

순서 CMOS Domino Logic Array의 설계 및 테스트

○ 박진관, 김윤홍, 정준모, 한석봉, 임인철
한양대학교

Design and Test of Sequential CMOS Domino Logic Array

○ J.K. Park, Y.H. Kim, J.M. Jung, S.B. Han, I.C. Lim
Hanyang University

ABSTRACT

This paper proposes a design method for SCLA(Sequential CMOS Domino Logic Array) using 1-level CMOS Domino Logic and Stable Shift Register Latch. Also an algorithm to generate a test sequence and a test procedure for the SCLA are presented.

The SCLA has advantages of low power consumption, high density and high speed, and performs hazard- and race-free logic operation, because of using SSRL(Stable Shift Register Latch)

By using the proposed test method, all of stuck-at, cross-point, stuck-on and stuck-open faults in SCLA are detected by short test sequence.

I. 서론

최근 LSI/VLSI 기술의 급속한 발전으로 칩의 집적도가 비약적으로 증가하고 있다. 특히 PLA(Programmable Logic Array)는 구조가 간단하고 규칙적이므로 임의의 논리회로의 즉각적인 실현과 변경이 가능하여 많이 이용되고 있다.[3, 4]

PLA의 구성방법은 대부분 바이폴라 및 NMOS를 이용하여 구성해왔으나, 최근에는 비교적 낮은 전력소모, 고집적도 등의 성결과 속도의 개선 및 soft-error가 적음등의 장점으로 인하여 CMOS가 점차 VLSI의 중요한 구성소자로 등장하고 있으며, 그 사용 범위가 확대됨에 따라 PLA에도 CMOS 기술이 채택되고 있다.[9, 11]

또한 CMOS의 특징을 가지면서 time skew의 문제점을 해결하고 기존의 CMOS 회로보다 칩 면적이 작고, 속도가 빠르며, NMOS 기술을 그대로 이용할 수 있는 장점을 가진 CMOS Domino Logic에 대한 연구가 활발히 행해지고 있다.[5, 6, 7]

본 논문에서는, 1-레벨의 구조로써 PLA와 동일한 기능을 수행하며 CMOS의 장점을 갖는 동시에, 테스트가 용이

한 CMOS Domino Logic Array(이하 CLA)[10]에 래치를 결합한, Sequential CMOS Domino Logic Array(이하 SCLA)의 설계방식을 제안한다.

즉, 조합논리회로 부분은 1-레벨의 CMOS Domino Logic 들을 사용함으로써, 전력소모가 매우 적고, 밀도가 높을 뿐 아니라, 속도 또한 빠르게 되며, PLA의 AND-OR logic 과 동일한 기능을 갖게 된다. 한편 순서논리 기능을 수행하기 위한 래치는 Level Sensitive 및 Scan Design 개념을 [1] 갖는 Stable Shift Register Latch(이하 SSRL)[2]를 사용하여 설계함으로써 회로의 동작이 각 소자 또는 회로의 ac 특성과 무관한 안정된 동작을 수행하게 되고 [8] 테스트 패턴 생성과 테스트가 매우 간단해진다.

또한 특성행렬(personality matrix)을 이용한 테스트 집합 생성 알고리즘 및 전체 테스트 절차를 제안한다. 이 방법에 의하면, 짧은 테스트 패턴 생성 시간 및 테스트 시퀀스에 의해 고전적인 stuck-at 고장 및 접점(cross-point)고장뿐만 아니라, CMOS 회로 특유의 stuck-on 고장과 stuck-open 고장까지도 검출할 수 있다.

II. SSRL의 논리회로 구성 및 동작

Level Sensitive 및 Scan Design 개념에 의한 Stable Shift Register Latch(SSRL)는 쉬프트 클럭 A, B 이외에 클럭 P에 의해 제어되는 L3 래치를 사용함으로써, 래치내의 값

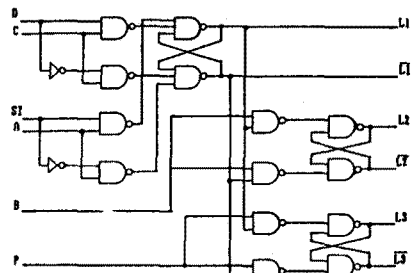


그림 1. Stable Shift Register Latch

을 임의로 선택할 수 있게 된다. 그림 1은 SSRL의 논리 회로를 나타낸다.

