論文 89-26-6-18

Domino CMOS NOR-NOR Array Logic의 Testable Design에 관한 연구

(A Study on Testable Design and Developement of Domino CMOS NOR-NOR Array Logic)

李 仲 鎬,** 趙 相 福,* 鄭 天 錫*

(Joong Ho Lee, Sang Bock Cho and Cheon Seok Jung)

要 約

본 논문에서는 CMOS 및 domino CMOS의 특징과 PLA등 array logic의 특징을 동시에 살리면서 동작특성이 좋고 집적도가 높으며 테스트 생성이 쉬운 domino CMOS NOR-NOR array logic의 설계방식을 제안하였다. 이 방식은 pull-down 특성을 개선하여 기생 커패시턴스의 문제점을 해결하며 간단한 부가회로를 사용하여 회로내의 모든 고장들을 검출할 수 있도록 한 testable design 방식이다. PLA의 적항군의 개념 및 특성 행렬을 이용한 테스트 생성 알고리듬과 절차를 제안하였고 이를 PASCAL 언어로 실현하였다. 또한 SPICE 및 P-SPICE를 이용하여 본 설계방식에 대한 검증을 행하였다.

Abstract

This paper proposes Domino CMOS NOR-NOR Array Logic design method which has the same as characteristic of CMOS and Domino CMOS in Array Logic like PLA, good operation feature, high desity, easy test generation. This testable design method can detect all of faults in the circuit using simple additional circuit and solve the parasitic capacitance problem by improving the pull-down characteristics. A Test generation algorithm and test procedure using concept of PLA product term and personality matrix are proposed, and it was implemented in PASCAL language. This design method is verified by SPICE and P-SPICE simulation.

*正會員, **準會員, 蔚山大學校 電子 및 電算機工學科
(Dept. of Elec. & Comp. Eng., Ulsan Univ.)

接受日字 : 1989年 2月 16日
(※ 본 연구는 1986년 한국과학 재단의 신진연구비 지원에 의해 연구되었음.)

I. 서 론

최근 VLSI 설계에 규칙적인 구조로 임의의 논리함수의 즉각적인 실현이 가능하며, 설계시간과 노력이 감소되고, 변경이 용이하게 이루어 질 수 있는 매우 효과적인 수단으로 PLA(programmable logic array), PPL(path programmable logic) 등과 같은 어

레이 로직이 널리 사용되고 있다.[1~3] PLA구성 방법으로 비교적 낮은 전력소모, 속도의 개선, 소프트에러가 적음등의 장점으로 인하여 CMOS가 점차 VLSI의 중요한 구성요소로 등장하고 있다.[4~6]

또한 타임 스큐의 문제점이 해결되고 칩 에리어의 축소, 속도가 빠르다는 점등의 장점을 갖고 있는 domino CMOS회로 구성방식이 최근에 많이 사용되고 있다.[7~8]

그러나 domino CMOS NAND-NAND PLA에서는 AND 어레이의 각 로우와 OR 어레이의 각 칼럼이 하나의 domino CMOS 기본구조를 형성해야 하므로 클락킹 게이트 및 인버터의 수가 너무많이 들게 된다. 따라서 상기의 어레이 로직의 특징과 CMOS 및 domino CMOS의 여러가지 장점을 동시에 살릴 수 있는 설계방식의 필요성이 증대되어 왔다.

본 연구에서는 어레이 로직과 CMOS 및 domino CMOS 회로의 특징을 살리면서 칩면적의 감소 및 속도개선 등 여러가지 장점을 가지며 design for testablity에 입각한 새로운 domino CMOS array logic의 testable design방식을 제안하였다. 이에 대한 검증을 위하여 SPICE 및 P-SPICE 시뮬레이션을 행하였다. 또한 제안된 설계방식에 대해 PLA의 특성 행렬과 적항군의 개념을 이용한 새로운 테스트 생성 알고리듬을 제안하였고 이를 PASCAL로 실현하여 부가회로까지 포함한 회로내의 모든 고장들을 검출할 수 있는 고장 시뮬레이터를 개발하였다.

Ⅱ. Domino CMOS logic의 동작과 특성

Domino CMOS 회로는 그림 1과 같이 함수형성을 위한 NMOS 함수블럭과 클락킹 게이트, 그리고 인버터로 구성되어 있다.

