스위치 레벨 회로 결함 모델

스위치 레벨 결함 시뮬레이션은 고전적인 논리적 결함 모델로 stuck-at 결함과 비고전적인 결함 모델로 트랜지스터 stuck-open, stuck-closed 결함을 고려한 결함 시뮬레이션들이 소개되고 있다. 여러 기법의 제시에서 설정되는 결함 모델은 고전적인 stuck-at 결함의 한계를 극복하기 위하여 다른 스위치 레벨 결함이 제안되고, 이를 위해 앞장에서 구현 제시된 논리 시뮬레이션을 기초로 결함 시뮬레이션을 구현한다. 구현하는 결함 시뮬레이션에 적용되는 결함 모델은 비고전적인 트랜지스터 stuck-open, stuck-closed 결함 모델을 대상으로 하였다.

(그림 7)은 CMOS회로의 각 트랜지스터에 각각의 stuck-open과 stuck-closed 결함이 존재하는 그래프 구조를 나타내었다. (그림 7)에서 트랜지스터 $T_i$의 stuck-open 결함 동작은 게이트 신호값($T_{ig}$)의 인가와 무관하게 $T_{is}$와 $T_{id}$가 개방되는 turn-off로 수행되도록 하였으며, Ti의 stuck-closed 결함 동작은 게이트 신호값($T_{ig}$)의 인가에 무관하게 $T_{is}$와 $T_{id}$가 단락 되는 turn-on이 수행되도록 하였다. 이와 같은 게이트 입력에 의해 Vdd부터 Gnd로 향한 지배적 경로를 계산하였다.




(a) $T_i$의 stuck-open fault

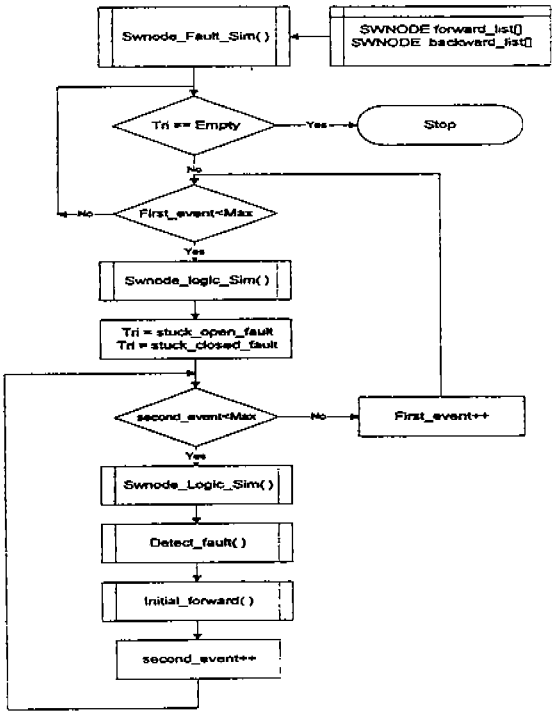



(b) $T_i$의 stuck-closed fault

(그림 7) 트랜지스터 $T_i$의 stuck-open, stuck-closed fault
(Fig. 7) Stuck-open, stuck-closed fault of the transistor $T_i$

### 3.2 스위치 레벨 회로 결함 시뮬레이션

스위치 레벨 결함 시뮬레이션은 주어진 테스트 벡터를 정상회로와 결함 모델이 부여된 결함회로에 인가하여 여기서 최후 출력의 결과는 정상회로와 결함회로가 다른 값을 가질 때에 그 결함회로의 결함이 검색되어진다. 2장에서 구현한 스위치 레벨 논리 시뮬레이션은 어떤 트랜지스터에 stuck-open, stuck-closed 결함을 부여하고 최초 입력에는 임의의 초기값을 반복적으로 인가하여 최종 출력의 결과값을 검사하게 된다. 이 때에 트랜지스터의 결함은 한 순간에 오직 한 개의 결함만이 존재한다는 가정 하에 시뮬레이션 수행이 진행된다.




