

# MTA 코드를 적용한 Testable CAM 설계에 관한 연구

## (A Study on the Design of Testable CAM using MTA Code)

鄭長源\*, 朴魯京\*, 文大哲\*

(Jang-Won Jung, Nho-Kyung Park, and Dai-Tchul Moon)

### 요 약

본 논문에서는 테스트가 용이하도록 ECC(error checking circuit)를 내장하여 테스트를 수행할 수 있는 CAM(content addressable memory)을 설계하였다. 즉, CAM에서 발생하는 읽기, 쓰기 및 매치 동작의 기능 고장을 검사할 수 있는 회로를 내장한 CAM을 설계하였다. 일반적으로 테스트 회로를 내장하면 전체 면적의 증가를 가져오게 된다. 본 논문에서는 기존의 병렬 비교기를 사용한 내장(built-in) 테스트 회로의 면적 오버헤드를 줄이기 위해서 새로 제안된 MTA 코드를 이용하였다. 설계한 회로는 VHDL 시뮬레이션을 통하여 검증하였으며, 0.8 $\mu$ m double-metal CMOS 공정을 이용하여 레이아웃을 수행하였다. ECC 회로의 경우 CAM의 기본셀에서 매치기능을 담당하고 있는 XOR회로를 이용함으로써 약 30%정도 면적 감소를 가져왔다.

### Abstract

In this work, the testable CAM(Content Addressable Memory) is designed to perform the test effectively by inserting the ECC(Error Checking Circuit) inside the CAM. The designed CAM has the circuit which is capable of testing the functional faults in read, write, and match operations. In general the test circuit inserted causes the increase of total circuit area, Thus this work, utilizes the new MTA code to reduce the overhead of an area of the built-in test circuit which has a conventional parallel comparator. The designed circuit was verified using the VHDL simulator and the layout was performed using the 0.8 $\mu$ m double metal CMOS process. About 30% reduction of a circuit area was achieved in the proposed CAM using the XOR circuit

### I. 서 론

반도체 제조 기술이 급속히 발전함에 따라 메모리의 집적도는 계속적으로 증가하고 있으며 이에 따라 테스트 소요 시간이 증가되고 있다. 즉, 회로가 커질수록 테스트를 수행하는데 소요되는 시간이 증가하게 된다. 따라서 반도체 테스트에 걸리는 소요시간을 줄이는 기술은 최근 반도체 제조 분야에서 매우 중요한 분야로

대두되고 있다. 실제로 테스트에 소요되는 비용도 고 집적화 될수록 전체 생산비용의 많은 부분을 차지하고 있다. 이러한 문제를 해결하기 위하여 설계 단계에서 부터 테스트를 고려한 것이 DFT(design for testability)이다. 즉, 기존에 사용하던 회로 설계를 테스트가 용이하도록 구조를 변경하거나 신호의 흐름을 방해하지 않는 범위에서 디바이스 사이에 테스트를 할 수 있는 장치를 삽입하는 것이다. 최근에는 외부의 테스트 장비에 의존하지 않고 자체적으로 테스트를 수행할 수 있는 BIST(built-in self test)등의 기술도 소개되었다. 이러한 기술들은 빠른 시간내에 테

\* 正會員, 湖西大學校 情報通信工學部  
(School of Information & communication Hoseo Univ.)  
接受日字:1997年8月1日, 수정완료일:1998年5月28日

스트를 할 수 있으며, 별도의 테스트 장치가 없이도 테스트를 수행할 수 있어 테스트 비용의 절감을 가져온다. 반면 회로가 차지하는 오버헤드가 크다는 단점이 있다.

일반적으로 테스트는 메모리 분야와 비메모리 분야로 나눈다. 메모리 분야는 매우 규칙적인 구조를 가지고 있어 다양한 구조를 갖는 조합 회로나 복잡한 프로세서등을 테스트하는 것보다 테스트하기가 쉬우나 테스트 하고자 하는 용량이 큰 만큼 빠른 테스트 수행 능력을 필요로 한다. 빠른 테스트 수행 능력을 위해서 설계 초기부터 테스트가 용이한 DFT를 고려하면서 전체 회로에서 차지하는 테스트 회로가 최적화가 설계해야 한다<sup>[8,9,10]</sup>.