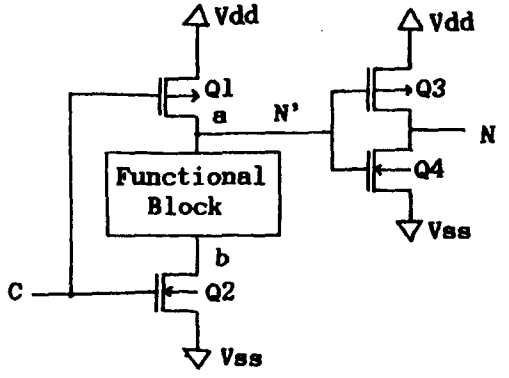


그림 1. Domino CMOS 회로의 기본구조
Fig. 1. Basic structure of domino CMOS circuit.

그림 1과 같은 기본구조에서 클럭C가 0일 때 Q2가 ON되어 N'은 1이 되고, 클럭C가 1일 때는 Q1이 ON되어 함수블럭 F에 의해서 N'이 0 또는 1로 결정된다. N'에는 출력이 N인 CMOS 인버터가 연결되어 있고, 따라서 클럭C가 0일 때, 출력 N은 항상 0이 된다. 클럭C가 1로 될 때 함수 F가 true이면 a와 b 사이에 path가 형성되어, N'은 F와 Q1에 의해 0이 된다. 이 때 Q3가 ON되고 Q4가 OFF되어, 출력 N은 0에서 1로 변화하게 된다. 즉 함수블럭이 true이고 클럭 C가 1일 때만 변화가 일어나고 그외에는 항상 0으로서 domino CMOS logic의 속도는 논리값 1이 전파되는 시간에 의해서 결정된다.

Domino CMOS logic의 형태는 각 함수의 길이가 길어질 수록 입력함수가 통과하기 위한 트랜지스터의 갯수가 많아진다. 각 함수의 트랜지스터 연결 방법은 직렬이므로 직렬연결로 인해 각 트랜지스터 사이의 기생커패시터 현상은 무시할 수 없다. 즉 domino CMOS logic의 구조적 특징으로 인해 precharge될때는 여러단의 커패시터 효과는 출력시간의 지연시간 문제에 지대한 영향을 미친다. 이로 인해 precharge되는 시간과 evaluate 시간의 불균형이 심각해지면 클럭이 일정시간 동안 1의 상태로 트리거되는 동안 evaluate 트랜지스터의 출력을 얻을 수 없게 되고 따라서 원하는 출력을 얻을 수 없게 된다. 다행히 클럭이 1상태 동안 evaluate 트랜지스터가 동작되어 출력을 얻었다 해도 응답속도가 늦어진다는 문제점이 있다.

Domino CMOS logic은 CMOS와 같이 전력소모가 적을 뿐 아니라, 칩면적이 작아서 집적도가 높고, 구조적 특징으로 인해 하나의 클럭 펄스에 의하여 함수값을 얻을 수 있으므로 타이밍 문제에 있어서 다른 다이내믹 회로보다 안정하며, 응답속도 또한 1.5에서 2배정도 빠르다는 잇점이 있다.

Ⅲ. Domino CMOS NOR-NOR array logic의 설계

1. Domino CMOS NOR-NOR array logic의 기본 회로

Domino CMOS NOR-NOR array logic의 함수 실현에서 하나의 product term에 대한 기본 회로 구성은 그림 2와 같다. 클럭이 0이면 Q1이 on되고, N'에 precharge되어 1이 되지만 Q3도 on되어 N은 1이므로 Z는 0이 된다. 클럭이 1이고 함수 입력이 true이면 N'이 1, 뒷단의 함수 block이 on되고, Q4도 on되어 N은 0, Z는 1이 된다. 그런데 기존의 logic방식은 앞단의 함수블럭이 false이고, 클럭이 0에서 1로 될 경우 glitch나 hazard 문제가 발생될 수 있는데, 이

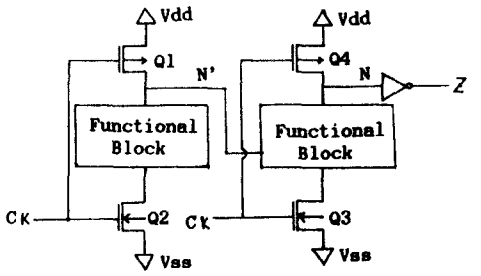


그림 2. Domino CMOS NOR-NOR array logic의 기본회로 구성

Fig. 2. Basic structure of CMOS NOR-NOR array logic.

logic방식에서는 first block의 function도 병렬로 연결되어 있으므로 이 문제를 해결할 수 있다. glitch 문제는 클럭이 0에서 1로 변하는 순간 N'에 precharge된 값이 discharge되는 시간에 의해 결정되는데, N'이 N보다 discharge되는 시간이 빠르면 해결할 수 있다. 이 discharge되는 속도는 앞단의 부하에 의해 결정되는데, first level의 부하를 second level의 부하보다 작게 해주면 된다. 제안된 회로는 first level과 second level의 discharge되는데 통과해야 할 트랜지스터 수가 똑같으므로 앞단의 부하를 작게 해주는 것은 충분히 가능하다.

