

내장 메모리 테스트 시스템 설계

김 지 호*, 윤 대 한*, 송 오 영*

*중앙대학교 전자전기공학부

e-mail : song@jupiter.cie.cau.ac.kr

Design of Embedded Memory Test System

Ji Hoo Kim*, Dae Han Youn*, Ohyoung Song*

*School of Electrical & Electronics Engineering, Chung-Ang University

요약

본 논문에서는 PC상에서 내장 메모리를 테스트 할 수 있는 테스트 시스템을 구현하였다. 테스트상으로는 Synchronous DRAM을 사용하였고 내장 자체 테스트 회로에 10N March C 알고리즘을 적용, DSRAM, SRAM을 제어하는 테스트 시스템 제어를 설계하였다. 본 테스트 시스템은 메모리 테스트 검증을 고가의 테스트 장비 없이 용이하게 하도록 설계되었다.

1. 서론

현재에는 칩 외부에 배치되었던 메모리 같은 모듈들도 이제는 하나의 칩에 내장되는 추세이다. 이와 같이 고집적화된 칩의 테스트에 있어서 가장 어려운 부분 중에 하나로 여겨지는 것은 내장 메모리(Embedded Memory)의 테스트이며, 이러한 문제들을 해결하기 위하여 내장된 자체 테스트 기법은 반드시 필요하게 되었다[10]. 내장된 자체 테스트 기법이 적용된 칩은 부수적으로 면적 증가와 같은 오버헤드를 갖게 되지만, 테스트의 복잡도가 크게 줄어들고 모듈별로 가장 적합한 테스트가 가능하며 고가의 외부 테스트 장치를 사용하지 않고도 빠른 시간에 테스트를 수행할 수 있다는 장점들을 갖게 된다. 더욱이 내장된 메모리의 크기가 점차 커져감에 따라 내장된 자체 테스트 회로의 단점인 면적 오버헤드가 상대적으로 크게 감소하게 되므로 그 장점이 더욱 부각되고 있다. 내장된 메모리의 테스트에서 고려해야 할 또 하나의 중요한 문제는 테스트 알고리즘의 문제이다. 효율적인 고장 모델을 설정하고, 정의된 고장 모델을 적절한 시간에 효율적으로 테스트 할 수 있는 알고리즘을 개발하는 것이 중요하다. 또한 알고리즘이 적용된 테스트 회로를 검증할 수 있는 방법이 필요하다.

본 논문에서 테스트 할 대상으로는 고속 동작이 가능한 Synchronous DRAM으로 정한다[4,5]. 본론 2장 1절에서는 테스트 대상이 되는 Synchronous DRAM에 대해서 간략히

설명한다[6,7]. 2장 2절에서는 Synchronous DRAM에서 테스트 할 고장 모델을 정의하고 알고리즘에 대해서 설명하겠다. 2장 3절에서는 내장된 자체 테스트 회로 검증 시스템의 구조와 동작에 대해 설명하겠다. 2장 4절에서는 시뮬레이터와 테스트 보드 검증을 통해 시스템의 정상 동작을 확인하고 3장에서 결론을내림으로써 논문을 마치도록 하겠다.

2. 본론

2.1 Synchronous DRAM 개요

SDRAM이 나오기 전의 DRAM은 자체에서 소모되는 지연(Delay)외에도 DRAM 제어가 DRAM의 특성을 시스템 클럭(System Clock)에 동기시키는데 소요되는 지연도 포함하고 있어 그 속도에 한계가 있었다. 그래서 DRAM이 시스템 전체의 병목 지점이 되어 DRAM의 속도 개선이 시급한 문제로 대두되었다.

이에 대한 해결책으로 DRAM 내에서의 동작도 최대한 버스 클럭(Bus Clock)에 동기되도록 함으로써 제어기에 의한 버스 클럭 동기(Synchronization)시에 소요되는 시간, 그리고 인터리빙(Interleaving) 제어에 추가되는 시간 등을 없애며 DRAM 속도를 버스 클럭의 정수배로 고정시켜 시스템 설계의 편리를 도모토록 한 것이 Synchronous DRAM이다 [6,7].

SDRAM은 EIA/JEDEC JC42.3 DRAM 표준 위원회에서 국내 반도체 회사, 일본 및 미국의 반도체와 컴퓨터 회사가

모여 협의를 통해 표준을 결정한 표준품이다.

