반도체 소자의 논리결함검출을 위한 pattern generator 회로설계에 관한 연구

노 영동*, 김 준식*

*호서대학교 전자공학과

초록

반도체 소자의 집적도의 발전에 따라 생산과정에서의 기능적인 오류 검사 소요시간이 증가하게 되어 비용절감에 커다란 장애 요인이 되고 있다. 이러한 문제점을 효과적으로 처리하기 위하여 일괄적인 패턴과 어드레스를 발생시키는 pattern generator를 연구하였다.

1. 서론

우리나라의 반도체 산업은 1983년 메모리 공정 사업에 국내 기업이 본격적으로 참여하여 급속한 발전을 이룩해 왔다. 발전에 따른 반도체 소자의 고집적화로 인해 테스트 소요시간이 증가하게 되었다. 이러한 고집적화 메모리에 대하여 검사를 할 경우 동일한 동작을 여러 가지 셀에 반복적으로 수행한다면 검사소요 시간이 증가함에 비례하여 생산공정에 따른 비용도 증가하게 된다. 이러한 문제점을 효과적으로 처리하기 위해선 일괄적인 패턴을 발생시켜 단시간에 기능적인 오류를 검사하여야 한다. 이에 사용되는 장비를 pattern generator라 한다. [1][2]

본 연구에서는 패턴을 발생시키기 위한 전체적인 개념과 그에 따른 설계 방법을 블록도와 시뮬레이션 결과를 통해 검증한다.

2. 실험 방법

PG의 주요작업은 주소 생성과 데이터 생성이다. 일반적으로 주소 생성은 카운터(counter), LFSR (Linear Feedback Shift Register) 또는 마이크로프로세서를 이용하며, 데이터 생성은 주소 또는 유한상태기 (finite state machine)를 이용한다. [2]

그림 1 은 pattern generator의 전체 블록이다. 컴퓨터에서의 데이터 입력을 받아들여 PC data bus를 통하여 PG의 main memory 와 program counter로 데이터가 들어간다. Program counter로 들어간 데이터는 PG main memory의 어드레스를 발생시키고 PG의 main memory로 들어간 데이터는 program Counter에서 발생된 어드레스 메모리 라인에 의해 할당된 어드레스로 데이터가 들어가 address part, data part, index part 그리고 control part로 입력될 memory file 신호를 발생하게 된다. 각각의 발생되어진 memory file은 address part에서는 테스트에 사용될 주소를 생성하여 순차적으로 제공하게 되고, data part에서는 address part에서 발생된 주소에 들어갈 데이터를 발생하게 된다. Index part에서는 memory의 출력을 제어를 하게 되고, control part에서는 PG를 제어할 명령과 메모리 제어에 관련된 신호를 발생하게 된다. 또 main

memory에서 출력된 신호는 MOD와 MUTC 신호 발생 부분으로 들어가 MOD 24개 신호와 MTC 24개 신호를 발생하게 된다.

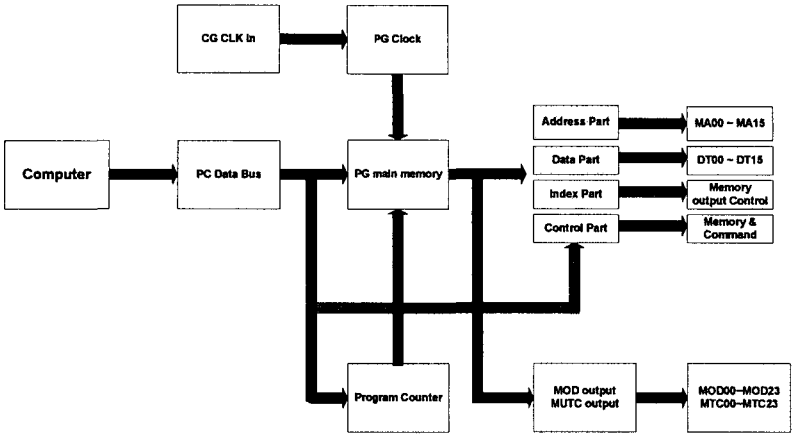


그림 1. Pattern generator의 전체 구성도

PG의 각 부분별 기능을 알아보겠다. 그림 2 는 PG에서 가장 중요한 역할을 하는 program counter이다. 메인 메모리에 주소를 할당하고 그 할당된 주소로 PC로부터 data가 입력되어 메인 메모리에서 발생되는 신호인 MF 신호와 data신호를 발생하게 된다.