(그림 8) 스위치 레벨 회로에 대한 결함 시뮬레이션의 흐름도

(Fig. 8) Flowchart of fault simulation for switch level circuits

스위치 레벨 결함 시뮬레이션은 정상회로에 대한 스위치 레벨 논리 시뮬레이션을 수행하고 특정한 트랜지스터의 stuck-open이나 stuck-closed 결함을 주입한 후 각 그룹별로 순차적 반복 수행으로 각 노드 상태값을 계산한다. 이때 그룹간의 상태값은 최초 입력부터 최종 출력까지 나누어진 블록 단위로 진행되어 그룹간 중간 게이트 입력 논리값으로 전달된다. 각 그룹 라벨은 입력파일에서 주어지며 라벨값의 증가에 따라 단계적으로 최종 출력까지 노드 상태값이 계산된다. (그림 6)을 예로서 설명하면 그룹1에 의해 계산된 $N_3$은 $T_3$, $T_5$의 동작을 결정하였다. 따라서 먼저 그룹1을 처리하고 다음으로 그룹2를 처리한다. 또한, 각 노드 상태를 결정하는 것은 논리 시뮬레이션과 같은 방법으로 (식 2)에 의해 low, high, unknown 상태 중에 하나로 결정된다. 스위치 레벨 회로에 대한 결함 시뮬레이션 처리 단계 흐름도는 (그림 8)과 같이 나타내었다.

처리 단계 흐름도에서 결함 시뮬레이션의 전체 루틴은 Swnode_Fault_Sim( )이며 전 처리 과정으로 만들어진 재구성의 자료구조인 SWNODE forward_list[ ], backward_list[ ]가 사용되어진다. 검색 대상의 결함을 트랜지스터의 stuck-open, stuck-closed 결함으로 하기 때문에 회로에 존재하는 모든 트랜지스터는 테스트를 마칠 때까지 하나씩 선택되어 결함 검색 처리가 진행된다. 또한 한 개의 결함이 주입되면 두개의 event(two pattern)로 처리가 진행되는데, 첫 번째는 first_event가 인가되어 정상회로에서 각 노드 상태값을 저장하고, 둘째는 결함을 가정한 회로에 second_event가 인가되고 상태값을 비교하여 결함 검색 여부를 결정하여 저장하게 된다. second_event의 반복 처리가 종료되면 first_event를 반복 처리하고 다음 트랜지스터 결함을 선택하여 진행하였다.

## 4. 모델 수행 및 결과 검토

2장과 3장에서 논의된 방법을 통하여 구현된 스위치 레벨 결함 시뮬레이션은 도전율이 제시된 A. Greiner가 설계한 CMOS 회로를 EDIF 파일 형식으로 변환하여 스위치 레벨 회로의 입력파일로 사용하였다[12]. 또한 데이터 파일은 주어진 CMOS 회로에 관련된 계수를 정의한 데이터를 작성하였다. 초기화로서 계수 정의 데이터 파일은 에지인 트랜지스터의 저항치와 Vdd, Gnd의 노드 세기값 및 그룹화를 위한 그룹 라벨 계수를 포함한다. 수행단계는 CMOS 회로를 그래프 모델로 변환하고 Vdd→Gnd 경로는 순방향

경로로, Vdd←Gnd 경로는 역방향 경로로 나타내었다. 경로 중에 존재하는 branch는 각 노드에 연결된 여러 개의 에지로 나누어진다. 예로서 (그림 6)에 보인 2-입력 AND CMOS 회로를 대상으로 경로 설정을 구한 결과는 (그림 9)과 같은 형태의 경로를 갖는다. 이때 주어진 경로는 지배적 경로임을 판정하고 최초 입력에 인가되는 입력벡터에 대한 최종 출력의 결과를 얻어내었다. 시뮬레이터 수행으로 얻어진 결과와 논리연산에 의해 얻어진 결과를 비교하였다.

```
(file-name a2y.edif) ----------< Forward Path >----------
No 0 Path : N1(s:100)=>T1(r:16)=>N3(s:84)=>T4(r:16)=>N6(s:68)=>T6(r:16)=>N7(s:100)
No 1 Path : N1(s:100)=>T2(r:16)=>N3(s:84)=>T4(r:16)=>N6(s:68)=>T6(r:16)=>N7(s:100)
No 2 Path : N1(s:100)=>T3(r: 8)=>N4(s:92)=>T5(r:16)=>N7(s:100)
(file-name a2y.edif) ----------< Backward Path >----------
No 0 Path : N7(s:100)=>T6(r:16)=>N6(s:84)=>T4(r:16)=>N3(s:68)=>T1(r:16)=>N1(s:100)
      Branch                                  =>N3(s:68)=>T2(r:16)=>N1(s:100)
No 1 Path : N7(s:100)=>T5(r:16)=>N4(s:84)=>T3(r: 8)=>N1(s:100)
```