SDRAM의 특징은

- 외부 클럭과 동기된 입출력 회로(Synchronous I/O)
- 연속된 데이터 접근(Burst Access)
- 다중 뱅크(Multibank)구성
- 명령어형식의 접근(Mode Register의 Programming)
- 파이프라인 데이터 경로(Pipelined Data Path)

등으로 요약해 볼 수 있다.

이러한 기능을 종래의 DRAM에 추가시키기 위해서는 여러 가지 새로운 회로들이 삽입되어야 한다. 물론 DRAM과 기본 골격은 동일하지만 주소, 제어 및 각종 데이터 I/O 회로, 그리고 고속 데이터 경로를 클럭에 동기시키기 위한 래치나 레지스터들이 삽입되어야 하는 점이 큰 차이를 이룬다.

2.2 Synchronous DRAM 내장 자체 테스트

내장된 자체 테스트 기법에서는 회로의 일부분이 회로 자체를 테스트하는데 쓰이게 된다. 내장된 자체 테스트 기법의 특징은 테스트 입력이 자체 내에서 생성되고 테스트 결과도 자체 내에서 평가된다는 것이다. 따라서 외부에서 필요한 동작은 테스트의 시작과 테스트가 종료된 이후에 테스트의 통과 유무만을 확인하는 것이다.

정상 동작 시에는 정상 입력(Normal Inputs)과 정상 출력(Normal Outputs) 단자를 통해 회로가 동작을 하게 되고, 테스트 동작 시에는 테스트 패턴 생성기(Pattern Generator)에서 생성된 테스트 패턴으로 회로를 테스트한 후 결과를 테스트 응답 분석기(Output Response Monitor)에서 분석하여 회로의 고장 유무를 파악하게 된다

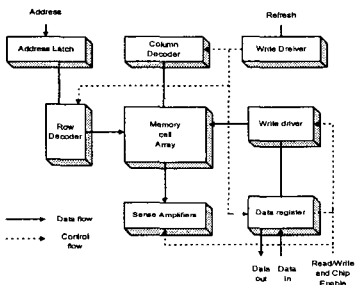


그림 1 메모리의 기능적 모델

실제 메모리에서의 고장은 매우 다양한 상태로 나타나게 된다. 따라서 메모리의 정상적인 동작에 영향을 미칠 수 있는 고장의 모든 경우에 대해서 테스트를 수행한다는 것은 실질적으로 불가능하다. 그러나 메모리 테스트의 목적은 특수한 경우를 제외하고는 고장의 유형이나 위치를 파악하기 보다는 단순히 고장의 발생 유무를 파악하는 것이다. 그러므로 일반적인 메모리 테스트에서는 먼저 메모리의 구조를 기능 모델로 단순화시킨다. 이 경우 메모리는 메모리 셀 배열(Memory Cell Array), 주소 디코더(Address Decoder), 읽기·쓰기 회로(Read·Write Logic)로 구성된다. 테스트를

위한 메모리의 기능적 모델은 그림 1과 같다.

본 논문에서 제시하는 내장 자체 테스트는 March C- 알고리즘을 적용하였다. 이 테스트 알고리즘은 고착 고장, 천이 고장 그리고 결합 고장 검출에 널리 사용되는 알고리즘이며, 전체 테스트는 N(전체 메모리 셀 수)에 비례한다. 다음 그림 3.3에 나타난 바와 같이 먼저 셀에 0(1)을 쓰고 이후 어드레스의 한 셀에 0을 읽은 후 다시 1을 쓴다. 이 동작을 오름 차순으로 반복하여 N번지 셀까지 완료되면 이번에는 어드레스를 내림 차순으로 각 셀의 데이터를 읽은 후 반대 데이터를 쓰는 동작을 반복한다. 오름 차순과 내림 차순에 의한 테스트는 현재 셀과 상위 어드레스 혹은 하위 어드레스 셀 사이에 존재할 수 있는 결합 고장을 검출할 수 있으며 또한 어드레스 디코더의 고장에 의한 multi-selection 고장을 검출할 수 있다.

M0, M1 등은 March 요소라고 하며 하나의 March 요소는 어드레스의 방향에 따라 전체 메모리 셀에 대하여 테스트를 수행한 후 다음 March 요소를 테스트하는 것을 의미한다. 또한, 한 차순에 대해서 2가지 레벨 패턴(0, 1)을 인가하는 것은 천이(Transition) 고장을 검출하기 위함이다.

그림 2는 March C Algorithm의 전체적인 과정을 나타낸다.

