

(19)대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl. ⁷ G01R 31/26	(45) 공고일자 (11) 등록번호 (24) 등록일자	2005년08월10일 10-0506770 2005년07월29일
--	-------------------------------------	--

(21) 출원번호 (22) 출원일자	10-2000-0035589 2000년06월27일	(65) 공개번호 (43) 공개일자	10-2001-0015068 2001년02월26일
------------------------	--------------------------------	------------------------	--------------------------------

(30) 우선권주장 09/340,371 1999년06월28일 미국(US)

(73) 특허권자 가부시키가이샤 어드밴티스트
일본국 도쿄도 네리마구 아사히초 1초메 32반1고

(72) 발명자 턴퀴스트제임스엘란
미국95054캘리포니아주산타클라라스코트볼러바드3201

스가모리시게루
미국95054캘리포니아주산타클라라스코트볼러바드3201

야모또히로아끼
미국95054캘리포니아주산타클라라스코트볼러바드3201

(74) 대리인 장수길
구영창

심사관 : 권호영

(54) 이벤트 기준 반도체 테스트 시스템

요약

전자 설계 자동조작(EDA) 환경에서 설계된 LSI 장치에 대하여 컴퓨터에서 실행된 장치 논리 시뮬레이션의 결과인 데이터에 따라 생성된 테스트 패턴을 발생하고, LSI 장치를 테스트하며, 그 테스트 결과를 EDA 환경으로 피드백하는 반도체 테스트 시스템이 제공된다. 반도체 테스트 시스템은 피테스트 LSI 장치의 설계 단계에서 장치 논리 시뮬레이션을 실행함으로써 얻은 이벤트 데이터를 오브젝트 코드로 변환하는 컴파일러, 컴파일러에 의해 변환된 이벤트 데이터를 저장하는 이벤트 파일, 타이밍에 관하여 이벤트 파일로부터의 이벤트 데이터를 저장하는 이벤트 메모리, 이벤트 메모리로부터의 이벤트 데이터에 따라 테스트 패턴을 발생하고 그 테스트 패턴을 피테스트 LSI 장치에 적용하는 수단, 피테스트 LSI 장치의 응답 출력을 평가하고 최종 평가 데이터를 저장하는 결과 데이터 파일, 및 결과 데이터 파일에 저장된 데이터에 따라 LSI 장치의 설계를 평가하는 수단을 포함한다.

대표도

도 7

색인어

반도체 설계, 컴파일러, 논리 시뮬레이션, EDA 환경

명세서

도면의 간단한 설명

도 1은 종래 기술에 있어서 반도체 테스트 시스템(LSI 테스터)의 기본 구성을 도시한 블록도.

도 2는 EDA 환경에서 CAD 시스템을 통해 얻은 LSI 설계 데이터에 대해 논리 시뮬레이션을 실행함으로써 생성된 덤프 파일을 사용하는 종래 반도체 테스트 시스템에 의해 LSI를 테스트하는 전체적인 관계를 도시한 개략도.

도 3은 종래 반도체 테스트 시스템에 의해 사이클 기준 포맷으로 테스트 패턴을 발생하는 패턴 데이터 및 타이밍 데이터의 기술 일예를 도시한 개략도.

도 4는 덤프 파일에서 얻은 데이터로 기술된 이벤트 기준 파형의 화상을 도시하고 EDA 환경에서 얻은 LSI 설계 데이터에 대해 논리 시뮬레이션을 실행함으로써 덤프 파일을 발생하는 처리를 도시한 블록도.

도 5는 본 발명의 반도체 테스트 시스템의 이벤트 기준 테스트 패턴을 발생하는 기술의 일예와 종래 반도체 테스트 시스템의 사이클 기준 테스트 패턴을 발생하는 기술의 일예를 비교하는 도면.

도 6은 EDA 환경에서 얻은 LSI 설계 평가 데이터에 따라, 종래 반도체 테스트 시스템에 의해, EDA 환경과 사이클 기준 테스트 패턴을 발생하는 테스트 시스템 사이의 주로 필요 소프트웨어에 의한 관계를 도시한 도면.

도 7은 도 6의 종래 기술과의 비교를 위해, 주로 필요 소프트웨어에 의한, 본 발명의 이벤트 기준 테스트 시스템과 DEA 환경 사이의 관계를 도시한 블록도.

도 8은 EDA 환경에서 얻은 LSI 설계 평가 데이터에 따라, 종래 반도체 테스트 시스템에 의해 사이클 기준 테스트 패턴을 발생하는 테스트 시스템과 EDA 환경 사이의 주로 필요 소프트웨어에 의한 관계를 상세히 도시한 블록도.

도 9는 도 8의 종래 기술과 비교를 위해, 주로 필요 소프트웨어에 의한 본 발명의 이벤트 기준 테스트 시스템과 EDA 환경 사이의 관계를 도시한 블록도.

도 10은 EDA 환경에서 얻은 LSI 설계 평가 데이터에 따라 테스트 패턴을 발생하고 피테스트 장치를 테스트하는 본 발명의 반도체 테스트 시스템과 종래 기술의 반도체 테스트 시스템을 비교하는 블록도.

도 11은 EDA 환경에서 얻은 LSI 설계 평가 데이터에 따라 테스트 패턴을 발생하고 피테스트 장치를 테스트하는 EDA 환경과 본 발명의 테스트 시스템 사이의 전체 관계 및 본 발명의 반도체 테스트 시스템의 기능을 도시한 블록도.

도 12는 본 발명의 이벤트 기준 반도체 테스트 시스템의 구성 일예를 도시한 블록도.

<도면의 주요 부분에 대한 부호의 설명>

57 : 덤프 파일

71 : EDA 환경

72 : 변환 소프트웨어

73 : 컴파일러

74 : 오브젝트 파일

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 대규모 집적(LSI) 회로 등의 반도체 집적 회로를 설계하고 테스트하는 반도체 설계 및 테스트 시스템에 관한 것으로, 특히 CAD(computer aided design) 처리를 통해 반도체 IC의 설계 단계에서 발생된 논리 시뮬레이션 데이터로부터 직접 발생하는 이벤트 기준 테스트 패턴을 발생하는 것에 의해 반도체 IC를 테스트하는 이벤트 기준 반도체 IC 설계 및 테스트 시스템에 관한 것이다.

도 1은 반도체 집적 회로(이하, "IC 장치", "피테스트 LSI" 또는 "피테스트 장치"라 함)를 테스트하는 반도체 테스트 시스템의 일예를 개략적으로 도시한 블록도이다.

도 1의 예에서, 테스트 프로세서(11)는 반도체 테스트 시스템 내에 마련되어 그 테스트 시스템의 동작을 테스트 버스를 통해 제어하는 전용의 프로세서이다. 테스트 프로세서로부터의 패턴 데이터에 따라, 패턴 발생기(12)는 타이밍 발생기(13) 및 파형 포맷터(14)에 타이밍 데이터 및 파형 데이터를 각각 제공한다. 테스트 패턴은 패턴 발생기(12)로부터의 파형 데이터 및 타이밍 발생기(13)로부터의 타이밍 데이터를 사용하여 파형 포맷터(14)에 의해 발생되고, 드라이버(15)를 거쳐 피테스트 장치(device under test)(DUT)(19)에 공급된다.

테스트 패턴으로부터 기인하는 DUT(19)로부터의 응답 신호는 아날로그 비교기(16)에 의해 소정의 스레시홀드 전압 레벨을 참조하여 논리 신호로 변환된다. 논리 신호는 논리 비교기(17)에 의해 패턴 발생기(12)로부터의 기대값과 비교된다. 논리 비교의 결과는 DUT(19)의 어드레스에 대응하는 고장 메모리(fail memory)(18)에 저장된다. 드라이버(15), 아날로그 비교기(16) 및 피테스트 장치의 핀 변경을 위한 스위치(도시하지 않음)는 핀 전자 장치(20)에 설치된다.

대규모 집적(LSI) 회로 등의 반도체 집적 회로를 개발하는 처리에 있어서, 거의 언제나 컴퓨터 이용 설계(CAD) 툴을 이용하는 설계 방법이 채용된다. 그러한 CAD 툴을 사용하는 설계 환경은 전자 설계 자동조작(EDA) 환경이라고도 한다.

그러한 EDA 환경에서의 반도체 개발 처리에 있어서, 원하는 반도체 회로는 VHDL 및 Verilog 등의 하드웨어 기술 언어를 사용하여 LSI로 제조된다. 또한, 이 처리에 있어서, 그렇게 설계된 반도체 회로의 기능은 장치 논리 시뮬레이터(device logic simulator)라고 하는 소프트웨어 시뮬레이터를 사용하여 컴퓨터 상에서 평가된다.

장치 논리 시뮬레이터는 테스트 데이터(벡터)가 의도한 반도체 회로를 나타내는 장치 설계 데이터(장치 모델)에 적용되는 통상 테스트벤치(testbench)라고 하는 인터페이스를 구비하고, 의도한 반도체 회로의 결과 응답이 평가된다.

LSI 회로의 설계 단계후, 실제 LSI 장치를 생산하여, LSI 장치가 의도한 기능을 적절히 실행하는가를 판정하기 위해 LSI 테스터 등의 반도체 테스트 시스템에 의해 테스트한다. 위에서 기술한 바와 같이, LSI 테스터는 피테스트 LSI 장치에 테스트 패턴(테스트 벡터)을 공급하고, LSI 장치의 결과 출력과 기대 데이터를 비교하여 LSI 장치의 통과/고장을 판정한다.

더 높은 수준의 기능 및 밀도를 갖는 LSI 장치를 테스트하기 위해, LSI 장치에 적용될 테스트 패턴도 그에 따라 복잡하고 장황하게 되어, 테스트 패턴 작성에 있어서 작업 부담 및 작업 시간이 매우 커지게 된다.

따라서, 반도체 집적 회로의 전체 테스트 효율 및 생산성을 개선하기 위해, 반도체 집적 회로의 실제 테스트에 있어서 장치 논리 시뮬레이터의 실행을 통해 생산된 데이터를 사용하려는 시도가 있었다. 이것은 실제 반도체 집적 회로 테스트에 있어서 LSI 테스터에 의해 실행된 테스트 과정은 상술한 CAD 처리시 반도체 회로의 설계 데이터를 테스트할때 장치 논리 시뮬레이터에 의한 테스트 과정과 실질적으로 유사성을 갖기 때문이다.

