

디지털 CMOS회로의 Multi-Level Test를 위한 범용 Test Set 생성

(Universal Test Set Generation for Multi-Level Test of Digital CMOS Circuits)

金 東 郁

(Dong Wook Kim)

要 約

CMOS회로 시대의 도래와 함께 CMOS회로에서의 FET Stuck-On고장과 FET Stuck-Off고장은 gate 회로의 Stuck-At고장과 더불어 test비용을 가중시켜 IC의 생산원가를 증가시키고 있다. 본 논문에서는 gate회로와 transistor회로를 모두 test할 수 있는 test set과 그 생성 알고리즘을 제안하여 IC 설계및 제조공정의 test절차를 획일화할 수 있는 방안을 제시한다. 이 알고리즘의 입력재원으로서는 transistor 또는 gate회로의 참조없이 주어진 논리함수만을 사용하며, 그 결과의 test set은 test절차의 획일화 뿐 아니라 알고리즘의 복잡도와 test 적용시간에 있어 기존의 방법에 비해 상당한 개선을 가져올 수 있다.

Abstract

As the CMOS technology becomes the most dominant circuit realization method, the cost problem for the test which includes both the transistor-level FET stuck-on and stuck-off faults and the gatelevel stuck-at faults becomes more and more serious. In accordance, this paper proposes a test set and its generation algorithm, which handles both the transistor-level faults and the gate-level faults, thus can unify the test steps during the IC design and fabrication procedure. This algorithm uses only the logic equation of the given logic function as the input resource without referring the transistor or gate circuit. Also, the resultant test set from this algorithm can improve in both the complexity of the generation algorithm and the time to apply the test as well as unify the test steps in comparing the existing methods.

1. 서 론

1970년대 후반부터 현재에 이르는 약 20년간 고집적회로를 위한 Bipolar시대로 부터 MOSFET시대로

의 전환기를 맞고 있다. 속도와 전력소모에 있어서 상관관계에 있는 두 회로실현기술은, 1988년 IC 시장의 대등한 점유율을 보였으며 MOSFET의 점유율은 점차 증가추세에 있다.^[1] 특히 CMOS 논리회로는 정상상태 (Steady State)시 정전류(Static Current)가 흐르지 않으므로 전력소모가 극히 적고 초고집적회로의 실현기술 발달로 1980년대 중반 이후 크게 각광받고 있다. Dual network을 형성시켜야 하는 CMOS회로는 같은 함수를 실현하는 nMOS

*正會員, 光云大學校 電子材料工學科
(Dept of Elec. Materials Eng., Kwangwoon Univ.)

接受日字: 1992年 7月 18日

회로보다 IC사용율이 떨어져 보이나 충분한 방전경로 형성을 위해 nMOS회로는 transistor의 크기를 조절하여야 하는데 반해 CMOS회로는 그 조절이 필요없어 두 회로의 크기는 필적할 정도이다. 디지털 회로의 test에 사용되는 고장(Fault) 모델로는 gate 레벨의 Stuck-At(S-A) 고장이 주로 사용되어왔으나 CMOS회로에서는 두 network의 Dual동작에 의해 Stuck-On(S-On) 고장과 Stuck-Off(S-Off) 고장의 새로운 고장형태가 발생한다.^[2]

S-Off 고장은 특정 FET가 항상 off상태로 고장되는 고장 형태로서 이로 인한 출력은 high impedance 상태, 즉 조합회로에서 sequential동작을 유발하게 된다. 또한 S-On 고장은 특정 FET가 항상 on되어 입력조합의 형태에 따라 두 network 모두에 전류경로가 형성, 전류원(VDD)에서 접지(GND)로의 정전류가 흐르게 된다. 이 두 형태의 고장을 test하기 위해 1970년대 후반 부터 많은 연구가 진행되어오고 있으며, 이 두 고장은 그 성격상 transistor 레벨에서 취급되어야 하는 것으로 알려져 왔다.^[3-8]

이 연구는 S-A고장을 위한 test 생성방법을 응용하거나^[9] S-A 고장의 test를 적용시키는 방법^[10, 11] 들 또한 연구되고 있으나, 이들의 고장검출능력을 높이기 위해서는 또 다른 생성 알고리즘을 적용하거나 transistor 레벨 회로를 gate레벨로 재 modeling하여야 한다. 본 논문에서는 IC 제조과정에서 요구되는 simulation과 test의 모든 단계에 적용할 수 있는 CMOS디지털 회로의 범용 test set을 제안한다.

이 test set은 gate 레벨 또는 transistor 레벨로 실현된 CMOS 회로에 적용할 수 있으며 이를 위해 test의 생성은 transistor회로나 gate회로 이외의 회로정보(본 논문에서는 논리함수)를 사용한다. 본 논문의 구성은, 먼저 제안되는 test set의 구조와 성질을 transistor회로에서 규명하고(II), 이 test set의 생성 알고리즘을 소개한 뒤(III), 이 test set을 gate회로에 적용할 수 있음을 보인다(IV).

마지막으로 결론 (V)에서는 본 논문에서 제안된 test set이 test 단계의 획일화 뿐 아니라 알고리즘의 복잡도(complexity)와 test 적용시간에 있어 기존의 방법들에 비해 상당한 개선을 가져올 수 있음을 보인다. 본 논문에서는 S-On고장을 test하기 위하여 전원에서 접지로 흐르는 정전류의 양을 측정하는 Current Monitoring(IDDQ 측정) 방법^[8, 12]을 사용한다. 또한 단일고장발생을 가정하며 고려되는 모든 회로는 최소화(Irreducible and Irredundant)된 것으로 간주한다.

II. Robust Three Patterns for Four Tests

(R3P4T)

Dual network으로 형성되는 static 디지털 CMOS회로는 그림 1에 나타낸 것과 같이 complex-gate 회로(b)와 primitive-gate회로(c)로 실현할 수 있다. Complex-gate회로는 주어진 논리함수가 한 쌍의 dual network으로 실현되나 primitive-gate회로는 미리 정의된 primitive gate들의 조합으로 논리함수가 실현된다. 이 두 방법 모두 p-channel FET로 구성되는 network(P-Net)과 n-channel FET로 구성되는 network(N-Net)이 dual형태를 이룬다. 본 논문은 주로 complex-gate회로에 대해 진행되나 그 이론과 결과는 primitive-gate회로에도 그대로 적용된다.^[13] 본 논문의 설명을 위해 다음을 먼저 정의한다.

(정의 1) Vertex는 don't care 조건의 입력을 가지지 않는 완전한 입력조합으로 정의하며, vertex i 는 V_j 로 표기한다. 이 때 j 는 i 의 십진수를 나타낸다. 또한 vertex의 값은 그 vertex를 입력으로 적용하였을 시의 출력값으로 정의한다.