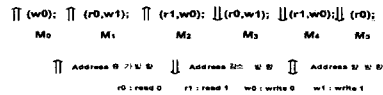


그림 2 March C Algorithm

March C 알고리즘은 어드레스 디코더의 고장, 고착 고장, 천이 고장, 결합 고장을 검출할 수 있다. 모든 고착 고장과 천이 고장은 1과 0을 차례로 씌으로써 검출될 수 있다. 그리고 어드레스 디코더 고장과 결합 고장은 (M1,M4) (M2,M3)의 조합에 의해서 검출된다.

2.3 내장 자체 테스트 시스템 구조

본 설계의 내장 자체 테스트 시스템은 그림 3에 나타난 바와 같이 중앙 제어부(TGC), 어드레스 발생부(Addgen), 테스트 패턴 생성부(Datagen) 그리고 데이터 비교 및 고장 검출부(Datacomp), 내장 자체 테스트 제어부(BIST Controller), Synchronous DRAM, Static RAM, PCI Interface, Driver, PC에서 동작하는 테스트 프로그램으로 구성되어 있다.

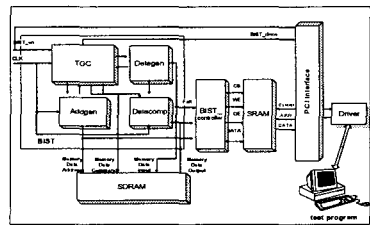


그림 3 내장 자체 테스트 시스템 구조

중앙 제어부는 외부에서 BIST_en 신호를 받아들여 테스트를 시작하며 March C- 알고리즘 상태 머신에 의해 알고

리즘을 종료하면서 BIST_done 신호를 발생시킨다. 테스트 중에 각 내부 회로들에 필요한 내부 제어 신호들을 발생시켜 어드레스 발생 시점과 데이터 생성 및 비교 검출 시점들을 조정한다. 중앙 제어부는 March C- 알고리즘 수행에 적합하도록 크게 4개의 상태 머신 회로로 구성된다. 메모리 초기화를 위한 파워 온 시퀀스 머신 회로(Power-on Sequence State Machine), 전 셀에 대해서 일정한 패턴으로 쓰기 동작을 진행하는 초기화 상태 머신 회로, 전 셀에 대해서 동시에 읽기와 쓰기 동작을 수행하는 읽기/쓰기 상태 머신 회로 그리고 전 셀의 데이터를 한꺼번에 읽기를 진행하는 읽기 상태 머신 회로로 구성되어 알고리즘의 각 동작을 담당한다. 그림 4는 중앙제어부의 구조를 나타낸 그림이다. 그림에서 나타난 각각의 블록들은 March C Algorithm에 따라 왼쪽에서부터 순차적으로 동작한다. 단 Auto Refresh 블록의 경우에는 독자적으로 동작한다.

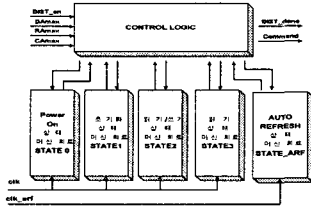


그림 4 중앙 제어부 블록 다이어그램

어드레스 발생부는 내부 카운터를 사용하여 구현되며 테스트 대상이 되는 메모리 셀의 어드레스를 발생시킨다. 이 과정에서 어드레스 Scramble을 적용시켜 셀들에 대해 테스트가 순차적으로 진행되도록 한다. 그림 5는 어드레스 발생부의 전체적인 구조를 나타낸 것이다. 각각의 블록은 BIST_mode를 제어 신호로 받아 Bank Address, Row Address, Column Address를 발생시킨다.

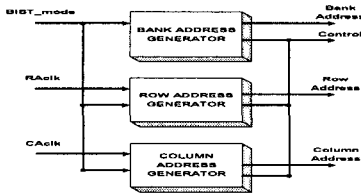


그림 5 어드레스 발생부 블록 다이어그램

비교 검출부는 테스트 대상 셀의 데이터와 내장 자체 테스트 패턴 생성부의 데이터를 비교하여 메모리 고장의 유무를 판별하는 상호 비교 방식을 취한다. 이와 같은 비교 방식은 다중 비트 구조에서 메모리의 정상 동작하에 발생할 수 있는 신호의 지연을 야기하지 않고 비교적 낮은 하드웨어 오버헤드를 가지나, 회로 자체가 XOR 게이트로 구성되어 높은 데이터 전송 폭을 지원하는 MML (Merged Memory Logic)이 테스트 대상인 경우에는 비교 검출부 회로의 면적에 대한 오버헤드가 전체 내장 자체 테스트 회로 면적에 상당 부분을 차지하는 단점이 있다. 테스트 패턴 생성부는 알고리즘의 인가 패턴을 생성시키

며, 비교 검출부와 함께 메모리 I/O 구성에 의존하는 면적 오버헤드 특징이 있다. 중앙 제어부에서 알고리즘 진행에 따른 1비트의 패턴 신호를 테스트 패턴 생성부에 인가하게 되면 각각의 출력 단자에 구성돼 있는 패턴 생성 셀들이 응답하여 테스트 패턴을 출력하게 된다.

