(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0105984
(43) 공개일자 2016년09월08일

(51) 국제특허분류(Int. Cl.)
G01R 31/319 (2006.01) *G01R 31/26* (2014.01)
G01R 31/3183 (2006.01)
(52) CPC특허분류
G01R 31/31908 (2013.01)
G01R 31/2601 (2013.01)
(21) 출원번호 10-2016-7023561(분할)
(22) 출원일자(국제) 2010년02월05일
심사청구일자 2016년09월01일
(62) 원출원 특허 10-2012-7009910
원출원일자(국제) 2010년02월05일
심사청구일자 2015년01월14일
(85) 번역문제출일자 2016년08월26일
(86) 국제출원번호 PCT/US2010/023280
(87) 국제공개번호 WO 2011/043832
국제공개일자 2011년04월14일
(30) 우선권주장
12/575,800 2009년10월08일 미국(US)

(71) 출원인
테라다인 인코퍼레이티드
미국 01864 매사추세츠주 노스 리딩 엔알700-2-3 리버파크 드라이브 600
(72) 발명자
코너, 조지, 더블유.
미국 93010 캘리포니아주 카마릴로 이스트 하이랜드 드라이브 570
(74) 대리인
양영준, 백만기

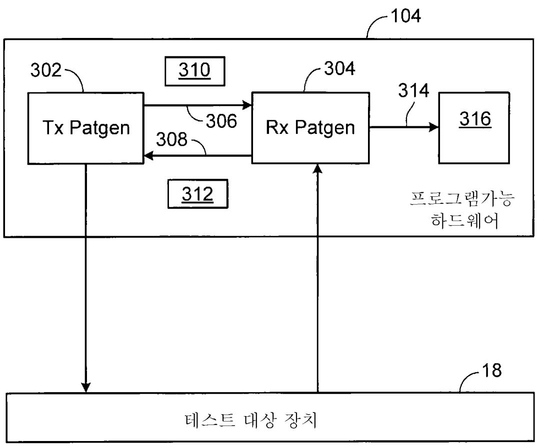
전체 청구항 수 : 총 1 항

(54) 발명의 명칭 **프로그램가능 프로토콜 발생기**

(57) 요 약

반도체 장치 테스터는 테스트 대상 반도체 장치를 테스트하도록 구성된 프로그램가능 하드웨어를 포함한다. 프로그램가능 하드웨어는 테스트 대상 반도체 장치로의/로부터의 데이터의 흐름을 제어하도록 2개 이상의 패턴 발생기로 프로그램된다.

대 표 도 - 도5



(52) CPC특허분류
G01R 31/318314 (2013.01)

명 세 서

청구범위

청구항 1

반도체 장치 테스터로서,

테스트 대상 반도체 장치를 테스트하도록 구성되는 프로그램가능 하드웨어를 포함하는, 반도체 장치 테스터.

발명의 설명

기 술 분 야

[0001] 본 발명은 반도체 장치 테스터에 관한 것이다.

배 경 기 술

[0002] 반도체 장치들(예컨대, 메모리 칩들 및 마이크로프로세서들)에 대한 상업적 수요가 증가함에 따라, 이들 장치의 테스트가 장치 제조업자들에게 중요하게 되었다. 그러한 테스트를 수행하기 위해, 제조된 반도체 장치들의 성능을 특성화하고 검증하도록 자동 테스트 장비(automatic testing equipment, "ATE")와 같은 반도체 장치 테스터들이 사용된다. 많은 ATE는 ATE의 하드웨어가 테스트 대상 반도체 장치(semiconductor device under test)("DUT")에 대해 특별히 구성되어 있는 시스템 레벨 테스트를 수행한다.

[0003] 일부 유형의 테스트에서, ATE는 직류 신호들과 같은 DUT 신호들을 전송한다. 일부 ATE들은 DUT로 전송되는 신호를 발생시키는 데 사용되는 단일 패턴 발생기(pattern generator)를 포함한다. 패턴 발생기에 의해 발생된 신호들에 기초하여, ATE는 신호를 DUT로 전송하고, DUT는 일부 데이터를 다시 ATE로 전송하는 것으로 응답한다.