(정의 2) Vertex의 order는 그 vertex가 포함되는 Prime Implicant(PIm) 또는 Prime Implicate(PIx)의 수로 정의한다.

(정의 3) 두 vertex간의 거리(distance)는 두 vertex간의 Hamming distance로 정의하며, Hamming distance가 i 인 두 vertex들을 distance- i vertices라 한다.

(정의 4) Vertex i 의 주위 vertices(Surrounding Vertices:SVi)는 vertex i 에서 거리가 1인 모든 vertex로 정의한다.

(정의 5) 간략화된 형태(reduced form)의 회로는 두 개 이상의 PIm 또는 PIx가 결합한 회로구성의 형태로 정의한다. 예로써, 두 개의 PIm의 합인 $AB+AC$ 가 $A(B+C)$ 의 형태로 회로가 구성되는 것을 의미하며 이 때 A에 해당하는 FET를 공통(common) FET, B 또는 C에 해당하는 FET를 비공통(uncommon) FET라 정의한다. 그림 1 (b), (c)의 간략화된 형태는 그림 2 (a), (b)에 각각 나타내었다.

1. Robustness Test 조건

S-Off고장의 가장 큰 영향은 조합회로내에서 sequential동작을 유발하는 것이다. 따라서 한 개의 S-Off고장을 검출하기 위해서 최소 두 개의 test 패턴이 필요하며 이를 초기패턴(T1)과 검출패턴(T2)이라 한다. 초기패턴은 고장이 존재하는 network의 dual network에 전류경로를 형성시키며 검출패턴

은 고장이 존재하는 network에서 고장난 FET를 경유하는 경로중 하나를 연결시켜야 한다. Test를 수행하는 과정은 T1과 T2를 연쇄적으로 적용하는 것인데, 이 때 만약 T1에서 T2로의 천이과정중 한 개 또는 그 이상의 천이패턴을 경유하고 그 중 하나라도 T2와 같은 논리치를 갖는다면 이 test는 robust하지

않다.¹⁴⁾

예를 들어 그림 3 (b)에서 P5의 S-Off고장 (P5/Off)을 검출하기 위한 T2는 (ABCD)=(1010) (V_{10})이며 T1은 (0101) (V_5), (0110) (V_6), (1001) (V_9)이 가능하다. 따라서 (V_5, V_{10}), (V_6, V_{10}), (V_9, V_{10})의 세 test 패턴 쌍이 존재한다.

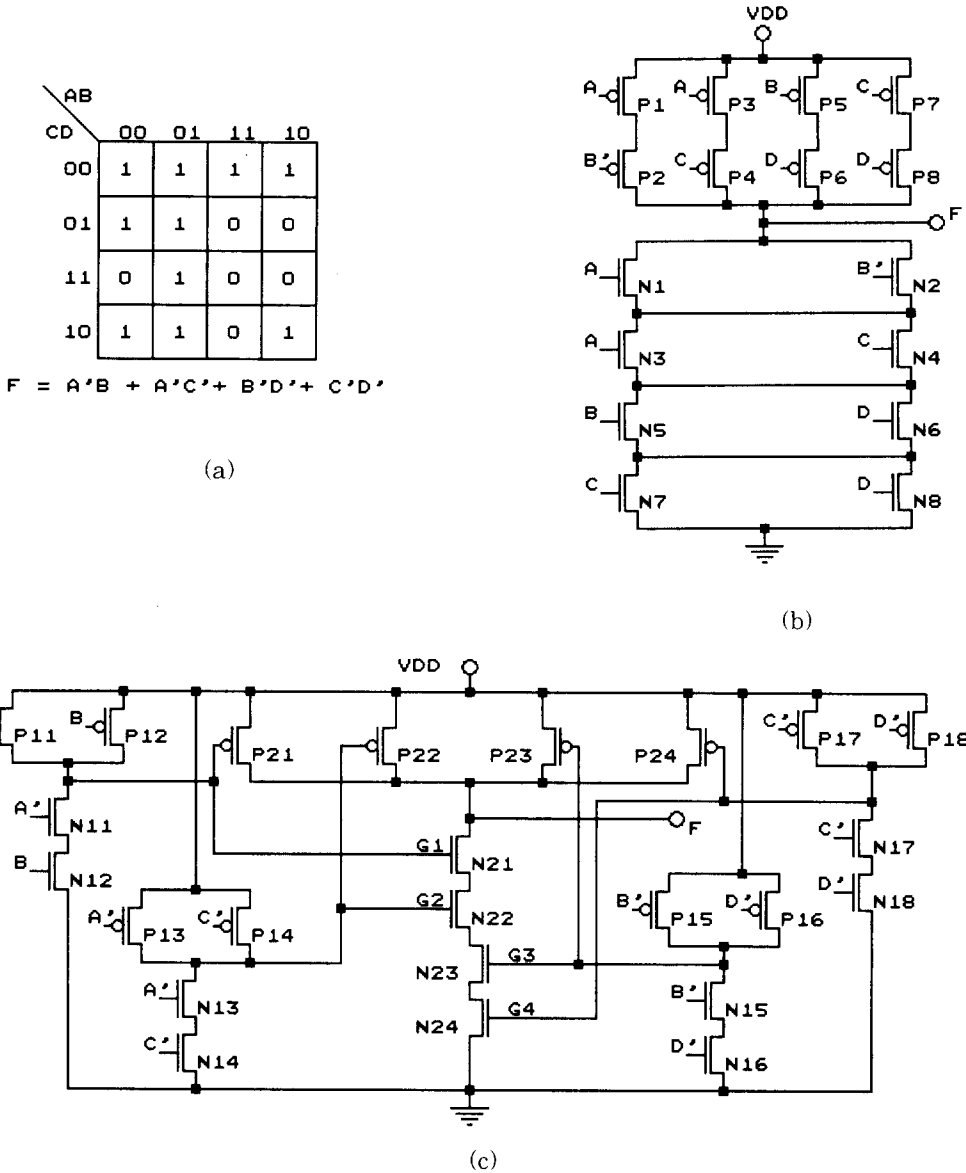
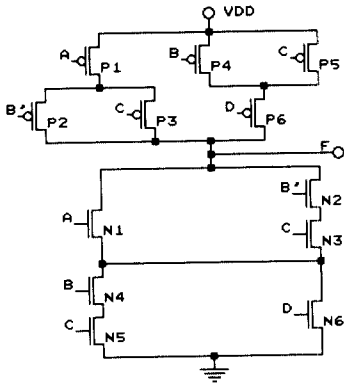


그림 1. Complex-gate회로와 Primitive-gate회로

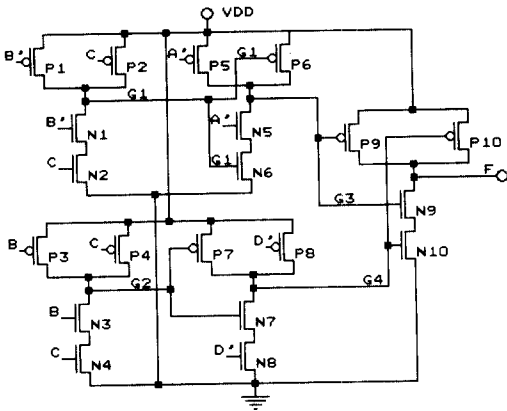
(a) 논리함수 (b) Complex-gate 실현 (c) Primitive-gate 실현

Fig. 1. Complex-gate Circuit and Primitive gate Circuit.

