

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(21) 출원번호	10-2001-7011122	(65) 공개번호	10-2001-0104719
(22) 출원일자	2001년08월31일	(43) 공개일자	2001년11월26일
번역문 제출일자	2001년08월31일		
(86) 국제출원번호	PCT/US2000/004865	(87) 국제공개번호	WO 2000/52488
국제출원일자	2000년02월24일	국제공개일자	2000년09월08일

AP ARIPO특허 : 케냐, 레소토, 말라위, 수단, 스와질랜드, 우간다, 가나, 감비아, 짐바브웨, 시에라리온, 탄자니아,

EA 유라시아특허 : 아르메니아, 아제르바이잔, 벨라루스, 키르키즈스탄, 카자흐스탄, 몰도바, 러시아, 타지키스탄, 투르크맨,

EP 유럽특허 : 오스트리아, 벨기에, 스위스, 독일, 덴마크, 스페인, 프랑스, 영국, 그리스, 아일랜드, 이탈리아, 룩셈부르크, 모나코, 네덜란드, 포르투칼, 스웨덴, 핀란드, 사이프러스,

OA OAPI특허 : 부르키나파소, 베니, 중앙아프리카, 콩고, 코트디브와르, 카메룬, 가봉, 기니, 말리, 모리타니, 니제르, 세네갈, 차드, 토고, 기니 비사우,

(30) 우선권주장 09/260,463 1999년03월01일 미국(US)

(73) 특허권자 폼팩터, 인크.
미국, 캘리포니아 94551, 리버모어, 7005 싸우스프론트 로드

밀리,찰스,에이.
미국94539캘리포니아주프레먼트세밀런드라이브48881

(74) 대리인 주성민
장수길

심사관 : 김세영

(54) 단일 테스터 채널을 이용하여 다수의 디바이스의 테스트를 병렬 테스트하기 위한 분산형 인터페이스

요약

복수의 테스트 대상 집적 회로(IC) 디바이스(DUT)를 테스트하기 위한 시스템은 단일 또는 멀티-채널 테스터에 결합되고, 테스터로부터 데이터 값을 수신하며 DUT에 관련한 여러 정보를 제공하기 위한 인터페이스 회로를 구비하고 있다. 인터페이스 회로는 단일 채널을 통해 테스터로부터 수신된 데이터 값을 복수의 DUT에 병렬로 전송한다. 회로는 DUT로부터 판독된 데이터 값을 이용하여 비교를 수행하고, 응답으로서 비교를 나타내는 여러 값을 생성한다. 그리고 나서, 여러 값을 동일한 또는 다른 채널을 통해 테스터에 리턴된다.

대표도

도 4

색인어

테스트, DUT, 병렬 테스트, 집적 회로, 인터페이스 회로

명세서

기술분야

본 발명은 반도체 테스터를 이용한 집적 회로 디바이스의 테스트에 관한 것으로, 특히 더 나은 효율과 처리량을 위해 테스터의 단일 채널을 이용하여 다수의 디바이스를 병렬로 테스트하는 것에 관한 것이다.

배경기술

집적 회로(IC) 디바이스는 현대의 거의 모든 전자 또는 컴퓨터 시스템의 중요한 부분이다. 그러한 시스템의 제조 단가를 낮추기 위해, 제조사들은 각 구성 요소 IC 디바이스가 결함이 없고 그 사양에 따라 동작되기를 원한다. 그러므로, 모든 IC 디바이스가 시스템 제조사들에게 보내지기 전에 엄격한 테스트를 받는다고 예상하는 것은 특별한 것이 아니다.

그러나, IC 디바이스를 제조하는 전체 비용의 상당 부분이 테스트에 기인한다고 판단되어 왔다. 그것은 많은 현대의 IC 디바이스들이 복잡한 기능들을 수행하고, 많은 수의 입력과 출력을 구비하며, 고속으로 동작하기 때문이다. 예를 들면, 256Mb 메모리 디바이스는 16개의 데이터 라인과 22개의 어드레스 라인을 가질 수 있다. 그러한 디바이스를 테스트하는 가장 간단한 접근법은 주지의 데이터 값을 각 메모리 로케이션(location)에 기입한 후, 각 로케이션으로부터 판독하여, 판독된 값과 예상되는 값 또는 기입된 값을 비교하여 소정의 에러를 판단한다. 그러나, 각각 수개의 비트를 포함하는 다수의 로케이션이 존재하기 때문에, 이러한 각 로케이션의 각 비트를 테스트하는 기술은 매우 많은 시간을 소비하게 된다. 그러므로, 테스트 엔지니어링의 분야에서는 가장 적은 수의 테스트 시퀀스를 이용하면서도 가능한 한 많은 에러를 검출할 수 있는 효율적인 기술을 개발하는데 주력하고 있다.

메모리 디바이스는 자동화된 반도체 테스터를 이용하여 테스트될 수 있다. 도 1은 DUT(Device Under Test, 118)와 같은 다수의 테스트 대상 디바이스를 병렬 테스트하기 위한 다수(N)의 채널을 갖는 테스터(108)를 도시하고 있다. 일반적으로, 테스터(108)는 테스트 프로그램을 실행하고, 그에 응답하여 특정 DUT를 테스트하기 위해 설계된 복잡한 테스트 시퀀스(106)를 정의하는, 각 채널상의 데이터와 어드레스를 생성한다. 테스터(108)의 각 채널이 각 DUT에 제공되어, 채널 수에 대응하는 다수의 DUT가 동시에 테스트된다. DUT들이 반도체 웨이퍼(116)에 존재하고 있는 상태에서, 모두 N개의 채널들을 수신하는 프로브 카드(도시되지 않음)가 테스트 시퀀스(106)의 어드레스와 기입 데이터를 N개의 서로 다른 DUT의 로케이션에 동시에 전달한다. 그리고 나서, 테스터(108)는 이들 로케이션으로부터 판독하여, 생성되는 예상 데이터와 비

교를 수행한다. 비교 결과는 DUT의 로케이션으로부터 판독된 특정 비트가 에러인지 여부를 결정하는데 도움을 준다. 테스터(108)는 상기 판독 및 기입 사이클을 동일하거나 또는 다른 데이터 패턴으로 여러번 수행하여, 주어진 시간과 경비 제한 하에서, 가능한 한 많은 DUT의 로케이션들을 검증한다.

단위 시간당 테스트되는 DUT의 개수의 측면에서 처리량을 증가시키기 위해서는, 더 많은 채널을 갖는 더 큰 테스터가 제조되어야 한다. 그러나, 그러한 해결책은 비용이 과다하게 소요된다. 테스터는 복잡하고 고속 기계이며, 변형하거나 개선하는데 많은 시간과 비용이 필요하다. 더구나, 현대의 테스트의 하나의 채널은 50 내지 100개의 신호 배선들로 구성되므로, 테스터와 프로브 카드간의 채널 수를 증가시키면 모든 신호 배선을 프로브 카드에 연결하는 것이 물리적으로 비현실적 이게 된다. 따라서, IC 테스트 시스템의 처리량을 증가시키기 위한 더 효율적인 해결책이 필요하다.

발명의 상세한 설명

따라서, 본 발명의 실시예는 본질적으로 테스터와 다수의 DUT간의 릴레이로서 동작하는 인터페이스 회로에 관한 것으로, 각 채널상의 테스트 벡터들은 다수의 DUT로 팬아웃(fan-out)된다. 일반적으로, 테스트 벡터는, DUT 상으로 전달되는 어드레스, 데이터 값, 및 제어 신호와 같은 스티뮬러스(stimulus)를 포함하는 한편, 테스터에 의해 설정되었던 스티뮬러스들 간의 임의의 타이밍 제한을 유지한다. 이들 스티뮬러스에 대한 DUT의 응답은 인터페이스 회로에 의해 수집되어 테스터에 다시 릴레이된다. 원한다면, 인터페이스 회로는 응답에 기초한 에러 정정 능력을 추가로 구비할 수도 있다. 예를 들면, 각 DUT로부터의 응답은 DUT내(within-DUT) 및 DUT간(cross-DUT) 비교에 의해 내부 일관성(consistency)에 대해 평가되거나, 테스터로부터 수신된 예상 응답에 대한 비교에 의해 평가될 수도 있다. 그리고 나서, 평가 결과는 요약 또는 세부화된 형태로 테스터에 다시 제공된다.