예를 들면, 장치 논리 시뮬레이션을 실행하는 것에 기인하는 데이터(덤프 파일)를 사용하여 의도한 반도체 집적 회로를 테스트하기 위해 LSI 테스터에 대하여 테스트 패턴 및 기대치 패턴을 작성하려는 시도가 있다. 도 2는 반도체 집적 회로의 설계 단계와 반도체 집적 회로의 테스트 단계 사이의 전체적인 관계를 도시한 개략도이다. 이 예는 시스템-온-칩(SoC)(23) 등의 대규모 집적 회로(LSI)가 전자 설계 자동조작(EDA) 환경하에서 설계되는 상황을 보여준다.

EDA 환경하에서 반도체 집적 회로(23)를 설계한 후, 설계 데이터 파일(25) 및 테스트 데이터 파일(33)이 얻어진다. 여러 가지 데이터 변환 처리를 거쳐서, 설계 데이터는 설계된 반도체 집적 회로의 각 게이트를 나타내는 물리적 레벨 데이터로 변환된다. 물리적 레벨 데이터에 기초하여, 실제 집적 회로(29)가 반도체 집적 회로 생산 처리(실리콘 처리)에서 만들어진 다.

그렇게 생산된 집적 회로는 반도체 테스트 시스템(30)에 의해 테스트된다. 집적 회로의 설계 단계를 통해 구한 테스트 데이터를 사용하여 테스트벤치(34)에 의해 논리 시뮬레이션을 실행하는 것에 의해, 집적 회로의 입출력 관계를 나타내는 데이터 파일(35)이 생성된다. 그러한 데이터 파일의 일례는 Verilog의 VCD(Value Change Dump)이다.

다음에 상세히 설명하는 바와 같이, 포맷 변환 처리는 변환 소프트웨어(37)에 의해, 이벤트 기준 포맷으로 기술된 VCD 데이터 파일(35)이 사이클 기준 포맷의 테스트 신호로 변환되도록 실행된다. 결과적으로, 사이클 기준의 테스트 패턴은 반도체 테스트 시스템(30)의 파일(38)에 저장된다. 하드웨어 테스터(39)는 장치 기능 등을 테스트하기 위해 테스트 패턴을 피테스트 장치(29)에 적용한다.

위에서 간략히 언급한 바와 같이, 그러한 장치 시뮬레이션 데이터에 있어서, 장치 모델의 최종 출력(기대치 패턴)뿐만 아니라 장치 모델에 적용될 테스트 패턴은 이벤트 기준 포맷으로 표현된다. 여기서, 이벤트 기준 데이터는 시간의 경과를 참고로 테스트 패턴의 논리 "1"에서 논리 "0" 또는 그 반대로의 변화점(이벤트)을 표현한다. 일반적으로, 그러한 시간 경과는 소정의 기준점으로부터의 기간(절대 시간 차분) 또는 이전 이벤트로부터의 기간(상대 시간 차분)으로 표현된다.

그와 반대로, 실제 LSI 테스터(반도체 테스트 시스템)에서는 테스트 패턴이 사이클 기준 포맷으로 기술된다. 사이클 기준 포맷 데이터에 있어서, 테스트 패턴의 각 변수는 LSI 테스터의 각 테스트 사이클(테스터 레이트)에 대하여 정의된다. 따라서, 다음에 상세히 기술하는 바와 같이, 대표적인 LIS 테스터에 있어서, 대응하는 테스트 사이클의 테스트 패턴은 테스트 사이클(테스터 레이트), 파형(파형의 종류 및 에지 타이밍) 및 테스트 패턴 데이터의 벡터의 기술에 기초하여 형성된다.

상기에서와 같이, 기존의 LSI 테스터는 사이클 기준의 데이터를 취급하는 반면, EDA 환경을 통해 생성된 데이터는 이벤트 기준이다. 따라서, 반도체 장치의 설계 단계에서 얻은 CAD 데이터에 기초하여 실제로 생산된 반도체 장치를 테스트하기 위한 테스트 패턴을 효과적으로 만들기 위해서는 이벤트 기준 데이터를 사이클 기준 데이터로 변환할 필요가 있다.

따라서, 상술한 도 2에 있어서, 변환 소프트웨어(37)는 반도체 장치의 설계 단계에서 장치 논리 시뮬레이션을 실행하는 것에 의해 구해진 덤프 파일(35)로부터 패턴 데이터 및 타이밍 데이터를 추출한다. 변환 소프트웨어(37)는 추출된 데이터를 사이클 기준 데이터로 변환한다. 그렇게 사이클 기준 포맷으로 변환된 패턴 데이터 및 타이밍 데이터는 테스트 사이클(테스터 레이트), 파형(파형의 종류, 에지 타이밍) 및 벡터에 관한 기술을 구비한다. 패턴 데이터 및 타이밍 데이터는 테스트 시스템(30)의 패턴 파일(38)에 저장된다.

패턴 파일(38)에 저장된 데이터에 따라, 사이클 기준 테스트 시스템인 테스터(39)는 DUT(29)에 적용될 테스트 패턴을 발생한다. 상술한 바와 같이, 기존 LSI 테스터를 위한 테스트 패턴은 패턴 데이터 및 타이밍 데이터에 포함되어 있는 테스트 사이클(테스터 레이트) 데이터, 파형(파형의 종류, 에지 타이밍) 데이터 및 벡터 데이터를 사용하여 포맷된다.

도 3은 사이클 기준 데이터의 구조를 도시한 것이다. 도 3의 예에서, 사이클 기준 테스트 패턴은 테스트 패턴 파일(41) 및 테스트 플랜 파일(42)에 저장된다. 이들 파일로부터의 데이터를 사용하는 것에 의해, 파형(45)으로 도시한 의도의 테스트 패턴은 도 1의 파형 포맷터(4)에 의해 포맷된다. 여기서, 테스트 패턴 파일(41)은 벡터 데이터(패턴 데이터라 함)을 저장하고, 테스트 플랜 파일(42)은 타이밍 데이터, 즉 테스트 사이클(테스터 레이트) 및 파형(파형의 종류 및 에지 타이밍)에 관한 데이터를 저장한다.

도 3에 있어서, 의도한 파형(45)을 포맷하기 위해, "1", "0", "X" 등의 벡터 데이터는 패턴 기술(46)에 세트되고, 테스트 사이클 마다의 파형 및 지연 데이터는 타이밍 데이터 기술(47)에 세트된다. 상기 데이터는 테스트 사이클마다 정의될 필요가 있다.

한편, 장치 논리 시뮬레이터를 실행하는 것에 기인하는 데이터 등의 EDA 환경하의 LSI 설계 단계에서 생성된 데이터는 상술한 바와 같은 이벤트 기준 데이터 구조를 갖고, 이것은 도 3의 사이클 기준 데이터의 데이터 구조와 실질적으로 상이하다. 그러한 상황은 도 4를 참조하여 더 설명하기로 한다. 이 예에서, 도 4의 파형(58)은 덤프 파일(57)로부터의 이벤트 데이터를 나타내고, 이것은 이벤트 타이밍 시퀀스로 논리 시뮬레이션을 실행하는 것으로부터 구해진다. 즉, 덤프 파일의 이벤트 데이터는 시간에 대한 세트 및 리셋의 리스트에 의해 파형(58)의 각 변화점을 표현한다.

도 4를 참조하여, 덤프 파일을 형성하는 처리에 관하여 간략히 설명한다. EDA 환경하의 LSI 설계에 있어서, 설계된 데이터는 다른 레벨(계층)의 기술로 표현된다. 더 높은 레벨에 있어서, 더 많은 기능 기술이 사용되고, 이것은 지정된 데이터의 기술이 반도체 장치의 물리적 구조의 게이트의 레벨로 되는 더 낮은 레벨로 변환된다. 도 4에 있어서, RTL 파일(54)은 레지스터 변환 언어(RTL)로 기술된 설계 데이터를 저장한다. RTL 설계 데이터는 변환 소프트웨어(37)에 의해 넷 리스트 파일(56)에 저장되는 넷 리스트 데이터로 변환된다. 넷 리스트는 RTL 설계 데이터보다 반도체 장치의 물리적 구조에 더 가까운 게이트 중에서 접속 관계를 기술하는 데이터의 종류이다.

그러한 설계 데이터에 기초하여, 테스트벤치(51)는 테스트 벡터를 예를 들면, HDL 시뮬레이션(52) 또는 게이트 레벨 시뮬레이션(53)에 제공하여 설계 데이터를 검증한다. 설계 데이터 평가의 결과, 상술한 덤프 파일에는 입력 이벤트, 그 입력 이벤트에 대응하는 출력 이벤트 및 그러한 이벤트의 시간을 기술하는 데이터가 제공된다.

상기에서와 같이, 설계된 LSI를 테스트하기 위한 테스트 패턴은 LSI의 CAD 설계 단계에서 얻은 데이터, 예를 들면 LSI 장치의 설계 단계에서 생성된 장치 논리 시뮬레이터를 실행하는 것에 의해 얻은 설계 평가 데이터를 사용하여 생성된다. 그러나, 그러한 처리는 상이한 데이터 구조(이벤트 기준 대 사이클 기준) 사이에서의 테스트 패턴 변환을 수반하기 때문에, 또한 LSI 테스트의 하드웨어 및 소프트웨어에 있어서의 기능적 제한때문에, 이벤트 기준 설계 평가 데이터는 사이클 기준 테스트 패턴 데이터로 완전히 변환될 수 없다.

따라서, 설계 평가 데이터로부터 변환된 테스트 패턴을 사용하여 반도체 장치를 테스트함에 있어서, 피테스트 반도체 장치에서의 문제가 항상 발견되는 것은 아니다. 즉, 결함 검출율(디버그 효율)이 줄어들 수 있다. 더 완전한 테스트 패턴으로 변환하기 위해서는 더 많은 시간과 더 심한 작업 부하가 투자되어야 하여, 테스트 비용이 상승하고(오버헤드) 테스트 효율이 저하한다.