(a) Logic Function, (b) Complex-gate Realization, (c) Primitive-gate Realization.



(a)



(b)

그림 2. 그림 1의 간략화된 형태

(a) Complex-gate 회로

(b) Primitive-gate 회로

Fig. 2. Reduced Form of Fig. 1.

(a) Complex-gate Circuit.

(b) Primitive-gate Circuit.

그러나 이 세 쌍 모두 천이패턴을 거치게 되는데, 예로써 (V_9, V_{10})은 V_8 과 V_{11} 의 천이패턴을 거친다. V_8 과 V_{11} 모두 $T2(V_{10})$ 와 같은 논리 1을 갖고 있고, V_9 은 P7-P8의 경로를, V_{10} 은 P9-P10의 경로를 연결시켜 출력 F를 VDD로 충전시킨다. 이 때 출력의 전압치는 V_8 또는 V_{11} 의 duration time에 의존하며 따라서 이 test는 실패의 가능성이 높다.

Robustness 문제의 유발을 막기 위해 distance-1의 test(Robust Two Pattern Test: R2PT)만 사용되는 데, 그림 3 (b)의 P5 또는 P6처럼 S-Off고장에 대한 robust test가 존재하지 않는 FET가 존

재할 수 있다. 주어진 회로의 모든 FET에 대해 S-Off고장의 robust test 존재여부는 다음의 세 조건으로 판별 가능하다.

<조건 1> 간략화되지 않은 형태의 CMOS회로에서 주어진 p(n)-channel FET(FET A)에 대해, V_i 는 다음의 세 조건을 만족하면 A/Off에 대한 R2PT의 T2가 될 수 있다.

1. V_i 의 값이 1(0)이다.
2. V_i 는 오직 해당 PIm 또는 PIx에만 속한다.
3. SV_i중 최소한 개의 vertex는 V_i 와 다른 값을 갖는다.

<조건 2> m 개의 PIm 또는 PIx가 결합하여 간략화된 CMOS회로의 경우 공통의 p(n)-channel FET(FET B)에 대해 B/Off를 test할 때, V_j 는 다음의 세 조건을 만족하면 B/Off에 대한 R2PT의 T2가 될 수 있다.

1. V_j 의 값이 1(0)이다.
2. V_j 는 m 개의 PIm 또는 PIx중 한 개이라도 포함된다.
3. SV_j중 최소한 vertex는 V_j 와 다른 값을 갖는다.

<조건 3> m 개의 PIm 또는 PIx가 결합하여 간략화된 CMOS회로의 경우 비공통인 p(n)-channel FET(FET C)에 대해 C/Off를 test할 때, V_k 는 다음의 세 조건을 만족하면 C/Off에 대한 R2PT의 T2가 될 수 있다.

1. V_k 의 값이 1(0)이다.
2. V_k 는 m 개의 PIm 또는 PIx중 해당 PIm 또는 PIx에만 속한다.
3. SV_k중 최소한 vertex는 V_k 와 다른 값을 갖는다.

주어진 CMOS회로의 각 FET는 위의 세 경우중 하나에 해당되며 그 경우의 조건을 만족하면 최소한한 개의 R2PT가 존재한다. 그 T2는 각 경우에 명시된 vertex이며 해당 T1은 T2와 다른 값을 갖는 한 개의 SV로 결정할 수 있다.

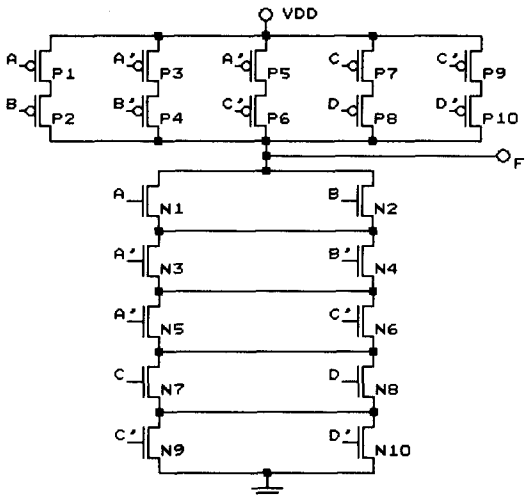
그러나 그림 3 (b)의 P5와 P6의 경우와 같이 모든 FET가 위의 조건을 만족하지 않는 CMOS회로도 존재한다. 이런 회로에 대해서는 다음의 정리로 해결될 수 있다.

[정리 1] 디지털 CMOS회로에서 해당 S-Off고장을 검출하는 R2PT가 존재하지 않는 FET가 한 개라도 존재할 경우, 회로실현의 형태를 SP-PS(PS-SP)에서 PS-SP(SP-PS)로 변환하면 각 FET의 S-Off 고장을 검출할 수 있는 R2PT가 최소한 개씩은 존재한다.

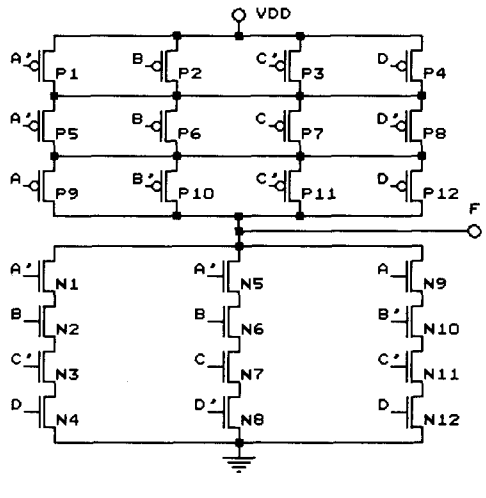
이 정리는 [13]에 증명되어 있으며 본 논문에서

	AB			
CD	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	1	1	1	1
10	1	0	1	1

(a)



(a)



(b)

그림 3. Robustness 조건의 예제 회로

(a) 논리함수 (b) SP-PS형 실현 (c) PS-SP형 실현

Fig. 3. Example Circuit for Robustness Conditions.

(a) Logic Function, (b) SP-PS Form Realization, (c) PS-SP Form Realization.