발명의 내용

[0004] 일반적으로, 일 태양에서, 반도체 장치 테스터는 테스트 대상 반도체 장치를 테스트하도록 구성된 프로그램가능 하드웨어를 포함한다. 프로그램가능 하드웨어는 테스트 대상 반도체 장치로의/로부터의 데이터의 흐름을 제어하도록 2개 이상의 패턴 발생기로 프로그램된다. 2개 이상의 패턴 발생기는 테스트 대상 반도체 장치로 데이터를 전송하고 테스트 대상 반도체 장치로부터 데이터를 수신하도록 프로그램된다. 패턴 발생기들 중 하나의 패턴 발생기는 테스트 대상 반도체 장치로 데이터를 전송하는 전송 프로그램가능 패턴 발생기를 포함한다. 패턴 발생기들 중 다른 패턴 발생기는 테스트 대상 반도체 장치로부터 데이터를 수신하는 수신 프로그램가능 패턴 발생기를 포함한다. 수신 프로그램가능 패턴 발생기는 테스트 대상 반도체 장치로부터 수신된 데이터에 대해 하나 이상의 연산 코드(operation code)를 실행하도록 구성된다. 전송 프로그램가능 패턴 발생기와 수신 프로그램가능 패턴 발생기는 통신하고 있다. 수신 패턴 발생기는 전송 패턴 발생기가 테스트 대상 반도체 장치로 데이터를 전송해야 한다는 것을 나타내는 메시지를 전송 패턴 발생기로 전송한다. 전송 패턴 발생기는 전송될 데이터를 지정하는 하나 이상의 연산 코드를 실행하도록 프로그램된다. 수신 프로그램가능 패턴 발생기는 전송 프로그램가능 패턴 발생기가 테스트 대상 반도체 장치로 데이터를 전송해야 한다는 것을 나타내는 데이터를 포함하는 메시지를 전송 프로그램가능 패턴 발생기로 전송하도록 구성된다.

[0005] 구현예들은 하기의 특징들 중 하나 이상의 특징을 포함할 수 있다. 반도체 장치 테스터는 또한 프로그램가능 하드웨어로부터의/로의 데이터의 흐름을 제어하는 데이터 제어기를 포함한다. 데이터 제어기는 전송 프로그램가능 패턴 발생기와 통신하고 있고, 데이터 제어기는 전송 프로그램가능 패턴 발생기에 의해 실행될 하나 이상의 연산 코드를 지정하는 데이터를 포함하는 명령을 전송 프로그램가능 패턴 발생기로 전송하도록 구성된다. 데이터 제어기는 수신 프로그램가능 패턴 발생기와 통신하고 있고, 데이터 제어기는 수신 프로그램가능 패턴 발생기에 의해 실행될 하나 이상의 연산 코드를 지정하는 데이터를 포함하는 명령을 수신 프로그램가능 패턴 발생기로 전송하도록 구성된다.

[0006] 일반적으로, 일 태양에서, 반도체 장치를 테스트하는 방법은 반도체 장치를 테스트하는 하드웨어를 프로그램하는 단계를 포함한다. 하드웨어는 테스트 대상 반도체 장치로 데이터 비트를 전송하는 연산 코드로 프로그램될 수 있는 전송 패턴 발생기, 및 수신된 데이터 비트에 대해 데이터 연산을 수행하는 연산 코드로 프로그램될 수

있는 수신 패턴 발생기를 포함한다. 방법은 또한 테스트 대상 반도체 장치로 하나 이상의 데이터 비트를 전송하는 단계, 및 테스트 대상 반도체 장치로부터 하나 이상의 데이터 비트를 수신하는 단계를 포함한다. 방법은 또한 전송 패턴 발생기에 의해, 전송될 데이터를 지정하는 하나 이상의 연산 코드를 실행하는 단계를 포함한다. 방법은 또한 테스트 대상 반도체 장치로부터 수신된 하나 이상의 데이터 비트에 대해 하나 이상의 연산 코드를 실행하는 단계를 포함한다.

[0007] 전송 패턴 발생기는 테스트 대상 반도체 장치로 데이터 비트를 전송하기 전에 수신 패턴 발생기로부터 전기 신호를 수신하기 위해 기다린다.

[0008] 일반적으로, 일 태양에서, 반도체 장치는 수신 패턴 발생기 및 전송 패턴 발생기로 프로그램되는 프로그램가능 하드웨어를 포함한다. 전송 패턴 발생기는 하나 이상의 데이터 비트를 테스트 대상 반도체 장치로 전송하도록 구성된다. 수신 패턴 발생기는 전송 패턴 발생기와 통신하도록 구성된다. 수신 패턴 발생기는 하나 이상의 데이터 비트를 테스트 대상 반도체 장치로부터 수신하도록 구성된다. 전송 패턴 발생기는 테스트 대상 반도체 장치로 하나 이상의 제2 데이터 비트를 전송하기 전에 수신 패턴 발생기로부터 신호를 수신하기 위해 기다리도록 구성된다.