본 논문에서는 CAM(content addressable memory, 연상메모리)<sup>[2,3,4,7]</sup>을 테스트가 용이하도록 ECC를 내장한 Testable CAM을 설계하였다. 고성능 데이터 통신이나 고속의 데이터 검색이 필요한 분야, 패턴 인식 분야 등에서 광범위하게 사용되어지고 있는 CAM은 고성능 인공 지능 컴퓨터에서 연상 처리(associative operation)<sup>[14]</sup> 등으로 인한 방대한 양의 데이터 처리 시간에 따른 병목 현상을 줄이기 위해서 사용되어지고 있다. 또한 최근에는 신경망에서 프로토타입(prototype)과 초기 상태로 사용되는 벡터(probe vector)들 사이의 비선형 전송을 반영하기 위해서 사용되어지고 있다<sup>[15]</sup>.

이와 같이 다양한 응용 분야에서 기존의 메모리를 대신하고 있는 CAM은 단지 입력된 데이터를 읽고 쓰는 기능만을 가진 기존의 메모리와는 달리 고속의 병렬 비교 기능을 함께 가지고 있는 메모리로서 기존의 SRAM에 입력 데이터와의 비교를 위한 매치(match) 기능이 추가되어 있어서 일반적인 메모리에서 발생하는 읽기/쓰기에서의 오류 뿐만 아니라 매치 기능에서의 오류를 초래할 수 있다. 따라서 CAM의 읽기/쓰기 동작 뿐만 아니라 중요한 기능 중의 하나인 매치 기능 오류를 검사할 필요가 있을 뿐만 아니라 시간의 단축을 고려하여 병렬 테스트가 가능해야 한다.

따라서 본 논문은 기존의 해밍코드를 이용하여 설계한 병렬 비교기가 전체 테스트 회로에서 대부분을 차지하고 있는 점을 착안하여 병렬 비교기가 차지하는 면적을 줄일 수 있는 방법을 제안한다. MTA 코드를 이용하여 회로를 설계할 경우 비교기는 간단하게 되고, CAM의 특수 동작인 매치 기능을 효과적으로 사용할

수 있다. 그러므로 병렬 비교기 없이 CAM 자체내의 매치 기능을 이용하여 데이터의 궤환을 통한 결과를 비교한 뒤 오류 검출기에서 이를 검사함으로써 기능 오류를 판단할 수 있도록 하였다.

본 논문의 구성은 1장 서론과 2장에서 Testable CAM의 구성과 동작에 대해서 알아 본다. 3장에서는 2장을 바탕으로 Testable CAM 회로를 설계하고, 4장에서는 3장에서 설계한 CAM에 대한 동작 및 성능 평가를 한다. 그리고 5장에서 결론을 다루었다.

## II. Testable CAM의 구성과 동작

### 1. Testable CAM의 구성

Testable CAM은 그림 1과 같이 입출력 버퍼와 감지 증폭기, 그리고 CAM 셀 어레이의 기본적인 구조와 모든 워드 선을 선택할 수 있는 어드레스 디코더와 테스트를 수행할 수 있는 ECC로 구성되어 있다. 그림 1에서 알 수 있듯이 출력 버퍼/레지스터단의 출력이 ECC단을 거쳐 입력 버퍼/마스크 레지스터단의 입력으로 궤환됨으로써 테스트하고자 하는 워드 선(word line)의 내용과 비교하여 오류를 검사할 수 있도록 구성하였다.

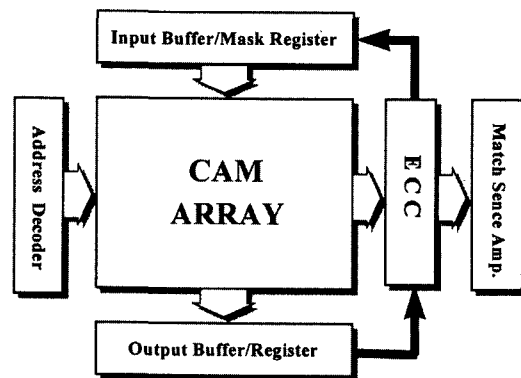


그림 1. Testable CAM의 블록도  
Fig. 1. Block diagram of Testable CAM.

### 2. Testable CAM의 동작

#### 1) 테스트 동작 흐름도

그림 2는 CAM을 테스트하기 위한 동작 흐름도이다. 먼저 변수로 작용하는  $j$ 와  $k$  ( $j$ : 테스트 패턴의 수,  $k$ : 워드 수)을 초기화 시킨 뒤,  $j$  번째 테스트 패턴(test pattern : MTA 코드)을 모든 CAM 셀에 저장한다. (CAM 셀에 테스트 패턴을 입력하는 초기

화 과정) 초기화 과정이 끝나면, 테스트 동작에서는 모든 워드의 내용(테스트 패턴)을 읽은 뒤 그 내용의 상위 비트와 하위 비트를 교차시켜 CAM의 입력값이 되며, 입력된 데이터는 모든 CAM 셀에 저장된 테스트 패턴과 매치 동작을 수행하게 된다.