다른 실시예에서, 인터페이스 회로는 테스터로부터 테스트 데이터, 예상 데이터(테스트 벡터) 및 제어 값을 수신하기 위한 입력을 구비하는 것을 특징으로 하고 있다. 출력은 테스트 데이터를 다수의 DUT로 구동한 후, 이어서 DUT로부터 데이터를 판독한다. 비교 회로는 각 DUT로부터 판독된 데이터 값과 테스터로부터 수신된 예상 데이터간의 비교를 수행한 것에 응답하여 에러 정보를 제공한다. 에러 정보에 대한 저장 영역은 인터페이스 회로의 일부로서 제공된다. 그러므로, 인터페이스 회로는 종래 테스터의 각 채널이 단일 DUT 뿐만 아니라 복수의 DUT를 양호하게는 병렬로 테스트하는 것을 가능하게 한다.

본 발명의 다른 실시예에 따르면, 복수 셋트의 테스터 입/출력(I/O) 라인을 구비하는 종래 테스터, 및 2개 이상의 DUT의 다수의 신호 로케이션을 접촉하기 위한 복수의 프로브 소자를 포함하는 복수의 DUT를 테스트하기 위한 시스템이 개시된다. 인터페이스 회로는 프로브 카드 위에 장착되고, 테스터 I/O 라인의 셋트들 중 하나에 결합되는 입력 및 프로브에 결합되는 출력을 구비하고 있다. 인터페이스 회로는 데이터 값을 그 입력으로부터 그 출력으로 전송하고, DUT로부터 판독된 데이터 값을 이용한 비교를 수행하여 DUT내의 에러(존재하는 경우)들을 결정한다. 그러므로, 단일 DUT를 테스트하기 위해 이전에 생성되었던 테스트 시퀀스를 방해하지 않고, 다수의 DUT들이 테스터의 각 채널에 의해 테스트된다. 테스터 프로그램은 테스트 시퀀스가 완료된 후에, 동일한 셋트의 테스터 I/O 라인을 통해 에러 정보를 판독하도록 변형된다.

특정 실시예에서, 비교 결과(에러 정보)는 이전에 기록된 데이터의 판독을 요구하는 테스터에 응답하여 동일한 채널을 통해 테스터에 리턴된다. 테스터 프로그램은 판독 요구에 응답하여 수신된 에러 정보가 단일 DUT보다는 복수의 DUT에 관련된다는 것을 인식하도록 변형된다.

특정 실시예에서, DUT는 메모리 디바이스이고, 에러 정보는 소정 어드레스/로케이션에 대해, 메모리 디바이스로부터 판독된 데이터 값과, 테스터로부터 인터페이스 회로에 의해 수신된 예상 데이터 값간의 차이를 나타낸다.

본 발명의 다양한 실시예들의 이러한 특징 및 장점 뿐만 아니라 다른 특징 및 장점들은 이하의 청구의 범위, 실시예, 및 도면을 참조하면 더 잘 이해될 것이다.

도면의 간단한 설명

도 1은 테스터의 채널당 하나의 DUT를 갖는, IC 디바이스를 테스트하기 위한 종래의 시스템을 도시한 도면.

도 2는 본 발명의 실시예에 따라, 종래의 테스터의 각 채널에 의해 다수의 DUT를 동시에 테스트하기 위한 시스템의 블럭다이어그램을 도시한 도면.

도 3은 본 발명의 실시예에 따라 인터페이스 회로의 블럭 다이어그램을 도시한 도면.

도 4는 본 발명의 실시예에 따라, 테스터의 단일 채널을 이용하여 16개의 DUT를 동시에 테스트하기 위한 채널-DUT 인터페이스 회로의 블럭 다이어그램을 도시한 도면.

도 5는 본 발명의 실시예에 따라, 테스터로부터 수신된 예상 데이터를 이용하여 IC 디바이스를 테스트하기 위한 시스템에서 수행되는 동작의 플로우다이어그램을 도시한 도면.

도 6은 테스터로부터 예상 데이터를 수신하지 않고 디바이스를 테스트하기 위한 플로우 다이어그램.

도 7은 메모리 디바이스의 16 비트 워드를 테스트하기 위한 종래 기술을 도시한 도면.

도 8은 본 발명의 다른 실시예에 따라, DUT들 간에서 수행되는 비교를 이용하여 16 비트 폭의 채널상에 4개의 DUT에 관련된 에러값을 제공하기 위한 기술을 도시한 도면.

도 9a 및 9b는 본 발명의 다른 실시예에 따라, 워드내 비교 및 DUT간 비교의 조합을 이용하여 4개의 DUT를 테스트하기 위한 기술을 도시한 도면.

도 10은 본 발명의 다른 실시예에 따른 프로브 카드를 도시한 도면.

실시예

도 2는 종래 반도체 테스터의 단일 채널을 이용하여 다수의 DUT를 테스트하기 위한 시스템으로서 본 발명의 실시예를 도시하고 있다. DUT는 메모리 칩과 같이 완전한 IC 다이가 될 수도 있고, 또는 메모리 부를 가지고 있는 임의의 반도체 디바이스일 수도 있다. DUT들은 통상은 유사한, 바람직하게는 동일한 디바이스들이다. 테스트 시스템은 테스트 프로그램(206)에 따라 동작하는 N개의 채널 테스터(108)를 특징으로 한다. 테스터(108)는 시스템 컨트롤러(104)를 특징짓는 IC 제조 및 테스트 라인의 일부일 수도 있다. 시스템 컨트롤러(104)는 DUT에 대한 제조 및 테스트 동작의 처리 흐름을 조율하는 기능을 한다. 테스트 프로그램(206)을 실행하고 있는 동안에, 테스트 시퀀스(106)가 이어지고, 데이터 값 및 관련 어드레스 및 다른 제어 신호를 포함하는 테스트 벡터들이 종래 기술에 따라 각 채널상에 제공된다. 예를 들면, 도 2에 도시된 실시예에서, 어드레스 버스(244)에 22개의 어드레스 라인, 및 데이터 버스(240)에 16개의 데이터 라인이 있다 (제어 신호는 도시되지 않음). 물론, 본 기술 분야의 통상의 지식을 가진 자라면, 원하는 전체 성능 및 테스트되는 DUT의 특정 타입에 따라 다른 버스 구성 및 폭이 선택적으로 사용될 수도 있다는 것을 알고 있을 것이다. 데이터 라인은 테스트 시퀀스 데이터를 인터페이스 회로(226)에 전송하는데 이용된다. 본 발명의 소정의 버전에서는, 여러 정보를 테스터(108)에 리턴시키는데 동일한 데이터 라인들을 이용할 수 있다. 이것은 테스터 채널의 데이터 라인에 대한 종래의 양방향 이용과 일치한다.

각 채널에 대한 인터페이스 회로(226)는 서브 회로(216a, 216b, ...)로 분할된다. 각 서브 회로(216)는 하나 이상의 DUT의 입력들의 조합을 구동하고, 출력들의 조합을 모니터링할 수 있다. 본 특정 실시예에 있어서, 각 서브 회로(216)는 4개의 DUT와 동시에 통신하도록 구성된다. 본 기술 분야의 통상의 지식을 가진 자라면, 각 서브 회로는 4개보다 많거나 적은 수의 DUT들과 인터페이싱하도록 구성될 수 있다는 것을 알고 있을 것이다. 각 서브 회로(216)는 단일 채널로부터 수신된 테스트 벡터를 4개의 DUT 각각에 전송한다. 데이터 값 및 이와 관련된 어드레스는 DUT 각각에 인가되고, DUT 각각의 대응하는 어드레스로의 어드레스 매핑이 이어진다.

테스터(108)가 판독 사이클에 들어간 경우, 테스터(108)는 예상 데이터 및 관련 어드레스를 각 서브 회로(216)에 제공한다. 각 서브 회로(216)는 DUT 내의 대응하는 어드레스로부터 데이터를 판독한다. 그 후, 예상 데이터와 판독 데이터간의 비교가 서브 회로에 의해 수행되어, DUT로부터 판독된 데이터가 임의의 에러를 포함하고 있는지 여부를 결정한다. 일부 경우에, 서브 회로는 여러 정보를 실시간으로 테스터(108)에 다시 제공하여, 테스터(108)가 추가적인 테스트를 중지할 수 있게 설계된다. 또한, 테스터는 예를 들면 DUT 내의 여분의 회로 소자 중에서 선택함으로써 DUT의 후속 분석이나 복구를 위해 에러를 수집할 수도 있다.