는 생략한다. 위의 정리에서 SP(PS)는 Sum-of-Products(Product-of-Sums), 즉 직렬연결된 FET들의 병렬연결(병렬연결된 FET들의 직렬연결)을 의미한다. 또한 선행의 회로 형태는 P-Net의 연결형태를, 후행의 것은 N-Net의 연결형태를 각각 나타낸다. 그림 3 (b) 회로는 정리 1에 의해 그림 3 (c)로 변환할 수 있고 이 회로의 모든 FET는 최소한 한 개의 R2PT를 갖는다. 따라서 정리 1과 조건 1, 2, 3에 의해 모든 FET는 R2PT를 가질 수 있으며 차후의 설명은 모든 FET가 R2PT를 갖는 것으로 간주한다.

2. Robust Two Patterns for Three Tests (R2P3T)

앞 절의 설명에서와 같이 한 개의 S-Off고장을

test하는 데에는 최소 두 개의 패턴이 필요하다. 또한 VDD에서 GND에 이르는 전류경로를 연결시키는 S-On고장은 고장발생을 가정하여 그 전류경로를 연결하는 한 개의 패턴을 적용, VDD에서 흐르는 전류의 양을 측정(Current Monitoring 또는 IDDQ 측정)하여 고장을 검출할 수 있다.^(8, 12)

주어진 회로의 한 FET(FET A)에 대해 A/Off test의 T1은 FET A가 속한 network의 모든 경로를 차단하고 그 dual network에는 최소한 한 개의 경로를 연결시켜야 한다. 또한 T2는 FET A가 속한 network에서 FET A를 경유하는 경로를 연결시켜야 하며 그 dual network의 모든 경로를 차단해야 한다. A/On test는, 고장발생을 가정하지 않았을 때 FET A가 속한 network에서는 FET A를 경유하는 경로중 FET A 이외의 모든 FET를 도통시켜야 하

며 그 dual network에서는 최소한 한 개의 경로를 연결시켜야 한다.

이 두 고장의 test 조건은 단일 FET에 대한 것이나 CMOS회로는 두 network의 동작과 구성에 있어 duality 성질을 갖고 있으므로 한 network의 주어진 FET에 대해 반드시 다른 network에 dual FET가 존재한다. 따라서 본 논문에서는 두 dual FET를 동시에 고려하여 새로운 test 단위를 제안한다. 먼저 두 network중 SP형태의 network을 고려한다. SP network 내의 한 FET(FET A)에 대한 A/Off의 T2 조건은 SP network의 모든 전류경로를 차단하고 PS network의 전류경로를 연결하는 것이며, T1의 조건은 PS network의 경로를 차단하고 SP network에서 FET A를 경유하는 경로를 연결하는 것이다. 이 test 패턴 쌍은 정리 1과 조건 1, 2, 3에 의해 distance가 1이며 따라서 이 test는 R2PT이다. 이 R2PT에서 T1과 T2간에 서로 다른 bit는 임의의 입력변수일 수 있다. 그러나 그 중 서로 다른 bit가 FET A의 입력변수인 R2PT를 고려하자. 이 경우 T1은 PS network에서 단지 FET A의 dual FET(DFET A)를 경유하는 경로만을 연결시키고 다른 모든 경로는 단절시키며 SP network의 경로는 모두 차단시킨다. 반면 T2는 SP network에서 FET A를 경유하는 경로만 연결시키고 PS network의 모든 경로를 차단시킨다. 즉, T1과 T2간의 서로 다른 bit가 FET A의 입력변수에 해당하는 R2PT(Dual R2PT)의 두 패턴은 적용시의 회로 동작이 dual 성질을 갖게 된다. 이 Dual R2PT의 존재는 다음의 정리로 규명한다.

[정리 2] 최소화된 CMOS 논리회로의 모든 FET가 조건 1, 2, 또는 3을 만족하면, 모든 FET에 대해 반드시 Dual R2PT가 최소한 한 개 존재한다.

(증명) 조건 1, 2, 3에 의해 모든 FET가 최소한 개의 T2를 갖는다. 이 T2(V_i)는 test되는 FET (FET Under Test: FUT)를 경유하는 한 개 이상의 경로를 연결시킨다. T2에서 FUT의 입력변수를 변화시킨 패턴을 V_i 라 하자. 만약 V_i 가 FUT가 속한 network에 또다른 경로를 연결시킨다면, 최소화(irreducible and irredundant)된 논리식의 성질에 의해 V_i 로 연결되는 경로는 V_i 로 연결되는 경로와 포함관계가 성립된다. 따라서 CMOS회로가 최소화된 논리식에 의해 실행되었다면 V_i 는 FUT가 속한 network의 모든 경로를 차단한다. 결국 최소화된 CMOS 회로에서는 (V_i, V_i)는 Dual R2PT를 형성한다. Q. E. D.

정리 1, 2와 조건 1, 2, 3에서 Dual R2PT는 FUT의 S-Off test임이 증명되었다. 이제 T1을 고

려하자. T1은 dual network에서 FUT의 dual FET(DFUT)를 경유하는 한 개 이상의 경로를 연결한다. 이 경로는 직렬연결로 볼 수 있으며 이 경로에 속한 FET들의 dual FET들은 FUT가 속해있는 network에서 병렬연결되어 있고 FET의 동작에 의해 이들은 모두 off되어 전류경로를 모두 차단하게 된다. 만약 FUT에 S-On고장이 발생하였다면 FUT가 도통되어 FUT가 속한 network에 최소한 한 개의 전류경로를 연결시키게 되어 VDD에서 GND로 흐르는 경로를 연결시키게 되어서 T1이 DFUT/On의 test와 동일한 조건을 갖게된다. 또한 위에서 설명한 바와 같이 T2는 T1과 duality 동작을 유발한다. 따라서 T2의 조건은 T1의 조건과 dual 조건이며 이는 DFUT와 dual인 FUT의 S-On test의 조건과 동일하다. 결국 Dual R2PT로써 다음의 보조정리가 성립한다.

[보조정리 1] Static CMOS 디지털 회로에서 임의의 FET(FET A)의 S-Off고장에 대한 dual R2PT는 FET A의 S-Off고장, FET A의 S-On고장, DFET A의 S-On고장을 검출한다 (Robust Two Patterns for Three Tests: R2P3T).

예로써 그림 2 (a)의 P1/Off의 R2PT는 (V_3, V_1), (V_9, V_1), (V_{13}, V_5), (V_{14}, V_6), (V_{15}, V_7)이다. P1의 입력변수가 A이므로 P1/Off의 dual R2PT는 (V_9, V_1), (V_{13}, V_5), (V_{14}, V_6), (V_{15}, V_7)이며, $V_9, V_{13}, V_{14}, V_{15}$ 는 P1/On의 test이고 V_1, V_5, V_6, V_7 은 N1/On의 test이다. 따라서 P1, N1에 대한 R2P3T는 (V_9, V_1), (V_{13}, V_5), (V_{14}, V_6), (V_{15}, V_7) 이 된다.