만약 매치 동작에서 오류가 발생되면 ECC(오류 검출기)는 오류가 발생(ERROR=1)했음을 출력시킴으로써 하나의 테스트 패턴에 대한 검사가 끝나게 된다. 계속적으로 오류 테스트를 하고자 하는 경우는 초기화 과정을 거쳐서 다시 모든 워드에 대한 테스트를 수행하여 모든 테스트 패턴에 대한 검사가 끝나면 CAM에 대한 모든 오류 검사는 끝나게 된다.

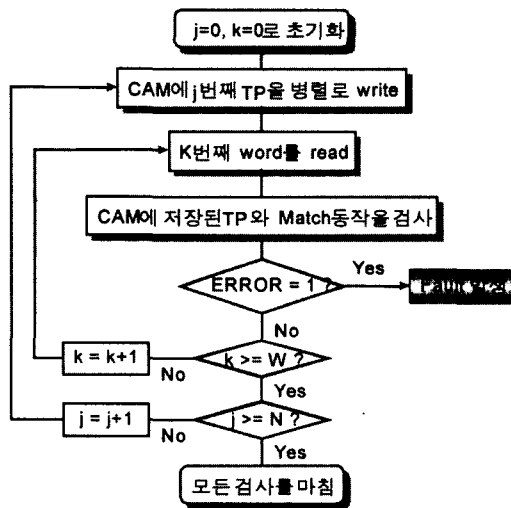


그림 2. 테스트 동작 흐름도  
Fig. 2. Flowchart of test operations.

2) 테스트 동작 파형

테스트 동작위한 파형은 그림 4와 같으며, 테스트 동작은 크게 setup 상태와 test 상태로 나눌 수 있다. Setup 상태는 MTA 코드에 의해서 발생된 테스트 패턴을 CAM셀에 저장하는 단계이며, test 상태는 저장된 테스트 패턴을 사용하여 각각의 워드 선에 대하여 검사를 하는 단계이다.

CAM의 어드레스 선은 테스트 패턴이 입력으로 사용될 때(setup 상태) 모든 워드를 선택함으로써, 병렬로 같은 내용의 데이터(테스트 패턴)를 저장하며, 데이터 입력으로 사용되는 테스트 패턴은 저장신호(write)가 발생할 때 CAM 셀에 저장되고, b/2(b: 비트수) 만큼의 테스트 패턴(T1, T2, ... , Tb/2)이 사용된다. 읽기 신호(read)는 하나의 테스트 패턴에 대해서 워드

수 만큼이 발생하여 각각의 워드선에 대한 매치 결과를 통해서 오류를 검사하게 된다. 이 때 어드레스 선은 순차적으로 각 워드선을 선택하게 된다.

3) MTA 코드

오류 정정 및 수정기법<sup>[5,6]</sup>에서 매우 중요한 복호화 과정은 부가회로의 면적을 감소시키고, 속도의 향상과 신뢰도를 유지하는 것이다. 따라서 본 논문에서 사용한 MTA 코드<sup>[1]</sup>는 메모리의 기능 고장(functional fault)을 검출하기 위해서 제안된 새로운 코드로서, 오류 검출시 기존의 해밍 코드(hamming code)나 Hsiao 코드에 비해 복호 회로의 면적이나 처리 속도를 개선시킨 코드로 알려져 있다. 또한 병렬 테스트의 적용에 용이하고 검사비트를 저장하기 위한 부가 면적을 필요로 하지 않으며, 삼중고장까지도 검출할 수 있는 이점을 가지고 있다. 여기서 MTA 코드의 행렬(H)과 오증(Syndrome)은 다음과 같으며, 이를 이용하여 생성한 패턴은 행렬이 보수적으로 대칭적인 형태를 가지고 있어서 생성이 용이하다.

$$H = \begin{matrix} V_0 & & & & & & & & & & & & & & & & V_{15} \\ \begin{pmatrix} 11111111010000000 \\ 11111110101000000 \\ 11111011001000000 \\ 11110111000100000 \\ 11011111000010000 \\ 11011111000001000 \\ 10111111000000100 \\ 01111111000000010 \end{pmatrix} \end{matrix} \quad S_n = V_{7-n} \oplus V_{8+n} \quad \text{단, } n=(0,1, \dots, 7)$$

여기서 Syndrome S를 이용하여 복호 회로를 구현하기 위해서는 별도의 XOR 회로를 필요하다. CAM의 메모리 셀내에 XOR 회로를 이용하여 간단한 데이터의 경로 제어만으로도 저장된 패턴의 오류를 검출한다. 즉, MTA 코드의 행렬(H)은 중앙을 기준으로 서로 보수적인 관계를 가지면서 대칭으로 이루어져 있어, 이를 CAM에 저장시킨 뒤, 저장된 내용을 읽어 LSB와 MSB를 서로 교차시켜 매치 입력으로 사용한다.