(그림 9) CMOS회로의 경로 설정 결과
(Fig. 9) Results of dominant path for CMOS circuit

CMOS 회로의 경로 설정을 위한 모델링 수행에서 두개의 소오스 노드(Vdd, Gnd) 세기는 큰 정수값인 "100"으로 가정하고, 각 노드 세기 초기값은 "0"으로 설정한다. 각 에지의 저항치는 시뮬레이션에 필요한 이산적 정수값을 얻기 위해 "(length/width)*100"인 식의 계산으로 설정하였다. 이 때 직렬 연결 경로는 임의 노드 세기값을 계산할 때 앞에서 논의한 것처럼 시작 노드 세기($S_j$)에서 경로의 총 저항값($\Sigma r_i$)을 뺀다. 병렬 연결의 경우는 CMOS 회로의 각 트랜지스터 저항치가 동일하므로 초기 설정을 존재하는 모든 경로에 연결하며 각각 다른 패스로 설정하였다.

스위치 레벨 결함 시뮬레이션 수행은 임의의 트랜지스터 한 개를 stuck-open 결함으로 설정하여 진행하였다. 결함으로 가정한 트랜지스터 저항치는 노드 최대값의 근접치로 "99"를 설정하여 turn-OFF로 결함 시뮬레이션을 실시하였다. 또한 같은 방법으로 임의의 트랜지스터 한 개를 stuck-closed 결함으로 설정할 때, 트랜지스터의 저항치는 최소치의 근사값으로 "1"을 설정하여 turn-ON으로 결함 시뮬레이션을 실시하였다.

본 연구에서는 위에서 설정되는 각 파라미터와 2장, 3장에서 논의된 방법을 토대로 32bit급 intel 486 PC에 설치된 free software인 LINUX system에서 스위치 레벨 결함 시뮬레이터를 구현하고 수행하였다. CMOS 회로의 각 트랜지스터 stuck-open, stuck-closed 결함을 대상으로한 스위치 레벨 결함 시뮬레이션은 모든 입력 벡터를 적용하여 수행한 결과는 〈표 1〉과 같다.

〈표 1〉 트랜지스터의 stuck-open, stuck-closed fault 검색 결과
〈Table 1〉 Results of detected transistor stuck-open, stuck-closed fault

| circuit | Logic function | Number of transistor | St-open fault | | St-closed fault | |
|---|---|---|---|---|---|---|
| | | | Total | Detect | Total | Detect |
| a2_y | i0 AND i1 | 6 | 6 | 6 | 6 | 6 |
| o2_y | i0 OR i1 | 6 | 6 | 6 | 6 | 6 |
| annup_y | (i1 OR i2) NAND (i3 OR i4) | 8 | 8 | 8 | 8 | 8 |
| cry_y | NOT((si AND ci) OR (pi AND (si OR ci))) | 10 | 10 | 10 | 10 | 10 |
| mx2_y | ((j0 AND j1) OR (i0 AND i1) | 10 | 10 | 10 | 10 | 10 |
| nmx2_y | ((j0 AND j1) NOR (i0 AND i1) | 8 | 8 | 8 | 8 | 8 |
| na2_y | i0 NAND i1 | 4 | 4 | 4 | 4 | 4 |
| nao3_y | (i0 AND i1) NOR i2 | 6 | 6 | 6 | 6 | 6 |
| no2_y | i0 NOR i1 | 4 | 4 | 4 | 4 | 4 |
| nxr2_y | NOT(i0 XOR i1) | 10 | 10 | 10 | 10 | 9 |
| xr2_y | i0 XOR i1 | 10 | 10 | 10 | 10 | 10 |
| sum_y | NOT((si AND pi) AND ci) OR (cob AND ((si OR pi)OR ci))) | 14 | 14 | 14 | 14 | 13 |

〈표 1〉에서 보인 것처럼 "nxr2_y"와 "sum_y"회로는 stuck-closed 결함 1개씩이 검색되지 않아 95%, 96.4%의 결함 시험도를 얻었다. 그 외의 모든 회로는 결함 검색이 완전하게 처리됨을 본다. 일반적으로 stuck-open 결함 보다 stuck-closed 결함의 결함 시험도는 낮다. 예를 들면 2-입력 AND게이트 회로에서 최초 입력으로 두 개의 입력 패턴 벡터의 초기값이 인가되는 경우에 각 트랜지스터 stuck-open, stuck-closed 결함의 검색 결과는 〈표 2〉와 같이 얻었다.