2. Domino CMOS NOR-NOR array logic의 설계

입력함수들이 많아짐에 따라 domino CMOS의 NAND-NAND 구조로 인한 함수들의 각 소자들의 직렬연결이 불가피하다. 그로인해 출력에 지연시간문제 때문에 원하지 않는 출력이 나타날 수 있는데, 이러한 경우를 방지하기 위해서 모든 소자를 병렬연결시켜 NOR-NOR 구조로 설계한다.

〔정의 1〕 입력가능한 decoder 수를 y(입력가능한 최대의 수)라 놓고 각 입력을 X라 놓고, 여러 입력들의 조합을 Py라 놓으면 이것은 출력의 함수가 된다.

$$Py = X1' + X2' + \cdots + Xy' = X1X2\cdots Xy$$

정의1에 의해 가능한 출력(F)을 행렬식으로 표시하면 아래와 같다.

$$F1 = P11 + P12 + \cdots + P1y$$
$$F2 = P21 + P22 + \cdots + P2m$$
$$\cdots\cdots\cdots\cdots$$
$$Fn = Pn1 + Pn2 + \cdots + Pnk$$

여기서 m<=y, k<=y이고 n은 출력가능한 최대의 수 즉 true수이다.

〔정의 2〕 위와 같이 표현된 함수에서 동일한 열(column)에 속하는 적항들을 하나의 적항군이라 한다. 이 때 위의 함수는 Y개의 적항선을 갖는다. Pmn은 Py의 조합에 의해 이루어 진다.

위와 같은 개념에 입각하여 domino CMOS array logic의 testable design 방식은 다음과 같다.

1) 각 출력함수는 각각 하나의 CMOS-domino logic의 함수 블럭내에 실현된다. 즉 함수 블럭은 NMOS의 NOR logic으로 적항선을 구성하여 병렬연결하며, 전체적으로 array구조를 이룬다.
2) 함수내의 동일한 적항들은 동일군에 둔다. 예를 들면 F1과 F2에 동일 적항이 있을 경우, P11=Pn1 또는 P12=P21가 되도록 같은 열에 둔다.
3) 첫번째 적항군부터 차례로 적항선을 구성한다.
4) 각 출력함수에 해당하는 적항선들을 각각 병렬연결하여, 함수 블럭을 구성하고, CMOS-domino logic으로 연결한다.
5) 같은 출력군에 해당하는 함수들은 한군으로 묶어서 위와 똑같이 한다.
6) 출력은 PMOS와 함수사이에서 연결된다.

〈설계방식에 따른 예〉

X=3이고, n=4인 입력과 출력을 가지는 함수를 domino CMOS NOR-NOR 회로로 그림 3에 부가회로와 함께 나타내었다. 아래는 그 함수를 나타낸 것이다.

$$F1 = X1X2 + X1X2'X3'X4$$
$$F2 = X1' + X2'X3$$
$$F3 = X1X2 + X1X4'$$

여기서 각 적항선은 어레이 구조로 구성되어 있어 PLA의 AND 어레이와 같은 기능을 수행한다. 또한 각 적항선은 병렬로 연결되어 OR logic을 실현시킴으로서, PLA의 OR 어레이 기능을 대신한다. 그래서 domino-CMOR NOR-NOR array logic의 구조는 AND-OR logic의 PLA와 동일한 기능이 실현될 수 있다.

그림 3 에서 S1과 S2는 shift register로써 전체회로를 테스트하기 위한 부가회로로 첨가되었다. 그림 3 의 회로는 그림 2 의 기본구조를 가진다. S1, S2옆의 클럭이 들어가는 line의 N, P형 소자는 그림 2 의 Q1, Q2에 해당되고 second level도 마찬가지로 구성했다. S1, S2에 의해 제어되는 pass 트랜지스터는 NMOS를 사용했는데, 이는 S1, S2의 신호가 1일 때 동작하기 위해서 이다.

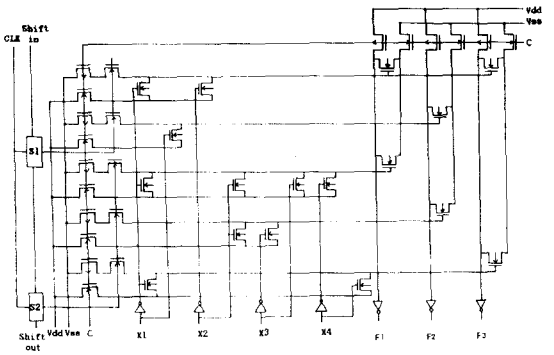


그림 3. Domino CMOS NOR-NOR 회로의 testable design

Fig. 3. Testable design for domino CMOS NOR-NOR circuit.