특정 실시예에서, 인터페이스 회로(226)에 의해 생성된 여러 정보는 예상 데이터를 수신하는데 이용되었던 동일한 채널을 통해 테스터(108)에 다시 제공된다. 그러나, 인터페이스 회로(226)에 의해 제공된 추가의 팬아웃은 모든 DUT에 대한 여러 정보를 동시에 리턴하는데 이용될 수 있는 비트수를 제한한다. 예를 들면, 16개의 DUT들이 도 1에 도시된 바와 같이 16 비트(데이터 버스) 채널에 의해 테스트되는 경우, 각 DUT에 채널이 할당되면, DUT당 단지 1비트의 여러 정보만이 제

공될 수 있다. 테스터(108)로 리턴하는데 필요한 에러 정보량은 처리 흐름이 요구하는 테스트의 타입에 따라 좌우될 수도 있다. 예를 들면, 최종 테스트에서는, DUT당 단순한 "고우-노 고우(go-no go)" 비트가 적절할 수 있다. 여분 복구를 위한 테스트의 경우, 복구가 가능한 회로 소자의 사이즈 및 구성은 요구되는 에러 정보의 대역폭에 영향을 준다.

도 3은 테스터 DUT 인터페이스 서브 회로(216a)의 내부 구성의 하이 레벨 블럭 다이어그램을 도시하고 있다. 채널 어드레스 포트(304)는 어드레스 버스(244)로부터의 어드레스 및 단일 채널의 제어 버스(도시되지 않음)로부터의 제어 신호를 수신하도록 제공된다. 테스터 데이터 I/O 포트(308)는 단일 채널의 데이터 버스(240)로부터 데이터 값을 수신하도록 제공된다. 예상 데이터 레지스터(309)는 채널의 제어 버스(도시되지 않음)로부터 수신된 판독 제어 신호에 의해 클럭킹되어, 예상 데이터를 래칭(latch)한다. 이 실시예에서, 16라인의 데이터 버스 중 4개가 서브 회로(216a)에 의해 이용되어, 4개의 DUT(310a, 310b, 310c, 및 310d)에 관련된 테스터(108)에 에러 정보를 리턴한다. 나머지 12개 라인들은 다른 서브 회로(216b, 216c, 및 216d)에 의해 동일한 방식으로 이용된다.

DUT 측에서, DUT I/O 포트(314a, 314b, 314c, 및 314d)는 대응하는 DUT와의 데이터 인터페이스로서 제공된다. 유의 할 점은 테스터 I/O 포트(308)를 통해 수신되는 16개의 데이터 비트는 모두 4개의 DUT I/O 포트에 복사되어, 상기 데이터를 각각의 DUT에 다시 전송한다는 점이다. 다수의 DUT 어드레스 포트(320a, 320b, 320c, 및 320d)는 수신된 어드레스 또는 매핑된 어드레스를 각각의 DUT에 전달하기 위해 제공된다.

비교 회로(330)는 DUT 각각으로부터 판독 데이터를 수신하고 비교를 수행하여, 판독 데이터가 임의의 에러를 나타내는지 여부를 결정한다. 도 2 내지 4의 실시예에서, 16:1의 압축이 압축 회로(330)에 의해 수행된다. 이것은 DUT의 각 로케이션에서 데이터의 16비트마다, 각 서브 회로(216)에 의해 1 에러 비트가 생성되어, 각 어드레스에 대해 총 16 에러 비트가 인터페이스 회로(226)에 의해 테스터 채널의 데이터 버스(240) 상에 제공된다는 것을 의미한다. 도 4는 4개의 테스터-DUT 인터페이스 서브 회로(216a, 216b, 216c, 및 216d)의 완전한 셋트 및 그들이 단일 테스터 채널의 어드레스 및 데이터 버스에 결합되는 방법을 도시하고 있다. 각 서브 회로(216)는 4비트의 에러 정보를 데이터 버스상에 제공하여, 4개의 DUT 각각의 4개 셋트에 대응하는 총 16비트를 제공한다. 테스터(108)의 단일 채널을 이용하여, 더 많은 수의 DUT를 동시에 테스트하기 위한 인터페이스 회로(226)를 구성하는 다른 방법은 본 기술 분야의 통상의 지식을 가진 자라면 충분히 고안할 수 있다. 이들 구성은 처리 흐름에 의해 요구되는 에러 데이터량에 따라 좌우된다. 예를 들면, 최종 테스트 시나리오에서, 각 DUT에 대해 단일 오류 비트가 충분하다. 다른 경우에는, 어드레스/로케이션당 하나의 오류 비트가 충분할 수 있다. 또한, 도 4에 도시된 병렬 버스 접근법을 이용하는 것보다, 선택적으로, 직렬 링크가 인터페이스 회로(226)를 테스터(108) 또는 시스템 컨트롤러(104)에 접속하는 데에 이용될 수도 있다.

다시, 도 2를 참조하면, 인터페이스 회로(226)는 테스터(108)의 개별적인 채널에 결합되는 프로브 카드(212)의 일부일 수도 있다. 도 2에 도시된 프로브 카드(212)는 M개의 팬아웃을 제공하므로, 그 채널은 웨이퍼(116) 상에서 M개의 다른 DUT에 동시에 결합된다. 그러므로, 채널당 하나의 DUT로, 16개의 DUT를 테스트할 수 있는 종래의 N-채널 테스터(108)를 이용한 시스템은 N개의 프로브 카드를 구비한다면 $M \times N$ 개의 DUT를 동시에 테스트할 수 있을 것이다.

도 5는 본 발명의 실시예에 따라, IC 디바이스를 테스트하기 위한 시스템에서 수행되는 동작의 플로우 다이어그램을 도시하고 있다. 동작은 테스터(108)가 단일 DUT에 입력으로서 공급될 테스트 시퀀스에 대한 어드레스 및 데이터를 생성하는 단계 504에서 시작할 수 있다. 이것은 테스터(108)의 각 채널 상에 데이터 값 및 관련 어드레스를 생성하는 종래 테스트 엔지니어링 기술에 따라 수행된다. 동작은, 마찬가지로 종래 기술에 따라, 테스터가 데이터 및 관련 어드레스를 다수의 테스터 채널 상으로 동시에 전송하는 단계 508로 진행한다. 그리고 나서, 동작은 인터페이스 회로(226)가 단일 채널을 통해 데이터 및 어드레스를 수신하고, 그에 응답하여 데이터를 다수의 DUT에 동시에 인가하는 단계 512로 진행한다. 각 DUT에 제공된 어드레스는 테스터로부터 수신된 어드레스와 동일할 수 있다. 선택적으로, 테스터(108)로부터 수신된 어드레스는 DUT 내의 상이한 대응 어드레스/로케이션들에 매핑될 수 있다. 이러한 매핑은 인터페이스 회로(226)가 단일 DUT 동작에 대해 구성될 수 있게 하고, 그렇게 프로그래밍된 경우, 바람직하게는 테스트 시퀀스가 종료된 이후에, 테스터(108)는 임의의 DUT 내의 임의의 로케이션에 액세스할 수 있다. 데이터를 DUT에 전송하는 경우에 인터페이스 회로(226)로 인해 가산된 레이턴시(latency)(각 채널이 단지 단일 DUT만을 취급했을 경우에는 나타나지 않을 수도 있음)는 테스트 프로그램이나 인터페이스 회로(226) 내의 연속적인 판독 또는 연속적인 기입 명령들 사이에 시스템적인 지연을 삽입함으로써 용이하게 핸들링될 수 있다.

테스터(108)가 채널을 통해 어드레스와 데이터를 전달한 후, 동작은 단계 516으로 진행하여, DUT가 메모리 디바이스 또는 메모리부를 포함하는 임의의 반도체 디바이스인 경우에, 테스터(108)는 선택적으로 리프레시 사이클을 작동시킨다. 또한, 단계 512와 같이, DUT의 전기적 파라미터를 스트레스 주도록 설계된 긴 사이클 테스트 및 다른 테스트 패턴이 인터페이스 회로(226)에 전달된다. 인터페이스 회로(226)는 그러한 시퀀스들을 모든 DUT에서 모방하도록 구성된다. 그리고 나서, 동작은 단계 519로 진행하여, 테스터는 예상 데이터 값을 인터페이스 회로(226)에 전송하고, 각 DUT로부터의 판독을

동시에 개시한다. 예상 데이터 값 및 관련 어드레스는 일반적으로, 단계 508에서 테스터에 의해 전송되었던 데이터 및 어드레스와 동일하다. 그 후, 동작은 단계 520으로 진행하여, 인터페이스 회로(226)는 채널로부터 예상 데이터를 래치하고, DUT의 대응하는 로케이션으로부터 데이터를 판독하며, 판독 데이터와 예상 데이터를 비교하여 판독 데이터에 임의의 에러가 존재하는지 여부를 결정한다.