위의 회로는 다음의 성질을 갖는다.

1) 각 래치들은 어떤 특정한 값으로 제어(controllability)될 수 있으며, 또한 그 래치내의 값을 외부에서 관찰(observability)할 수 있다.

2) 순서논리회로에 대한 테스트는 조합논리회로에 대한 테스트로 귀착되므로 테스트가 훨씬 용이해진다.

3) CMOS로 구성된 조합논리회로에 대한 테스트 패턴 생성시, L3 래치로써 중간값을 제거하여 초기화 값의 변동을 막을 수 있다.

4) 회로의 정상동작은 논리회로의 지연(delay) 특성이나 과도특성에 영향을 받지 않는 안정된 동작을 수행한다. 이때, $T1 \geq T2$, $T1$: 클락 C와 P사이의 시간.

$T2$: 조합회로의 최대 지연시간.

III. 순서 CMOS Domino Logic Array의 설계

1. CMOS Domino 회로의 동작

CMOS Domino 회로는 그림 2와 같이, 함수형성을 위한 NMOS 함수블럭과 클락킹 게이트, 그리고 인버터로 구성되어 있다. 이러한 구조로 인하여, 일반 CMOS 회로보다 칩 면적이 작고 전력소모가 적을 뿐아니라, NMOS 기술을 그대로 이용할 수 있으며, 타이밍 문제에 있어서도 다른 다이내믹 회로보다 안정하게 된다.

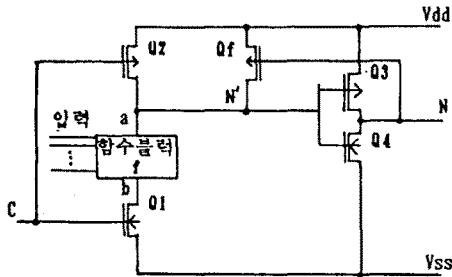


그림 2. CMOS Domino Logic의 기본구조

그림 2의 기본구조에서, 클락 C가 0일 때는 Q2가 ON되어 N'는 1이 되고, 클락 C가 1일 때는 Q1이 ON되어 함수블럭 f에 의하여 N'가 0 또는 1로 결정된다. N'에는 출력 N을 갖는 CMOS 인버터가 연결되어 있고 따라서 클락 C가 0일때 출력 N은 항상 0이 되며, 전체적으로 CMOS Domino Logic은 비반전 논리(noninverting logic) 회로이다.

2. 순서 CMOS Domino Logic Array의 설계

PLA는 ROM과 같은 array 구조로써, 다양한 기술로 임의의 조합논리회로 또는 순서논리회로를 간단하게 실현할 수 있다.

특히, Domino PLA[5]를 이용하여 임의의 논리함수를 실현할 경우, 그 출력함수들이 동일한 적항을 많이 포함하지 않는다면, CMOS Domino Logic의 특징으로 인하여 OR array의 논리회로는 그 존재가치가 상당히 감소하게 된다.

이와같은 경우에 OR array부분의 소자를 제거하고, 1-레벨의 CMOS Domino Logic과 래치를 사용하여 테스트가 용이한, 새로운 형태의 array 구조를 갖는 순서 PLA를 설계할 수 있다.

이 새로운 형태의 array 구조를 갖는, 순서 PLA의 설계에 대하여 다음과 같이 정의한다.

[정의 1] 1-레벨의 CMOS Domino Logic 들을 병렬로 배열하여, 각각의 NMOS 함수블럭들을 PLA의 AND array와 같은 array 구조로 구성하고 SSRL을 이용하여 순서논리기능을 갖도록 하며, 매우 적은 양의 부가회로를 사용하여 테스트가 용이하도록 설계한 논리회로를 Sequential CMOS Domino Logic Array(SCLA)라 한다.