[0009] 하나 이상의 실시예의 상세 사항이 첨부 도면 및 이하의 설명에 기술되어 있다. 다른 특징, 목적 및 이점이 이러한 설명 및 도면, 그리고 특허청구범위로부터 명백하게 될 것이다.

도면의 간단한 설명

[0010] <도 1>

도 1은 장치를 테스트하기 위한 ATE의 블록 다이어그램.

<도 2>

도 2는 ATE에서 사용되는 테스터의 블록 다이어그램.

<도 3>

도 3은 반도체 장치 테스터의 개략도.

<도 4, 도 6 및 도 8>

도 4, 도 6 및 도 8은 반도체 장치 테스터에 의해 수행되는 프로세스의 프로세스 흐름도.

<도 5 및 도 7>

도 5 및 도 7은 프로그램가능 하드웨어의 다이어그램.

<도 9 및 도 10>

도 9 및 도 10은 패턴 발생기들에 의해 실행되는 연산 코드의 예들의 도면.

발명을 실시하기 위한 구체적인 내용

[0011] 도 1을 참조하면, 반도체 장치와 같은 테스트 대상 장치(DUT)(18)를 테스트하기 위한 ATE 시스템(10)은 테스터(12)를 포함한다. 테스터(12)를 제어하기 위해, 시스템(10)은 하드와이어(hardwire) 연결(16)을 통해 테스터(12)와 인터페이스(interface)하는 컴퓨터 시스템(14)을 포함한다. 전형적으로, 컴퓨터 시스템(14)은 DUT(18)를 테스트하기 위한 루틴 및 기능의 실행을 시작하기 위해 테스터(12)에 명령을 전송한다. 그러한 테스트 루틴의 실행은 테스트 신호들을 발생하여 DUT(18)로 전송하는 것을 시작하고 DUT로부터 응답들을 수집할 수 있다. 다양한 유형의 DUT들이 시스템(10)에 의해 테스트될 수 있다. 예를 들어, DUT들은 집적 회로(IC) 칩(예컨대, 메모리 칩, 마이크로프로세서, 아날로그-디지털 변환기, 디지털-아날로그 변환기 등)과 같은 반도체 장치들일 수 있다.

[0012] 테스트 신호들을 제공하고 DUT로부터 응답들을 수집하기 위해, 테스터(12)는 DUT(18)의 내부 회로에 대한 인터페이스를 제공하는 하나 이상의 커넥터 핀에 연결된다. 일부 DUT들을 테스트하기 위해, 예컨대 64개 또는 128개만큼 많은 커넥터 핀(또는 그 이상)이 테스터(12)에 인터페이스될 수 있다. 예시의 목적을 위해, 이러한 예에서, 반도체 장치 테스터(12)는 하드와이어 연결을 통해 DUT(18)의 하나의 커넥터 핀에 연결된다. 도체(20)(예컨대, 케이블)가 핀(22)에 연결되고, 테스트 신호들(예컨대, 파라미터 측정 유닛(parametric measurement unit, "PMU") 테스트 신호들, 핀 전자 장치(pin electronics, "PE") 테스트 신호들 등)을 DUT(18)의 내부 회로에 전달하는 데 사용된다. 도체(20)는 또한 반도체 장치 테스터(12)에 의해 제공되는 테스트 신호들에 응답하여 핀(22)에서의 신호들을 감지한다. 예를 들어, 전압 신호 또는 전류 신호가 테스트 신호에 응답하여 핀(22)에서 감지되고 분석을 위해 도체(20)를 통해 테스터(12)로 전송될 수 있다. 그러한 단일 포트 테스트는 또한 DUT(18) 내에 포함된 다른 핀들에서 수행될 수 있다. 예를 들어, 테스터(12)는 테스트 신호들을 다른 핀들에 제공하고 (제공된 신호들을 전달하는) 도체들을 통해 다시 반사되는 연관된 신호들을 수집할 수 있다. 반사된 신호들을 수집함으로써, 핀들의 입력 임피던스가 다른 단일 포트 테스트 양과 함께 특성화될 수 있다. 다른 테스트 시나리오들에서, DUT(18)에 디지털 값을 저장하기 위해 디지털 신호가 도체(20)를 통해 핀(22)으로 전송될 수 있다. 저장되면, 저장된 디지털 값을 검색하여 도체(20)를 통해 테스터(12)로 전송하기 위해 DUT(18)가 액세스될 수 있다. 이어서 적절한 값이 DUT(18)에 저장되었는지를 판정하기 위해 검색된 디지털 값이 확인될 수 있다.