자체 내장 테스트에서 오류 발생시 그 정보를 저장해 두어야 한다. 본 설계에서는 이 정보를 SRAM에 저장하기로 한다. 자체 내장 테스트에서 출력되는 데이터의 속도가 가장 빠를 경우는 연속 읽기(Burst Read)의 경우이다. 연속적인 읽기에 의해서 데이터가 나올 경우는 11개의 상태를 지난다. 그러므로 11개의 상태를 지나는 시간전에 정보를 저장해야 한다. 그림 6은 자체 내장 테스트 제어기의 상태전도를 나타낸 그림이다. Fail을 검사하고 저장하기 위해서는 3개의 상태를 거쳐야 한다. 그러므로 연속적인 Fail이 발생되더라도 데이터를 안전하게 저장할 수 있다.

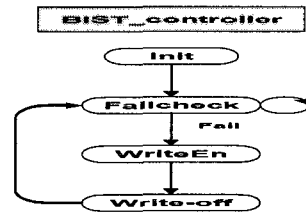


그림 6 내장 자체 테스트 제어기의 상태전도

2.4 실험

설계된 내장 자체 테스트 회로를 먼저 verilog 시뮬레이터로 검증한 다음 내장 자체 테스트 보드를 구성하여 PC상에서 테스트 실험을 하였다.

2.4.1 Verilog 시뮬레이션

자체 내장 테스트 시스템 설계는 Verilog HDL로 기술되어 있으며 동작의 검증은 Verilog-XL™으로 하였다[11]. 시스템 클럭은 100MHz로 Auto refresh 클럭은 25MHz로 넣어 주었다 BIST_en 신호가 0에서 1로 되면서 내장 자체 메모리의 테스트가 시작된다. 그리고 자체 내장 테스트 회로에서 나오는 cke_o, csb_o, rasb_o, casb_o, web_o의 신호들이 명령어가 되어 Synchronous DRAM의 상태를 변화시키고 동작을 제어한다. 파워 온 시퀀스 상태와 초기화 상태, 읽기/쓰기 상태, 읽기 상태를 거치면서 10N March C Algorithm을 수행한다. 10N March C Algorithm이 끝나게 되면 BIST_done 신호가 0에서 1로 되면서 테스트의 종료로 알려준다. 메모리의 오류의 검사는 읽기 명령어가 수행될 때 이루어 진다. 메모리로부터 읽어들이는 데이터에 오류가 있으면 내장 자체 테스트 회로에서 Fail 신호를 1로 변화시킨다. Fail 신호가 0에서 1로 되면 내장 자체 테스트 제어기가 오류가 난 정보를 Static RAM에 저장시킨다.

자체 내장 테스트 시스템의 정상적인 동작 시뮬레이션 결과가 그림 7에 나와 있다. 그림에서 읽기와 쓰기가 반복되면서 데이터의 오류를 검사하고 있다. 정상적인 메모리 모델을 사용하였기 때문에 오류는 발견되지 않고 Fail 신호

는 0로 유지된다.

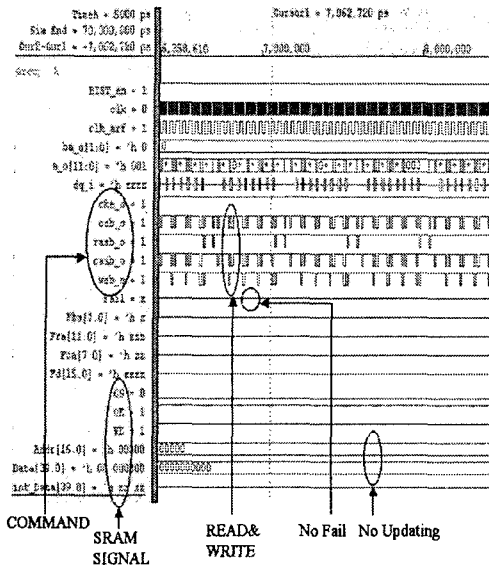


그림 7 Verilog 시뮬레이션 파형 (정상동작)

그림 8은 메모리에 오류가 있을때의 자체 내장 테스트 회로의 동작을 나타낸 그림이다. 강제적으로 오류를 발생시키기 위해서 데이터 버스에 임의의 데이터(FFzz)를 입력으로 넣어 주었다. 오류 발생시의 정보인 오류 주소와 오류 데이터가 SDRAM에 저장됨을 확인할 수가 있다.

