

Memory Tester용 ASIC 칩의 설계

The Design of ASIC chip for Memory Tester

정지원*, 강창현**, 최창***, 박종식***
(J.W.Joung, C.H.Kang, C.Chi, J.S.Park)

경북대학교 전자공학과 VLSI Design lab
(전화:(053)940-8839, E-mail : jjw@vlsilab.knu.ac.kr)

Abstract – In this paper, we design the memory tester chip playing an important role in the memory tester as central parts. Memory tester has the sixteen inner instructions to control the test sequence and the address and data signals to DUT. These instructions are saved in memory with each block such as sequencer and pattern generator. Sequencer controls the test sequence according to instructions saved in the memory. And Pattern generator generates the address and data signals according to instructions saved in the memory, too. We can use these chips for various functional test of memory.

1. 서 론

디지털 시스템에서 전체 칩 중에 메모리 칩이 차지하는 비율이 높은 경우는 메모리가 디지털 시스템의 장애율(failure rate)에 큰 영향을 미친다. 따라서 메모리 테스트는 디지털 시스템의 신뢰도(reliability) 및 가용도(availability)를 높이는 데 중요한 역할을하게 된다.

메모리 테스트는 조립되기 전 웨이퍼 상태에서 이루어지는 웨이퍼 테스트와 조립 공정 이후에 패키지 상태로 이루어지는 패키지 테스트로 구분할 수 있다. 패키지 테스트는 제품의 불량여부를 판별하는 가부(go/no-go) 테스트와 메모리의 성능을 정확하게 판별하는데 목적이 있는 특성(Characterization) 테스트가 있다. 특성 테스트는 광범위하고 정밀한 메모리의 각 특성을 확인하는데 목적이 있고, 또한 대부분 소량의 메모리에 대하여 테스트를 하기 때문에 테스트에 소요되는 시간이 다소 길다 하여도 무방하다. 반면 가부 테스트의 경우 많은 수의 메모리에 대하여 제품의 불량여부를 판별하여야 하기 때문에 무엇보다도 테스트 소요시간이 중요한 요소가 되며 따라서, 짧은 시간에 많은 고장을 검출하도록 하는 것이 중요하다.

기능 테스트(Functional test)의 목적은 메모리가 논리적으로 정확하게 동작하는지를 검증하는 것이다. 대부분의 기능 테스트 방법은 레이아웃에 독립적인 기능 고장 모델만을 다룬다. 이러한 고장 모델은 고착 고장, 결합 고장 등과

같이 포괄적인 고장 형태의 수학적인 축약 형태로써 메모리의 물리적인 레이아웃 결함과 논리 고장간의 관계를 표현하지는 않는다. 이러한 고장의 형태와 물리적인 결함의 형태간 관계가 명확히 규정되지는 않더라도 이러한 고장 모델은 높은 고장 검출율을 얻을 수 있으므로 매우 효과적이다.

본 논문에서는 메모리 테스터의 핵심 부품인 메모리 테스터용 칩을 설계하고 그 기능을 검증하였다. 설계된 메모리 테스터용 칩은 논리 고장 모델의 기능 테스트 알고리즘을 생성하여, 주어진 Timing Rate에 맞춰 DUT(Device Under Test : memory)에 인가할 Write, Read 신호를 생성한다. 이 신호들은 몇 가지 처리과정을 거쳐 driver IC를 통해 DUT에 인가되어 메모리를 테스트하게 된다.

2. 본 론

2.1 Memory Tester의 전체 개요

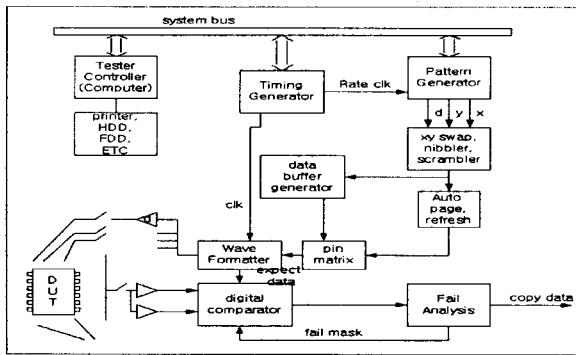
본 논문에서 제시한 memory tester용 chip의 구성 및 동작에 대한 이해를 돋기 위해 Memory tester의 전반적인 구조와 특징들을 설명하고 Memory tester용 chip의 구성 및 역할에 대해 설명한다.