[0013] 1-포트 측정들을 수행하는 것과 함께, 2-포트 테스트가 또한 반도체 장치 테스터(12)에 의해 수행될 수 있다. 예를 들어, 테스트 신호가 도체(20)를 통해 핀(22)에 주입될 수 있고, 응답 신호가 DUT(18)의 하나 이상의 다른 핀으로부터 수집될 수 있다. 이러한 응답 신호는 이득 응답, 위상 응답, 및 기타 처리율 측정 양과 같은 양을 결정하기 위해 반도체 장치 테스터(12)에 제공된다.

[0014] 또한 도 2를 참조하면, DUT(또는 다수의 DUT)의 다수의 커넥터 핀으로부터 테스트 신호들을 전송하고 수집하기 위해, 반도체 장치 테스터(12)는 다수의 핀과 통신할 수 있는 인터페이스 카드(24)를 포함한다. 예를 들어, 인터페이스 카드(24)는 테스트 신호들을, 예컨대 32개, 64개, 또는 128개 핀으로 전송하고 대응하는 응답들을 수집할 수 있다. 핀으로의 각각의 통신 링크는 전형적으로 채널로 지칭되며, 테스트 신호들을 많은 수의 채널에 제공함으로써, 테스트 시간이 감소되는데, 그 이유는 다수의 테스트가 동시에 수행될 수 있기 때문이다. 인터페이스 카드에 많은 채널을 갖는 것과 함께, 테스터(12)에 다수의 인터페이스 카드를 포함함으로써, 전체 채널들의 수가 증가되며, 그로써 테스트 시간을 추가적으로 감소시킨다. 이러한 예에서, 테스터(12)에 다수의 인터페이스 카드가 있을 수 있다는 것을 설명하기 위해 2개의 부가적인 인터페이스 카드(26, 28)가 도시되어 있다.

[0015] 각각의 인터페이스 카드는 특정 테스트 기능을 수행하기 위한 전용 집적 회로(IC) 칩(예컨대, ASIC(application specific integrated circuit))을 포함한다. 예를 들어, 인터페이스 카드(24)는 파라미터 측정 유닛(PMU) 테스트 및 핀 전자 장치(PE) 테스트를 수행하기 위한 IC 칩(30)을 포함한다. IC 칩(30)은 PMU 테스트를 수행하기 위한 회로를 포함하는 PMU 스테이지(32) 및 PE 테스트를 수행하기 위한 회로를 포함하는 PE 스테이지(34)를 갖는다. 또한, 인터페이스 카드들(26, 28)은, 각각 PMU 및 PE 회로를 포함하는 IC 칩들(36, 38)을 포함한다. 전형적으로, PMU 테스트는 입력 및 출력 임피던스, 전류 누설, 및 기타 유형의 DC 성능 특성과 같은 양을 결정하기 위해 DC 전압 또는 전류 신호를 DUT에 제공하는 것을 포함한다. PE 테스트는 AC 테스트 신호들 또는 파형들을 DUT(예컨대, DUT(18))로 전송하는 것 및 DUT의 성능을 추가적으로 특성화하기 위해 응답들을 수집하는 것을 포함한다. 예를 들어, IC 칩(30)은 DUT에 저장하기 위해 이진 값들의 벡터를 나타내는 AC 테스트 신호들을 (DUT로) 전송할 수 있다. 이들 이진 값이 저장되면, 올바른 이진 값들이 저장되었는지를 판정하기 위해 DUT가 테스터(12)에 의해 액세스될 수 있다. 디지털 신호들이 전형적으로 급격한 전압 천이를 포함하기 때문에, IC 칩(30) 상의 PE 스테이지(34)에서의 회로는 PMU 스테이지(32)에서의 회로와 비교하여 상대적으로 높은 속도로 동작한다.

[0016] DC 및 AC 테스트 신호들 둘 모두를 인터페이스 카드(24)로부터 DUT(18)로 전달하기 위해, 전도성 트레이스(40)가 IC 칩(30)을 인터페이스 기판 커넥터(42)에 연결시켜, 신호들이 인터페이스 기판(24)으로/으로부터 전달되게 한다. 인터페이스 기판 커넥터(42)는 또한 인터페이스 커넥터(46)에 연결된 도체(44)에 연결되어, 신호들이 테스터(12)로/로부터 전달되게 한다. 이러한 예에서, 도체(20)는 테스터(12)와 DUT(18)의 핀(22) 사이에서의 양방향 신호 전달을 위해 인터페이스 커넥터(46)에 연결된다. 일부 구성들에서, 테스터(12)로부터 DUT로 하나 이상의 도체를 연결시키기 위해 인터페이스 장치가 사용될 수 있다. 예를 들어, DUT(예컨대, DUT(18))는 각각의 DUT 핀에의 액세스를 제공하기 위한 장치 인터페이스 기판(DIB) 상에 장착될 수 있다. 그러한 구성에서, 테스트 신호들을 DUT의 해당 핀(들)(예컨대, 핀(22))에 주기 위해 도체(20)가 DIB에 연결될 수 있다.