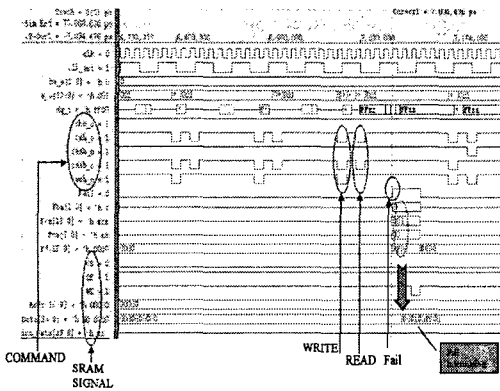


그림 8 Verilog 시뮬레이션 파형(오류발생)

두 가지의 경우를 종합해 볼 때 자체 내장 테스트 회로가 정상적으로 설계되었음을 알 수가 있다.

2.4.2 Test 보드

PCI Interface는 PLX9050 소자를 사용해서 테스트 보드를 구성하였다. 설계하고자 하는 Test 프로그램 특성상 40 bit의 정보를 사용할 수 있어야 하기 때문에 우선 32bit으로 한번 읽고 다시 8bit을 읽는 형식으로 구현을 하였다. 또한 PCI Bus를 이용하기 때문에 132MB/sec의 속도를 낼 수 있

기 때문에 빠른 메모리 테스트가 가능하다. 내장메모리 회로는 Altera FPGA를 사용하여 구성하였고 SDRAM은 현대 4 bank 32Mbit SDRAM을 사용하였으며 SPAM은 256kbit 5개를 사용하였다. 그림 9는 테스트 보드의 사진을 보여준다.

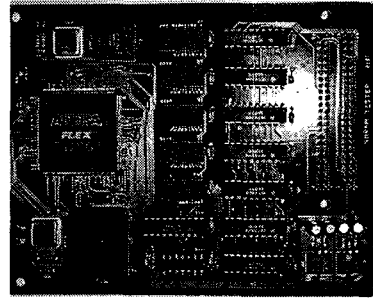


그림 9 내장 자체 테스트 시스템보드

3. 결론

Synchronous DRAM의 특징과 동작들은 자체 내장 테스트 시스템 설계를 위해 연구되었다. 실제 메모리를 기능적 모델로 단순화 시켜, 발생 가능한 고장들을 각각 정의하였다. 그리고, 정의된 고장들을 검출할 수 있는 알고리즘에 대한 연구를 하였다. 내장 자체 테스트 회로에 적용된 10N March C 알고리즘에 대한 연구를 하였고, 내장 자체 테스트 시스템은 자체 내장 테스트 회로와 제어기, Synchronous DRAM과 Static RAM으로 구현되었다. 그리고 시뮬레이션과 테스트 보드를 구성을 통하여 내장 자체 테스트 시스템의 동작을 검증하였다.

참고문헌

- [1] Daniel P. van der velde and A.J. v.d. Goor, "Designing a Memory Module Tester," *IEEE Press*, 1999.
- [2] Chih-tsun Huang, Jing-Reng Huang, "A Programmable BIST Core for Embedded DRAM," *IEEE Press*, 1999.
- [3] "HDL250 eDRAM IP SPEC 0.25um 16M 2Bank," *HYUNDAI Electronics Industries co. Ltd*, June 1999.
- [4] "HY57V651620B 4 Banks×1M×16Bit Synchronous DRAM," *HYUNDAI Electronics Industries co. Ltd*, Nov 1999.
- [5] Hideo Fujiwara, "Logic Testing and Design for Testability," *The MIT Press*, 1985.
- [6] Miron Abromovico, Melvin A. Breuer, Author D. Friedman, "Digital Systems Testing and Testable Design," *IEEE Press*, 1990.
- [7] Michael Keating, Pierre Bricaud, "Reuse Methodology Manual for System-on-a-Chip Designs," *Kluwer Academic Publishers*, 1998.
- [8] Educational Services Group, "Verilog-XL™ Training Manual," *CADENCE*, June 1995.
- [9] 유희준, "고성능 DRAM," 시그마프레스, 1997년 7월.