〈표 2〉에서 최초 입력 테스트 벡터($i_1$ $i_0$)인 "1 1 / 1 0"이 스위치 레벨 결함 시뮬레이터에 인가될 때 first

vector는 정상 상태값을 저장하고 second vector로 결함 상태를 판단하게 된다. 여기서 트랜지스터 stuck-open 결함은 논리 상태값으로 정확하게 검색하였다. 또한 트랜지스터 stuck-closed 결함을 검색할 때, $T_3$, $T_4$와 $T_6$의 트랜지스터는 정상회로의 결과가 '0' 또는 '1'인 논리 상태값에 대해 결함회로의 결과가 unknown으로 차이를 갖는 애매한 검색을 갖기도 한다. 그러나 스위치 레벨 결함 시뮬레이션에서 애매한 결함 검색을 갖는 트랜지스터는 two pattern vector가 주어질 때, 결과로서 최종 출력 상태값의 전 처리 과정인 노드 세기값은 정상회로와 결함회로에서 차이를 보이므로 결함 검색 판단을 내릴 수 있었다.

〈표 2〉 2-입력 AND게이트 회로의 stuck-open, stuck-closed fault 검색 결과

〈Table 2〉 Results of detected stuck-open, stuck-closed fault for two-input AND gate

| transistor | stuck-open fault | | stuck-closed fault | |
|---|---|---|---|---|
| | first vector/ second vector | good/fault | first vector/ second vector | good/fault |
| $T_1$ | 11/10 | 0/1 | 00/11 | 1/0 |
| $T_2$ | 11/01 | 0/1 | 00/11 | 1/0 |
| $T_3$ | 00/11 | 1/0 | 00/00 | 0/X |
| $T_4$ | 00/11 | 1/0 | 00/10 | 0/X |
| $T_5$ | 11/00 | 0/1 | 00/11 | 1/0 |
| $T_6$ | 00/11 | 1/0 | 00/01 | 0/X |

## 5. 결 론

본 연구에서는 논리 게이트 레벨까지 제한된 생각을 물리적 레벨인 스위치 레벨로 확장하는 기반 연구로 적용된다. CMOS 회로는 그래프 모델로 변환하는데, 이때 스위치 레벨 회로의 신호 흐름이 약 95%정도가 단방향으로 설정되어 평가되기 때문에 스위치 회로를 모델화하는데 양방향 신호 흐름을 단방향으로 재구성하였다. 스위치 레벨 결함 시뮬레이션은 CMOS 회로의 대칭적인 특징을 적용한 지배적 경로 기법에 의해 수행하였다. 또한 스위치 레벨 시뮬레이션 입력 파일은 CMOS 회로에 적용한 표준형식의 EDIF 컴파일러를 이용하여 구축한 자료구조 형태를 갖는다. 논리 시뮬레이션은 새롭게 제시된 지배적 경로 기법을 적용하여 Vdd부터 Gnd 혹은 Gnd부터 Vdd로 향하는 지배적 경로 판정에 따라 최종 출력 상태값을 결정하였다. 스위치 레벨 결함 시뮬레이션은 논리 시뮬레이션에 기초하여 회로 경로상 임의 트랜지스터에 stuck-open, stuck-closed 결함을 주입시켜 지배적 경로를 판정하였다. 이 때 최초 입력은 두 개의 패턴 벡터(first event, second event)를 인가하여 정상회로의 최종 출력 상태값과 결함회로의 출력 상태값과 비교하여 상이한 결과로 결함을 검색하였다. 본 논문에서 구현한 스위치 레벨 결함 시뮬레이터는 CMOS회로 12개를 입력회로로 사용하여 결함 시뮬레이션을 수행한 결과로서, 결함 시험도가 낮은 stuck-closed 결함에 의해 "nxr2_y"와 "sum_y"회로에 대해서는 95%, 96.4%의 결함 시험도를 얻었으며, 그 외는 100%의 결함 시험도를 얻어 완전한 결함 검색을 보였다.