그러면 다음과 같은 sum of products 형태로 표현되는 $m+n$ 개의 출력함수를 실현하는 SCLA의 설계방식을 제안한다.

$$D1 = P11 + P12 + \dots + P1p$$

$$D2 = P21 + P22 + \dots + P2q$$

$$\vdots$$

$$Dm = Pm1 + Pm2 + \dots + Pmr$$

$$F1 = Q11 + Q12 + \dots + Q1a$$

$$F2 = Q21 + Q22 + \dots + Q2b$$

$$\vdots$$

$$Fn = Qn1 + Qn2 + \dots + Qnc$$

$Pij(1 \leq i \leq m, 1 \leq j \leq p)$, $Qk1(1 \leq k \leq n, 1 \leq l \leq a)$; 귀환 입력 및 주입력의 지향. 단 $p \geq q \geq \dots \geq r$, $a \geq b \geq \dots \geq c$

[정의 2] 위와 같이 표현된 함수에서, 동일한 열(column)에 속하는 적항들을 적항군이라 한다. 이때, 위의 함수는 p 또는 a 개의 적항군을 갖게 된다.

[SCLA의 설계방식]

- (1) 각 출력함수는 각각 하나의 CMOS Domino Logic의 함수블럭내에서 실현된다. 즉, 함수블럭은 NMOS의 NAND logic으로 적항선을 구성하여 병렬로 연결하며, 전체적으로 array 구조를 이룬다.
- (2) 함수내의 동일한 적항 및 적어도 하나의 complement bit 또는 true bit가 같고 각 적항에 동일 입력선에 의한 complement bit와 true bit가 동시에 존재하지 않을 경우 동일 적항군으로 한다.
- (3) 첫번째 적항군의 적항부터 차례로 적항선을 구성한다.
- (4) 각 적항선의 끝에 하나의 트랜지스터를 추가로 직렬 연결하고, 각 적항군의 트랜지스터들을 공통 입력으로 묶어, 각 적항군의 입력에 p -bit 또는 a -bit의 쉬

프트 레지스터의 각 단을 연결한다.

(5) 각 출력함수에 해당하는 적항선들을 각각 병렬 연결하여 함수블럭을 구성한다.

(6) Level Sensitive 및 Scan Design 개념을 갖는 SSR을 사용하여 순서논리기능을 부가한다.

위의 설계방식에 의한 구체적인 예로써 다음과 같은 함수를 SCLA로 실현하여 그림 3에 나타내었다.

$$\begin{aligned}
 D1 &= \bar{X}1Q2 + X1\bar{X}2\bar{Q}1 \\
 D2 &= X3\bar{Q}1 \\
 F1 &= \bar{X}1Q1\bar{Q}2 + X1\bar{X}2\bar{X}3Q1 + X2X3\bar{Q}1 \\
 F2 &= X1X2X3Q1Q2 + X1\bar{X}2\bar{Q}1Q2 + X2\bar{Q}1Q2 \\
 F3 &= \bar{X}1Q1\bar{Q}2 + X1X3\bar{Q}1
 \end{aligned}$$

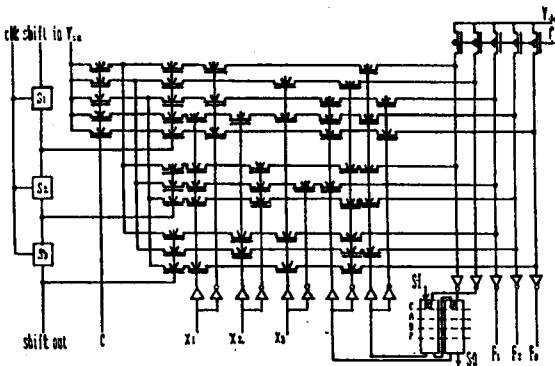


그림 3. Sequential CMOS Domino Logic Array

한편, $(y+z)$ 개의 입력, $(m+n)$ 개의 출력 및 k 개의 적항선을 가진 SCLA의 특성을 나타내기 위하여 PLA에서의 특성행렬(personality matrix)의 개념을 도입한다.

즉, array 부분에 대한 논리회로의 특성을 나타내는 $\{k * (y+z)\}$ 행렬을 입력특성행렬이라 하고, 각 적항선이 출력선에 연결된 상태를 나타내는 $\{k * (m+n)\}$ 행렬을 출력특성행렬이라 한다.

위의 개념에 따라, 예시한 그림 3의 SCLA에 대한 입력 및 출력특성행렬을 구하면 그림 4와 같다.