[0017] 이러한 예에서, 신호들을 전달하고 수집하기 위해 전도성 트레이스(40) 및 도체(44)만이, 각각 IC 칩(30) 및 인터페이스 기판(24)을 연결시킨다. 그러나, IC 칩(30)은 (IC 칩들(36, 38)과 함께) 전형적으로 (DIB를 통해) DUT로부터의 신호들을 제공하고 수집하기 위해 다수의 전도성 트레이스 및 대응하는 도체에 각각 연결되는 다수의 핀(예컨대, 8개, 16개 등)을 갖는다. 또한, 일부 구성들에서, 테스터(12)는 인터페이스 카드들(24, 26, 28)에 의해 제공되는 채널들을 하나 또는 다수의 테스트 대상 장치로 인터페이스하기 위해 2개 이상의 DIB에 연

결될 수 있다.

[0018] 테스터(12)는 테스터(12)가 DUT(18)에 응답하도록 프로그램될 수 있는 프로그램가능 하드웨어(104), 예컨대 "FPGA"(field-programmable gate array) 반도체 장치를 포함한다. 프로그램가능 하드웨어(104)는 DUT(18)에 연결된 핀 전자 장치 칩에 연결된다.

[0019] 도 3을 참조하면, 테스터(12)는 하드웨어 전송 기술을 포함하는 물리 계층(106)을 포함한다. 예를 들어, 물리 계층(106)은 데이터가 어떻게 DUT(18)로 전송되고 그로부터 수신되는지를 제어한다. 일부 예들에서, 물리 계층(106)은 DUT(18)로부터 테스터(12)로의 비트별 전달(bit-by-bit delivery)을 지정한다. 물리 계층은 프로토콜에 고유하지 않고, DUT(18)의 포트들에서 사용되는 다양한 유형의 프로토콜 및 인터페이스, 예컨대 "SATA"(Serial Advanced Technology Attachment) 및 "HDMI"(High-Definition Multimedia Interface)와 관련하여 사용될 수 있다. 테스터(12)에서, 타이밍 및 클록 복원의 상세 사항들은 물리 계층(106)에서 처리되고, 프로그램가능 하드웨어(104)는 테스터(12)와 DUT(18) 사이에서 전달되는 정보의 레벨(즉, 데이터 비트)을 제어한다.

[0020] 프로그램가능 하드웨어(104)는, 예를 들어 테스터(12)의 사용자에 의해 프로그램될 수 있고, 따라서 테스터(12)는 테스트 동안 DUT의 포트들에서 다양한 프로토콜을 사용한다. 프로그램가능 하드웨어(104)는 적절한 프로토콜로 프로그램될 수 있고 그 프로토콜을 사용하여 DUT(18)와 통신한다. 프로그램가능 하드웨어(104)를 통해, 테스터(12)는 "프로토콜을 인식"하는데, 그 이유는 테스터(12)가 DUT(18) 상의 포트로, 그 포트에 대한 프로토콜을 사용하여 데이터를 전송하기 때문이다. 또한, 테스터(12)는 일련의 DUT 문제점을 디버그하거나, 테스터(12)가 내장된 프로토콜 지원을 갖지 않는 DUT(18)를 테스트할 수 있다.

[0021] 테스터(12)가 DUT(18)에 응답하도록 프로그램될 수 있기 때문에, 테스터(12)는 DUT(18)와 연결("핸드쉐이크(handshake)")을 설정할 수 있다. 핸드쉐이크는 DUT(18)에서 사용되는 연결, 속도, 프로토콜 상세, 및 알고리즘의 검증을 포함한다. 일부 예들에서, 테스터(12)가 DUT(18)를 테스트하기 전에 DUT(18) 상의 직렬 포트들을 초기 설정(bring up)하는 데 핸드쉐이크가 필요하다.