그리고 나서, 그러한 비교 결과는 압축되어, 테스터(108)나 시스템 컨트롤러(104)에 다시 전송되거나, 메모리(399, 도 3 참조)에 저장된다. 에러 정보는 예상 데이터가 도착하는 동일한 채널을 이용하여, 실시간으로 또는 후처리로서 테스터에 리턴된다. 압축은 각 채널에 의해 테스트되는 DUT의 개수 및 채널내의 데이터 버스의 폭에 좌우된다. 예를 들면, 16개의 DUT들이 16 비트 데이터 버스를 갖는 각 채널에 의해 테스트되는 경우, 16:1 압축이 수행되어 DUT당 1비트의 에러 정보가 이용 가능하다. 물론, 이것은 DUT로부터 판독된 소정의 로케이션에서 16 비트 중 하나 이상에 에러가 존재하는 경우에, 에러들의 정확한 비트 로케이션이 실시간으로 테스터에 다시 전송되지 않는다는 것을 의미한다. 원한다면, 인터페이스 회로(226)는 테스터(108)가 임의의 특정 DUT의 임의의 오류발생 로케이션을 조회하여 에러의 정확한 비트 로케이션을 결정할 수 있도록 하는 바이패스 모드로 구성될 수도 있다. 이것은 통상 테스트 시퀀스의 종료 이후에 수행될 것이다. 선택적으로는, 테스터(108)나 시스템 컨트롤러(104)에 의해 후처리를 위한 에러 데이터를 수집하는데 메모리(399)를 이용할 수도 있다.

도 6은 예상 데이터를 테스트 채널을 통해 전달하지 않고, 인터페이스 회로(226)를 이용하여 에러 정보를 결정하기 위한 선택적인 기술의 플로우 다이어그램이다. 인터페이스 회로는 테스터로부터 수신된 명령에 응답하여, 이와 같이 동작하고 또한 일반적으로는 다른 모드들로 동작하도록 구성될 수 있다. 예를 들면, Ray and Miller에 의한 "Efficient parallel testing of integrated circuit devices using a known good device to generate expected responses(P078)"라는 제목의 미국 특허 출원서를 참조하라. 단계 618에서, 인터페이스 회로(226)가 테스트 데이터를 그에 대응하는 DUT에 인가하면, 동작은 단계 619로 진행하여, 테스터(108)는 각 채널상에서 판독 사이클을 개시하여 이전에 기입된 로케이션으로부터 판독한다. 단계 620에서, 인터페이스 회로(226)는 대응하는 DUT로부터 판독함으로써 응답하고, DUT간 및/또는 DUT내의 데이터 값의 비교를 수행하여 DUT 내의 임의의 에러를 결정한다. 예를 들면, 인터페이스 회로(226)는 동일한 DUT내의 로케이션으로부터 판독된 비트들의 그룹의 비교를 수행하도록 구성될 수 있다(각 그룹은 단계 618에서 기입된 동일한 비트 패턴을 가짐). 그러한 종래 기술은 도 7을 참조하여 이하에 설명한다. 종래 기술에 더하여, 또는 종래 기술을 대신하여, 인터페이스 회로(226)는 상이한 DUT들 내의 로케이션들로부터 판독한 비트의 비교를 수행하도록 추가로 구성될 수 있다. 이 후자의 기술은 도 8과 관련하여 이하에 설명된다. "워드 내(within word)" 및 "DUT간" 비교의 이들 2개의 기술을 조합하는 것은 도 9a 및 9b에 도시된다. 그러므로, 도 5의 실시예와는 대조적으로, 도 6의 테스터(108)는 테스트 시퀀스 동안에 예상 데이터를 인터페이스 회로(226)에 전송하지 않는다. 오히려, 인터페이스 회로(226)는 아래의 도 7 내지 9와 같은 DUT간 및 DUT내 비교와 선택적인 통계를 수행하여, 상당히 높은 신뢰성으로 DUT 내의 에러를 예측한다. 에러 데이터의 적절한 저장 및 압축도 또한 수행된다. 예상 데이터를 전송하는 사이클을 제거하는 것은 DUT를 테스트하는 데 필요한 시간을 더 감소시키고, 따라서, 더 효율적인 테스트 방법을 제공한다.

단계 520/620에서 에러 정보를 생성하는데 어느 기술을 이용하는지에 관계없이, 동작은 도 5의 단계 524 및 528(및 도 6의 대응하는 단계)로 진행하여, 인터페이스 회로(226)는 특정 어드레스들로부터 판독하기 위한 테스터(108)로부터의 요청에 응답하여 각 DUT에 관련된 에러 정보를 제공한다. 그리고 나서, 동작은 단계 532로 진행하여, 수정된 테스트 프로그램에 따라, 테스터(108)는 새로운 에러 정보가 단일 DUT가 아닌 다수의 DUT에 관련된 것임을 인식하고, 그에 따라 각 DUT에 대해 저장된 에러 정보를 갱신한다. 상기 설명한 단계 504-532는 테스트 시퀀스(106, 도 1 참조)에 의해 요구되는 횟수만큼 다수 반복된다.

상기 설명한 바와 같이, 도 5의 단계 520은 비교 회로(330, 도 3 참조)에 의한 비교를 수행하여 DUT로부터 얻어진 판독 데이터에 임의의 에러가 존재하는지 여부를 결정한다. 비교를 수행하는 수개의 기술이 본 명세서에 제공된다. 상기 소개되었던 그러한 하나의 기술에서, 비교 회로(330)는 테스터(108)로부터 수신된 예상 데이터 및 DUT로부터의 판독 데이터값의 대응하는 비트에 대해 배타적 OR(XOR) 연산을 수행한다.

반면에, 도 7은 DUT의 로케이션이나 데이터 워드 내에서 비교를 수행하기 위한 종래 기술을 도시하고 있다. 이것의 목적은 단일 DUT의 16비트 워드 내에 에러가 존재하는 경우, 에러를 표시하는 4비트 압축 에러값을 얻는 것이다. 이 기술을 이용하여, 4개의 DUT들이 병렬로 테스트되어, 16 비트의 에러 데이터가 채널의 16 비트 데이터 버스를 통해 테스터(108)로 리턴될 수 있게 된다. 도 7에서, 비교되어야 할 16 비트 데이터 워드는 W, X, Y, Z로 표시된 4비트의 4개 그룹으로 분할된다. 이 경우에, 이 데이터 워드에 기입되었던 데이터의 패턴이 매 4 비트마다 반복되어 각 그룹은 동일한 비트 패턴을 가지고 있는 것으로 가정한다. 도 7의 특정 예에서, C1OW, C1OX, C1OZ는 모두 양의 결과를 나타낸다. 이것은 데이터 워드 내의 W, X, Z 비트들 모두에 에러가 없다는 것을 의미한다. 반면에, C1OY는 비트 로케이션 Y에 대해 음의 결과를 나타내

지만, 어느 그룹이 비트 Y에 에러를 포함하는지까지는 지정하지 않는다. 이러한 탑입의 압축된 에러 정보는 데이터 워드 내에서 발견되는 에러들에 대한 완벽한 그림을 제공해 주지는 못하지만, 그럼에도 불구하고, DUT가 오류 발생 Y 비트를 제공하는 회로를 대체할 수 있는 여분 회로를 포함하는 경우와 같이 특정 상황에서는 유용하다.

도 7의 종래 접근법의 하나의 단점은, 소정의 로케이션의 대응하는 모든 비트가 에러 상태인 경우에, 비교 CIO의 잘못된 양의 결과가 발생할 수 있다는 점이다. 예를 들면, 음의 결과를 나타내는 CIOY에 있어서, 4개의 대응하는 Y 비트들 중 많아야 1, 2, 또는 3개가 에러 상태가 될 수 있다. 그러나, 4개의 모든 Y 비트들이 에러 상태라면, CIOY는 양의 결과를 나타낼 것이다. 왜냐하면, XOR 게이트의 4개의 모든 입력이 동일한 값을 가지기 때문이다. 4개의 대응하는 모든 비트들이 에러 상태인 경우는 드물지만, 그러한 잘못된 양의 결과의 발생을 제거하거나 최소한 감소시키는 것이 바람직하다. 이것은 인터페이스 회로(226)를 바이패스 모드로 구성하고, 테스터(108)를 이용한 종래 테스트를 절대 정확도를 위해 다중 디바이스 중 단일 디바이스를 테스트하도록 운영함으로써 달성될 수 있다.