앞으로 본 연구는 상위 레벨 회로에 대한 계층적 결함 시뮬레이터를 본 연구에서 보인 스위치 레벨 결함 시뮬레이터와 결합하여 큰 VLSI 회로의 스위치 레벨 결함을 다룰 수 있는 확장된 계층적 결함 시뮬레이터를 개발에 큰 도움이 되리라 본다.

## 참 고 문 헌

[1] R. L. Wadsack, "Fault Modeling and Logic simulation of CMOS and MOS Integrated Circuits," *Bell System Technical Journal*, Vol. 57, pp. 1449-1474, May-June, 1978.

[2] S. M. Reddy, V. D. Agrawal and S. K. Jain, "A Gate Level Model for CMOS Combinational Logic Circuits with Application to Fault Detection," Proc. 21st Design Automation Conference, pp.504-509, June 1984.

[3] M. D. Schuster and R. E. Bryant, "Concurrent Fault Simulation of MOS Digital Circuits," Proc. MIT Conf. on adv. Research in VLSI, pp. 129-138, Jan. 1984.

[4] J. P. Shen, W. Maly and F. J. Ferguson, "Inductive Fault Analysis of MOS Integrated Circuits", IEEE Design & Test of Computers, Vol. 2, pp. 13-26, Dec. 1985.

[5] C. Y. Lo, H. N. Nham, and A. K. Bose, "Algorithms for an Advanced Fault Simulation in MOTIS," IEEE Trans. on CAD, Vol. CAD-6, pp.232-240, Mar. 1987.

[6] J. P. Hayes, "Fault Modeling of Digital MOS Integrated Circuits," IEEE Trans. on CAD, Vol. CAD-3, pp.200-208, Jul. 1984.

[7] R. E. Bryant, "A Switch Level Model and Simulator for MOS Digital Systems," IEEE Trans. Computers, Vol. C-33, pp.160-177, Feb. 1984.

[8] D. Adler, "Switch-Level Simulation Using Dynamic Graph Algorithms," IEEE Trans. Computers, Vol. 10, pp.346-355, Mar. 1991.

[9] J. P. Hayes, "Pseudo-Boolean Logic Circuits," IEEE Trans. Computers, Vol. C-35, pp.602-612, Jul. 1986.

[10] N. P. Jouppi, "Derivation of Signal Flow Direction in MOS VLSI," IEEE Trans. on CAD, Vol CAD-6, pp.480-489, May 1987.

[11] E. D. Fabrricius, *Introduction to VLSI design*, McGraw-Hill, N. Y., 1990.

[12] A. Greiner, F. Pëcheux, "ALLIANCE: A complete Set of CAD Tools for Teaching VLSI Design," Laboratorie MASI/CAO-VLSI, 1993.

[13] S. D. Sherelekar and P. S. Subramanian, "Conditional Robust Two-Pattern Tests and CMOS Design for Testability", IEEE Trans. on CAD, Vol.7, No.3, pp.325-332, Mar. 1988.

[14] S. Sastry, "Detectability of CMOS Stuck-open Faults Using Random and Pseudorandom Test Sequences," IEEE Trans. on CAD, Vol. 7, No. 9, pp.933-946, Sep. 1988.

**연 윤 모**

1982년 2월 충북대학교 공과대학 전기공학과 공학사
1989년 2월 성균관대학교 대학원 전기공학과 공학석사
1993년 2월 성균관대학교 대학원 전기공학과 박사수료
1982년 4월~1990년 3월 충북대학교 전자계산소 조교
1990년 3월~현재 청주전문대학 전자계산과 조교수, 부교수
주관심분야: VLSI Testing, CAD 시스템

**민 형 복**

1980년 2월 서울대학교 공과대학 전자공학과 공학사
1982년 2월 한국과학기술원 전기 및 전자공학과 공학석사
1990년 12월 The University of Texas at Austin 전기 및 컴퓨터 공학과 공학박사
1981년 3월~1985년 4월 금성통신(주)연구소 주임연구원
1985년 8월~1986년 7월 미국 Columbia대학교 연구원
1991년 3월~현재 성균관대학교 전기공학과 조교수, 부교수
주관심분야: VLSI Testing, CAD 시스템