[0022] 프로그램가능 하드웨어(104)는 DUT(18)로 전송할 데이터 비트를 지정하는 다양한 연산 코드("opcode")로 프로그램될 수 있다. 프로그램가능 하드웨어(104)는 또한 DUT(18)로부터 데이터를 수신한 후에 프로그램가능 하드웨어(104)에 의해 수행될 동작을 지정하는 연산 코드로 프로그램될 수 있다. 도 4를 참조하면, 프로그램가능 하드웨어(104)는 DUT(18)의 포트들과의 핸드쉐이크를 수행하거나 그 포트들을 초기 설정할 시에 다양한 동작(200)을 수행한다. 프로그램가능 하드웨어(104)는 데이터 비트를 DUT(18)로 전송하는 연산 코드를 실행한다(202). 프로그램가능 하드웨어(104)는 DUT(18)로부터 다시 데이터 비트를 수신한다(204). 프로그램가능 하드웨어(104)는 수신된 데이터를 검증하거나 그에 대해 다른 기능을 수행하는 추가의 연산 코드를 실행한다(206). 이들 기능의 예들은 DUT(18)를 테스트하기 위한 알고리즘을 테스트하는 것을 포함한다. 프로그램가능 하드웨어는 또한 추가의 데이터 비트를 DUT(18)로 전송하도록 부가적인 연산 코드를 실행한다(208). 이들 동작(202, 204, 206)은 DUT(18)가 테스트될 때까지 계속될 수 있다.

[0023] 도 5를 참조하면, 프로그램가능 하드웨어(104)는 2개의 패턴 발생기, 즉 전송 패턴 발생기(302)("Tx Patgen") 및 수신 패턴 발생기(304)("Rx Patgen")를 포함할 수 있다. 프로그램들의 라이브러리가 프로그램가능 하드웨어(104)에 제공된다. 프로그램들의 라이브러리는 Tx Patgen(302) 및 Rx Patgen(304)을 정의하는 코드를 포함한다. 프로그램가능 하드웨어(104)가 FPGA 카드인 경우, 코드는 FPGA 코드를 포함한다.

[0024] 패턴 발생기들(302, 304)은 테스터(12)의 사용자에 의해 프로그램가능 하드웨어(104)에 프로그램되는 일련의 연산 코드를 저장하고 실행한다. 예를 들어, Tx Patgen(302)은 데이터 비트를 DUT(18)로 전송하기 위해 Tx Patgen(302)에 의해 실행되는 일련의 연산 코드를 포함한다. Rx Patgen(304)은 또한 DUT(18)로부터 데이터 비트를 수신하기 위해 Rx Patgen(304)에 의해 실행되는 일련의 연산 코드를 포함한다.

[0025] 2개의 패턴 발생기(302, 304)의 사용을 통해, DUT(18)로의/로부터의 데이터의 흐름이 제어가능하다. 일례에서, Tx Patgen(302)은 데이터를 DUT(18)로 전송하고, Rx Patgen(304)은 DUT(18)로부터 데이터를 수신한다. Tx Patgen(302) 및 Rx Patgen(304)이 서로 데이터 통신을 하기 때문에, Tx Patgen(302)은 Rx Patgen(304)이 수신한 데이터를 Rx Patgen(304)이 검증할 때까지 기다리고 추가의 데이터를 전송하지 않도록 프로그램될 수 있다.

[0026] 테스터(12)의 사용자는 테스터(12) 상에 제공된 컴퓨터 인터페이스(52)(도 2)를 통해 Tx Patgen(302) 또는 Rx Patgen(304)에 의해 실행될 연산 코드의 시퀀스(및 연산 코드와 연관된 데이터)를 지정한다. 예를 들어, 사용자는 "WAIT" 및 "Set_TX"를 포함하도록 Tx Patgen(302)에 의해 실행되는 연산 코드의 시퀀스를 정의할 수 있다.

유사하게, 사용자는 "RPT" 및 "WAIT_RX"를 포함하도록 Rx Patgen(304)에 의해 실행되는 연산 코드의 시퀀스를 정의할 수 있다.

[0027] 일부 예들에서, Tx Patgen(302) 및 Rx Patgen(304)은 케이블들, 전기 와이어들 또는 커패시터들과 같은 통신 링크들(306, 308)를 통해 서로 통신한다. Tx Patgen(302)은 DUT(18)로의 데이터의 전송을 제어한다. Rx Patgen(304)은 DUT(18)로부터의 데이터의 수신을 제어한다. Tx Patgen(302)은 Tx 플래그(Flag)(310)로 지칭되는 데이터를 통신 링크(304)를 통해 Rx Patgen(304)으로 전송한다. Rx Patgen(304)은 Rx 플래그(312)로 지칭되는 데이터를 통신 링크(308)를 통해 Tx Patgen(302)으로 전송한다.