도 8은 비교를 수행하는 다른 기술을 도시하고 있다. 그러한 기술은 상이한 DUT들로부터의 비트들을 비교하는 것을 포함한다. 도 8은 4개의 DUT간 비교를 수행하는 것을 도시하고 있지만, 이러한 개념이 실용적인 한계 내에서 임의의 수의 DUT로 구현될 수 있는 것은 당연하다. 다시 한번, 이 구성은, 각 XOR 게이트에 의해 수행된 비교는 동일해야만 하는 비트 값에 관한 것이므로, 모든 비트가 동일한 0 또는 1의 값을 가지고 있는 경우에만 양의 결과가 발생한다고 가정한다. 이 예에 있어서, 각 데이터 워드는 16비트로 구성된다. 제1 XOR 게이트(880)는 각 DUT에서 비트 0의 Compare Across Duts (CAD0)를 제공한다. 마찬가지로, XOR 게이트(881)로부터의 CAD1은 각 DUT에서 비트 1을 비교하는 결과를 제공하고, CAD15까지 계속된다. 도 7의 종래 기술에 요구되는 반복 패턴과는 대조적으로, 이 접근법은 데이터의 임의의 패턴이 각 워드에 기입되는 것을 가능하게 한다. 그러나, CAD 결과는, 어떤 DUT가 에러를 포함하고 있는지를 나타내지 않고, 하나 이상의 비트가 에러 상태에 있다는 것만을 나타낸다. 4개의 서로 다른 DUT에 4개의 대응하는 비트가 모두 에러 상태에 있는 경우, CAD 결과는 잘못된 양의 결과를 나타낼 것이다. 이전에 언급한 바와 같이, 이러한 상황은 절대 정확도를 위해 DUT 중 적어도 하나를 바이패스 모드로 테스트함으로써 회피될 수 있다.

도 9a 및 9b는 도 7의 종래 기술이나 도 8의 DUT간 비교보다 양의 결과의 신뢰성이 몇 단위 정도 증가된 워드내 및 DUT 간 비교의 조합을 도시하고 있다. 도 9a 및 9b는 각 데이터 워드가 4개의 그룹을 포함하고, 한 그룹이 4비트를 갖는 경우에, 그룹 내의 4 비트 중 최초의 비트인 비트 W에 대한 비교 회로를 도시하고 있다. 그러므로, 도 9a 및 9b에 도시된 회로는 3번 반복되어, 비트 X, Y, 및 Z에 대한 비교 결과를 제공한다.

XOR 게이트(910, 912, 914, 및 916)의 제1 열은 각각 결과 CIOW0, CIOW1, CIOW2, 및 CIOW3를 제공한다. CIOW0은 DUT0의 단일 워드 내의 비트 W의 워드내 비교의 결과이다. 마찬가지로, CIOW1은 DUT1의 단일 워드 내의 비트 W의 비교의 결과이다. 나머지는 위와 같은 방식이다.

도 9a 및 9b의 배타적 OR 게이트(920, 922, 924, 926)의 제2 열은 각각 CADW0, CADW1, CADW2, 및 CADW3을 제공한다. CADW0은 각 DUT의 셋트 0 내에서 비트 W의 비교이고, CADW1은 각 DUT의 셋트 1의 비트 W를 비교한다. 그러므로, CADW 결과는 단지 4 비트(W 비트)만이 비교된다는 것을 제외하고는 도 8의 CAD 결과와 동일하다.

마지막으로, 도 9a 및 9b의 게이트의 제3 열(930-945)은 전체 16개의 NOR 게이트이고, 각각이 OKW_결과를 제공한다. 처음 4개의 결과 OKW00_ 내지 OKW03_는 DUT0 내의 워드를 정의하는 4셋트 중 하나 이상에서 비트 1의 에러(존재하는 경우)를 나타낸다. 마찬가지로, OKW10_ 내지 OKW13_는 DUT1의 대응하는 워드를 정의하는 4 셋트의 비트 W 내의 에러를 나타낸다. 그리고, 나머지는 위와 같은 방식이다. 유의할 점은, 소정의 DUT 내의 데이터 워드의 특정 셋트의 비트 W의 에러를 표시하는데 하나의 비트가 이용되므로, 에러의 정확한 로케이션이 주어진다는 점, 즉 압축이 수행되지 않는다는 점이다. 도 9a 및 9b의 회로가 비트 X, Y, 및 Z에 대해 반복된 경우, 임의의 소정의 DUT 내에서, 데이터 워드의 임의의 비트 내의 에러들을 정확하게 나타내는데 총 $16 \times 4 = 64$ 에러 비트가 이용 가능하다.

2개 이상의 DUT에 관련된 에러 정보를 단일 채널을 통해 테스터(108)로 리턴하기를 원한다면, 채널의 16 비트 어드레스 버스를 효율적으로 이용하기 위해 16 비트의 에러 정보를 4비트로 감소시키는 압축이 수행될 수 있다. 예를 들면, 각 DUT에 대해 단지 4비트의 압축된 에러 데이터에 있어서, 각 비트는 16 비트 데이터 워드의 대응하는 4비트 셋트 내의 에러(존재하는 경우)를 나타낸다. 이전에 언급한 바와 같이, 그럼에도 불구하고 테스터(108)는 예를 들어, 오류 발생된 회로를 교체할 수 있는 여분 회로가 DUT 내에서 이용 가능한 경우 DUT를 복구함으로써 그러한 정보를 가치있게 이용할 수 있다.

도 10은 본 발명의 다른 실시예에 따라, 인터페이스 회로(226)의 서브 회로(216a, 216b)를 포함하는 프로브 카드(1000)를 도시하고 있다. 프로브 카드(1000)는 그 특징으로서, DUT의 신호 포인트를 각각의 서브 회로에 전기적으로 접속하기 위한 다수의 프로브 소자(1004)를 구비하고 있다. 프로브 소자(1004)는 서브 회로의 IC 다이가 부착되는 반대측 상에서

스페이스 트랜스포머(1008, space transformer)에 부착된다. 신호는 탄력성이 있는 인터포우저(1016, interposer)를 이용하여, 임피던스 제어된 테스터 인터페이스 보드(1012)로 송수신된다. 테스터 채널은 인터페이스 보드(1012) 내의 트레이스(traces)로서 나타난다. 스페이스 트랜스포머(1008) 및 인터페이스 보드(1012)는 일반적으로, 예를 들면 패스너(1024, fastener)를 이용하여, 서로 고정된 관계를 유지한다. 모든 프로브 소자들이 테스트 대상 웨이퍼(116)와 평면 정렬 상태가 되도록, 평탄화기(1020, planarizer)가 제공된다. 동작시, 프로브 카드는 웨이퍼의 표면 위까지 낮춰져, 프로브 소자들이 웨이퍼를 형성하는 DUT의 신호 포인트에 접속한다. 이러한 실시예에 대한 더 상세한 내용은 미국 특허 출원 번호 제08/554,902호(95.9.11), "Probe Card Assembly With Space Transformer and Interposer" 또는 대응하는 PCT 출원 공개 번호 제WO96/15458호(POO6)(1996.5.23 공개)에 나타나 있다.

요약하면, 종래 테스터의 단일 채널을 이용하여 다수의 DUT를 병렬로 테스트하기 위한 본 발명의 여러가지 실시예들이 개시되어 있다. 본 기술 분야의 통상의 지식을 가진 자라면, 본 발명은 여러가지 다른 조합 및 환경에서 이용될 수 있고, 여기에 설명된 발명의 개념 내에서 여러가지 변화와 변경을 할 수 있다는 것을 잘 알고 있을 것이다. 예를 들면, 상기 다른 실시예에서 설명된 인터페이스 회로는 일반적으로, 프로브 카드 상에 존재하고, 각각이 서브 회로에 대응하는, 하나 이상의 접적 회로칩으로서 구현될 수 있을 것이다. 이와 같이, DUT 포트 내의 구동 및 감지 전자 장치들이 물리적으로 실제 DUT에 더 근접함으로써, DUT와 비교 회로간에 문제가 덜 발생하고, 보다 비용 효율적인 전기적 접속을 제공한다. 웨이퍼 프로브 카드 실시예에 대한 선택적 방법은, 테스터 채널과 패키징된 IC 디바이스의 트레이(tray) 간의 테스트 설치물 상에 인터페이스 회로를 배치한다. 여기서, 각 DUT는 웨이퍼의 일부라기 보다는 패키징된 디바이스의 일부이다. 따라서, 모든 그러한 변형 및/또는 변화는 청구의 범위의 범주 내에 포함된다.

(57) 청구의 범위

청구항 1.