2.1.1 Memory tester의 특징 및 구성

메모리 테스터는 기본적으로 테스터를 제어할 control system과 테스터 그리고 DUT(device under test)로 구성된다. control system은 pc 환경에서 작동 될 수 있으며 메모리 테스터의 감시 및 제어와 동시에 test vector를 Memory tester를 통하여 DUT(Device Under Test)에 인가하고 Memory tester의 Pass/Fail detector의 결과로부터 메모리 가부 판정 여부를 분석하는 기능을 수행한다.

저자 소개

- * 鄭智元 : 慶北大學 電子工學科 碩士課程
- ** 姜昶憲 : (주)일렉스
- *** 崔昶 : 慶北大學 電子工學科 博士課程
- ****朴宗植 : 慶北大學 電子工學科 教授 · 工博



<그림 2-1. 메모리 테스터의 개요>

일반적인 메모리 테스터는 ALPG(Algorithmic Pattern Generator)라는 별도의 회로가 있어 테스트 알고리즘을 생성해내지만 본 논문에서 설계한 칩은 ALPG를 프로세서로 설계하여 사용자가 미리 프로그램해 놓은 알고리즘을 내부 메모리에 저장한 뒤 instruction을 수행하여 자동으로 테스트 할 수 있도록 하였다.

Memory tester Chip은 sequence controller와 address/data pattern generator로 구성된다.

2.1.2 Memory tester chip의 동작

테스트 수행 전에 프로그램된 테스트 알고리즘은 테스트 제어 시스템에서 Memory tester chip의 내부 메모리에 저장된다. Sequencer는 내부 메모리에 저장된 instruction을 수행하여 PC(Program Counter)를 생성한다. Pattern generator에서는 PC를 입력 받아 해당되는 메모리의 데이터를 읽고 address와 data를 생성해서 DUT로 출력한다. 일반적인 테스트 과정은 DUT에 Write한 뒤, Read하여 Write data와 비교하고 pass/fail을 판별하게 된다.

Sequencer와 Pattern generator 각 블록은 알고리즘을 생성하기 위한 프로그램을 입력받아 저장하기 위한 256×32 bits의 내부 메모리를 가지고 있다. 그리고 매 Timing Rate마다 내부에서 opcode fetch와 execution을 수행하여 PC와 test pattern을 생성해낸다. 각 블록은 Timing Rate마다 하나의 instruction을 수행하도록 Pipeline 구조로 설계하였다. 그 밖에 Timing Generator는 Timing Rate 신호를 각 블록에 보내어 전반적인 Test Cycle을 제어하게 된다.

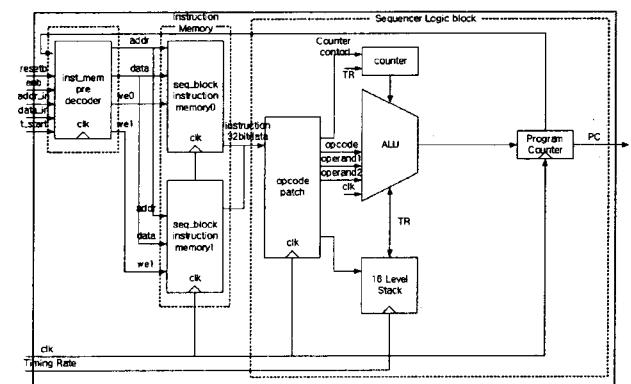
2.2 Tester chip의 설계

Test Control Chip은 메모리 테스트 알고리즘 생성을 주요 기능으로 하며, Sequencer, Pattern generator 그리고 Timing generator로 구성된다. 설계된 칩은 내부 메모리에 테스트 수행에 필요한 명령어를 미리 저장한 뒤, 테스트 수행 시에 내부 메모리를 읽어서 opcode patch와 execution을 수행하는 구조로 되어있다. 따라서 sequencer와 Pattern generator의 구성은 크게 내부에 명령어를 저장하기 위한 메모리와 이를 처리하는 logic 부분으로 구성되어 있다. Timing generator는 지정된 Test Cycle을 생성하도록 내부 레지스터와 카운터 회로로 구성되어 있다.