[0028] Tx 플래그(310)는 Rx Patgen(304)이 DUT(18)로부터 데이터를 수신하기 위해 기다리기 시작해야 한다는 것을 Rx Patgen(304)에 알려주는 데이터를 포함한다. 일부 예들에서, Tx Patgen(302)은 Tx Patgen(302)이 데이터를 DUT(18)로 전송하는 것과 동시에 Tx 플래그(310)를 전송한다. 다른 예들에서, Tx Patgen(302)은 데이터를 DUT(18)로 전송한 후에 Tx 플래그(310)를 전송한다.

[0029] 나중에 Rx Patgen(304)이 수신할 것으로 예상되는 모든 데이터를 수신하고 그 데이터의 정확성 또는 무결성을 검증할 수 있을 때, Rx Patgen(304)은 Rx 플래그(312)를 Tx Patgen(302)으로 전송한다. Rx Patgen(304)이 예상하고 있는 데이터를 수신한 후에, Rx Patgen은 Rx 플래그(308)를 Tx Patgen(302)으로 전송한다. Rx 플래그(312)는 Tx Patgen(302)이 그 다음 연산 코드를 실행하기 시작하고 추가의 데이터를 DUT(18)로 계속하여 전송해야 한다는 것을 Tx Patgen(302)에 신호한다.

[0030] 일부 예들에서, Rx Patgen(304)은 Rx Patgen(304)이 어떻게 처리할지를 모르는 데이터를 DUT(18)로부터 수신한다. 일례에서, Rx Patgen(304)은 DUT(18) 상의 아날로그-디지털 변환기로부터 출력되는 데이터를 판독할 수 없다. 이러한 경우에, Rx Patgen(304)은 데이터를 통신 링크(314)를 통해 캡처 메모리 장치(316)로 전송한다. 캡처 메모리 장치(316)는 데이터를 저장하고, 따라서 반도체 장치 테스터(100)는 나중에 그 데이터에 액세스할 수 있다. 이와 관련하여, 캡처 메모리 장치(316)는 Rx Patgen(304)이 직접 처리할 수 없는 데이터를 저장하는 데이터 저장소로서 기능한다.

[0031] 일부 예들에서, Tx Patgen(302) 및 Rx Patgen(304)에 들어오는 데이터는 48 비트 폭의 600 메가헤르쯔("㎒") "DRAM"(dynamic random access memory)으로부터 온 것이다. 이러한 예에서, 600 ㎒의 클록 주파수는 Tx Patgen(302)과 Rx Patgen(304) 사이에 분할된다. 그러나, 테스터(12)의 DRAM 내에 저장된 데이터를 리프레시하는 얼마간의 비효율이 있기 때문에, Tx Patgen(302)에 의해 실행되는 연산 코드는 266 ㎒로 48 비트의 데이터를 할당받는다. 유사하게, Rx Patgen(304)에 의해 실행되는 연산 코드는 266 ㎒로 48 비트의 데이터를 할당받는다. 48 비트의 데이터 중에서, 40 비트는 심볼 데이터이고, 8 비트는 제어 데이터로서 사용된다. 이하의 표 1은 Tx Patgen(302)에 프로그램될 수 있는 전송 연산 코드의 예를 제공한다.

표 1

명령(4 비트)	피연산자 (4 비트)	심볼 데이터 (40 비트)			
NEXT	xxxx	10b Sym0	10b Sym1	10b Sym2	10b Sym3
RPT	xxxx	10b Sym0	10b Sym1	10b Sym2	10b Sym3
BLOOP	xxxx	10b Sym0	10b Sym1	10b Sym2	10b Sym3
ELOOP	xxxx	10b Sym0	10b Sym1	10b Sym2	10b Sym3
WAIT	xxxx	10b Sym0	10b Sym1	10b Sym2	10b Sym3
WAIT_SPD	L,M,H	10b Sym0	10b Sym1	10b Sym2	10b Sym3
SET_TX	xxxx	10b Sym0	10b Sym1	10b Sym2	10b Sym3
SET_FL	xxxx	10b Sym0	10b Sym1	10b Sym2	10b Sym3
SET_ISL	xxxx	10b Sym0	10b Sym1	10b Sym2	10b Sym3
SET_MODE	P,L,D	10b Sym0	10b Sym1	10b Sym2	10b Sym3
STOP		10b Sym0	10b Sym1	10b Sym2	10b Sym3

[0032]

[0033] 상기 표 1에 나타낸 바와 같이, 연산 코드의 처음 4 비트는 Tx Patgen(302)에 의해 수행될 명령, 예컨대 NEXT 또는 RPT를 지정한다. 그 다음 4 비트는 다양한 제어 비트를 설정하는 데 사용되는 피연산자를 지정한다. 나머지 40 비트는 심볼 데이터로서 지정된다. 표 1에 제공된 예에서, 40 비트는 4개의 심볼로서 전송되고, 각각의 심볼은 10 비트를 포함한다.