Ⅳ. Domino CMOS NOR-NOR 구조의 지연시간과 특징의 정량적 분석

1. 지연시간의 정량적 분석

입력함수들의 각 트랜지스터들을 병렬연결시킴으로 인해 각 함수의 길이가 아주 길어져도 트랜지스터 한개에 해당하는 커패시터효과 밖에 나타나지 않으므로 트랜지스터의 길이가 길어짐으로 인한 커패시턴스 현상을 아주 효과적으로 제거할 수 있게 된다. 그렇게 됨으로 인해 정상적인 동작을 얻을 수 있고 나아가서 더욱 빠른 속도의 출력을 얻을 수 있게 된다. 아래에 정량적인 지연시간을 계산했다.

〔정의 3〕 입력이 m개 일 때 P형소자 한개에 해당하는 최소의 상승시간을 Tr, N형 MOS소자 전체에 해당하는 소자의 하강시간을 Tf라 한다. 여기서 트랜지스터의 기생 커패시턴스 효과는 무시한다.

- 정의 3 에 의해 Domino CMOS NAND-NAND logic의 전체지연시간;

  .Tr = Rp(Cd+C1)

  Rp = 4/(BpVdd) : P-디바이스의 저항

  Bp : PMOS 트랜지스터의 게이트 이득요소

  Cd = 단위 드레인 면적의 커패시턴스

  C1 = 게이트의 기타 로드 커패시턴스(라우팅과 팬아웃)

  .Tf = mRn(Cd+C1)

  Rn = 4/(BnVdd) : N-디바이스의 저항

  Bn : NMOS 트랜지스터의 게이트 이득요소

- Domino CMOS NOR-NOR logic의 전체지연시간;

  .Tr = Rp(Cd+C1)

  .Tf = Rn(Cd+C1)

위에서 보는 바와 같이 상승시간은 NOR-NOR logic이나 NAND-NAND logic이나 같음을 알 수 있다. 왜냐하면 domino CMOS logic의 상승시간을 발생시키는 트랜지스터는 PMOS 소자 한개 뿐이기 때문이다. 그리고 하강 시간은 NOR 구조가 NAND 구조보다 m배 나 응답 속도가 빠르다는 것을 알 수 있다. 왜냐하면 NAND 구조에서 함수의 각소자 길이가 m개 이면 NOR 구조는 m개의 각소자를 병렬연결 시킴으로 인해 소자 한개에 해당되는 커패시턴스 효과를 내기 때문이다.

2. 기존의 구성방식들과의 비교

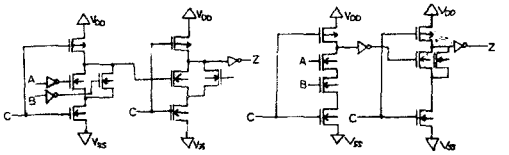
기존의 구성방식들과 제안된 구성방식의 기본형태를 살펴보면 그림 4 와 같다. 그림 4 의 예는 모두 P-LA에 적용할 수 있는 회로로써 기본 logic circit의 예를 보였다. 그림 4 에서 알 수 있듯이 입력단의 트랜지스터 수는 모두 같으므로 정량분석 대상에서 생략하였다. 출력이 m개일 때 출력 한개당 인버터가 들어가므로 인버터 갯수가 m, 출력level의 인버터를 제외한 트랜지스터 갯수를 3m, 적항선의 갯수를 k, 그리고 함수들의 중복이 d일 때 분석은 표1 과 같다. 또한 각 구성방식과 domino CMOS NOR-NOR logic방식과의 특성 비교는 표 2 와 같다.

Ⅴ. Domino CMOS NOR-NOR logic에 대한 Testable Design

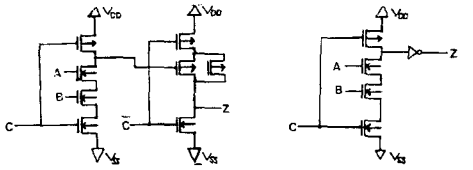
1. Domino CMOS 회로의 고장모델

- 부가회로에서의 고장

1. 쉬프트 레지스터의 stuck-at고장



(a) Domino CMOS NOR-NOR logic　　(b) Domino CMOS NAND-NAND logic



(c) Simplified domino CMOS logic　　(d) CMOS domino logic array

그림 4. 기존의 구성 방식들과 본 방식과의 비교
Fig. 4. Comparison with each design structure.