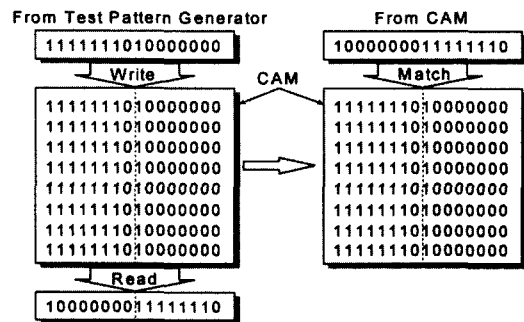


그림 3. MTA 코드를 적용한 CAM  
Fig. 3. MTA Code applied to CAM.

그림 3은 LSB와 MSB가 교차되어 매치 입력으로 사용되고 있음을 보여주고 있으며, 저장된 내용의 변화가 있을 때 매치선은 low가 되어 오류의 발생 여부를 알 수 있다. 이렇게 모든 테스트 패턴에 대해서 메모리에 저장/읽기 그리고 매치 동작을 수행하여 오류를 검사할 수 있다.

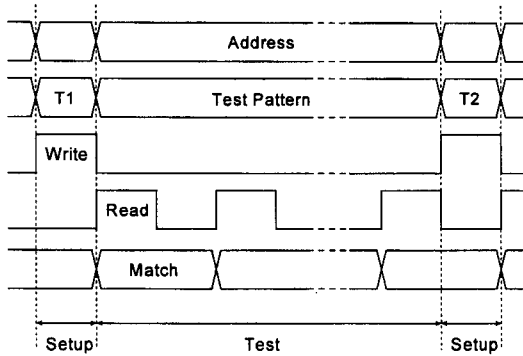


그림 4. 테스트 동작 파형  
Fig. 4. Waveforms of Test operations.

### III. Testable CAM 회로 설계

3장에서는 CAM의 매치기능을 이용하여 test회로의 면적을 줄이고, 테스트 시간을 빠르게 하기위해서 기존 CAM에 ECC 회로와 병렬 테스트를 위해 수정된 어드레스 디코더를 추가한 16word × 16bit의 Testable CAM을 설계하였다. 그 중 핵심 블록인 CAM 셀과 수정된 어드레스 디코더, 그리고 ECC에 대한 설계 방법을 설명한다.

#### 1. CAM 셀

그림 5는 CAM 셀 구조로서 읽기나 쓰기 동작은 SRAM의 동작과 같으며, 저장된 데이터와 입력 데이터와의 매치 기능을 수행하기 위해서 XOR 게이트를 첨가시킨 구조로 테스트 동작을 수행할 때 사용함으로써 별도의 병렬 비교기 없이 저장된 내용과 테스트 패턴을 병렬로 비교할 수 있으며 매치 기능의 오류 또한 검사할 수 있다.

그림 6에 나타난 CAM의 시뮬레이션 결과에서 보여주듯이 저장된 내용(stored data)과 입력 데이터의 내용이 서로 같으면 매치선이 high가 되고, 그렇지 않으면 low를 유지하며 테스트 동작 수행시 기능 오류가 발생하지 않도록 매치선은 low를 유지한다.

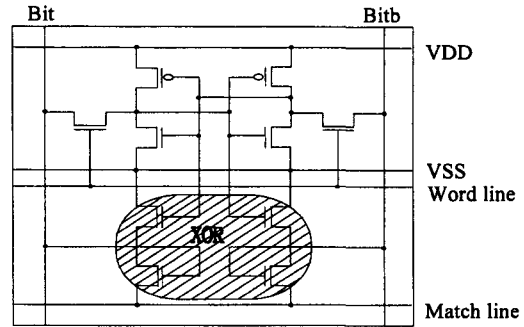


그림 5. CAM 셀  
Fig. 5. CAM cell.