3. Robust Three Patterns for Four Tests (R3P4T)

보조정리 1의 설명에 덧붙여 이제 DFUT의 S-Off 고장을 고려한다. 이 고장을 test하기 위한 T1의 조건은, T1의 적용시 DFUT를 경유하는 한 개 이상의 경로가 연결되고 FUT가 속한 network의 모든 경로는 차단된다. 또한 T2는 FUT를 경유하는 한 개 이상의 경로를 연결시키고 DFUT가 속한 network의 모든 경로를 차단시킨다. 따라서 (T2, T1)은 DFET A의 S-Off test이며 다음의 정리에 의해 명백해진다.

[정리 3] 최소화된 static CMOS회로에서 (T1, T2)가 주어진 FET(FET A)의 S-Off고장에 대한 R2PT이고 정리 2를 만족하면(즉, R2P3T), 그 역순인 (T2, T1)은 DFET A의 S-Off고장에 대한 R2PT이다.

이 정리의 증명은 위의 설명으로 대신한다. 또한

정리 2, 보조정리 1, 정리 3에 의해 다음의 보조정리가 성립한다.

[보조정리 2] 최소화된 CMOS 디지털 회로에서 (T1, T2)가 주어진 FET(FET A)의 S-Off고장에 대한 R2PT이고 정리 2를 만족하면, 세 패턴 test 단위, 즉 (T1, T2, T1) 또는 (T2, T1, T2)는 네 개의 FET 고장(FET A의 S-Off고장, FET A의 S-On고장, DFET A의 S-Off고장, DFET A의 S-On고장)을 검출한다(Robust Three Patterns for Four Tests: R3P4T).

보조정리 2의 R3P4T는 양방성을 띄고 있다. 즉, (T1, T2, T1)과 (T2, T1, T2)는 test 수행에 있어 동일한 효과를 가지며 따라서 이들은 방향성에 관계없이 <T1, T2> 또는 <T2, T1>으로 표기하기로 한다. 예로써, 그림 2 (a)의 P1, N1 쌍에 대한 R3P4T는 <1, 9>, <5, 13>, <6, 14>, <7, 15>중 어느 것이든 가능하다. 그림 2 (a)의 모든 FET에 대한 가능한 모든 R3P4T set을 표 1에 나타내었다. 두 개 이상의 R3P4T를 갖고 있는 FET 쌍에 대해서는 그 중 어느 것을 택하여도 무방하다.

이 상의 설명은 모두 complex-gate회로에 대한 것이었으나 이는 primitive-gate회로에도 그대로 적용된다. 따라서 재 설명은 생략하고 ([13] 참조), 예를 들어 그 타당성만 설명하기로 한다. Complex-gate회로에서 각각의 FET 연결상태는 primitive-gate에서 한 개의 primitive gate에 해당한다. 예로써 그림 2 (a)의 P2, P3와 N2, N3의 연결은 그림 2 (b)의 G1에 해당하고 그림 2 (a)의 P1, (P2, P3)와 N1, (N2, N3)는 그림 2 (b)의 G3에 해당한다. 표 1의 N1, P1에 대한 R3P4T중 하나를 그림 2 (a)에 적용하면 R3P4T의 한 패턴은 P1을 통과하는 VDD에서 F로의 경로를 연결하고 N1을 off시키므로써 F에서 GND로의 모든 경로를 차단하는 반면, 다른 패턴은 VDD에서 F의 경로는 모두 차단하고 N1을 경유하는 F에서 GND의 경로를 연결시킨다. 동일 R3P4T에 대한 그림 2 (b)의 동작은, 두 패턴 모두에 대해 test되는 FET쌍을 제외한 나머지 회로는 F에 영향을 미치지 않는 논리치(Nondominant Logic Value)를 갖는다. 즉, NAND gate인 G1은 논리 1로 고정되고 G4 역시 논리 1로 고정된다. R3P4T중 한 패턴은 N5를 on 시키고 P5를 off시키며 결국 P9를 on시키고 N9를 off시킨다. R3P4T의 다른 패턴에 의해서는 P5, N5, P9, N9의 동작이 반전되어 결국 이 R3P4T는 P5, N5의 R3P4T가 된다. 여기서 주목할 점은 이 R3P4T에 의한 P9, N9의 동작으로 이 R3P4T는 P9, N9의 R3P4T로도 사용가능하다는 점이다.

그림 1, 2, 3의 primitive-gate회로는 NAND, NOR, NOT gate만을 primitive gate로 가정한 것이다. 그러나 primitive gate의 종류를 확대시키면 결국 primitive-gate회로는, complex-gate회로를 분할 실현하여 cascade형태로 연결시킨 것으로 볼 수 있다. 지금까지의 설명을 토대로 한 이 일반화된 회로의 R3P4T에 대한 성질은, R3P4T의 한 패턴이 VDD에서 회로의 최종출력에 이르는 한 개의 경로(패턴의 order에 따라 두 개 이상의 경로를 연결할 수도 있으나 FUT의 견지에서 보면 단일 경로이므로)만 연결시키고 또다른 패턴은 이 경로를 차단하고 GND에서 최종출력의 경로를 연결한다(One Path Activation and Deactivation: OPAD). 또한 primitive-gate회로의 견지에서 보면 test되는 FET쌍을 포함하는 primitive gate에서 출력에 이르는 gate의 열을 동기, 비동기시키게 된다(One Gate Activation and Deactivation: OGAD). OPAD 또는 OGAD의 결과는, 동기 또는 비동기되는 경로 또는 gate 열 상의 모든 FET가 R3P4T의 두 패턴에 의해 on, off동작을 수행하므로 cascade시키는 FET쌍의 R3P4T는 cascade되는 FET쌍의 R3P4T가 된다는 것이다. 예로써 그림 2 (b)의 P1, N1쌍과 P2, N2쌍의 R3P4T는 표 1에 의해 각각 <3, 7>, <1, 3>이며, 이 두 R3P4T는 P6, N6쌍과 P9, N9쌍의 R3P4T 이기도 하다. 결과적으로 일반적인 CMOS회로의 모든 FET의 S-On고장과 S-Off고장을 검출하기 위해서는 cascade되는 FET들을 제외한 나머지 FET들에 대한 R3P4T만으로 충분하다.