제1 소자들의 2개 이상의 셋트 -상기 제1 소자들의 셋트 각각은 개별 테스트 대상 디바이스(Device Under Test, DUT)의 복수의 신호 로케이션에 접촉하기 위한 것임-,

테스터의 I/O 라인들의 셋트에 연결되는 입력 및 상기 제1 소자들의 2개 이상의 셋트에 연결되는 출력을 가지며, 상기 테스터로부터 수신된 기입 데이터값을 상기 소자들의 셋트 각각에 제공하는 제1 회로 - 상기 제1 회로는 각각의 DUT로부터 판독 데이터값을 판독하고, 상기 I/O 라인들의 셋트를 통하여 상기 테스터로부터 판독 명령을 수신한 것에 응답하여, 상기 판독 데이터값과 상기 테스터로부터 수신된 예상 데이터값을 비교하여, 각각의 DUT 내의 에러를 결정하고, 상기 각각의 DUT 내의 에러를 나타내는 에러값을 상기 I/O 라인들의 셋트를 통하여 상기 테스터로 전송하도록 구성되며, 상기 제1 회로는 상기 I/O 라인들의 셋트 중 하나 이상의 데이터 라인을 통하여 상기 기입 데이터값을 수신하도록 연결되고, 상기 에러값은 상기 기입 데이터를 전달하는 데에 사용된 상기 하나 이상의 데이터 라인에 의해 전달됨-,

제2 소자들의 2개 이상의 셋트 -상기 제2 소자들의 셋트 각각은 개별 DUT의 복수의 신호 로케이션에 접촉하기 위한 것임 -, 및

상기 테스터의 다른 I/O 라인들의 셋트에 연결되는 입력 및 상기 제2 소자들의 2개 이상의 셋트에 연결되는 출력을 갖고, 상기 테스터로부터 수신되고 상기 다른 I/O 라인들의 셋트 중 하나 이상의 데이터 라인에 의해 전달된 기입 데이터값을 상기 제2 소자들의 셋트들 각각에 제공하는 제2 회로 -상기 제2 회로는 상기 제2 소자에 의해 접촉되는 각각의 DUT로부터 판독 데이터값을 판독하고, 상기 다른 I/O 라인들의 셋트를 통하여 상기 테스터로부터의 판독을 수신한 것에 응답하여, 상기 판독 데이터값을 이용한 비교를 수행하여 각각의 DUT 내의 에러를 결정하고, 상기 제2 소자에 의해 접촉되는 각각의 DUT 내의 상기 에러를 나타내는 제2 에러값을 상기 다른 I/O 라인들의 셋트를 통해 상기 테스터에 전송하도록 구성되며, 상기 제2 에러값은 상기 다른 I/O 라인들의 셋트 중에서 상기 기입 데이터값을 전달하는 데에 사용된 상기 하나 이상의 데이터 라인에 의해 전달됨-

를 포함하는 컨택트 어셈블리.

청구항 2.

제1항에 있어서,

상기 제1 및 제2 회로는 별개의 ASIC(Application Specific Integrated Circuit)으로서 형성되는 컨택트 어셈블리.

청구항 3.

제1항에 있어서,

상기 제1 회로의 상기 입력에 연결된 I/O 라인들의 셋트를 갖는 테스터와 조합하여 사용되는 컨택트 어셈블리.

청구항 4.

제1항에 있어서,

메모리 디바이스인 DUT를 테스트하도록 구성된 테스터와 조합하여 사용되는 컨택트 어셈블리.

청구항 5.

제4항에 있어서,

상기 에러값의 각 비트는 상기 I/O 라인들의 셋트 중 상기 하나 이상의 데이터 라인 상에서 전달되고, 각 DUT의 에러 상태는 상기 에러값의 개별 비트에 의해 표현되는 컨택트 어셈블리.

청구항 6.

제1항에 있어서,

상기 제1 및 제2 소자들의 셋트는 각각의 DUT를 패키지화된 반도체 집적 회로 디바이스의 일부로서 접촉시키도록 구성된 컨택트 어셈블리.

청구항 7.

제1항에 있어서,

상기 판독 데이터값 및 상기 에러값은 각각 M 비트 이하의 길이이고, 상기 각각의 I/O 라인들의 셋트 각각의 데이터 라인들은 M 비트 폭이며, 상기 에러값은 각각 M 비트 길이까지의 데이터값을 저장할 수 있는 2개 이상의 DUT의 대응 로케이션들에서의 에러에 관한 정보를 포함하는 컨택트 어셈블리.

청구항 8.

제1항에 있어서,

상기 에러값은 상기 DUT 내에서의 에러의 정확한 비트 로케이션을 나타내지는 않는 컨택트 어셈블리.

청구항 9.

제1항에 있어서,

상기 제1 및 제2 회로는 메모리 디바이스인 각각의 DUT와 통신하고, 상기 에러값은 소정의 메모리 어드레스에 대하여 메모리 디바이스로부터 관독된 데이터값과 상기 제1 및 제2 회로에 의하여 수신된 예상 데이터값 간의 차이를 나타내는 컨택트 어셈블리.

청구항 10.

삭제

청구항 11.

삭제

청구항 12.

삭제

청구항 13.

삭제

청구항 14.

삭제

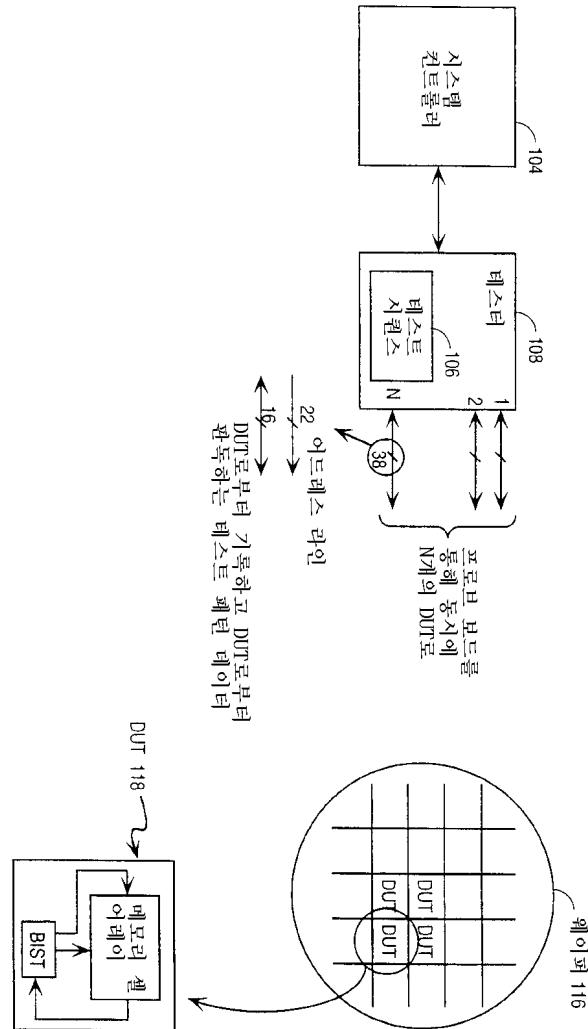
청구항 15.