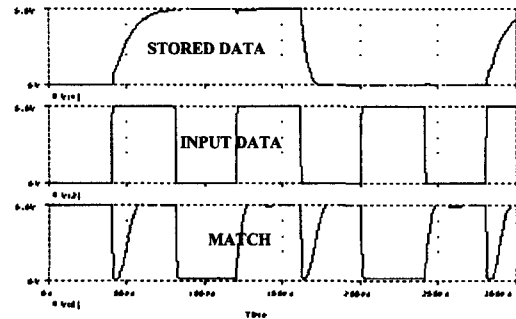


그림 6. CAM의 동작 파형  
Fig. 6. Waveforms of CAM operation.

#### 2. 수정된 어드레스 디코더

수정된 어드레스 디코더는 테스트 동작시 모든 워드 선을 선택할 수 있도록 기존의 회로를 변형하여 그림 7과 같이 설계하였다. Q1에서 Q7까지는 일반적인 디코더를 나타내고 있도록, Q8과 Q9를 첨가하여 어드레스 제어 입력(A1~A4)에 상관없이 모든 워드 선을 선택할 수 있게 하였다.

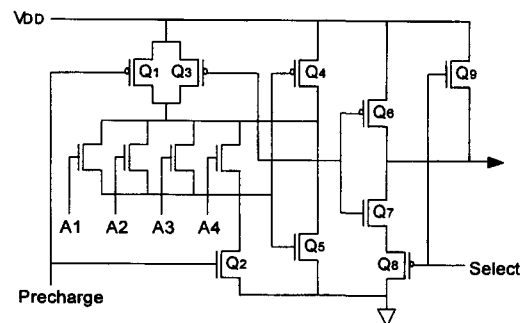
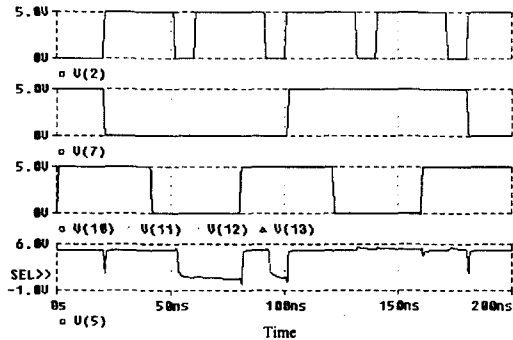


그림 7. 수정된 어드레스 디코더  
Fig. 7. Modified Address Decoder.

즉, Select 신호가 low이면 어드레스 입력에 따라

워드 선이 선택되어지며, Select 신호가 high이면 어드레스 입력에 상관없이 모든 워드 선이 선택되게 된다. 그림 8은 수정된 어드레스 디코더가 정상적인 동작을 수행하고 있음을 보여주는 시뮬레이션 결과이다.



V(2) : Precharge V(7) : Select V(5) : Output  
V(10), V(11), V(12), V(13) : Input

그림 8. 수정된 어드레스의 시뮬레이션 결과  
Fig. 8. Simulation results of modified address decoder.

3. ECC 회로

그림 9는 ECC회로 구조로서 제어 신호(TEST)가 high가 되면 CAM은 기존의 정상적인 동작을 수행하나, low가 되면 CAM은 읽기 쓰기 및 매치 동작 기능의 고장을 테스트하기 위한 테스트 동작으로 수행한다. 스위칭 회로는 CAM의 데이터 경로를 제어하는 회로로서 LSB와 MSB를 교차시켜 CAM의 매치 입력으로 사용하도록 하였다.

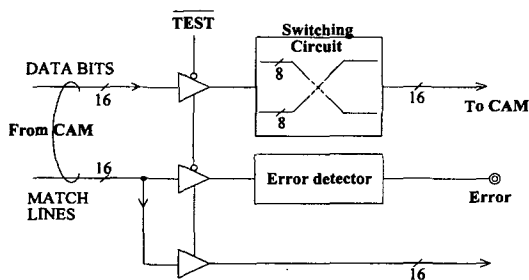


그림 9. ECC 블럭도  
Fig. 9. Block diagram of Error Checking Circuit.

오류 검출 회로(error detector circuit)는 매치 결과를 검사하여 오류가 발생했는지의 여부를 판단하는 회로로서 그림 10과 같다. 오류 검출 회로의 입력은 각 매치선(0 ~ n)과 연결되어 모든 매치선이 low를 나타내면 출력(ERROR)은 low가 되어 오류가 발생되

지 않았음을 출력하나 만약 어느 하나의 매치선이라도 high를 가르키면 출력은 high가 되어 오류가 발생했음을 나타낸다.