표 1. 그림 2 (a)에 대한 R3P4T Set
Table 1. R3P4T Set for Fig.2 (a).

test되는 FET들	R3P4T들
N1, P1	<1, 9> <5, 13> <6, 14> <7, 15>
N2, P2	<3, 7>
N3, P3	<1, 3>
N4, P4	<10, 14>
N5, P5	<12, 14>
N6, P6	<2, 3> <8, 9> <10, 11> <12, 13>

III. R3P4T 생성 알고리즘

Cascade되지 않은 FET들에 대한 정보는 논리함수에 충분히 나타나 있다. 즉, 논리함수에 나타나 있는 변수명은 전체 회로에서 cascade되지 않은 한 쌍

의 FET에 해당한다. 그림 2 (a), (b)의 논리함수는, $F = A'(B + C) + (B' + C')D'$ (1)이며 각 변수는 한 쌍의 cascade되지 않은 FET에 해당한다. 따라서 일반적인 CMOS회로에 대한 완전한 R3P4T set은 회로의 구성을 참조하지 않고 논리함수만으로 생성가능하다. 본 논문에서는 논리함수를 재원으로 하는 R3P4T 생성 알고리즘을 제안하며, 이는 그림 4에 나타내었다. 이 알고리즘은 {R3P4T_GEN}의 주 procedure와 {T1_GEN}, {SER_T1_GEN},

{PAR_T2_GEN}의 세 subroutine으로 구성된다. R3P4T는 SP network를 기준으로 생성되며 T2는 {R3P4T_GEN}에서, T1은 {T1_GEN}에서 생성된다. [7]에서 밝힌 바와 같이 직렬연결된 두 FET는 동일 T2를 갖고 병렬연결된 두 FET는 동일 T1을 갖는다. 따라서 직렬 또는 병렬연결된 FET들은 {R3P4T_GEN}에서 검출하여 해당 패턴을 {SER_T1_GEN} 또는 {PAR_T2_GEN}에서 생성된다. 이 알고리즘은 주어진 논리함수에 대해 모든 가

PROCEDURE {R3P4T_GEN}

Objective : Test Generation for a CMOS Digital Circuit
Input : Logic Equation of Digital Function (OC.EQN)
Output : All Possible Sets of R3P4Ts

```
begin
  Make the OC.EQN unreduced form (NC.EQN);
  Number all P1m's and P1x's in NC.EQN;
  If OC.EQN is reduced form,
    Mark the reduced P1m or P1x on OC.EQN;
  for each P1m or P1x (OPli) in OC.EQN,
    begin
      Find ordered list of variables (OPli.VAR);
      for each variable (x) in OPli.VAR,
        begin
          Find all serial variables to x (Xi.SER.VAR);
          Find all parallel variables to x (Xi.PAR.VAR);
          Find all including P1m's or P1x's from NC.EQN;
          Get all vertices of including P1m's or P1x's (T2.LST);
          Call {T1_GEN};
          for each variable (y) in Xi.SER.VAR,
            Call {SER_T1_GEN};
          for each variable (z) in Xi.PAR.VAR,
            Call {PAR_T2_GEN};
        end
      end
    end
end
```

(a)

SUBROUTINE {T1_GEN}

```
begin
  for each vertex in T2.LST,
    begin
      Take it as XiT2s if OC.EQN is SOP(Sum of Product) form;
      Take it as XiT1s if OC.EQN is POS(Product of Sum) form;
      Change the corresponding bit to x;
      Check if it produces correct output value,
        if OC.EQN is SOP form, the value is 0;
        if OC.EQN is POS form, the value is 1;
      if correct,
        Put it into T1.LST;
      else,
        Delete it;
    end
  end
end
```

(b)

SUBROUTINE {SER_T1_GEN}

```
begin
  for each vertex in T2.LST,
    begin
      Take it as YiT2;
      Change the corresponding bit to y;
      Check if it produces correct output value,
        if OC.EQN is SOP, the value is 0;
        if OC.EQN is POS, the value is 1;
      if correct,
        begin
          Take it as YiT1;
          Put < YiT1, YiT2 > into R3P4T.LST for Yi;
          Remove y from OPli.VAR;
        end
      end
    end
end
```

(c)

SUBROUTINE {PAR_T2_GEN}

```
begin
  for each vertex in T1.LST
    begin
      Take it as ZiT1;
      Change the corresponding bit to z;
      Check if it produces correct output value,
        if OC.EQN is SOP, the value is 1;
        if OC.EQN is POS, the value is 0;
      if correct,
        begin
          Take it as ZiT2;
          Put < ZiT1, ZiT2 > into R3P4T.LST for Zi;
          Remove z from OPli.VAR;
        end
      end
    end
end
```

(d)

그림 4. R3P4T 생성 알고리즘

(a) {R3P4T_GEN} (b) {T1_GEN} (c) {SER_T1_GEN} (d) {PAR_T2_GEN}

Fig. 4. R3P4T Generation Algorithm.

(a) {R3P4T_GEN}, (b) {T1_GEN}, (c) {SER_T1_GEN}, (d) {PAR_T2_GEN}.

능한 R3P4T set을 생성하며 그림 2 (a)와 (b)에 대한 알고리즘의 결과는 표 1과 동일하다.

IV. R3P4T에 의한 gate 레벨의 test

서론에서 언급한 바와 같이 지금까지의 IC 설계 및 제조공정중의 test는 gate 레벨 설계에 대한 S-A 형태의 고장검출과 transistor 레벨의 S-On과 S-Off 고장검출이 따로 수행되었다. 따라서 test 생성 노력의 과다로 test비용의 증가를 초래, 결국 IC의 생산 원가 증가의 결과를 갖고 왔다. 본 논문에서는 test 비용 절감의 일 책으로 R3P4T를 제안하고 이를 gate 레벨 회로에 적용, gate 레벨의 S-A고장을 모두 검출함을 보여 두 레벨의 test를 단일화할 수 있음을 보인다.

CMOS transistor회로의 gate회로로의 modeling은 [15]의 Sutton이 제안한 방법을 따른다. Sutton의 방법에 의해 modeling된 그림 2 회로의 gate 레벨 회로는 그림 5에 나타내었다. Sutton의 modeling 방법은 FET의 직렬연결을 AND gate로, 병렬연결을 OR gate로 변환한다. PFET는 동작특성에 의해 입력에 NOT gate를 첨가하고 N-Net은 방전경로의 생성에 관계되므로 출력에 NOT gate를 부착한다. 그림 2 (a)의 P-Net과 N-Net은 그림 5 (a)와 (b)로 각각 변환되고 그림 2 (b)는 primitive gate들을 해당 gate로 대체하여 그림 5 (c)로 변환되었다. 그림에서 볼 수 있듯이 이 세 회로 모두 식 (1)과 그 부(complement) 논리식으로 구성될 수 있다.