표 1. Domino CMOS NOR-NOR logic의 정량 분석
Table 1. Quantitative analysis for domino CMOS NOR-NOR logic.

Array / Logic 정량 분석	출력 1개만 고려		출력 m개일 때	
	인버터 수	인버터 제외 TR수	적항선의 수	인버터 포함 총 TR수
Domino CMOS NOR-NOR logic	1	3	k-d	m+3m−d (1+3)
Domino CMOS NAND-NAND logic	2	3	k-d	2m+3m−d (2+3)
Simplified Domino CMOS logic	0	3	k−d	3m−3d
CMOS Domino logic array	1	0	k	m

2. 트랜지스터의 단선고장(stuck-open)
3. 트랜지스터의 도통고장(stuck-on)

.Pull-up 및 pull-down 회로의 고장

4. stuck-on 고장
5. 첫번째 단의(입력단) n형 트랜지스터고장 (stuck-open)
6. 첫번째 단의(출력단) p형 트랜지스터고장 (stuck-open)

표 2. 각 구성 방식들과의 특성 비교
Table 2. Charactristic comparison with each structure.

	Domino CMOS NAND-NAND logic	Simplified domino -CMOS (SDC)	CMOS domino logic array (CLA)
Domino CMOS NOR-NOR logic	·하강시간 개선 ·인버터단이 빠지므로 ·전력소모 적다 ·집적도 낮다 ·속도개선	·하강시간 개선 ·클럭조절이 제안되는 방식보다 복잡하다 ·기생 커패시턴스 현상 방지	·하강시간 개선 ·중복되는 함수가 많아지면 중복되는 모두를 설계해야 한다는 단점을 보완

7. 두번째 단의(출력단) n형 트랜지스터고장 (stuck-open)
8. 두번째 단의(입력단) p형 트랜지스터고장 (stuck-open)
9. Stuck-at고장(출력단)

.array에서의 고장

10. 입력단의 stuck-at고장
11. missing device고장(트랜지스터가 존재해야 될 부분에 없는것)
12. extra device고장(트랜지스터가 없어야 될 부분에 있는것)
13. 트랜지스터의 단선고장(stuck-open)
    N형 트랜지스터의 고장(입력, 출력단)
    P형 트랜지스터의 고장(입력, 출력단)
14. 트랜지스터의 도통고장(stuck-on)
    N형 트랜지스터의 고장(입력, 출력단)
    P형 트랜지스터의 고장(입력, 출력단)

2. 함수실현을 위한 특성행렬

임의의 함수를 실현한 domino CMOS NOR-NOR logic의 PLA 특성행렬 개념을 다음과 같이 정의한다.

〔정의 4〕 n개의 입력과 k개의 적항선을 가진 domino CMOS NOR-NOR logic에 대하여 다음과 같은 요소들로 구성되는 (k*n) 행렬을 domino CMOS NOR-NOR logic의 입력 특성행렬이라 한다.

.X(i, j) = 1; true bit line과 적항선(i)의 교점에 소자 존재

.X(i, j) = 0; complement bit line과 적항선(i)의 교점에 존재

.X(i, j) = −; 어느 bit line과의 교점에도 존재하지 않을때

〔정의 5〕 k개의 적항선과 m개의 출력을 가진 domino CMOS에 대하여 다음과 같은 요소들로 구성되는 (k*m) 행렬을 domino CMOS NOR-NOR logic의 출력 특성행렬이라 한다.

.f (i, j) = 1; 적항선 (i)와 출력선 (j), 즉 Fj에 연결 되었을때.

.f (i, j) = 0; 출력선 F (j)에 적항선 (i)이 연결되지 않았을때.

여기서, $1 <= i <= k$, $1 <= j <= m$

정의 4와 정의 5에 의하여 앞의 그림 3의 회로에 대한 입력 및 출력특성행렬을 구하여 보면 표 3과 같다. 여기서 점선은 각 적항군을 구분한 것이다. 이렇게 적항군을 구분하는 것은 테스트를 빠르게 하고 또한 쉬프트 레지스터를 적절히 배치하기 위함이다. 아래 특성행렬에서도 보다시피 결과가 중복되지 않게 최대 적항들을 묶어서 하나의 적항군으로 둔다. 결과가 중복되는 적항을 같이 묶어 두면 고장 검출이 불가능 하기 때문이다.

표 3의 특성행렬을 테스트 하기 위한 특성행렬로 변환하기 위해 정의 4와 정의 5에 의해 출력 특성행렬의 0을 don't care (−)로 바꾼다. 이렇게 함으로써 해당 적항선이 연결되어 있는 출력만을 테스트 하기 위함이다. 그리고 하나의 적항선에 속하는 같은 형태의 함수들을 하나로 묶음으로써 동시에 테스트가 가능하다. 표 4에 변형된 입, 출력 행렬을 나타내었다.