또한, 종래 반도체 테스트 시스템에 있어서, 상기에서와 같이, 테스트 사이클, 테스트 신호 파형, 타이밍 및 논리 벡터 등의 여러가지 파라미터를 패턴 데이터 및 타이밍 데이터에 따로따로 기술해야만 한다. 따라서, 따로 마련된 기술에 기초하여 테스트 패턴을 발생하기 위해서는 복잡하고 고가의 하드웨어 및 소프트웨어를 사용해야만 한다.

따라서, 산업계에서는 EDA 환경에서 구한 평가 데이터와 동일한 고려 방식에 기초하여, 테스트 패턴을 발생하고, 피테스트 반도체 장치를 평가하며, 그 테스트 결과를 EDA 환경으로 피드백할 수 있는 고효율의 반도체 설계 및 평가 시스템이 필요하다. 또한, 산업계에서는 그러한 새로운 개념을 갖는 새로운 반도체 테스트 시스템을 사용하여 LSI 설계 및 그의 평가에 필요한 기간을 더 줄이는 것이 요망된다.

발명이 이루고자 하는 기술적 과제

따라서, 본 발명의 목적은 전자 설계 자동조작(EDA) 환경하에서 LSI의 설계 단계에서 형성되는 논리 시뮬레이션 데이터와 유사한 데이터 구조(이벤트 기준)의 테스트 데이터 또는 피테스트 LSI 장치의 논리 시뮬레이션 데이터를 직접 사용하는 것에 의해 테스트 패턴을 발생하여, 피테스트 LSI 장치를 테스트하는 반도체 테스트 시스템을 제공하는 것이다.

본 발명의 다른 목적은 의도한 피테스트 LSI 장치의 설계 단계에서 구한 CAD 데이터에 기초하여 생성된 이벤트 기준 시뮬레이션 데이터를 직접 사용하는 것에 의해 피테스트 LSI 장치를 테스트하는 테스트 패턴을 생성할 수 있는 이벤트 기준 반도체 테스트 시스템을 제공하는 것이다.

본 발명의 또 다른 목적은 의도한 피테스트 LIS 장치의 설계 단계로부터 구한 CAD 데이터에 따라 작성된 이벤트 기준 시뮬레이션 데이터를 직접 사용하는 것에 의해 피테스트 LSI 장치를 테스트하기 위한 테스트 패턴을 생성할 수 있어, LSI 설계와 테스트 패턴 형성 사이의 턴 어라운드 시간을 실질적으로 저감하는 반도체 테스트 시스템을 제공하는 것이다.

본 발명의 또 다른 목적은 의도한 피테스트 LSI 장치의 설계 단계로부터 구한 CAD 데이터에 기초하여 작성된 이벤트 기준 시뮬레이션 데이터를 직접 사용하는 것에 의해 피테스트 LSI 장치를 테스트하기 위한 테스트 패턴을 생성할 수 있어, 반도체 테스트 시스템에 수반되는 소프트웨어 및 하드웨어를 단순화할 수 있는 이벤트 기준 반도체 테스트 시스템을 제공하는 것이다.

본 발명의 또 다른 목적은 피테스트 LSI 장치의 EDA 리서치 및 설계 환경과 직접 관련되고, 설계 데이터로부터 직접 발생된 테스트 패턴에 의해 피테스트 LSI 장치를 테스트할 수 있고, 테스트 결과를 DEA 리서치 및 설계 환경으로 직접 피드백할 수 있는 이벤트 기준 반도체 테스트 시스템을 제공하는 것이다.

본 발명의 또 다른 목적은 의도한 피테스트 LSI 장치의 설계 단계로부터 구한 CAD 데이터에 기초하여 생성된 이벤트 기준 시뮬레이션 데이터를 직접 사용하는 것에 의해 피테스트 LSI 장치를 테스트하기 위한 테스트 패턴을 생성할 수 있어 테스트 비용을 저감하고 테스트 효율을 증가시킬 수 있는 이벤트 기준 반도체 테스트 시스템을 제공하는 것이다.

발명의 구성 및 작용

본 발명의 반도체 테스트 시스템에 있어서, 테스트 패턴은 전자 설계 자동조작(EDA) 환경에서 설계된 LSI 장치 상에서 장치 논리 시뮬레이션을 실행하는 것을 통해 얻은 이벤트 기준 데이터인 덤프 파일 데이터를 직접 사용하는 것에 의해 생성된다. 장치 테스트는 그렇게 생성된 테스트 패턴을 피테스트 LSI 장치에 적용하는 것에 의해 실행되고, 설계된 LSI 장치의 평가는 테스트 결과에 따라 EDA 환경으로 피드백된다.

본 발명의 반도체 테스트 시스템에 있어서, 반도체 장치 테스트는 전자 설계 자동조작(EDA) 환경에서 컴퓨터의 도움으로 설계된 LSI 장치 상에서 장치 논리 시뮬레이션을 실행하는 것을 통해 얻어진 데이터를 직접 사용하는 것에 의해 높은 테스트 효율로 실행된다. 반도체 테스트 시스템은 피테스트 LSI 장치의 설계 단계에서 장치 논리 시뮬레이션을 실행하는 것에 의해 얻어진 이벤트 데이터를 상기 LSI 장치의 설계 데이터 상에 저장하는 이벤트 파일, 상기 이벤트 파일로부터 직접 얻어진 이벤트 데이터를 기준 클럭 사이클의 정수배를 나타내는 것과 상기 기준 클럭의 분수를 나타내는 것인 두 가지 다른 종류의 데이터로 저장하는 이벤트 메모리, 상기 이벤트 메모리로부터의 이벤트 데이터를 직접 사용하는 것에 의해 테스트 패턴을 발생시키고, 상기 테스트 패턴을 상기 피테스트 LSI 장치에 적용하는 수단, 상기 피테스트 LSI 장치에 적용된 상기 테스트 패턴을 표시하고, 상기 이벤트 파일 내의 상기 이벤트 데이터를 부분적으로 수정하는 것에 의해 상기 테스트 패턴을 실시간으로 변경하는 수단, - 상기 테스트 패턴에 있어서의 상기 변경은 이벤트의 특정 타이밍의 변경을 포함함 - 상기 피테스트 LSI 장치의 응답 출력을 저장하는 결과 데이터 파일 및 상기 결과 데이터 파일에 저장된 데이터에 따라 상기 LSI 장치의 설계를 평가하고, 상기 피테스트 LSI 장치의 설계 단계에서의 장치 논리 시뮬레이션 실행을 위한 평가 결과들에 따른 테스트 벤치를 생성하며, 이에 의해 상기 LSI 장치의 설계에 있어서 상기 테스트 결과들을 통합하는 수단을 구비한다.

상술한 바와 같이, 본 발명의 반도체 테스트 시스템에 있어서, 테스트 패턴은 전자 설계 자동조작(EDA) 환경에서 장치의 설계 단계에서 얻은 피테스트 장치의 논리 시뮬레이션 데이터를 직접 사용하는 것에 의해 생성된다. 장치 테스트는 그렇게 생성된 테스트 패턴을 피테스트 장치에 적용하는 것에 의해 실행된다. 본 발명의 반도체 테스트 시스템에 있어서, LSI 장치의 설계 단계에서 생성된 CAD 설계 데이터로부터 발생된 이벤트 기준 시뮬레이션 데이터는 LSI 장치를 테스트 하기 위한 테스트 패턴을 형성하도록 직접 사용될 수 있다.

따라서, 본 발명의 반도체 테스트 시스템에 따르면, LSI 설계에서 테스트 패턴 생성까지의 턴어라운드 시간을 상당히 저감하는 것이 가능하다. 또한, 테스트 시스템은 매우 단순한 형태의 하드웨어 및 소프트웨어에 의해 구성될 수 있다. 또한, 장치의 설계 단계의 논리 시뮬레이션 및 장치에 적용되는 테스트 패턴은 동일한 기본아이디어에 따라 구성되고, 테스트 결과는 EDA 설계 환경으로 직접 피드백될 수 있다. 또한, 본 발명의 반도체 테스트 시스템에서, 장치 테스트는 낮은 테스트 비용 및 높은 테스트 효율로 실행될 수 있다.

본 발명의 실시예 및 종래 기술과 다른점을 도면을 참조하여 설명한다. 도 5는 종래 반도체 테스트 시스템에서 사이클 기준 테스트 패턴을 생성하기 위한 기술의 일예와 본 발명의 반도체 테스트 시스템에서 이벤트 기준 테스트 패턴을 생성하기 위한 기술의 일예를 비교하는 도면이다.

도 5의 위 부분에는 집적 회로의 설계 단계에서 생성된 논리 시뮬레이션의 결과 데이터를 저장하는 덤프 파일(57)이 도시되어 있다. 덤프 파일의 대표적 일예는 Verilog의 VCD(Value Change Dump)이다. 덤프 파일의 출력은 설계된 LSI 장치의 입력 및 출력에서의 변경을 나타내고 예를 들면 도 5의 파형(61)을 표현하는 기술을 갖는 이벤트 기준 포맷의 데이터이다. 이 예에서는 파형(61)으로 나타난 바와 같은 테스트 패턴이 형성되는 것으로 한다. 파형(61)은 각각 핀(테스터 핀 또는 테스트 채널) Sa 및 Sb에 의해 발생된 테스트 패턴을 나타낸다. 파형을 기술하는 이벤트 데이터는 세트 예지 Sa, Sb 및 그들의 타이밍(예를 들면, 기준점으로부터의 시간)과 리셋 예지 Ra, Rb 및 그들의 타이밍으로 형성된다.

덤프 파일로부터의 덤프 데이터에 따라 종래 반도체 테스트 시스템에 사용되는 사이클 기준 테스트 패턴을 생성하기 위해, 덤프 데이터는 테스트 사이클(테스터 레이트), 파형(파형의 종류, 예지 타이밍) 및 벡터를 나타내는 기술로 변환되어야 한다. 그러한 기술의 일예는 도 5의 중간 및 왼쪽에 도시되어 있다. 도 5의 왼쪽 부분에 파형(63)으로 도시한 바와 같은 사이클 기준 테스트 패턴에 있어서, 테스트 패턴은 테스트 사이클마다 파형 및 타이밍이 정의되도록 각 테스트 사이클(TS1, TS2, TS3)로 분할된다.