III 장에서 제안된 R3P4T의 S-A고장 검출기능을 보이기 위해서 gate회로의 S-A고장 검출조건과 transistor회로의 S-Off고장 검출조건을 R3P4T의 적용을 가정하여 비교한다. 이는 표 2에 나타내었으며 S-A고장 검출을 위해서는 D-알고리즘의 Fault Provoking, Fault Propagation, Backward Line Justification의 세 절차를 분리하여 각각의 절차에 해당되는 R3P4T에 의한 S-Off고장 검출조건을 비교하였다. 표 2에 나타난 것과 같이 R3P4T에 의한 S-Off고장 검출조건은 OPAD 또는 OGAD에 의해 S-A 고장 검출조건과 동일함을 알 수 있고, 따라서 R3P4T set은 모든 gate 레벨 S-A고장을 검출할 수 있다. 예로써, 그림 5 (a)의 회로에 표 1의 R3P4T set을 적용하였을 때 각각의 패턴에 의해 검출되는 S-A고장을 표 3에 나타내었다. 여기서 i/k 는 선(line) i 의 S-A-고장을 뜻한다. 동일한 방법으로 그림 5 (b), (c) 회로 또한 표 1의 R3P4T set으로 test 가능하다. [13] 결과적으로 III 장에서 논리함수에

의해 생성된 R3P4T set은 transistor 레벨의 S-On 고장과 S-Off고장뿐 아니라 gate 레벨의 S-A고장 또한 검출할 수 있다. 따라서 R3P4T set은 IC 설계 및 제조과정의 gate 레벨과 transistor 레벨의 모든 test(또는 simulation)에 적용가능하다.

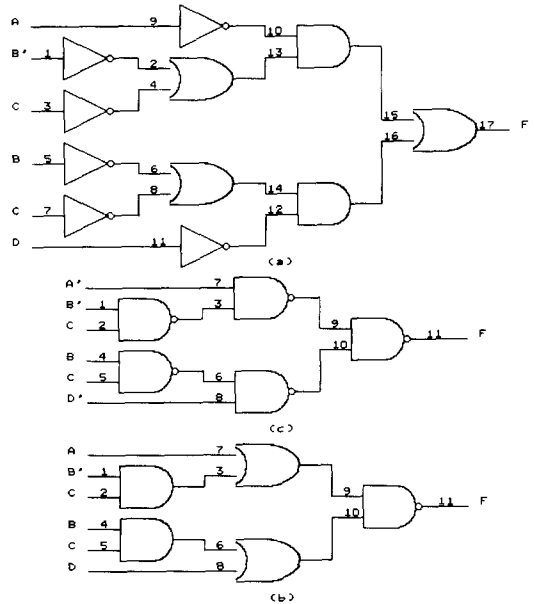


그림 5. 그림 2의 transistor회로에 대한 gate 레벨 model
(a) 그림 2 (a)의 P-Net에 대한 gate 회로
(b) 그림 2 (a)의 N-Net에 대한 gate 회로
(c) 그림 2 (b)에 대한 gate 회로

Fig. 5. Gate-Level Models for the Transistor Circuits in Fig. 2.
(a) Gate Circuit for P-Net of Fig. 2(a),
(b) Gate Circuit for N-Net of Fig. 2(a),
(c) Gate Circuit for Fig. 2 (b).

V. 결론

본 논문에서는 디지털 CMOS IC 설계 및 제조공정에서의 gate 레벨과 transistor 레벨 test를 확일화할 수 있는 test(R3P4T) set과 그 생성알고리즘을 제안하였다. 제안된 test set은 세 패턴을 하나의 test단위로 사용하며 생성 알고리즘의 입력재원으로 논리식이 사용되므로 gate회로나 transistor회로의 설계전에 test set을 생성할 수 있고 이 test set의 적용시 gate회로의 고장검출조건과 transistor회로의 고장검출조건이 동일함을 보임으로써, 제안된

표 2. Gate 레벨의 S-A고장 검출조건과 transistor 레벨의 S-Off고장 검출조건의 비교

Table 2. Comparison of Gate-Level S-A Test Conditions with Transistor-Level S-Off Test Conditions.

Table 2

gate 회로 (선 i의 S-A-X 고장)		transistor 회로 (선 i의 입력과 동일입력을 가진 FET i의 S-Off 고장)
Fault Provoking	1. S-A-1에 대해 0 적용 2. S-A-0에 대해 1 적용	* PFET 1. T1에서 1 적용 2. T2에서 0 적용 * NFET 1. T1에서 0 적용 2. T2에서 1 적용
Fault	1. 다른 모든 gate는 비 주(Nondominant) 논리치를 갖는다.	* Complex-gate 회로 1. 다른 모든 경로는 차단된다. * Primitive-gate 회로 1. 다른 모든 gate는 비 주 논리치를 갖는다.
Backward Line Justification	1. test되는 gate의 다른 모든 입력은 비 주 논리치를 갖는다. 2. 이 조건과 Fault Propagation 조건을 만족하기 위해 모든 선의 값이 규명된다.	* Complex-gate 회로 1,2. test되는 FET를 포함하는 경로의 다른 모든 FET는 on이다. * Primitive-gate 회로 1. test되는 gate의 다른 모든 입력은 비 주 논리치를 갖는다. 2. Cascade 회로의 입력들의 값 결정

test set은 gate회로와 transistor회로 모두에 적용 가능함을 보였다. 따라서 gate 레벨 test와 transistor 레벨 test를 분리 생성하던 종래의 방법과는 달리, 한 개의 test set으로 설계 및 제조공정의 대부분 test절차를 수행할 수 있어 test생성비용의 상당한 절감이 기대된다.

또한 본 논문에서 제안된 test set의 생성 알고리즘에 대한 복잡도(complexity)와 test 수행시간(application time)의 개선 정도를 보이기 위하여 기존의 방법들과 비교하여 표 4에 나타내었다. 이 표는 보다 정확한 비교를 위하여 검출되는 고장의 종류, 검출방법 test를 위해 부가되는 회로를 첨가하여 나타내었다. 복잡도와 test 수행 시간의 계산은 [14]의 방법을 기준으로 수행되었으며 각 방법에 대한 계산절차는 [13]에 상세히 나타나 있다. 표 4에 나타난 바와 같이 복잡도와 test 수행시간 모두에 있어 본 논문에서 제안된 방법이 가장 작은 값을 나타내었으며, 이는 본 논문의 방법을 사용 시 test절차의 확실확만 아니라 test생성 자체 비용의 절감과 test 수행시간 단축의 결과를 가져와 test를 위한 전체 비용의 상당한 감소를 꾀할 수 있음을 보여준다.

표 3. 표 1의 R3P4T들에 의한 그림 5 (a)의 gate레벨 S-A고장 검출

Table 3. Gate-Level S-A Fault Detection for Fig.5 (a) with R3P4Ts in Table 1.