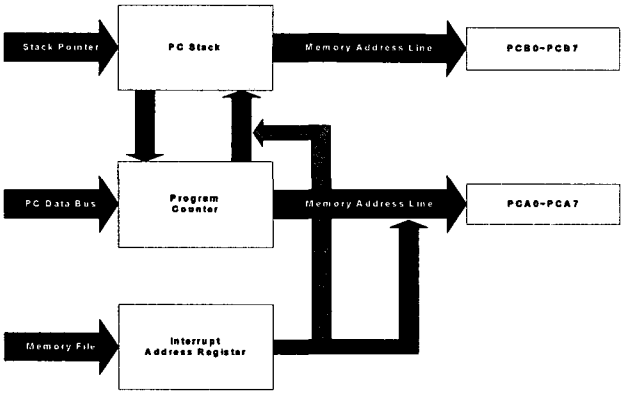


그림 2. Program counter 구성도

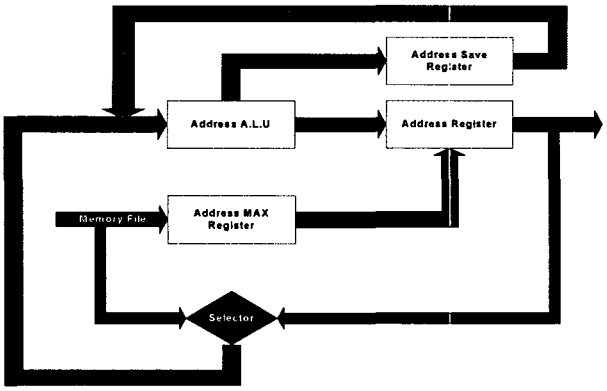


그림 3. Address part 구성도

그림 3 은 address part의 구성도이다. 메모리에서 생성된 memory file이 입력으로 들어가 전반적인 신호를 구성하게 되는데, 이 부분에서는 검사할 소자에 데이터가 입력될 어드레스를 발생하게 된다.

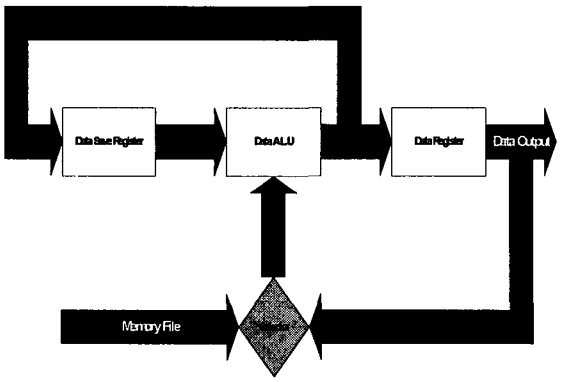


그림 4. Data part 구성도

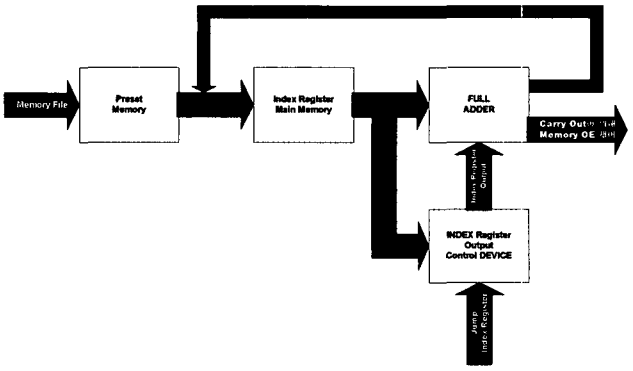


그림 5. Index part 구성도

그림 4는 PG의 중요한 기능 중 하나인 어드레스에 들어갈 data를 출력하는 data part의 구성도이다. memory file이 입력되면 data A.L.U에서 정의된 논리식에 의해서 16 bit data값이 출력하여 테스트할 device에 가해진다. 그림 5는 index part의 구성도이다. MF신호를 입력으로