그러한 파형, 타이밍 및 테스트 사이클에 대한 데이터 기술의 일례는 타이밍 데이터(테스트 플랜)(66)에 도시되어 있다. 파형의 논리 "1", "0", "Z"의 일례는 벡터 데이터(패턴 데이터)(65)에 도시되어 있다. 예를 들면, 타이밍 데이터(66)에서, 테스트 사이클은 테스트 사이클 사이의 기간을 규정하는 "레이트"에 의해 기술되고, 파형은 RZ(return to zero), NRZ(non-return to zero) 및 XOR(exclusive OR)에 의해 기술된다. 또한, 각 파형의 타이밍은 대응하는 테스트 사이클의 소정의 예시로부터의 지연 시간에 의해 규정된다.

그와 반대로, 덤프 파일로부터의 덤프 데이터에 따라 이벤트 기준 반도체 테스트 시스템에 사용되는 이벤트 기준 테스트 패턴을 형성하는 것은 매우 간단하다. 이것은 덤프 데이터가 이벤트 포맷으로 기술되어 있기 때문이다. 도 5의 아래 오른쪽 쪽에 도시된 이벤트 데이터(68)를 도 5의 위쪽에 도시된 덤프 데이터로부터 직접 구할 수 있다는 것은 명백하다.

도 6은 EDA 환경에서 얻은 LSI 설계 평가 데이터에 따라, 종래 반도체 테스트 시스템에 의해, 사이클 기준 테스트 패턴을 생성하기 위한 테스트 시스템과 EDA 환경 사이의 주로 필요한 소프트웨어에 의한 관계를 도시한 것이다. EDA 환경(71)에서 얻은 LSI 설계 평가 데이터는 변환 소프트웨어(72)에 의해 사이클 기준 데이터로 변환된다. 변환된 데이터는 또 오브젝트 파일(74)이 생성되도록 컴파일러(73)에 의해 오브젝트 코드로 변환된다. 오브젝트 파일(74)은 예를 들면, 타이밍 오브젝트 코드, 패턴 오브젝트 코드 및 파형 오브젝트 코드를 구비한다.

오브젝트 파일(74)의 적절한 양의 데이터는 LSI 테스터의 패턴 메모리에 마련되어 있는 타이밍 메모리 및 파형 메모리로 전달되어 그곳에 저장된다. 메모리에 저장된 데이터는 도 5에 도시한 바와 같은 사이클 기준 포맷을 갖는다. 피테스트 반도체 장치를 테스트할 때, 데이터는 메모리에서 추출되고, 따라서 테스트 패턴이 생성된다. 테스트 패턴은 LSI 테스터의 핀 전자 장치(PE)를 통해 피테스트 반도체 장치에 적용된다.

결과 테스트 데이터는 예를 들면 고장 메모리(failure memory)(도시하지 않음)에 저장된다. 필요하다면, 테스트 결과 데이터의 파형은 재구성되어 파형 데이터 파일(77)에 저장되고, 고장 분석 툴(78)에 의한 고장 분석을 위해 이용된다. 고장 분석의 결과는 EDA 설계 환경으로 피드백된다. 상술한 바와 같이, EDA 설계 환경(71)을 통해 얻은 데이터는 반도체 테스트 시스템(76)에 사용될 데이터 구조와 다른 구조를 가지므로, 데이터 구조의 변환을 실행하기 위한 여러가지 종류의 소프트웨어가 필요하다. 또한, 데이터 구조 차이때문에, EDA 설계 환경(71)으로 피드백된 데이터를 사용하는 것이 어렵다.

도 7은 도 6의 종래 기술과 비교하기 위해 본 발명의 이벤트 기준 테스트 시스템과 EDA 설계 환경 사이의 관계를 도시한 것이다. 이벤트 기준 반도체 테스트 시스템에 있어서, EDA 설계 환경(81)에서 얻은 덤프 출력 데이터는 컴파일러(83)에 의해 직접 컴파일되어 오브젝트 코드가 생성된다. 이벤트 오브젝트 코드는 파일(84)에 저장되고, 그의 필요한 양이 테스트 시스템(86)의 이벤트 메모리로 전달된다.

이벤트 메모리에 저장된 이벤트 데이터는 도 5에 도시한 바와 같이 세트 및 리셋과 그들의 타이밍의 단순한 리스트인 기술의 형태를 갖는다. 따라서, 테스트 패턴은 이벤트 메모리로부터의 데이터를 사용하여 용이하게 형성된다. 테스트 패턴이 피테스트 장치에 인가될 때 생성된 테스트 결과 데이터는 고장 분석 툴(82)에 의해 고장 분석 목적으로서 직접 사용될 수 있다. 따라서, 고장 분석 결과는 EDA 설계 환경(81)으로 직접 피드백된다. 이 방식으로, 본 발명의 이벤트 기준 테스트 시스템에서는 매우 단순하고 적은 양의 소프트웨어를 사용하여 테스트 패턴을 생성할 수 있다. 또한, 테스트 패턴은 이벤트 데이터에 따라 세트 및 리셋을 단순히 발생하는 것에 의해 발생되므로, 발생된 테스트 시스템의 하드웨어도 크게 단순화된다.

따라서, 본 발명의 이벤트 기준 테스트 시스템에 따르면, 테스트 패턴을 발생하기 위한 하드웨어 및 소프트웨어의 비용을 실질적으로 줄일 수 있다. 또한, 테스트 시스템에 의해 생성된 테스트 패턴은 설계 데이터에 의해 생성된 테스트 패턴과 일치하므로, 피테스트 장치의 결함 검출의 효율(디버깅의 효율)이 증가된다. 또한, 장치의 설계 단계에서 생성된 이벤트 기준 시뮬레이션 데이터를 직접 사용하여 피테스트 장치를 테스트하기 위한 테스트 패턴을 발생하므로, LSI 설계에서 테스트 패턴 생성까지의 턴어라운드 시간을 크게 줄일 수 있다.

그러나, 도 8과 도 9에 도시한 상황 사이의 관계는 도 6과 도 7에 도시한 것과 유사하며, EDA 환경과 LSI 테스터 사이에 수반되는 소프트웨어를 더욱 상세히 기술하기로 한다. 도 8은 EDA 환경에서 얻은 LSI 설계 평가 데이터에 따라 종래 반도체 테스트 시스템에 의해 사이클 기준 테스트 패턴을 발생하기 위해 필요한 테스트 시스템과 EDA 환경 사이의 소프트웨어를 도시한 블록도이다.

도 8의 예에서, EDA 환경(91)에서 얻은 LSI의 설계 유효 데이터는 덤프 파일(92)에 저장된다. 덤프 파일(92)로부터의 데이터 및 반도체 테스트 시스템(106)의 핀 구성에 관한 핀 데이터 파일(93)로부터의 데이터는 변환 소프트웨어(97)에 주어짐으로써, 사이클 기준 데이터로 변환된다. 또한, 데이터 파일(94, 95, 96)로부터의 테스트 시스템의 여러가지 규격을 기술하는 데이터는 변환 소프트웨어(97)에 주어져서 사이클 기준 데이터로 변환된다.

이러한 과정을 거쳐, 테스트 플랜 파일(98) 및 테스트 패턴 파일(99)이 만들어진다. 본 발명 명세서의 문맥내에서, 테스트 플랜 데이터는 테스트 패턴 파형, 테스트 사이클 및 파형의 타이밍을 기술하는 타이밍 데이터와 동일하다. 패턴 데이터는 테스트 패턴의 벡터를 기술한다. 상술한 변환된 데이터는 컴파일러(101, 102)에 의해 오브젝트 코드로 또 변환되고 이것에 의해 오브젝트 코드 파일(103, 104)이 형성된다. 오브젝트 코드 파일(103, 104) 내의 데이터는 로더(loader)에 의해 테스트 시스템(LSI 테스터)(106)의 대응하는 메모리(파형, 타이밍 및 패턴 메모리)로 전달된다.

LSI 테스터(106)의 파형, 타이밍 및 패턴 메모리에 저장된 데이터는 도 5에 도시한 바와 같은 사이클 기준 데이터 구조를 갖는다. 피테스트 장치를 테스트할 때, 데이터는 이들 메모리로부터 판독되고, 이것에 의해 테스트 패턴이 생성된다. 테스트 패턴은 핀 전자 장치(PE)를 거쳐 피테스트 장치에 적용된다.

도 6의 경우와 마찬가지로, 테스트 결과 데이터는 고장 메모리(도시하지 않음)에 저장된다. 테스트 결과 데이터는 이벤트 기준 데이터와 동일한 입력 및 출력 파형을 보여주도록 데이터 파일(107)에서 재구성된다. 파일(107)로부터의 데이터는 분석 툴(108)에 의한 고장 분석에 이용된다. 고장 분석의 결과는 EDA 환경으로 피드백되지만, 고장 분석 결과는 그곳에서 직접 사용될 수 없다.

상술한 바와 같이, EDA 설계 환경(91)에서 얻은 데이터와 반도체 테스트 시스템(106)에서 사용되는 데이터는 데이터가 구조가 다르므로, 데이터 변환을 위해 여러가지 소프트웨어를 사용해야 한다. 특히, 사이클 기준 시스템에서 데이터 변환을 위해 점선 영역의 소프트웨어가 필요로 된다. 이들 소프트웨어 모두는 본 발명의 이벤트 기준 LSI 테스터에서는 필요하지 않다.

도 7의 경우와 마찬가지로, 도 9는 도 6 및 도 8의 종래 기술의 사이클 기준 반도체 테스트 시스템과 비교되는 본 발명의 이벤트 기준 반도체 테스트 시스템을 도시한 블록도이다. 이벤트 기준 반도체 테스트 시스템에 있어서, EDA 설계 환경(115)에서 얻은 덤프 파일(111)의 덤프 출력 데이터 및 핀 데이터 파일(112)로부터의 핀 정보는 컴파일러(113)에 의해 직접 컴파일되어 오브젝트 코드가 생성된다. 이벤트 오브젝트 코드는 파일(114)에 저장되고 그의 필요량은 로더를 거쳐 테스트 시스템(118)의 이벤트 메모리로 전달된다.