삭제

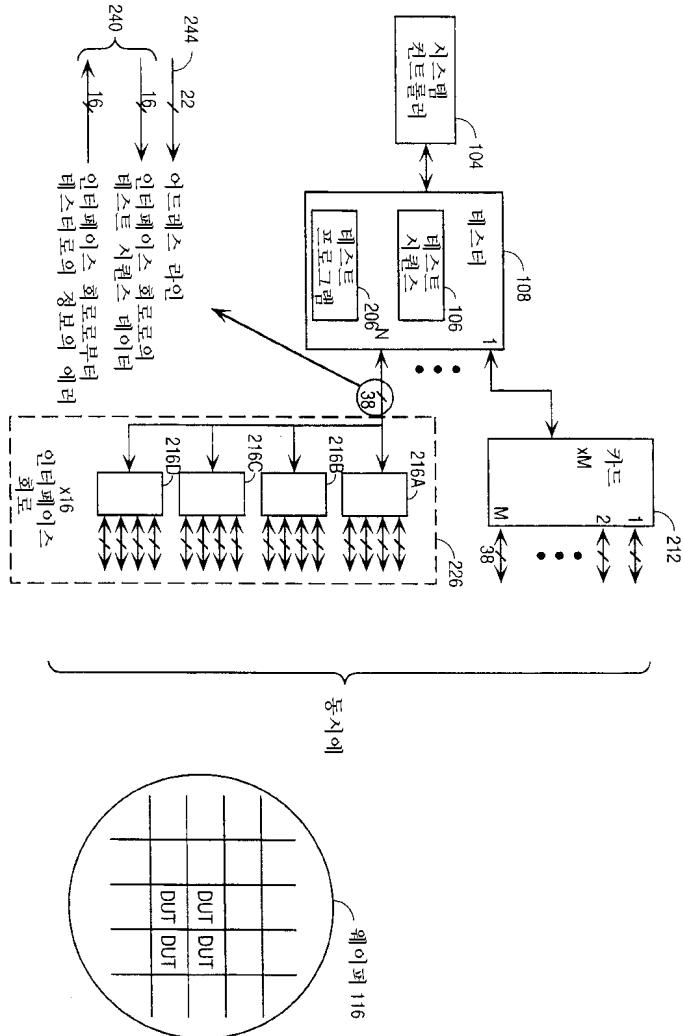
도면

도면1

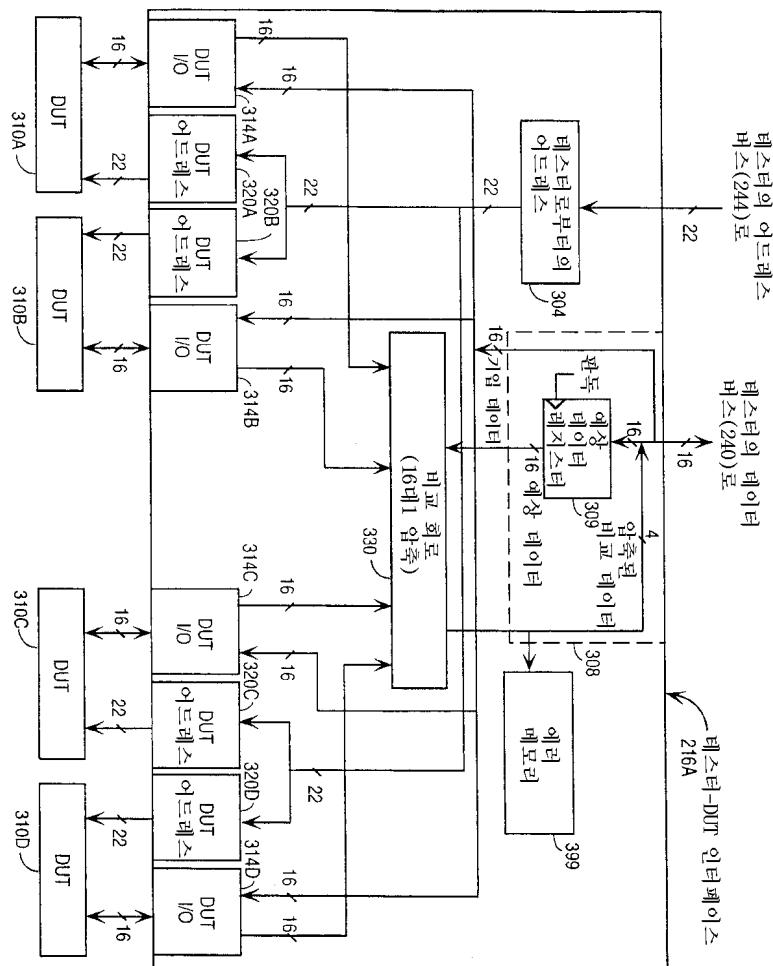
(총래 기술)