2.2.1 Sequence controller

Sequencer의 입력 단은 테스트 제어부의 address/data 버스를 통하여 테스트 수행 전에는 내부 메모리에 instruction

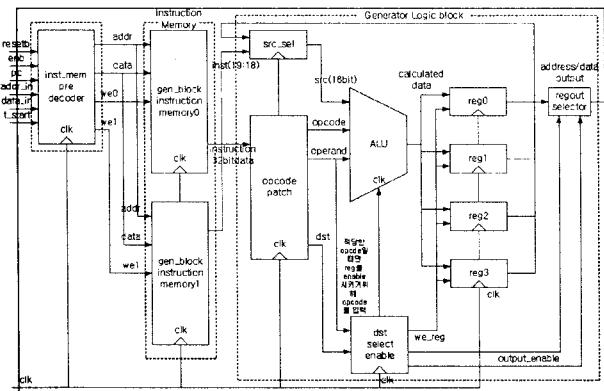
과 data를 저장하고, 테스트 수행 시에는 명령어를 읽도록 메모리 컨트롤러로 구성된다. 내부 메모리는 256×32 bit로 이루어져 있다. Opcode Patch block은 메모리에서 출력된 32bit 명령어를 해석하고, ALU에서 빠른 PC 연산을 수행하도록 각종 제어 신호를 생성한다. Repeat 같은 프로그램 루프를 수행하기 위해서는 카운터 회로가 필요하고, Jump, Call, Return 같은 분기 명령어를 수행하기 위해서는 스택 메모리가 있어야 한다. 카운터 회로와 스택 메모리는 프로그램 루프의 오버헤드를 없애고, 한 사이클에 명령을 수행하도록 ALU 연산에 맞춰 작동해야 한다. 서브루틴 호출이나 루프를 수행할 때 현재 PC를 저장하기 위한 스택 메모리는 16 레벨로 설계하였다. 또한 Repeat 명령어를 수행하기 위해 16bit counter를 설계하였다. (그림 2-2 참조) 연산된 Program Counter는 레지스터에 저장되었다가 Timing Rate에 맞춰 각 Generator chip으로 출력된다.



<그림 2-2. Sequence controller의 블록도>

2.2.2 Pattern Generator

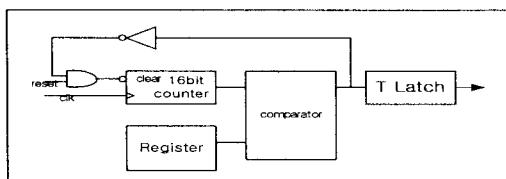
Sequencer와 마찬가지로 입력 단은 테스트 수행 전에 내부 메모리에 명령어와 데이터를 저장하고 테스트 수행 중에는 PC에 맞춰 메모리를 읽도록 테스트 제어 시스템의 bus와 연결된 메모리 컨트롤러로 설계되었다. 내부 메모리는 256×32 bit로 이루어져 있다. 내부 메모리를 읽어서 address 또는 data pattern을 생성하는 연산부는 한 Test Cycle 동안 하나의 instruction을 수행하도록 pipe line 구조로 설계하였다. ALU 부분에서는 address와 data를 생성하기 위해 산술 연산과 논리 연산을 수행하도록 10개의 명령어를 제공한다. 각 연산에 필요한 src는 내부 메모리나 내부 레지스터의 데이터가 될 수 있으며 src가 내부 레지스터의 데이터인 경우 opcode patch를 수행할 때 내부 레지스터에 저장된 src data를 가져오면서 발생하는 클럭 손실을 없애기 위해 opcode prepatch를 수행하여 opcode patch와 동시에 src data를 읽도록 설계하였다. Pattern generator에서의 연산은 메모리에 인가할 address와 data Pattern 생성을 위한 연산이 추가 되며 이 연산을 위해 내부에 네 개의 레지스터를 제공한다. ALU에서 연산을 수행한 뒤, 연산된 데이터 값이 저장될 dst 레지스터를 선택하기 위한 dst register selector block이 있으며 레지스터에 저장된 data와 address pattern을 Timing Rate에 맞춰 DUT로 출력하도록 output register selector block을 출력 단에 설계 하였다. (그림 2-3 참조)