이벤트 메모리에 저장된 이벤트 데이터는 세트 및 리셋과 그들의 타이밍의 단순한 리스트인 도 5에 도시한 바와 같은 기술의 형태를 갖는다. 따라서, 테스트 패턴은 이벤트 메모리로부터의 데이터를 사용하여 용이하게 형성된다. 테스트 패턴이 피테스트 장치에 적용될 때 발생된 테스트 결과 데이터는 고장 분석 목적으로서 고장 분석 툴(119)에 의해 직접 사용될 수 있다. 따라서, 고장 분석 결과는 EDA 설계 환경(115)으로 직접 피드백된다.

이 방식으로, 본 발명의 이벤트 기준 테스트 시스템에 있어서, 테스트 패턴은 매우 단순하고 적은 양의 소프트웨어를 사용하여 생성될 수 있다. 즉, 점선 영역으로 도시한 도 8의 종래 기술에서 필요한 소프트웨어는 도 9에 도시한 본 발명의 테스트 시스템에서는 완전히 불필요하다. 또한, 테스트 패턴이 이벤트 데이터에 따라 세트 및 리셋을 단순히 생성하는 것에 의해 발생되므로, 생성된 테스트 시스템의 하드웨어도 상당히 단순화된다.

따라서, 본 발명의 이벤트 기준 테스트 시스템에 따르면, 테스트 패턴을 발생하기 위한 하드웨어 및 소프트웨어의 비용을 실질적으로 줄일 수 있다. 또한, 테스트 시스템에 의해 생성된 테스트 패턴은 설계 데이터에 의해 생성된 테스트 패턴과 일치하므로, 피테스트 장치의 결함 검출의 효율(디버깅의 효율)이 증가한다. 또한, 장치의 설계 단계에서 생성된 이벤트 기준 시뮬레이션 데이터를 직접 사용하여 피테스트 장치를 테스트하기 위한 테스트 패턴을 발생할 수 있으므로, LSI 설계에서 테스트 패턴 생성까지의 턴어라운드 시간을 크게 줄일 수 있다.

도 10은 종래 기술과 본 발명 사이의 차이를 더욱 명확히 보이기 위해 도 8의 사이클 기준 아키텍처와 도 9의 이벤트 기준 아키텍처를 조합한 상황을 기본적으로 도시한 블록도이다. 도 10에서, EDA 환경을 나타내는 블록도는 도 4의 블록도와 대략 동일하다. 즉, 레지스터 변환 언어(RTL)로 설계 데이터를 저장하는 RTL 파일(121)은 변환 소프트웨어(122)에 의해 핀 데이터 파일(125) 및 레이아웃 데이터 파일(126)로 변환된다. 변환된 설계 데이터에 따라, 집적 회로(128)는 반도체 제조 공정(실리콘 공정)을 통해 만들어진다.

또한, 설계 데이터 각각으로부터 데이터를 추출하기 위한 처리(127)를 통해, 테스트벤치(124)가 생성된다. 테스트벤치는 설계 데이터로 테스트 벡터를 장치 모델에 제공하고, 이것에 의해 논리 시뮬레이션 처리(123)가 실행된다. 논리 시뮬레이션에 기인하는 덤프 파일인 VCD 파일(134)(그리고 이벤트 기준과 유사한 데이터 구조를 갖는 STIL 파일) 및 테스트 시스템의 핀 정보를 저장하는 핀 데이터 파일(133)을 사용하여 테스트 패턴을 작성한다. 덤프 파일의 일례는 VCD/Verilog 등의 VCD(value change dump)이다. STIL은 IEEE에 의한 "표준 테스트 인터페이스 언어"를 나타낸다.

본 발명에 있어서, 테스트 패턴은 점선 (A)로 나타난 처리에 따라 생성된다. 즉, 덤프 데이터 및 핀 데이터는 컴파일러(141)에 의해 오브젝트 코드로 변환되고, 이벤트 파일(132)에 귀착한다. 이벤트 파일(132)에 저장된 이벤트 데이터는 테스트 시스템(131)의 이벤트 메모리로 전달된다. 따라서, 테스트 시스템은 이벤트 기준 테스트 패턴을 발생함으로써, 피테스트 장치(128)를 테스트한다.

그와 반대로, 종래 반도체 테스트 시스템을 사용하는 과정에 있어서, 테스트 패턴은 점선(B)로 나타난 루트를 통해 생성된다. 즉, 테스트 시스템의 여러가지 규격 및 상태를 저장하는 덤프 파일(134, 135)로부터의 데이터 및 파일(137 내지 139)로부터의 데이터는 전처리(142)를 통해 사이클 기준 데이터를 형성하기 위해 필요한 정보가 추출되도록 처리된다. 이 처리에 있어서, 이벤트 기준 데이터는 대응하는 테스트 사이클내의 파형 데이터 및 타이밍 데이터가 생성되도록 각 테스트 사이클로 분할된다.

상기 처리에서 얻어진 데이터는 변환 소프트웨어(143)에 의해 테스트 시스템(152)의 포맷으로 변환되고, 이것은 테스트 패턴의 파형, 타이밍 및 벡터를 기술하는 포맷이다. 또한, 후처리(144)를 통해, 결과 데이터는 테스트 시스템 고유의 언어로 또 변환됨으로써, 테스트 패턴 파일(145, 146, 147)이 형성된다. 이들 패턴 파일로부터의 테스트 패턴 데이터는 테스트 소프트웨어(컴파일러를 포함)를 통해 테스트 시스템에 로드된다. 따라서, 테스트 시스템은 수신된 데이터에 따라 테스트 패턴을 발생하고 피테스트 장치(128)를 테스트한다.

상기에서와 같이, 본 발명의 테스트 시스템에서, EDA 환경 및 테스트 시스템은 점선 (A)의 루트로 나타난 바와 같이 서로 직접 결합될 수 있다. 그 루트중에 수반되는 소프트웨어는 컴파일러(141)뿐이다. 한편, 종래 기술의 테스트 시스템은 점선(B)로 나타난 루트에서 EDA 환경과 테스트 시스템 사이에 다양한 소프트웨어를 요구한다. 또한, 이벤트 기준 데이터 구조에서 사이클 기준 데이터 구조로의 변환은 항상 완벽하게 가능할 수 없으므로, 변환 에러가 생겨서, 장치 평가가 불충분하게 된다.

도 11은 EDA 환경에서 얻은 LSI 설계 평가 데이터에 따라 테스트 패턴을 발생하고 피테스트 장치를 테스트하기 위한 EDA 환경과 본 발명의 시스템 사이의 전체 관계 및 본 발명의 반도체 테스트 시스템의 기능을 도시한 것이다. 특히, 도 11은 이벤트 데이터를 표시할 뿐만 아니라 이벤트 데이터의 각 파라미터를 수정하는 본 발명의 테스트 시스템의 기능을 도시하고 있다.

테스트벤치(163)(제3자에 의한 것) 또는 테스트벤치(164)(테스터에 의해 생성됨)을 사용하여 LSI 설계 데이터에 대해 논리 시뮬레이션 처리(162)를 실행하는 것에 의해, 덤프 파일(168)이 생성된다. 인터페이스(171)를 통해, 덤프 파일(168)로부터의 이벤트 데이터 또는 테스트벤치(164)로부터의 이벤트 데이터(169)를 각 테스트 핀에 할당하고 각 이벤트의 신호 레벨을 규정하는 것에 의해 파일(172, 173)이 작성된다. 파일(172, 173)로부터의 데이터는 컴파일러(175)에 의해 오브젝트 코드로 변환됨으로써, 이벤트 파일(176)이 형성된다.

장치에 대해 테스트를 실행할 때, 이벤트 데이터는 로더(177)에 의해 파일(76)에서 테스트 시스템(178)으로 전달된다. 이벤트 메모리에 저장된 이벤트 데이터에 따라, 테스트 시스템(178)이 이벤트 기준 테스트 패턴을 발생함으로써, 피테스트 장치에 대한 테스트가 실행된다. 테스트 결과는 예를 들면 고장 분석에 사용되도록 테스트 결과 파일(166)에 축적된다. 이 방식으로 축적된 테스트 결과는 테스트벤치 발생기(165)를 통해 테스트벤치에 피드백될 수 있다.

상기에서와 같이, 이벤트 파일에 형성된 이벤트 데이터는 피테스트 장치에 적용될 테스트 패턴을 직접 기술한다. 따라서, 이벤트 데이터를 사용하여, 그래픽 유저 인터페이스(GUI)를 통해, 실제 테스트 패턴과 동일한 패턴 시퀀스가 표시되고 심지어 수정될 수 있다. 예를 들면, 테스트 패턴 및 테스트 핀을 보여주는 전체 화상(181) 및 테스트 패턴의 일부의 확대도(182), 소정의 인수만큼 감소(184) 또는 증가(185)되는 타이밍을 보여주는 도면(184, 185), 특정 이벤트의 타이밍(또는 위치)가 변경되는 오프셋 화상(183) 등이 있다. 그러한 표시 상에서의 파라미터 변경은 이벤트 파일의 데이터를 수정하는 것에 의해 실행될 수 있고 또한 피테스트 장치에 적용된 실제 테스트 패턴을 변경하여 피테스트 장치의 결과 응답을 모니터링하는 것을 가능하게 한다. 타이밍 데이터의 증가/감소(스케일링)에 관한 더 상세한 설명은 본 발명의 동일한 양수인이 소유한 미국 특허 출원 09/286,226에 개시되어 있다.

도 12는 본 발명의 이벤트 기준 반도체 테스트 시스템의 구성 일예를 도시한 블록도이다. 이 테스트 시스템에 관한 더 상세한 설명은 상기 미국 특허뿐만 아니라 본 발명의 양수인이 소유한 미국 특허 출원 09/259,401에 개시되어 있다. 도 12에 있어서, 호스트 컴퓨터(212)는 예를 들면 UNIX에 기초한 워크 스테이션이고, 유저 인터페이스로서 기능한다. 호스트 컴퓨터(212) 및 테스트 시스템의 하드웨어는 인터페이스 버스(213)를 거쳐 서로 접속되어 있다.