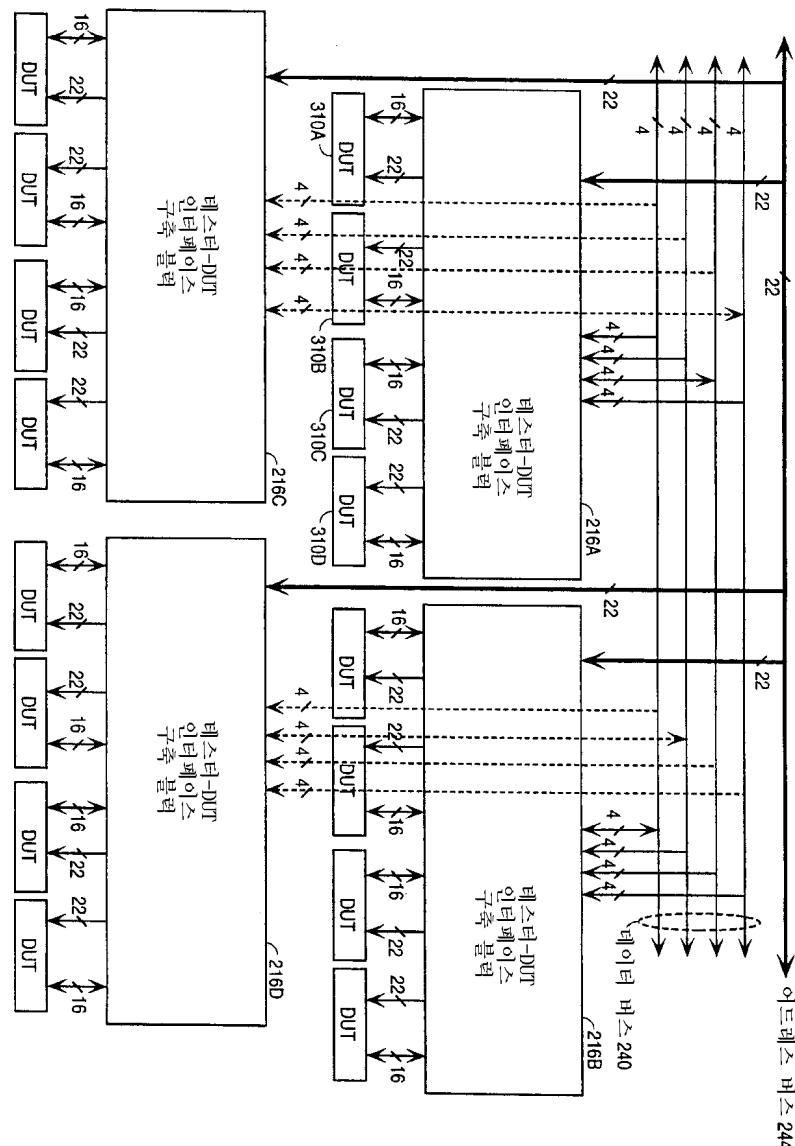
도면2



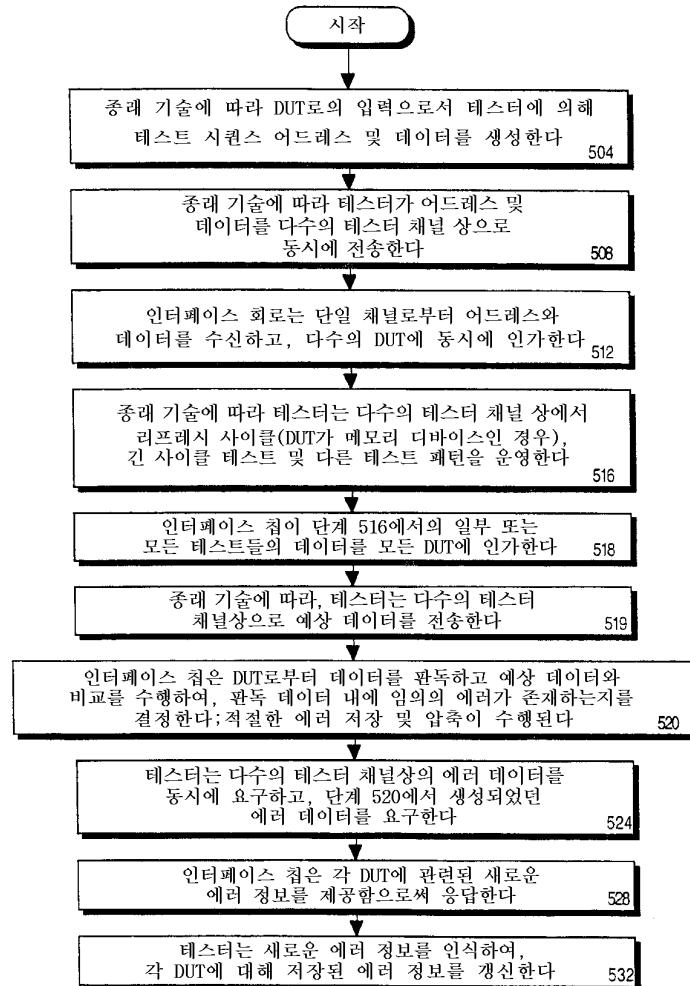
도면3



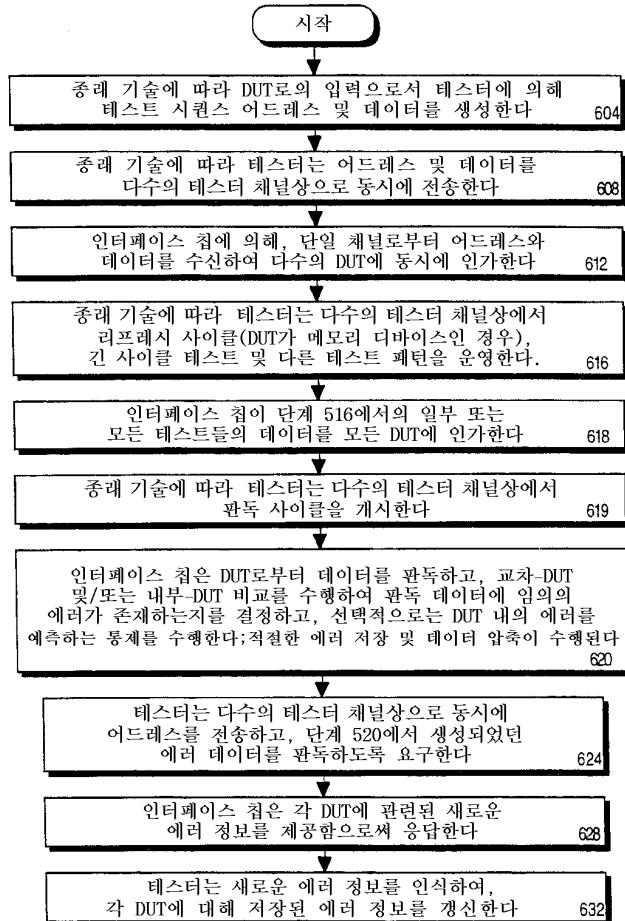
도면4



도면5

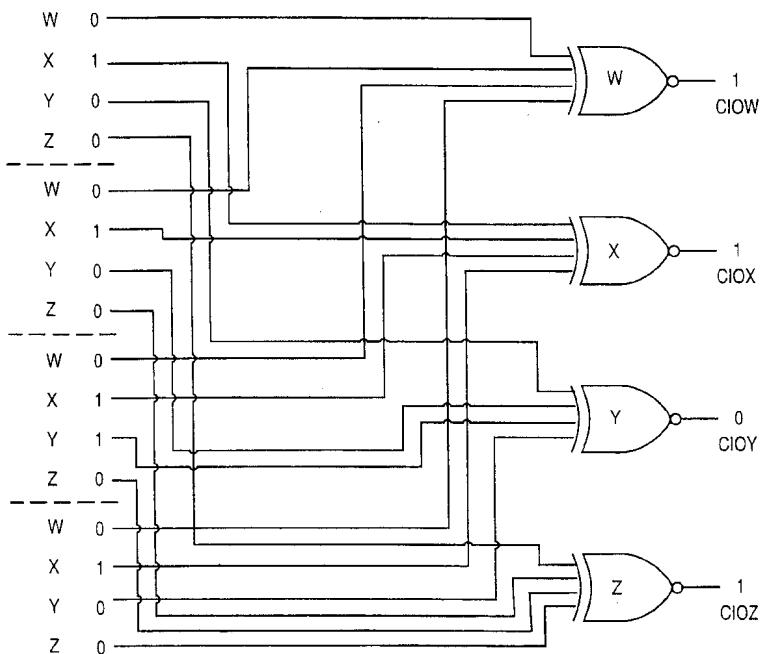


도면6

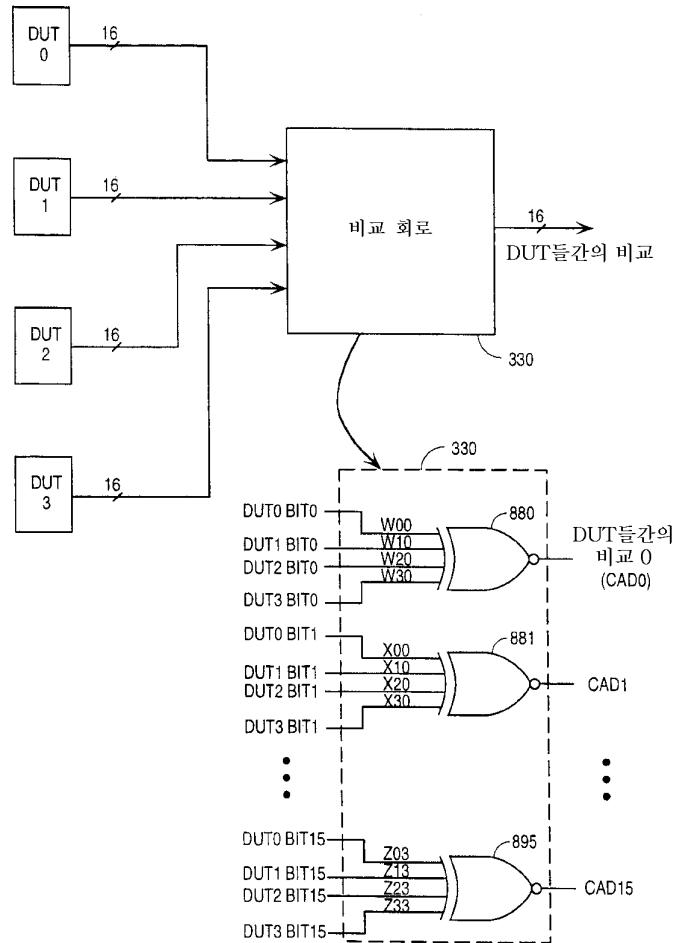


도면7

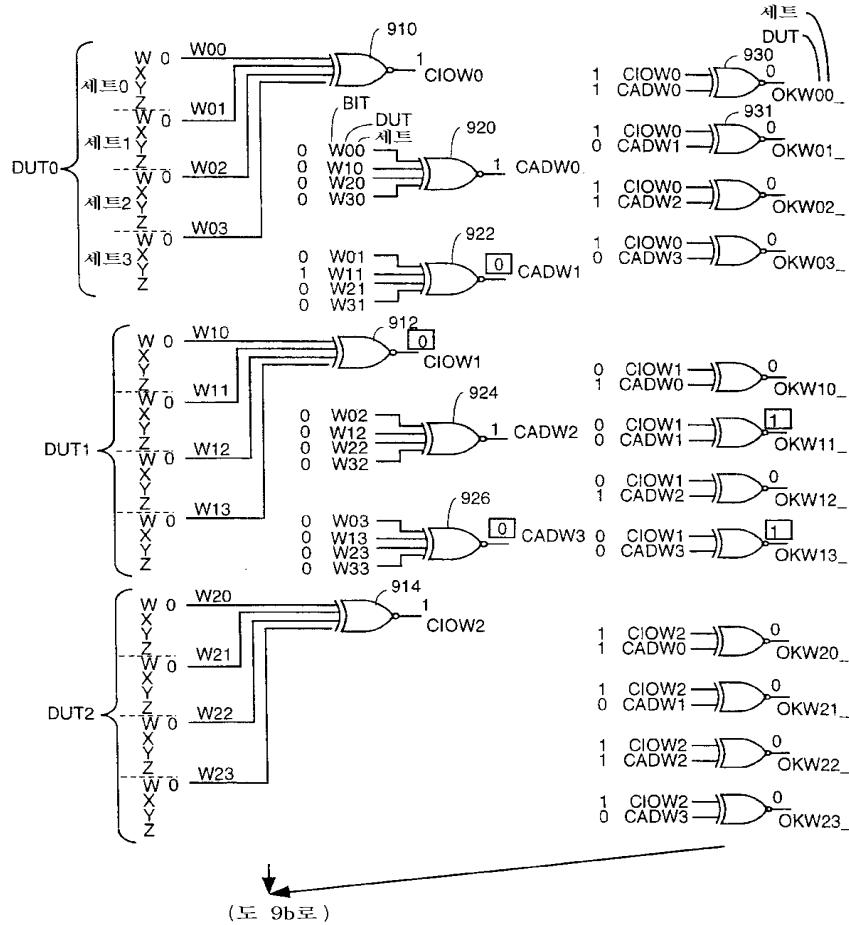
(종래 기술)



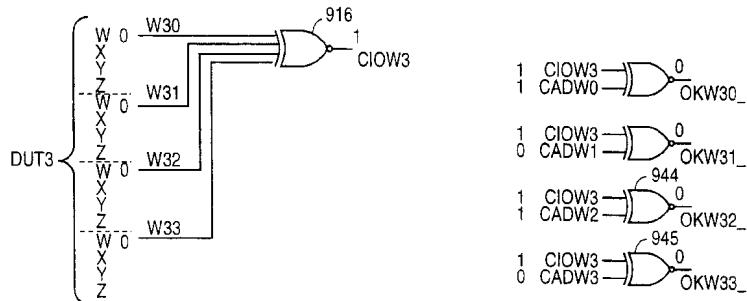
도면8



도면9a



도면9b



도면10