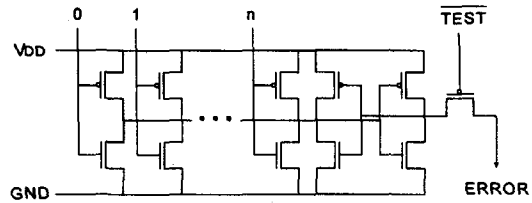
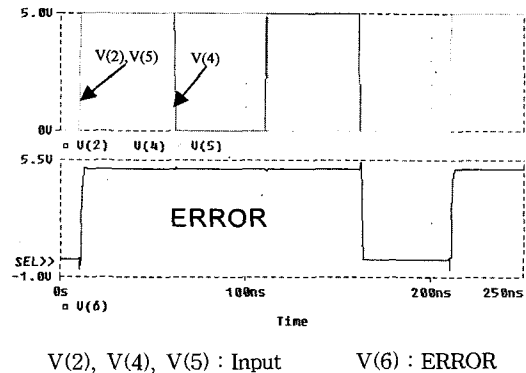


그림 10. 오류 검출 회로  
Fig. 10. Error Detector Circuit.

오류 검출 회로의 시뮬레이션 결과(그림 11)에서도 알 수 있듯이 입력 중에서 이 어느 하나라도 low가 되면, ERROR는 high가 되는 것을 알 수 있다.



V(2), V(4), V(5) : Input V(6) : ERROR

그림 11. 오류 검출 회로의 시뮬레이션 결과  
Fig. 11. Simulation result of Error detector circuit

IV. 시뮬레이션 결과 및 검증

본 논문에서 설계한 회로를 검증하기 위해서 VHDL를 이용하여 회로 및 논리 시뮬레이션을 수행하였다.

그림 12는 모든 워드선을 동시에 선택하기 위한 수정된 어드레스 디코더의 파형으로 EN(enable) 신호는 모든 어드레스 선을 선택하기 위한 신호이고, CK는 어드레스에 공급되는 클럭 신호이며, QOUT(0~15)는 어드레스 선에 연결되어 있는 출력을 나타내고 있다. 결과를 통해서 알 수 있듯이 EN 신호에서 모든 어드레스 선이 선택된 뒤 클럭이 발생할 때마다 어드레스가 하나씩 선택됨을 알 수 있다.

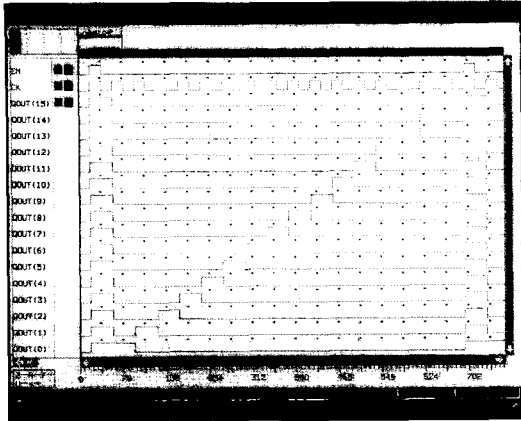


그림 12. 수정된 어드레스 디코더의 파형  
Fig. 12. Waveforms of Modified address decoder.

외부에서 생성되는 테스트 패턴(MTA 코드)은 그림 13과 같은 파형을 가지고 있다. 즉, 어드레스 디코더의 EN 신호를 동작 클럭으로 사용하여 EN 신호가 인가될 때마다 테스트 패턴이 MTA 코드의 순서대로 나타남을 보여주고 있다. 여기서 PRESET은 테스트 패턴 생성기의 초기화를 위해서 사용되며 TP(0~15)는 각 비트로 연결되는 출력을 의미한다.

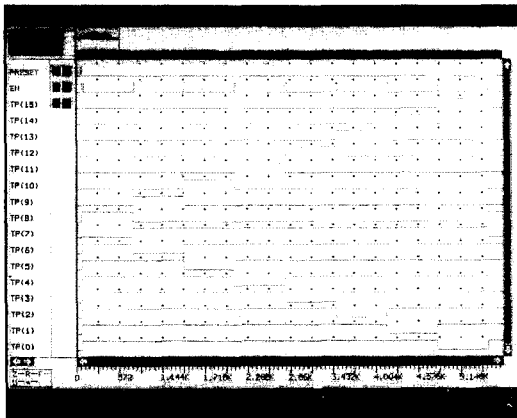
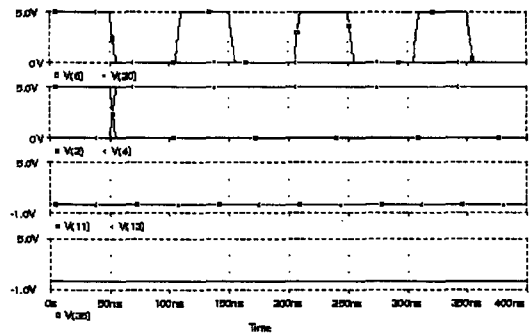


그림 13. 테스트 패턴(MTA 코드)의 파형  
Fig. 13. Waveforms of test pattern(MTA code).