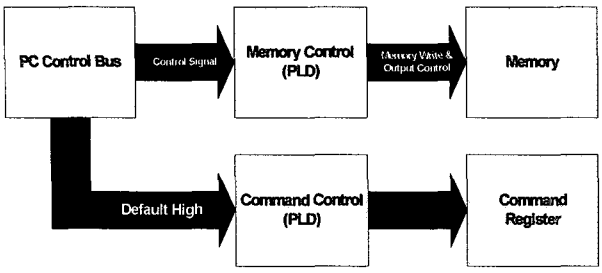


그림 6. Control part 구성도

받아들여 index register jump 신호를 발생시킨다. 이 신호에 의해 memory address line의 출력과 main memory output enable를 제어하여 PG에서의 입출력의 충돌을 제어하게 된다.그림 6은 PG의 제어를 담당하는 control에 관련된 구성도 이다. PC control Bus에서의 입력으로 인해 main memory제어신호와 program counter에서 발생되는 memory address line을 제어하게 된다.

3. 실험결과 및 고찰

그림 7 은 program counter 의 simulation 결과이다. 점선 부분은 각 조건에 의해 memory address line 인 PCA[7~0] 와 PCB[7~0] 의 출력을 나타낸다. 입력된 데이터 즉 counter 에서 발생되는 신호에 의해 main memory 의 address line 을 발생하는 program counter 는 main memory 에서 발생되는 신호인 MF 와 data 신호에 의해 interrupt address 를 제어하며 program counter 에서 발생하는 address line 과 main memory 와의 충돌을 방지하게 된다.

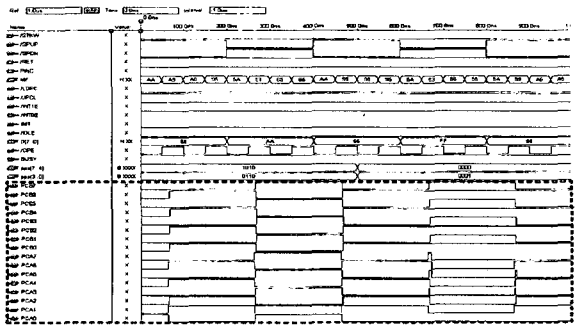


그림 7. Program counter의 simulation 결과

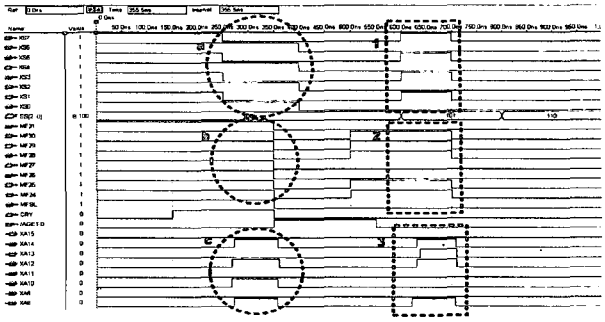


그림 8 Address part의 simulation 결과

그림 8 은 address part의 simulation 결과이다. MFSL 신호에 의해 이전신호를 유지할 것인지 아니면 새로운 신호를 사용할 지를 결정하게 된다. 결정된 신호는 address ALU device 에 입력되어 S0, S1, S2 신호에 의해 ALU device에서 정의해 놓은 연산을 하게 된다. 이 연산에 의해 출력되는 신호는 MF신호와 AND LOGIC 연산을 하여 최종 신호를 출력하게 된다. 그림 8의 결과에서 이 부분에 대해 설명한다면, 점선부분에서 보인 것처럼 a(1)의 입력으로 발생되는 신호와 b(2)의 입력신호와 AND 연산되어 최종결과인 c(3)가 출력하게 된다.