X1	X2	X3	Q1	Q2	D1	D2	F1	F2	F3
0	-	-	-	1	1	-	-	-	-
-	1	0	-	0	-	1	-	-	-
0	-	-	1	0	-	-	1	-	-
1	1	1	1	1	-	-	-	1	-
0	-	-	-	0	-	-	-	-	1
1	0	0	0	1	1	-	-	-	-
1	0	0	0	1	-	1	-	-	-
-	1	1	0	1	-	-	1	-	-
-	-	-	1	0	-	-	-	1	-
-	-	-	1	0	-	-	-	-	1

그림 4. 그림 3의 SCLA에 대한 입력 및 출력특성행렬

IV. 순서 CMOS Domino Logic Array의 테스트

1. SCLA의 고장모델

SCLA의 고장모델은 CMOS Domino Logic의 pull-up, pull-down 회로에서의 stuck-on 및 stuck-open 고장과 함수블럭, 출력선, 순서논리기능을 위한 SSR과 테스트를 위한 부가회로에서의 stuck-at 고장 및 함수블럭에서의 집적고장이 고려되어야 한다. 집적고장에는 missing device 고장과 extra device 고장이 있으며, stuck-open 고장이 발생할 경우 CMOS 회로 특유의 charge store 기능으로 인하여 전 상태값을 유지하게 되므로, 2개의 연속된 테스트 패턴이 필요하게 된다. 한편, stuck-on 고장이 발생할 경우 고장검출을 용이하게 하기위해 SCLA의 설계시 'p-dominant'가 되도록 한다고 가정한다.

2. 테스트 집합 생성 알고리즘 및 테스트 절차

SCLA의 고장검출을 하기위한 테스트 집합은 SCLA의 특성행렬로부터 구해질 수 있다. 그런데 SCLA의 테스트시, 적항군의 개념[설계방식(2)]을 이용하면 동시 테스트가 가능하다. 즉, 입력특성행렬에서 동일 적항군내의 동일 행은 하나로 통합하고, 동시에 출력특성행렬에서는 해당 행을 더하여 준다. 또한, 적항의 일부 bit가 같은 경우 입력특성행렬에서는 해당 행에 'v' 표시를 하고 출력특성행렬에서는 해당 행을 더하여 '('로 표시한다. 한편, 출력특성행렬에서는 모든 0을 '-'(don't care)로 바꾼다. 이것은 해당 적항선이 연결되어 있는 출력선만을 관찰하기 위해서이다.

또한 입력 및 출력특성행렬에서 Q1, Q2는 Scan In을 통해 입력을 시키고, D1, D2는 Scan Out에서 관찰을 하게 되므로 각각 S11, S12 및 SO1, SO2로써 나타낸다.

그림 5에 그림 4의 특성행렬을 변형시켜 나타내었다.

X1	X2	X3	S11	S12	SO1	SO2	F1	F2	F3
0	-	-	-	1	1	-	-	-	-
-	1	0	0	0	-	1	-	-	-
0	-	-	1	0	-	-	1	-	1
1	1	1	1	1	1	-	-	1	-
0	-	-	-	0	-	-	-	-	1
1	0	0	0	1	1	-	-	-	-
1	0	0	0	1	-	1	-	-	-
-	1	1	0	1	-	-	1	-	-
-	-	-	1	0	-	-	-	1	-
-	-	-	1	0	-	-	-	-	1

그림 5. 변형된 특성행렬

변형된 입력 및 출력특성행렬을 $\{k' * (y+z)\}$ 및 $\{k' * (m+n)\}$ 행렬이라 할때, 다음과 같은 테스트 집합 생성 알고리즘을 제안한다.

[테스트 집합 생성 알고리즘]

(단계 1) SCLA의 입력 및 출력특성행렬로부터 변형된 특성행렬을 작성한다.

(단계 2) 다음 단계에 의하여 테스트 패턴 $T(i, j)$ 를 생성한다. 여기서 j 는 i 번째 행이 속해있는 적항군의 번호이다. ($1 \leq j \leq p$ 또는 $1 \leq j \leq a$).