어드레스 컨트롤러(218)는 테스트 시스템을 제어하는 예를 들면, 테스터 프로세서이다. 도 12의 예에서, 어드레스 컨트롤러(218)는 이벤트 메모리(220, 221)에 공급되는 어드레스를 제어한다. 이벤트 데이터는 이벤트 메모리(220, 221)에서 호스트 컴퓨터로 테스트 프로그램으로서 전달된다. 예를 들면, 이벤트 메모리(220)는 기준 클럭 사이클의 정수배를 나타내는 이벤트 타이밍 데이터를 저장하고, 이벤트 메모리(221)는 기준 클럭 사이클의 분수를 나타내는 이벤트 데이터를 저장한다. 가산 및 스케일링(222)은 이벤트 메모리(220, 221)로부터의 타이밍 데이터를 가산 또는 수정하는 것에 의해 각 이벤트 데이터의 기간 데이터를 발생한다. 기간 데이터는 소정의 기준점으로부터의 기간(지연 시간)에 의해 각 이벤트의 타이밍을 표현한다. 이벤트 발생기(224)는 기간 데이터에 따라 테스트 패턴을 발생하고 그 테스트 패턴을 핀 전자 장치(226)를 통해 피테스트 장치(228)에 제공한다. 장치(228)는 그의 응답 출력을 평가하는 것에 의해 테스트된다.

상기에서 설명한 바와 같이, 본 발명의 반도체 테스트 시스템에 있어서, 테스트 패턴은 전자 설계 자동조작(EDA) 환경에서 장치의 설계 단계에서 얻은 피테스트 장치의 논리 시뮬레이션 데이터를 직접 사용하는 것에 의해 발생된다. 장치 테스트는 그렇게 발생된 테스트 패턴을 피테스트 장치에 적용하는 것에 의해 실행된다. 본 발명의 반도체 테스트 시스템에 있어서, LSI 장치의 설계 단계에서 생성된 CAD 설계 데이터에서 발생된 이벤트 기준 시뮬레이션 데이터를 직접 사용하여 LSI 장치를 테스트하기 위한 테스트 패턴을 형성할 수 있다.

발명의 효과

따라서, 본 발명의 반도체 테스트 시스템에 의하면, LSI 설계에서 테스트 패턴 생성까지의 턴어라운드 시간을 크게 줄일 수 있다. 또한, 테스트 시스템을 매우 단순한 형태의 하드웨어 및 소프트웨어에 의해 구성할 수 있다. 또한, 장치의 설계 단계에서의 논리 시뮬레이션 및 장치에 적용되는 테스트 패턴은 동일한 기본 아이디어에 따라 구성되므로, 테스트 결과를 EDA 설계 환경으로 직접 피드백할 수 있다. 또한, 본 발명의 반도체 테스트 시스템에서, 장치 테스트는 낮은 테스트 비용 및 높은 테스트 효율로 실행될 수 있다.

(57) 청구의 범위

청구항 1.

반도체 테스트 시스템에 있어서,

피테스트 LSI 장치의 설계 단계에서 장치 논리 시뮬레이션을 실행하는 것에 의해 얻어진 이벤트 데이터를 상기 LSI 장치의 설계 데이터 상에 저장하는 이벤트 파일;

상기 이벤트 파일로부터 직접 얻어진 이벤트 데이터를 기준 클럭 사이클의 정수배를 나타내는 것과 상기 기준 클럭의 분수를 나타내는 것인 두 가지 다른 종류의 데이터로 저장하는 이벤트 메모리;

상기 이벤트 메모리로부터의 이벤트 데이터를 직접 사용하는 것에 의해 테스트 패턴을 발생시키고, 상기 테스트 패턴을 상기 피테스트 LSI 장치에 적용하는 수단;

상기 피테스트 LSI 장치에 적용된 상기 테스트 패턴을 표시하고, 상기 이벤트 파일 내의 상기 이벤트 데이터를 부분적으로 수정하는 것에 의해 상기 테스트 패턴을 실시간으로 변경하는 수단; - 상기 테스트 패턴에 있어서의 상기 변경은 이벤트의 특정 타이밍의 변경을 포함함 -

상기 피테스트 LSI 장치의 응답 출력을 저장하는 결과 데이터 파일; 및

상기 결과 데이터 파일에 저장된 데이터에 따라 상기 LSI 장치의 설계를 평가하고, 상기 피테스트 LSI 장치의 설계 단계에서의 장치 논리 시뮬레이션 실행에 대한 평가 결과들에 따른 테스트 벤치를 생성하며, 이에 의해 상기 LSI 장치의 설계에 있어서 상기 테스트 결과들을 통합하는 수단

을 포함하는 것을 특징으로 하는 반도체 테스트 시스템.

청구항 2.

제1항에 있어서, 상기 이벤트 데이터는 소정의 기준점으로부터의 기간에 대하여 상기 피테스트 LSI 장치의 특정 위치에 서 입력 및 출력 신호의 연속적인 ON-OFF 변경을 기술하는 것을 특징으로 하는 반도체 테스트 시스템.

청구항 3.

제1항에 있어서, 상기 이벤트 파일 내에 저장되는 오브젝트 코드의 이벤트 데이터를 생성하기 위한 논리 시뮬레이션의 결과로서 발생하는 상기 이벤트 데이터를 컴파일하기 위한 컴파일러를 더 포함하고, 상기 컴파일러는 오브젝트 코드로 변환되어 상기 이벤트 파일 내에 저장되는 상기 반도체 테스트 시스템의 테스터 핀 데이터를 수신하는 것을 특징으로 하는 반도체 테스트 시스템.

청구항 4.

삭제

청구항 5.

제1항에 있어서, 상기 표시 수단은 상기 테스트 패턴의 전체 도면, 상기 테스트 패턴의 일부의 확대도, 상기 테스트 패턴의 각 이벤트의 타이밍 변경, 및 상기 이벤트 데이터가 그래픽 유저 인터페이스를 통해 변경될 때 실시간 방식으로 상기 테스트 패턴으로부터의 오프셋 추가 또는 제거를 나타내는 것을 특징으로 하는 반도체 테스트 시스템.

청구항 6.

반도체 테스트 시스템에 있어서,

컴퓨터 시스템의 도움으로 전자 설계 자동조작(EDA) 환경하에서 LSI 장치를 설계하는 수단;

상기 설계 단계에서 상기 LSI 장치에 대해 논리 시뮬레이션을 실행하는 것에 의해 얻은 덤프 파일;

상기 덤프 파일로부터 직접 얻어진 이벤트 데이터를 기준 클럭 사이클의 정수배를 나타내는 것과 상기 기준 클럭의 분수를 나타내는 것인 두 가지 다른 종류의 데이터로 저장하는 이벤트 메모리;

상기 이벤트 메모리로부터의 이벤트 데이터를 직접 사용하는 것에 의해 테스트 패턴을 발생시키고 상기 테스트 패턴을 상기 피테스트 LSI 장치에 적용하는 수단;

상기 피테스트 LSI 장치에 적용된 상기 테스트 패턴을 표시하고, 실시간으로 상기 이벤트 파일 내의 상기 이벤트 데이터를 부분적으로 수정하는 것에 의해 상기 테스트 패턴을 변경하는 수단; - 상기 테스트 패턴에 있어서의 상기 변경은 이벤트의 특정 타이밍의 변경을 포함함 -

상기 피테스트 LSI 장치의 응답 출력을 저장하는 결과 데이터 파일; 및

상기 결과 데이터 파일에 저장된 데이터에 따라 상기 LSI 장치의 설계를 평가하고, 상기 EDA 환경에서의 장치 논리 시뮬레이션 실행에 대한 평가 결과에 따라 테스트 벤치를 생성하는 것에 의해 상기 평가 결과들을 상기 EDA 환경으로 피드백하며, 이에 의해 상기 LSI 장치의 설계에 있어서 상기 테스트 결과들을 통합하는 수단

을 포함하는 것을 특징으로 하는 반도체 테스트 시스템.

청구항 7.

제6항에 있어서, 상기 이벤트 데이터는 소정의 기준점으로부터의 기간에 대하여 상기 피테스트 LSI 장치의 특정 위치에 서 입력 및 출력 신호의 연속적인 ON-OFF 변경을 기술하는 것을 특징으로 하는 반도체 테스트 시스템.

청구항 8.

제6항에 있어서, 상기 이벤트 파일 내에 저장되는 오브젝트 코드의 이벤트 데이터를 생성하기 위한 논리 시뮬레이션의 결과로서 발생하는 상기 이벤트 데이터를 컴파일하기 위한 컴파일러를 더 포함하고, 상기 컴파일러는 오브젝트 코드로 변환되어 상기 이벤트 파일 내에 저장되는 상기 반도체 테스트 시스템의 테스터 핀 데이터를 수신하는 것을 특징으로 하는 반도체 테스트 시스템.

청구항 9.

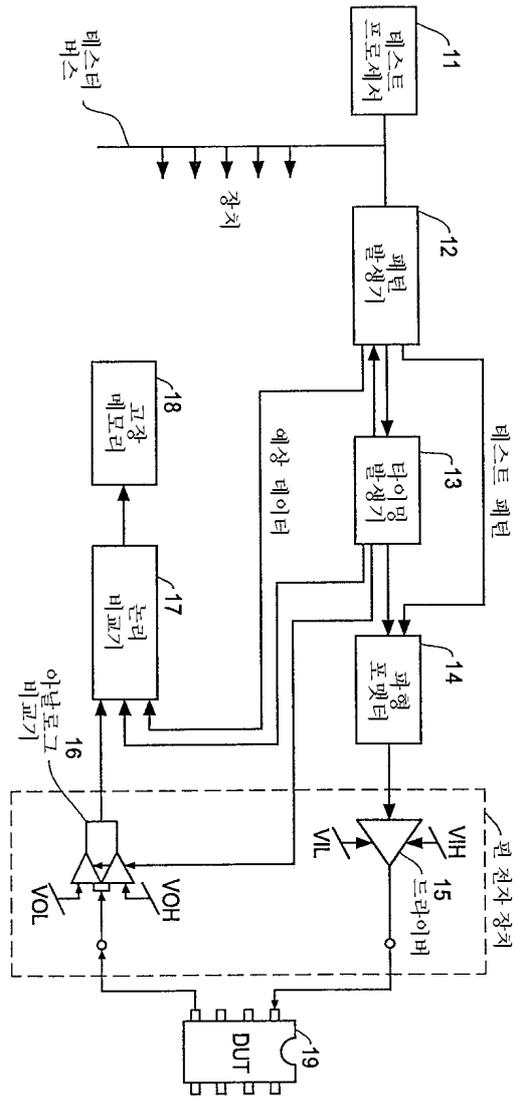
삭제

청구항 10.