그림 14는 오류가 없는 Testable CAM의 SPICE 테스트 결과로 2 단계의 동작을 수행하고 있음을 보여 주고 있다. 먼저 Write 신호가 발생하면 외부에서 발생한 테스트 패턴을 CAM에 저장하고, 다음에 발생하는 Read 신호는 저장된 내용(테스트 패턴)의 MSB와 LSB를 교차시켜 매치 입력으로 케환시킨다. 따라서 케환된 입력은 저장된 데이터와 비교하여 TEST 신

호는 low가 되면, 매치 결과를 참조로 기능 오류를 검사하게 된다. 그림 14에서 두 매치선이 모두 low가 되면, 매치 결과는 서로 다르게 되는데 이 경우는 케환된 입력이 오류가 없을 경우로 정상 동작을 하고 있음을 보여주고 있다. 따라서 ERROR 신호(그림 14가 low를 보여줌으로써 오류가 없음을 나타내고 있다.)



V(6) : R/W V(20) : TEST V(2), V(4) : Test Pattern  
V(11), V(13) : Match V(35) : ERROR

그림 14. Testable CAM의 시뮬레이션 결과  
Fig. 14. Simulation results of Testable CAM.

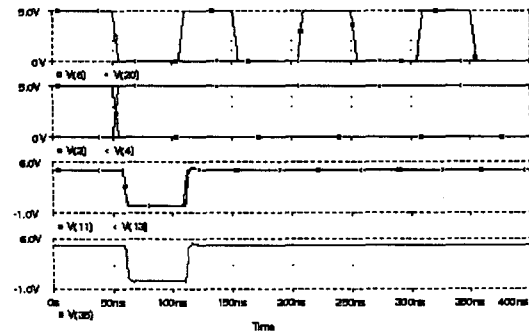


그림 15. 기능 오류 시뮬레이션 결과  
Fig. 15. Simulation results of functional fault.

그림 15는 오류를 가진 Testable CAM의 시뮬레이션 결과를 보여주고 있다. 오류가 있을 경우 서로 같은 비교 결과를 나타나게 되는데 이는 읽기 동작에서 어느 데이터가 변하였음을 나타내고 또한 모든 테스트 패턴에서 이러한 결과가 나올 경우에는 매치 동작에서도 오류일 확률이 높다. 그림 15에 나타난 결과는 매치선 모두가 high 오류가 발생되었음을 알 수 있으며, 그 결과 그림 15에서 알 수 있듯이 ECC는 ERROR 신호를 high로 출력시켜 오류가 발생되었음을 나타낸다.

## V. 결 론

본 논문에서는 CAM의 읽기, 쓰기 및 매치 동작의 기능 고장 검사를 할 수 있는 ECC를 내장한 Testable CAM을 설계하였다. 테스트 회로를 내장하는데 있어서 테스트 회로가 차지하는 오버헤드 즉, 테스트를 수행하기 위한 회로가 너무 많은 면적을 차지해서는 안된다는 것으로 이는 오버헤드가 증가하는 만큼 생산비용의 증가를 의미하기 때문이다. 따라서 본 논문에서는 기존의 CAM의 매치 기능과 구조적인 장점을 충분히 활용할 수 있는 간략화 된 MTA 코드를 사용함으로써 간단한 데이터의 경로 제어를 통해 테스트를 수행할 수 있었다. 그 결과, 기존의 방식을 사용한 CAM의 Built-in 테스트 회로보다<sup>[11]</sup> ECC회로의 경우 CAM의 기본셀에서 매치기능을 담당하고 있는 XOR 회로를 이용함으로써  $0.8\mu\text{m}$ 의 공정을 이용하여 layout을 수행한 결과 약 30% 정도의 면적 감소와 테스트 기능을 동작 클럭으로 수행할 수 있게 되었다. 그러므로 본 논문에서 설계한 Testable CAM은 읽기, 쓰기 및 매치 동작에 대한 테스트를 수행할 수 있다.