3. 테스트 집합 생성 알고리듬

(절차 1) PLA의 입력 및 출력 특성행렬로 부터 변형된 특성행렬을 작성한다.

(절차 2) 다음 단계에 의하여 테스트 패턴 T (i, j)를 생성한다.

표 3. 그림 3의 회로에 대한 입력, 출력 특성 행렬

Table 3. Input, Output characteristic matrix for the network in fig. 3.

X1	X2	X3	X4	F1	F2	F3
1	1	−	−	1	0	0
0	−	−	−	0	1	0
1	1	−	−	0	0	1
1	0	0	1	1	0	0
−	0	1	−	0	1	0
1	−	−	0	0	0	1

표 4. 표 3의 수정 특성행렬

Table 4. Modified characteristic matrix for the table 3.

X1	X2	X3	X4	F1	F2	F3
1	1	−	−	1	−	1
0	−	−	−	−	1	−
1	0	0	1	1	−	−
−	0	1	−	−	1	−
1	−	−	0	−	−	1

〔단계 1〕 X (i)의 내용중 don't care가 있으면 don't care인 입력을 모두 1로 두어 테스트 입력 t1으로 한다. t1은 해당되는 product line에 연결된 소자들을 모두 on시켜 경로를 활성화하는 입력이며 이 때의 출력은 F (i)가 된다.

〔단계 2〕 X (i)의 내용중 don't care인 입력을 모두 0으로 두어 테스트 입력 t2로 한다. 이 때도 정상 출력은 F (i)가 되며 X (i)에 don't care가 없으면 t2는 필요없다.

〔단계 3〕 X (i)의 don't care가 아닌 부분에 대해 하나씩 그 보수값을 취하여 t3 입력 패턴으로 한다. 출력은 F (i)'가 된다.

X (i) : 입력 특성행렬의 i번째 행

F (i) : 출력 특성행렬의 i번째 행

F (i)' : F (i)의 내용중 1을 모두 0으로 바꾸고 don't care는 그대로 둔값

(절차 3) i=1부터 절차 2를 반복수행 하여 i=k' 까지 수행후 종료한다. 여기서 j는 i번째 행이 속해 있는 product line의 번호로 한다. $(1 <= i <= p)$

수정 특성행렬 (표 4)로 부터 알고리듬 수행 결과는 표 5와 같다. 위의 테스트 집합 생성 알고리듬으로 부터 고장 검출은 다음과 같다.

절차 2의 단계 1과 단계 2에서 missing 및 extra device 및 N, P형 트랜지스터의 stuck-on, stuck-open 고장을 검출할 수 있다. 표 5의 T (1, 1) 예를 보면 함수값을 입력하고 나머지에 1이나 0을 인가했을 때 입력함수가 missing device인 경우 출력 F1과 F2가 동시에 0이 된다. 그런데 출력단의 F1 혹은 F3 적항선에 missing device인 경우 둘중 하나만 1이 출력된다. 출력단에 extra device가 있는 경우 1이 출력된다. 그리고 N, P형 pull-up, pull-down 트랜지스터가 고

장인 경우 테스트 패턴을 바꾸어도 항상 이전의 상태를 유지할 것이며 이 때 stuck-on, stuck-open 고장을 검출할 수 있다.

단계 2에서 각 함수의 stuck-at 고장과 N, P형 트랜지스터의 stuck-on, stuck-open 고장을 검출할 수 있으며 절차 3에서는 전체회로에 대해 위와 똑같은 방법으로 고장을 검출할 수 있다.

표 5. 그림 3의 회로에 대한 테스트 집합
Table 5. Test set for the network in fig. 3.

		X1	X2	X3	X4	F1	F2	F3
T(1, 1)	t1	1	1	1	1	1	–	1
	t2	1	1	0	0	1	–	1
	t3	1	0	–	–	0	–	0
		0	1	–	–	0	–	0
T(2, 1)	t1	0	1	1	1	–	1	–
	t2	0	0	0	0	–	1	–
	t3	1	–	–	–	–	0	–
T(3, 2)	t1	1	0	0	1	1	–	–
	t3	0	0	0	1	0	–	–
		1	1	0	1	0	–	–
		1	0	1	1	0	–	–
		1	0	0	0	0	–	–
T(4, 2)	t1	1	0	1	1	–	1	–
	t2	0	0	1	0	–	1	–
	t3	–	1	1	–	–	0	–
		–	0	0	–	–	0	–
T(5, 2)	t1	1	1	1	0	–	–	1
	t2	1	0	0	0	–	–	1
	t3	0	–	–	0	–	–	0
		1	–	–	1	–	–	0