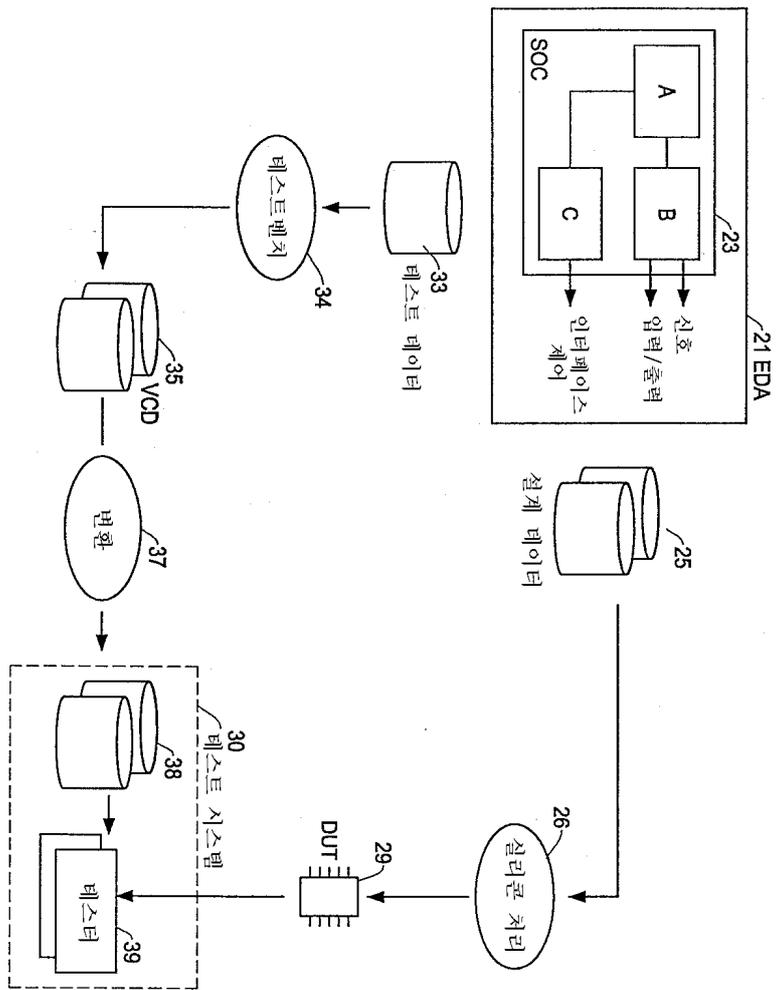
제6항에 있어서, 상기 표시 수단은 상기 테스트 패턴의 전체 도면, 상기 테스트 패턴의 일부의 확대도, 상기 테스트 패턴의 각 이벤트의 타이밍 변경, 및 상기 이벤트 데이터가 그래픽 유저 인터페이스를 통해 변경될 때 실시간 방식으로 상기 테스트 패턴으로부터의 오프셋 추가 또는 제거를 나타내는 것을 특징으로 하는 반도체 테스트 시스템.

도면

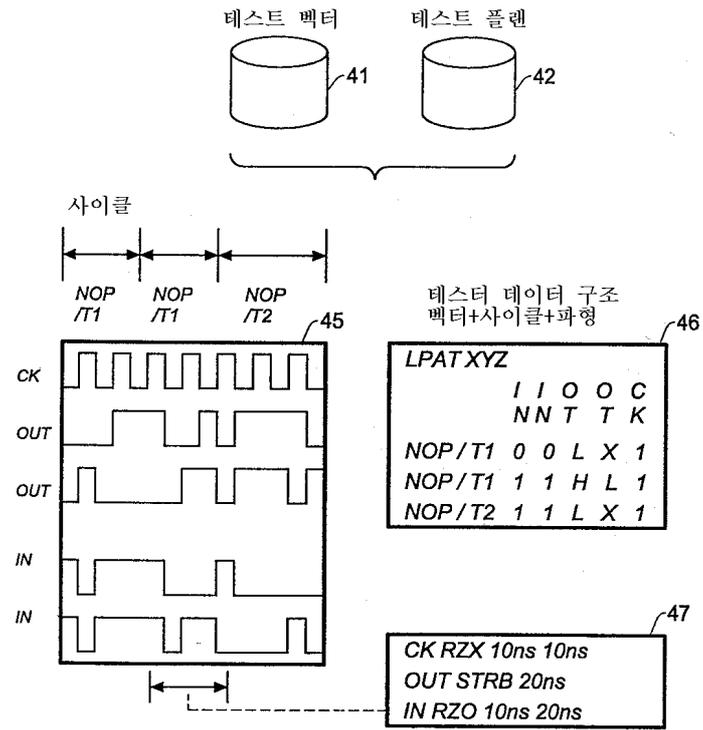
도면1



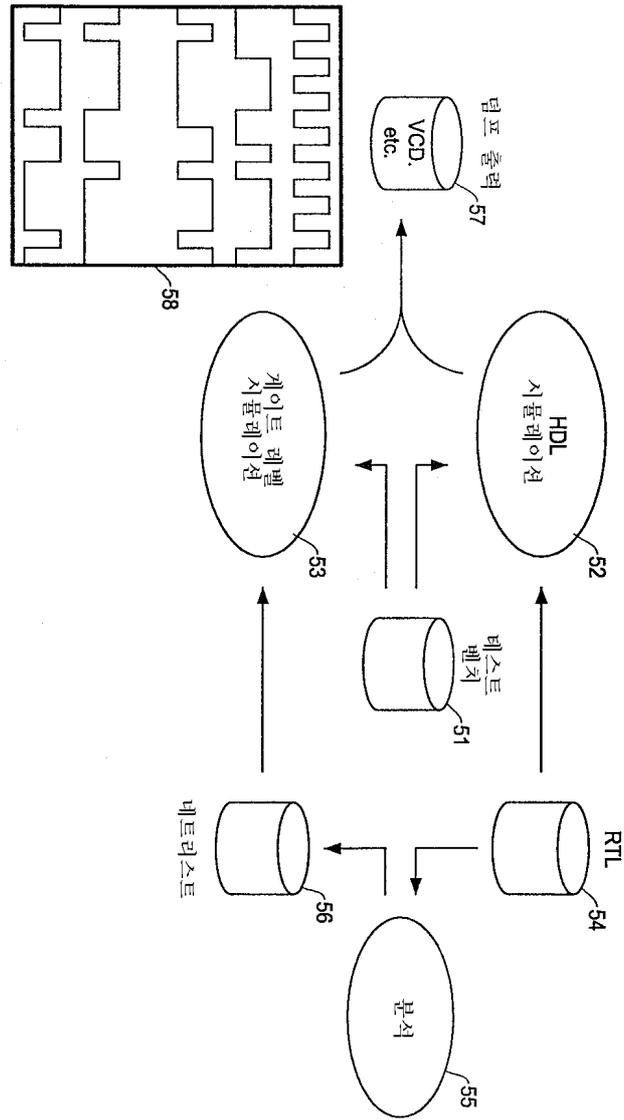
도면2



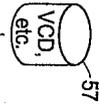
도면3



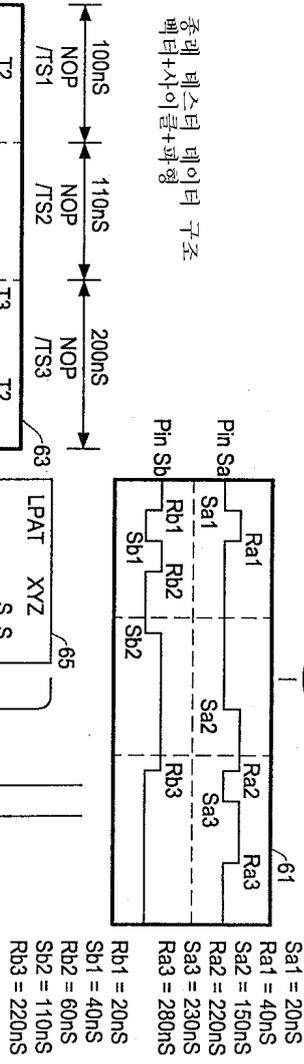
도면4



시플레이션
범프 출력
etc.



종래 테스트 데이터 구조
백터+사이클+위형



도면5

rate (TS1, 100ns); rate(TS2, 110ns); rate(TS3, 200ns);

Sa.wavekind(TS1, RZ); Sa.wavekind(TS2, NRZ); Sa.wavekind(TS3, XOR);
 Sb.wavekind(TS1, XORINV); Sb.wavekind(TS2, NRZ); Sb.wavekind(TS3, NRZINV);

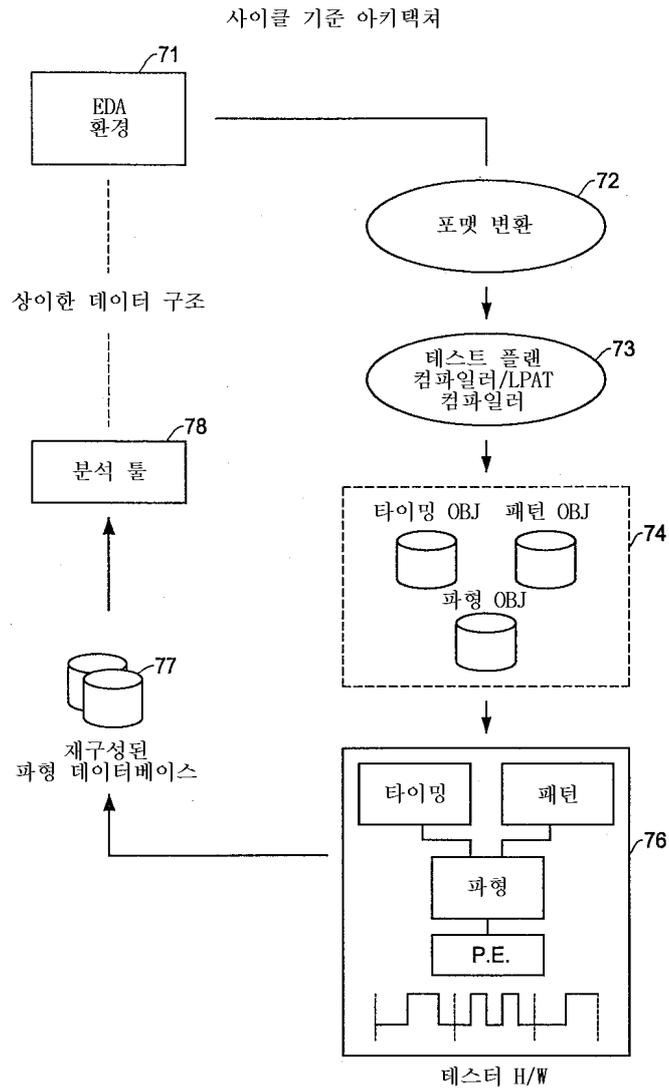
Sa.timing(TS1, T1, 20ns); Sa(TS1, T2, 40ns);
 Sa.timing(TS2, T1, 50ns);
 Sa.timing(TS3, T1, 30ns); Sa.timing(TS3, T2, 70ns); Sa.timing(TS3, T3, 10ns);
 Sb.timing(TS1, T1, 40ns); Sb.timing(TS1, T2, 60ns); Sb.timing(TS1, T3, 20ns);
 Sb.timing(TS2, T1, 10ns);
 Sb.timing(TS3, T1, 10ns);