설계한 Testable CAM은  $0.8\mu\text{m}$  이중 금속 CMOS 공정을 이용하여 레이아웃을(그림 16) 작성하였다. Testable CAM의 전체 칩 면적은  $689\mu\text{m} \times 759\mu\text{m}$ 이며, 이중 테스트회로로 소요되는 면적은  $70\mu\text{m} \times 121\mu\text{m}$  로서, 오버헤드는 약 2% 정도가 된다.

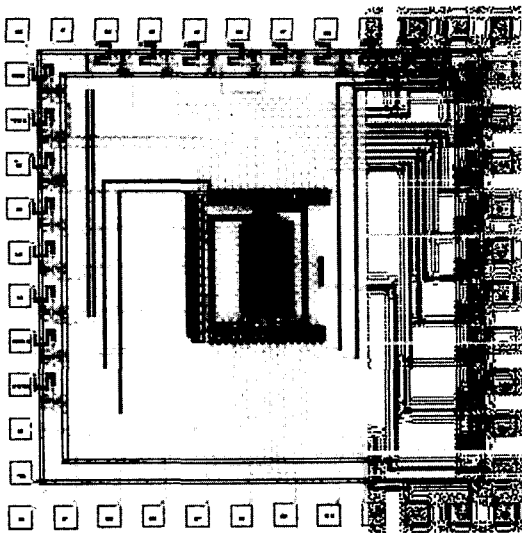


그림 16. 전체 회로의 레이아웃  
Fig. 16. Layout of CAM with ECC.

## 참 고 문 헌

- [1] 이중호, 조상복, 반도체 메모리의 테스트를 위한 MTA(Memory TesAble code) 코드, 전자공학회논문지, 제31권, A편, 제8호, 1994
- [2] 박노경, 차균현의, CAM(Content Addressable Memory)의 병렬테스팅을 위한 Built-in 테스트회로 설계에 관한 연구, 통신공학회논문지, 제19권, 제6호, 1994
- [3] 박노경, 차균현, 연상 메모리 설계 및 제작에 관한 연구, 한국 통신학회 논문지, 제 16권, 제 2호, 1991
- [4] 박노경, 차균현, CARM(Content Addressable Memory and Reentrant Memory)의 설계에 관한 연구, 한국 통신학회 논문지, 제 16권 제 1호, 1988
- [5] K.Furutani et al., A Built-In Hamming Code ECC Circuit for DRAMs, IEEE J. Solid-State Circuit, vol. 24, no. 1, pp. 50-56, Feb. 1989.
- [6] M. Asakura et al., An Experimental 1Mbit Cache DRAM with ECC, IEEE, J. Solid-State Circuit, vol. 25, no. 1, Feb. 1990.
- [7] Anthony J. McAyley & Charles J. Cotton, A Self-Testing Reconfigurable CAM, IEEE, J. Solid-State Circuit, vol. 26, no. 3, March. 1991.
- [8] T. W. Williams, VLSI TESTING, North-Holland.
- [9] Miron Abramivici, Arthur D.Friedman, Digital System Testing and Testable Design, IEEE PRESS, 1990.
- [10] Frank F.Tsui, LSI/VLSI Testability Design, McGRAW-HILL, 1986.
- [11] PINAKI MAZUMDER & JANAK K. PATEL, Parallel Testing for Pattern-Sensitive Faults in Semiconductor Random-Access Memories, IEEE, Tran. on Computer, vol. 38, no. 3, March 1989.
- [12] P. Marzumder, W.K Fuchs, Design and algorithms for parallel testing of random access and content addressable memories, 24th ACM/IEEE Design Automation Conf., pp. 688-694, 1987.
- [13] James T.Healy, Automatic Testing and Evaluation of Digital Intergrated Circuits,

- Prentice-Hall, 1981.
- [ 14 ] Takeshi Ogura et al, A 20-kbit Associative Memory LSI for Artificial Intelligence Machines, IEEE, J. Solid-State Circuit, vol. 24, no. 4, Aug. 1989.
- [ 15 ] R. Pergetti, Mapping Binary Associative Memories onto Sigmoidal Neural Networks Using a Modified Projection Learning Rule, IEEE, Circuit and Systems-II: Analog and Digital signal proc. vol. 41, no. 7, July. 1994.

---

저 자 소 개

鄭 長 源(正會員)

1971년생. 1995년 2월 호서대학교 정보통신공학과(학사). 1997년 2월 호서대학교 정보통신공학과(석사)

朴 魯 京(正會員) 第 35卷 S編 第 2號 參照

현재 호서대학교 정보통신공학부 부교수

文 大 哲(正會員) 第 35卷 S編 第 2號 參照

현재 호서대학교 정보통신공학부 정교수