4. 테스트 절차

테스트를 용이하게 하기 위해 쉬프트 레지스터를 사용한 부가회로를 첨가시키는데 이 부가회로는 3개의 외부단자 즉 쉬프트 레지스터의 shift, in, shift out, 및 클럭 입력단자이며, 정상적인 동작시에는 쉬프트 레지스터의 모든 비트를 1로 둔다. 테스트시 쉬프트 레지스터는 선택하고자 하는 적항군의 입력에 해당하는 비트만 1로 두고, 나머지는 모두 0을 인가함으로써, 각 출력선에 대해서 각각 하나의 적항선만을 활성화 시킬 수 있다.

테스트 집합 생성 알고리듬에 의하여 생성된 집합으로 수행되는 전체 테스트 절차는 다음과 같다.

(1) 쉬프트 레지스터의 테스트

i) 쉬프트 레지스터의 갯수(p)만큼에 해당되는 신호 1을 shift in한 후, 1비트씩 shift out하여 p개의 1이 출력됨을 확인한다.

ii) i)과 반대로 0을 shift in한 후, 1bit씩 shift out하여 shift in된 만큼 0이 출력됨을 확인한다.

(2) 쉬프트 레지스터의 상태를 테스트하려는 비트에만 1을 가하고, 나머지는 모두 0의 상태로 둔다. 그런후에 테스트 패턴 T(i, j)의 입력을 n개의 외부입력에 인가하고, m개의 외부출력을 통해 관찰하여 테스트 패턴의 출력과 비교한다. 이때 클럭이 0이면 모든 출력은 0이고, 클럭이 1일 때 출력이 나타난다.

Ⅵ. 시뮬레이션 결과(output)

본 논문의 설계방법을 회로적으로 확인하기 위하여 여러가지 회로에 대한 SPICE 및 P-SPICE 시뮬레이션 결과중 하나의 예로 다음과 같은 함수를 부가회로까지 포함하여 본 논문의 방식으로 구성한 그림 5에 대해 P-SPICE 시뮬레이션한 결과는 그림 6과 같다.

$F1 = XY + XZ$

$F2 = XZ + YZ$

각 파형에 대한 설명은 그림 6의 아래 부분에 나타내었다. 여기서 주의해야 할 점은 D-FF(쉬프트 레

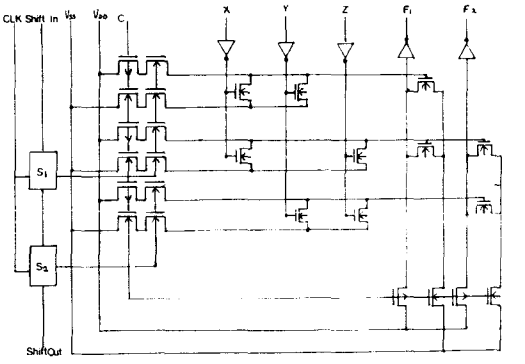
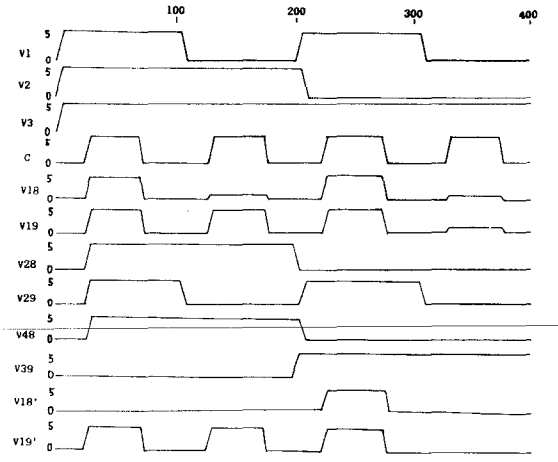


그림 5. Domino CMOS NOR-NOR array logic의 회로예
Fig. 5. Circuit example for domino CMOS NOR-NOR array logic.



[V1-V3(X, Y, Z) : 입력파형, C : 클럭, V18, V19(F1, F2) : 출력, V28, V29 : D-FF (D-flip flop)의 입력파형, V48, V39 : D-FF출력(테스트 입력을 형성, 즉 부가회로의 출력) V18', V19' : 부가회로의 출력을 입력으로 했을때 출력.]

그림 6. 부가회로를 포함한 시뮬레이션 결과
Fig. 6. Simulation result including the additional circuit.