LPAT	XYZ	S S
NOP /TS1	1 0	a b
NOP /TS2	1 1	
NOP /TS3	0 0	

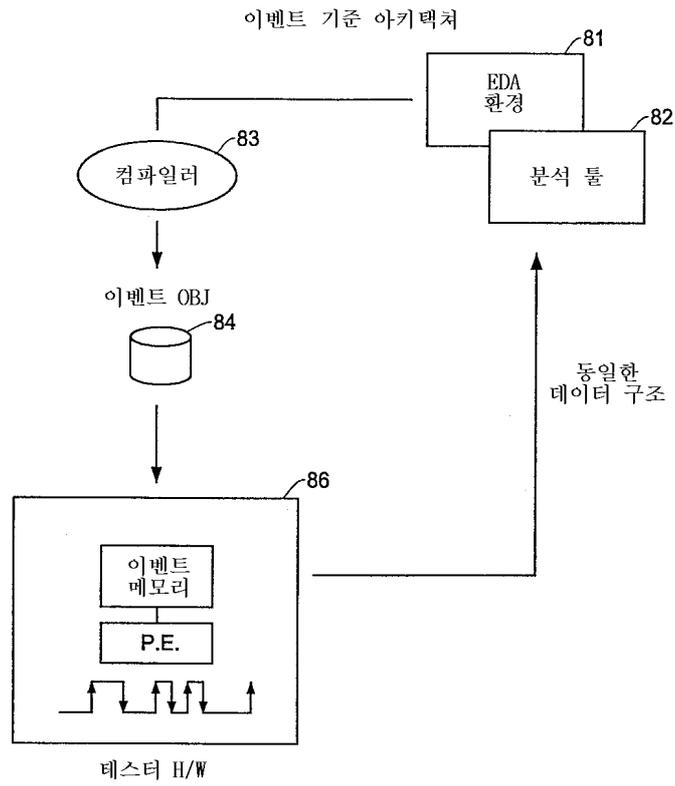
이벤트 기준 기술

Pin Sa	Sa1 ; 20
Ra1 ; 40	
Sa2 ; 150	
Ra2 ; 220	
Sa3 ; 230	
Ra3 ; 280	
Pin Sb	Rb1 ; 20
Sb1 ; 40	
Rb2 ; 60	
Sb2 ; 110	
Rb3 ; 220	

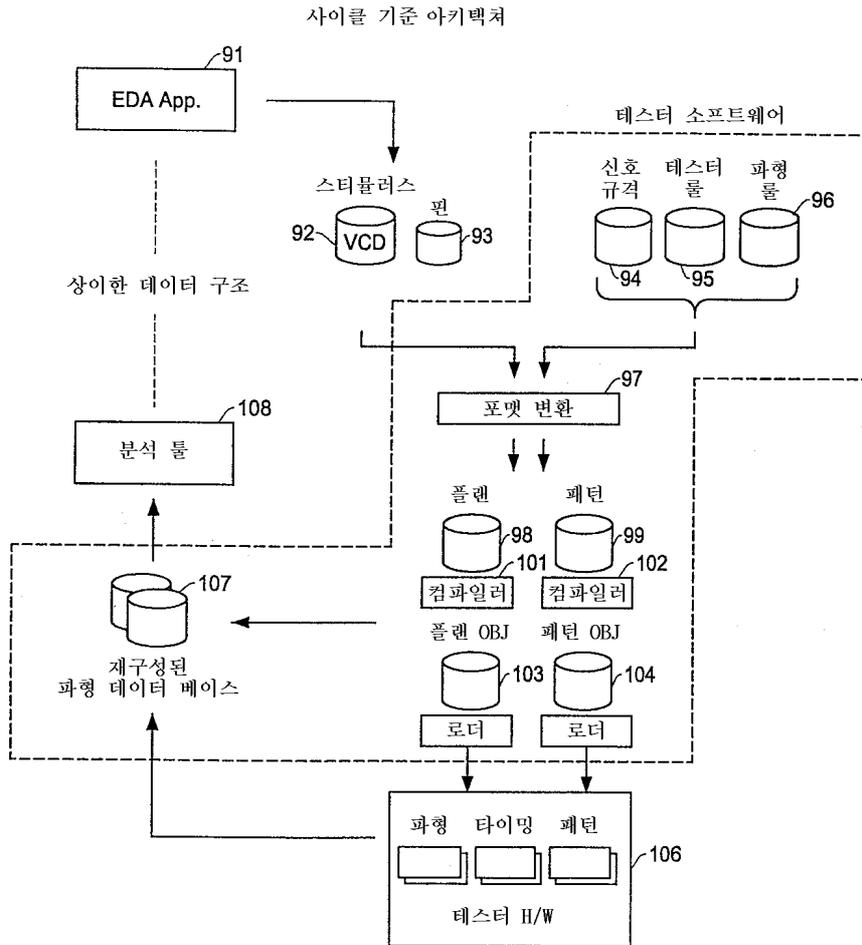
도면6



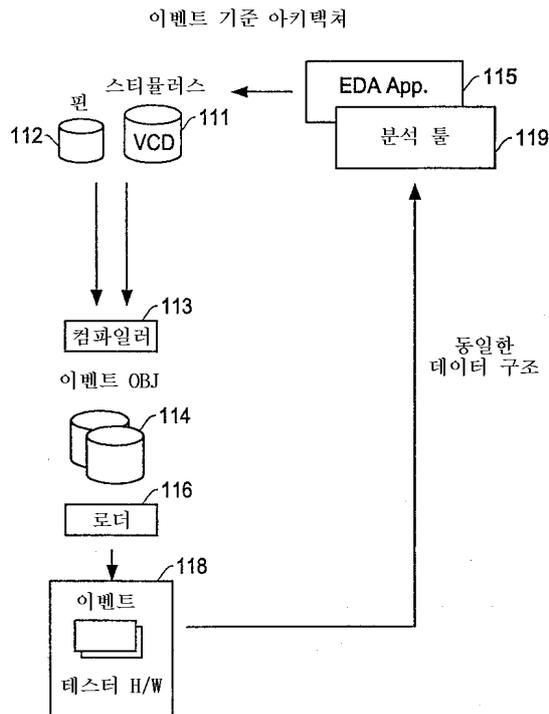
도면7



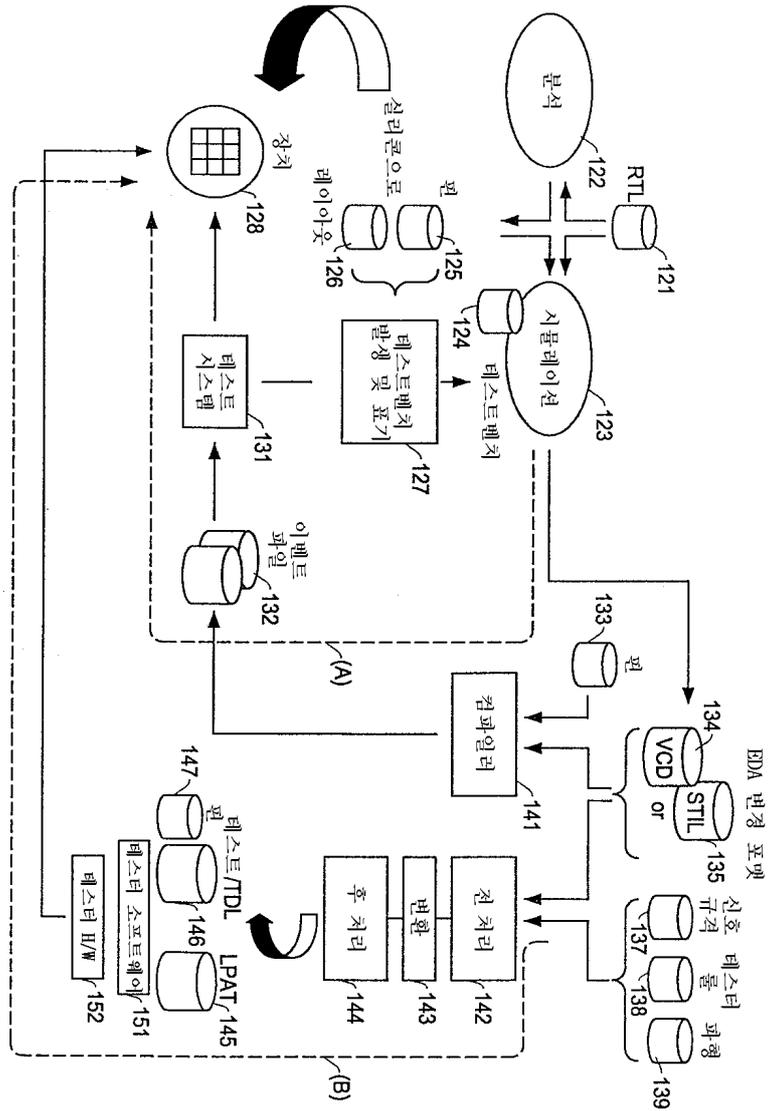
도면8



도면9



도면10



도면12