단계 2-1. $X(i), SI(i)$ 의 내용중 don't care가 있으면, don't care인 입력값을 모두 1(0)로 바꾸어 테스트 입력 $t1(t2)$ 로 하며, don't care가 없으면 $t2$ 를 생략한다.

이때의 정상출력은 $SO(i), F(i)$ 가 된다.

단계 2-2. $X(i), SI(i)$ 의 내용중, 적어도 하나의 complement bit(또는 true bit)가 같고 각 행에 동일 입력선에 의한 complement bit와 true bit가 동시에 존재하지 않을 경우 $t1$ (또는 $t2$)를 하나로 통합한다. 단 이 경우 $t1$ 과 $t2$ 는 서로 독립적이어야 한다.

단계 2-3. $X(i), SI(i)$ 의 내용중 don't care가 아닌 입력에 대하여 하나씩 그 보수값을 취한 패턴들을 테스트 입력 $t3$ 로 한다. 이때, $SI(i)$ 값은 $t2$ 와 동일한 것을 먼저 생성하며 첫번째 $t3$ 의 $SI(1)$ 의 내용이 다음 $t3$ 의 $SI(2)$ 에 쉬프트될 수 있도록 $t3$ 를 재구성한다. 이때, 정상출력은 $SO(i), F(i)$ 가 된다.

여기서, $X(i), SI(i)$; 입력특성행렬의 i 번째 행.

$SO(i), F(i)$; 출력특성행렬의 i 번째 행.

(단계 3) $i = 1$ 부터 절차 2를 반복하여 $i = k'$ 까지 수행한 후 종료한다.

위의 알고리즘에 의하여 생성된 테스트 집합의 일부를 표 1에 나타내었다. 생성된 테스트 집합에 의해 수행되는 SCLA의 전체 테스트 절차는 다음과 같다.

표 1. 그림 3의 SCLA에 대한 테스트 집합(일부)

Test pattern		input						output					
		X1	X2	X3	SI1	SI2	SO1	SO2	F1	F2	F3		
T(1,1)	t1	0	1	1	1	1	1	-	-	-	-		
	t2	0	0	0	0	1	1	-	-	-	-		
	t3	1	-	-	-	1	0	-	-	-	-		
T(4,1)	t1	1	1	1	1	1	-	-	-	1	-		
	t3	0	1	1	1	1	-	-	-	0	-		
		1	0	1	1	1	-	-	-	0	-		
		1	1	0	1	1	-	-	-	0	-		
T(7,3)	t1	1	1	1	0	1	-	-	1	(1)	(1)		
	t2	0	1	1	0	0	-	-	1	-	-		
	t3	-	0	1	0	-	-	-	0	-	-		
T(8,3)	t2	-	1	0	0	-	-	-	0	-	-		
	t3	-	0	0	0	1	-	-	0	-	-		
		-	1	-	1	1	-	-	-	0	-		
		-	1	-	0	0	-	-	-	0	-		

[테스트 절차]

(테스트 1) 쉬프트 레지스터의 shift in 단자를 통하여 p 또는 a개의 1(0)을 인가한 후 p 또는 a개의 1(0)이 출력됨을 확인한다.

(테스트 2) SSRL을 테스트한다. 즉, m개의 1(0)을 sc-

an in한 후, 1bit씩 shift out하여 m개의 1(0)이 scan out됨을 확인한다.

(테스트 3) 쉬프트 레지스터의 상태를 $S(j)$ 로 하고 테스트 패턴 $T(i, j)$ 의 입력을 인가하여, 그 출력과 테스트 출력을 비교한다.

이때, 클락이 0이면 모든 출력은 0이 되고, 1일때 출력이 나타난다.

여기서, $S(j)$ 는 j번째 bit만 1이고 나머지는 모두 0인 상태를 말한다.

SCLA내에 존재하는 모든 Stuck-at 고장, 점점고장, stuck-on 및 stuck-open 고장의 집합을 FS라 할때 다음과 같은 정리가 성립한다.

[정리] 생성된 테스트 집합에 의하여 SCLA의 테스트 절차를 수행하면, 어떠한 형태의 SCLA에 대해서도 집합 FS에 속하는 모든 고장을 검출할 수 있다.