지스터)의 출력이 본 회로의 테스트 데이타로 들어가는데, 결과 파형의 V48과 V49 파형처럼 주기가 엇갈려 있을 경우 순간 파형변화에서 처럼 지연시간 때문에 두 입력이 동시에 1인 상태가 잠시 발생한다. 이로 인해 테스트군 하나만 테스트해야 하는데 두단을 동시에 테스트하게 되는 오동작이 발생될 우려가 있지만 D-FF의 출력만을 임의로 버퍼로 통해서 시간을 지연시킬 수 있으므로 큰 문제는 생기지 않는다.

Ⅶ. 결 론

Domion CMOS 기술을 array logic에 적용하여 설계할 때 각 함수 구성에서 트랜지스터들의 직렬연결로 인한 기생 커패시턴스의 영향 때문에 지연시간의 불균형을 제거함으로서 하강시간이 m(입력 소자 갯수)배나 빨라져 출력의 응답이 안정되고 속도도 빨라지는 domino CMOS NOR-NOR array logic의 설계방식을 제안하였다. 아울러 부가회로를 사용하여 테스트가 용이 하도록 하였고, 이에 대한 테스트 집합 생성 알고리듬과 테스트 절차를 제안하였다. 이 방식은 비고전적인 stuck-on 고장 및 stuck-open 고장까지도 모두 고려함으로서 매우 높은 검출율을 기대할 수 있다. 그리고 SPICE 및 P-SPICE 시뮬레이션을 통해 회로 및 부가회로까지 정상 동작함을 보였다.

앞으로 domino CMOS 회로에 대한 기술이 발전함에 따라 회로동작의 지연시간 문제를 고려한 domino CMOS NOR-NOR array logic 방식이 많이 사용 되리라 기대된다.

參 考 文 獻

[1] Fujiwara, J., "A new PLA design for universal testability," *IEEE Trans. Compu.*, vol. c-33, pp. 745-750, Aug. 1984.

[2] Treuer, R., H. Fujiwara and V.K. Agrawal, "Implementing a built-in self test PLA design," *IEEE Design and Test of Computers*, Vol. 2, pp. 37-48, Apr. 1985.

[3] T. Ohtsuki, Layout Design and Verification, Advances in CAD for VLSI vol. 4, North-Holland, 1986.

[4] S.B. Cho, Y.C. Shin, I.C. Lim, "A test generation algorithm for CMOS Circuits," *Proc. IEEE 1985 Int. Symp. on Circuits and System, Kyoto, Japan*, pp. 1551-1554, June 1985.

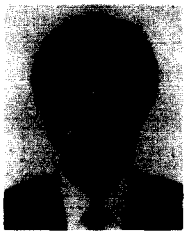
[5] D.J. Myers and P.A. Ivey, "A design style for VLSI CMOS," *IEEE Jour. of Solid State Circuits*, vol. Sc-20, no. 3, pp. 741-745, June 1985.

[6] S. Goto, Design Methodology, Advances in CAD for VLSI vol. 6, North-Halland, 1986.

[7] N. Weste and K. Eshraghian, Principles of CMOS VLSI Design, Addison-Wesley, 1985.

[8] E.J. Mc Cluskey, Logic Design Principles, Prentice-Hall, 1986. *

著 者 紹 介



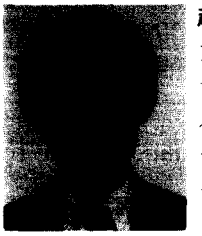
李 仲 鎬(準會員)

1965年 1月 23日生. 1988年 2月 울산대학교 전자 및 전산기공학과 졸업. 1988年 3月～현재 울산대학교 대학원 전자 및 전산기공학과 석사과정 재학중. 주관심분야는 Testable design 및 회로설계, VLSI의 System설계 등임.

●

鄭 天 錫(正會員)

1947年 6月 14日生. 1969年 2月 한국항공대학 항공전자공학과 졸업 공학사학위 취득. 1980年 9月 부산대학교 대학원 전자공학과 졸업 공학석사학위 취득. 1988年 8月 고려대학교 대학원 전자공학과 졸업 공학박사학위 취득. 현재 울산대학교 전자 및 전산기공학과 부교수. 주관심분야는 위성통신 및 마이크로 웨이브 통신 등임.



趙 相 福(正會員)

1955年 6月 10日生. 1979年 2月 한양대학교 전자공학과 졸업 공학사학위 취득. 1981年 2月 한양대학교 대학원 전자공학과 졸업 공학석사학위 취득. 1985年 2月 한양대학교 대학원 전자공학과 졸업 공학박사학위 취득. 1981年 2月～1986年 2月 광운대학교, 한양대학교 강사. 1986年 3月～현재 울산대학교 전자 및 전산기공학과 조교수. 주관심분야는 VLSI CAD 특히 VLSI Layout 설계 및 테스팅, Simulation등임.