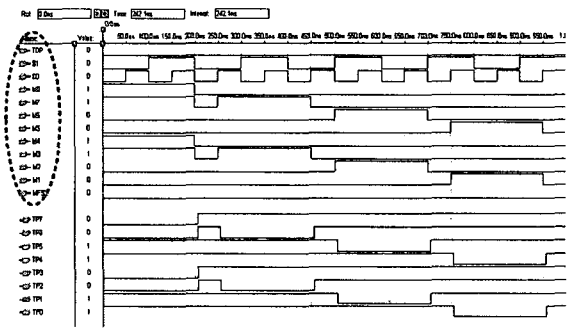


그림 9. Data part의 simulation 결과

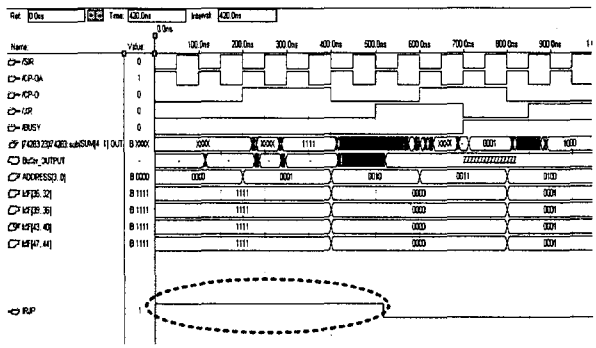
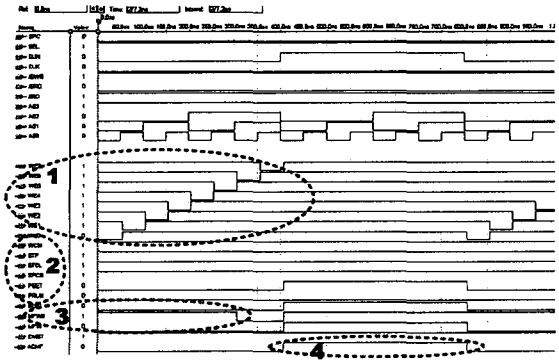


그림 10. Index part의 simulation 결과

그림 9은 data part의 simulation 결과이다. TP 라는 PAL device에 입력되는 MF 신호와 그 외의 제어 신호들이 정의해 놓은 논리 연산에 의해서 TP7~TP0 신호를 발생하게 된다. 그림 10은

index part의 simulation 결과이다. 여기에선 입력단의 신호에 의해 IRJP신호가 발생되는 과정을 중점으로 시뮬레이션 하였다. 그림 11은 control part의 simulation 결과이다. 각 입력에 의해 ①, ②, ③, ④를 발생하게 된다.



①은 memory write enable 신호에 대한 출력인데, 이 신호는 memory file write PAL device에서 정의된 논리 연산에 의해서 발생된다. ③, ④에서 발생되는 신호 또한 이 소자를 통하여 발생된다.

②의 출력은 command register 의 제어 신호이다.

그림 11. Control part의 simulation 결과

이 신호는 control PAL device 에 정의된 논리 연산을 통하여 발생하게 된다

③은 main memory part의 버퍼 방향을 결정하는 신호선 이고,

④는 memory output enable 제어 신호 발생부분이다

4. 결론

고집적화되고 기능이 복잡해진 반도체 소자 정상 동작 여부를 효율적으로 테스트하기 위해서는 테스트 패턴 생성기를 효율적으로 설계하여 빠른 시간 안에 테스트가 이루어지도록 하는 것이 중요하다.

본 연구에서는 pattern generator 를 연구하여 반도체 칩의 생산라인에서의 생산성 증대 및 품질 향상에 중점을 두었다. functional test 를 빠른 시간에 수행하기 위해 검사 할 소자의 특성에 따라 검사 패턴을 발생하게 된다. 이 연구에서는 알고리즘 패턴 적용을 위해 counter 를 사용하여 address 를 발생 시켰다. 또, 전체적인 구성에서 각 부분이 어떤 기능을 하는지에 중점을 두었다. control part 의 command part 에 의해 전체적인 PG 의 제어가 이루어짐을 확인 했으며, 각 부분별 출력 되는 신호가 어떤 과정을 거쳐 발생되는지를 알 수 있었다. 64Mb 메모리에 모두 0 을 쓴다고 하고 한 개의 셀에 0 을 쓰는 동작에 4 줄의 프로그램이 필요하다면 2 억 5600 만 라인의 길이를 갖는 프로그램이 필요하다. 이 pattern generator 를 사용하면 이러한 과정의 효율성을 증대 시킬수 있으며 보다 정확한 오류를 검출 할 수 있게 된다. 이로 인해 테스트 소요시간과 생산 비용이 감소되어 산업화의 경쟁력이 될 것이다.

후기

본 연구는 한국과학재단지정 호서대학교 반도체제조장비국산화연구센터의 연구지원으로 이루어진 것임.

참고문헌

[1] A.Stevens, "Introduction to Component Testing" Addison-Wesley Publishing Company, 1986.

[2] 강성호, 김규철. 소병세, 홍성제 공저 겔貸霽 테스트 대영사, 2001, p13~66.