(증명 생략)

변형된 입력특성행렬 $\{k' * (y + z)\}$ 에서 don't care가 아닌 모든 요소의 갯수를 Nx , don't care가 없는 행의 갯수를 D , 그리고 적어도 하나의 complement bit(또는 true bit)가 같고 각 적함에 동일 입력선에 의한 complement bit와 true bit가 동시에 존재하지 않는 경우의 수를 E , E 가 포함된 적함군의 수를 j 라 하면 전체 테스트 집합의 길이(LT)는 다음 식으로 표현된다.

$$LT = Nx + 2k' - D - E + j; E > j$$

여기서, 쉬프트 레지스터와 SSRL의 테스트는 제외하였다.

V. 결론

본 논문에서는 CMOS Domino Logic Array(CLA)와 L SSD개념을 갖는 SSRL을 사용하여, 순서 PLA와 동일한 기능을 수행하는 Sequential CMOS Domino Logic Array(SCLA)의 설계방식을 제안하였다. 제안된 SCLA는 CLA의 특징을 가지며, 회로의 동작이 그 구성소자 또는 ac특성과 무관한 안정된 동작을 수행하게 된다.

또한 매우 적은 부가회로를 사용하여 테스트가 용이하도록 설계하였으며, 이에 대한 테스트 집합 생성 알고리즘 및 테스트 절차를 제시하였다. 제시된 테스트 방법에 의하면, CMOS 회로에서의 초기화 값의 변동을 억제할 수 있으며, 짧은 테스트 패턴 생성시간 및 테스트 사이클스 $\{Nx + 2k' - D - E + j\}; E > j$ 에 의해, SCLA내의 stuck-at 고장, 점점고장, stuck-on 고장, stuck-open 고장을 모두 검출할 수 있다.

그리고, 제안된 SCLA를 기존의 TTL로 구성된 순서는 리회로(74LS160A)에 적용한 결과, SCLA의 설계방식 및 테스트 패턴 생성 알고리즘의 유용성을 입증할 수 있었다.

참 고 문 헌

1. E.B. Eichelberger and T.W. Williams, "A Logic Design Structure for LSI Testability", J. Design Automat. Fault Tolerant Comput., Vol. 2, pp. 165-178, May 1978.
2. S. Dasgupta, R.G. Walther, T.W. Williams, E.B. Eichelberger, "An Enhancement to LSSD and Some Applications of LSSD in Reliability, Availability and Serviceability," proc. International Symposium on Fault Tolerant Computing, June 1981, Maine, U.S.A.
3. Hideo Fujiwara and Kozo Kinoshita, "A Design of Programmable Logic Arrays with Universal Tests", IEEE Trans. on Computers, Vol. C-30, No. 11, pp. 823-828, Nov., 1981.
4. K.K. Saluja, K. Kinoshita and H. Fujiwara, "An Easily Testable Design of Programmable Logic Arrays for Multiple Faults", IEEE Trans. Comput., Vol. C-32, pp. 1038-1046, Nov., 1983.
5. David A. Hodges and Horace G. Jackson, "Analysis and Design of Digital Integrated Circuits", McGraw-Hill, 1983.
6. Vojin G. Oklobdzija and Predrag G. Kovijanic, "On Testability of CMOS-Domino Logic", 0731-3071/84/0000/0050, IEEE, pp. 50-55, 1984.
7. Neil H. E. Weste, Kamran Eshraghian, "Principles of CMOS VLSI Design-A Systems Perspective", Addison-Wesley, 1985.
8. 한서봉, "TEST 용이성을 고려한 LSI/VLSI 논리설계 방식과 그 응용에 관한 연구", 한양대학교 대학원 전자공학전공 공학석사 학위논문, 1983. 12.
9. 조상복, 임인철, "CMOS 회로의 테스트 생성 알고리즘", 대한전자공학회지, Vol. 21, No. 6, pp. 78-84, 1984. 11.
10. 박종로, 조상복, 임인철, "CMOS-Domino Logic Array (CLA)", 대한전자공학회 추계종합학술대회 논문집, vol. 8, No. 2, pp. 542-546, 1985. 11.
11. 김용두, 조상복, 임인철, "CMOS PLA 설계 및 Stuck-open 고장검출", 대한전자공학회 추계종합학술대회 논문집, Vol. 8, No. 2, pp. 547-550, 1985. 11.