Table 3

FETs	R3P4Ts	검출되는 gate 레벨의 S-A 고장
N1, P1	<1, 9>	1 3/1, 4/0, 9/1, 10/0, 13/0, 15/0, 17/0
		9 9/0, 10/1, 11/0, 12/1, 15/1, 16/1, 17/1
	<5, 13>	5 9/1, 10/0, 13/0, 15/0, 17/0
		13 9/0, 10/1, 11/0, 12/1, 15/1, 16/1, 17/1
	<6, 14>	6 1/1, 2/0, 9/1, 10/0, 13/0, 15/0, 17/0
		14 5/0, 6/1, 7/0, 8/1, 9/0, 10/1, 14/1, 15/1, 16/1, 17/1
	<7, 15>	7 1/1, 2/0, 9/1, 10/0, 15/0, 17/0
		15 9/0, 10/1, 15/1, 16/1, 17/1
N2, P2	<3, 7>	3 1/0, 2/1, 3/0, 4/1, 11/0, 12/1, 13/1, 15/1, 16/1, 17/1
		7 1/1, 2/0, 9/1, 10/0, 13/0, 15/0, 17/0
N3, P3	<1, 3>	1 3/1, 4/0, 9/1, 10/0, 13/0, 15/0, 17/0
		3 1/0, 2/1, 3/0, 4/1, 11/0, 12/1, 13/1, 15/1, 16/1, 17/1
N4, P4	<10, 14>	10 5/1, 6/0, 11/1, 12/0, 14/0, 16/0, 17/0
		14 5/0, 6/1, 7/0, 8/1, 9/0, 10/1, 14/1, 15/1, 16/1, 17/1
N5, P5	<12, 14>	12 7/1, 8/0, 11/1, 12/0, 14/0, 16/0, 17/0
		14 5/0, 6/1, 7/0, 8/1, 9/0, 10/1, 14/1, 15/1, 16/1, 17/1
N6, P6	<2, 3>	2 5/1, 6/0, 11/1, 12/0, 14/0, 16/0, 17/0
		3 1/0, 2/1, 3/0, 4/1, 11/0, 12/1, 13/1, 15/1, 16/1, 17/1
	<8, 9>	8 11/1, 12/0, 16/0, 17/0
		9 9/0, 10/1, 11/0, 12/1, 15/1, 16/1, 17/1
	<10, 11>	10 5/1, 6/0, 11/1, 12/0, 14/0, 16/0, 17/0
		11 11/0, 12/1, 15/1, 16/1, 17/1
	<12, 13>	12 7/1, 8/0, 11/1, 12/0, 14/0, 16/0, 17/0
		13 9/1, 10/1, 11/1, 12/1, 15/1, 16/1, 17/1

표 4. 알고리즘의 복잡도와 test 수행시간의 비교

Table 4. Comparison for Algorithm Complexity and Test Application Time.

Table 4

방법	검출가능한 고장	검출방법	부가회로	복잡도 (Complexity)	적용시간 (Application Time)
본 논문	S-On, S-Off	S-Off: 전입	-	n^2	$3n$
	S-A	S-On: 전류			
[9]	S-Off	S-Off: 전입	-	$4n^2$	$4n$
[3]	S-Off	S-Off: 전입	-	$2n^2$	$4n$
[5]	S-Off	S-Off: 전입	-	$4n^2$	$4n$
[7]	S-Off	S-Off: 전입	4 FETs 2 입력	$2n^2$	$4n$
[8]	S-On, S-Off	Buffer S-On: 전류 그 외: 전입	4 FETs 2 입력	$2n^2$	$6n$

n : 한 network내의 FET 수

參考文獻

- [1] US Market Report, "Looking for a Soft Landing after Hypergrowth," *Electronics*, pp 54-106, Jan. 1989.
- [2] R. L. Wadsack, "Fault Modeling and Logic Simulation of CMOS and nMOS Integrated Circuits," *The Bell System Technical Journal*, pp 1449-1474, May-June, 1978.
- [3] Kwang-Wei Chiang, Zvonko G. Vranesic, "On Fault Detection in CMOS Logic Networks," *IEEE Design Automation Conference*, pp 50-56, 1983.
- [4] Sudhakar M. Reddy, et. al., "Robust Tests for Stuck-Open Faults in CMOS Combinational Logic Circuits," *IEEE FTCS* pp 44-49, 1984.
- [5] Chantal Vivier, Georges Fournie, "Automatic Modeling of MOS Transistor Networks for Test Pattern Generation," *IEEE Test Conference*, pp 340-349, 1986.
- [6] R. Chandramouli, "On Testing Stuck-Open Faults," *IEEE FTCS*, pp 258-265, 1983.
- [7] Sudhakar M. Reddy, Mudhakar K. Reddy, "Testable Realizations for FET Stuck-Open Faults in CMOS Combinational Logic Circuits," *IEEE Trans. Computers*, Vol.C-35, N0.8, pp 742-754, August 1986.
- [8] Dick L. Liu, Edward J. McClusky, "Designing CMOS Circuits for Switch-Level Testability," *IEEE Design & Test*, pp 42-49, August 1987.
- [9] Sunil K. Jain, Vishwani D. Agrawal, "Test Generation for MOS Circuits using D-Algorithm," *IEEE Design Automation Conference*, pp 64-70, 1983.
- [10] Yacoub M. El-Ziq, Stephen Y. H. Su, "Fault Diagnosis of MOS Combinational Networks," *IEEE Trans. Computers*, vol.C-31, no.2, pp 129-139, Feb. 1982.
- [11] Steven D. Millman, Edward J. McClusky, "Detecting Stuck-Open Faults with Stuck-At Test Sets," *IEEE Custom Integrated Circuits Conference*, pp 22.3.1-22.3.4, 1989.
- [12] Kuen-Jong Lee, Melvin A. Bruer, "Design and Test Rules for CMOS Circuits for Facilitate IDDQ Testing of Bridging Faults," *IEEE Trans. Computer-Aided Design*, vol.11, no.5, pp 659-670, May 1989.
- [13] Dong-Wook Kim, "CMOS Digital Circuit Test Generation for Transistor-Level and Gate-Level Implementations," ph.D. Dissertation, Georgia Institute of Technology, July 1991.
- [14] Sreejit K. Chakravarty, "On the Complexity of Computing Tests for CMOS Gates," *IEEE Trans. Computer-Aided Design*, vol.8, no.9, pp 973-980, 1989.
- [15] Amar Mukherjee, *Introduction to nMOS & CMOS VLSI Systems Design*, Prentice-Hall, Englewood Cliffs, pp 62-75, 1986.

著者紹介



金 東 郁(正會員)

1983년 2월 한양대학교 전자공학과 졸업
1985년 2월 한양대학원 전자공학과 졸업
1991년 9월 Georgia Institute of Technology(Georgia 공과대학) Electrical Engineering (전기과)에서 박사학위 취득. 1992년 3월 부터 현재 광운대학교 전자재료공학과 재직중 주관심분야는 Digital VLSI System, VLSI Testability 등임.