<그림 2-3. Pattern Generator의 블록도>

2.2.3 Timing Generator

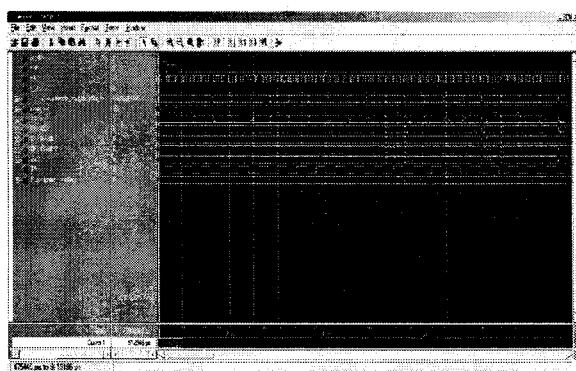
Timing Generator에서는 메모리 테스터의 Test Cycle을 제어하기 위한 Timing Rate를 생성한다. Timing Rate는 선택된 내부 레지스터 값과 카운터 값을 비교하여 생성해 내며 100MHz의 System clock을 사용할 경우 최저 1.5Khz에서 최대 16.6MHz의 Test Cycle을 생성할 수 있다. 최대 Test Cycle은 Sequence와 Pattern generator에서 연산에 소요되는 최저 Clock Phase에 의해 결정된다.



<그림 2-4. Timing generator의 submodule>

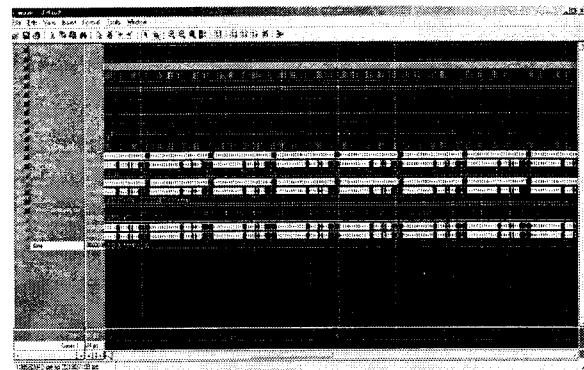
2.3 설계한 Chip의 검증

본 논문에서 설계한 칩은 Xilinx ise 6.1을 이용하여 Xilinx사의 Vertex xcv 300으로 구현 하였고, Model_sim XE 5.7C를 사용하여 검증 하였다. 그림2-7은 테스트 준비를 위해 내부 메모리에 명령어를 쓰는 과정이다.



<그림 2-5 chip 내부 메모리에 data 저장>

설계한 칩의 전체 검증을 위해 Scan test algorithm을 작성하여 내부 메모리에 저장하고 이를 수행한 결과를 그림 2-6에 나타내었다. 지면 관계상 32bit address와 16bit data 채널의 출력 패형을 모두 나타낼 수 없어서 일부만 표시하였다. 각 instruction으로 연산된 address 혹은 data의 패형이 출력되는 것을 확인할 수 있다.



<그림 2-6. 테스트 수행 과정중 output wave form>

3. 결 론

본 논문은 메모리 테스터용 프로세서 칩을 설계하고 간단한 test algorithm으로 동작을 검증하였다. 설계한 칩은 여러 가지 복잡한 알고리즘을 사용자가 임의로 구현할 수 있도록 함으로서, 유연하고 효율적인 테스트를 가능하게 하였다. 각 기능 블록으로 구현함으로 DUT에 따라 테스트 보드를 유연하게 구현하여 제조비 절감 효과도 볼 수 있으며 향후 수정이 용이하도록 하였다.

본 논문에서 설계한 Chip은 VHDL로 코딩하였으며 timing simulation을 수행 하였다. Address Matrix나 Scrambler, Auto refresh 등의 기능 블록과 메모리의 Pass/Fail을 판별하기 위한 회로가 추가되어야 하며, 고속 메모리 테스트를 위한 칩의 최적화 및 성능향상에 관한 연구가 계속 진행되어야 할 것이다.

참 고 문 헌

- [1] A. J. van de Goor, "Testing Semiconductor Memories," Theory and practice, John Wiley & Sons, 1991.
- [2] M. S. Abadir, and J. K. Reghbati, "Functional Testing of Semiconductor Random Access Memories," ACM Computing Surveys, 15,pp. 175-198. 1983
- [3] P. K. Veenstra, "Random Access Memory Testing: Theory and Practice," IEE Proceedings G, 1988.
- [4] M. Marinescu, "Simple and Efficient Algorithms for Functional RAM Testing," proc. of IEEE International Test Conference, 1982.
- [5] G. Perry, "The Fundamentals of Digital Semiconductor Testing," Soft Test Inc. 1996.
- [6] 강성호, 김규철, 소병세, 홍성제 공저, "메모리 테스트", 대영사, 2001
- [7] Y.W. Yoon, J.S. Moon, B.M. Moon, S.K. Ahn, and D.K. Jeong, "A study on the Architecture and Logic Block Design of FPGA," 전자공학회 논문지, pp.140-151, 1996.