[0034] NEXT 명령은 40 비트의 심볼 데이터를 취하여 이를 Tx Patgen(302) 상의 포트를 통해 송출한다. RPT 명령은 또한 40 비트의 심볼 데이터를 취하여 Tx Patgen(302) 상의 포트를 통해 송출한다. 또한, RPT 명령은 소정의 수의 사이클 동안 Tx Patgen(302) 상의 포트를 통해 데이터를 계속하여 송출하고, 여기서 사이클의 수는 그 다음 48 비트 명령에 지정되어 있다. BLOOP 명령은, 피연산자의 주소가 저장되고 따라서 ELOOP 명령이 이 주소로 다시 분기할 수 있다는 것을 제외하고는, NEXT 명령처럼 기능한다.

[0035] ELOOP 명령은 40 비트의 데이터를 Tx Patgen(302) 상의 포트를 통해 송출한다. ELOOP 명령은 또한 소정의 횟수 루프를 반복하는 루프 기능을 포함한다. ELOOP 명령에서, 루프가 다음과 같은 3개의 이벤트를 통해 영구적으로 종료될 수 있다: 일치 조건, 호스트 플래그 또는 ISL 명령. 그를 위해, 48b 데이터 필드 내의 나머지 비트가 이를 위해 다음과 같이 사용될 수 있다:

[0036] 비트 47 = RX 플래그를 기다림

[0037] 비트 46 = 호스트 플래그를 기다림

[0038] 비트 45 = ISL 명령을 기다림

[0039] 비트 44:10 = 예약됨

[0040] 설정된 비트 없음 = 정상 루프

[0041] WAIT 명령은 핀 데이터를 계속하여 제공하는 동안 이벤트가 발생할 때까지 기다린다. WAIT 명령은 또한 소정의 이벤트가 발생하기를 기다리고, 이벤트가 발생할 때, 실행이 그 다음 연산 코드로 전진한다.

[0042] WAIT_SPD 명령은 낮음("L"), 중간("M") 그리고 높음("H") 사이에서 패턴 속도를 변경하는 데 사용된다. 속도 변화가 안정될 수 있게 하도록 대기 카운트가 그 다음 연산 코드에 설정된다. 속도는 피연산자 비트에 의해 선택된다. SET_TX 명령은 Tx 플래그(310)를 Rx Patgen(304)으로 전송한다. 이것은 Tx Patgen(302)이 플래그 연산 코드에 도달했다는 것을 Rx Patgen(304)에 알려주는 데 사용된다. SET_FL 명령은 호스트 플래그를 설정한다. SET_ISL 명령은 ISL 완료 비트를 설정한다. SET_MODE 명령은 모드 비트를 다음과 같이 설정한다. 일부 예들에서, SET_MODE 명령은 Tx Patgen(302)을 "PRBS"(pseudo-random binary sequence) 데이터 패턴을 발생시켜 전송하도록 설정한다. 다른 예들에서, SET_MODE 명령은 루프백이 시작되어야 한다는 것을 지정한다. STOP 명령은 전송 데이터 패턴을 중단하지만 40 비트를 Tx Patgen(302) 상의 포트로 계속 전송한다.

[0043] Rx Patgen(304)에 프로그램될 수 있는 연산 코드는, 이하의 표 2에 예시된 바와 같이, Tx Patgen(302)에 대한 연산 코드와 유사할 수 있다.

표 2

명령(4 비트)	Care 필드 (4 비트)	심볼 데이터 (40 비트)			
NEXT	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
RPT	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
BLOOP	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
ELOOP	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
WAIT	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
SET_RX	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
SET_FL	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
SET_ISL	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
SET_CAP_ON	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
SET_CAP_OFF	Care 4b	10b Sym0	10b Sym1	10b Sym2	10b Sym3
RESET	A,M,V	Match A+	Match B+	Match A-	Match B-
STOP	xxxx				

[0044]

[0045] 피연산자 필드를 갖는 대신에, Rx Patgen(304)에 대한 연산 코드는 CARE 필드를 가질 수 있다. CARE 필드는 Rx Patgen(304)이 수신된 데이터에 대해 임의의 부가의 동작들을 수행함이 없이 데이터를 수신하는지를 지정한다. 일부 예들에서, DUT(18)로부터 수신되는 데이터는 예상된 값과 비교되지 않는다. 이들 예에서, 0이 CARE 필드 내의 각자의 심볼 위치에 배치된다. 다른 예들에서, CARE 필드는 캡처된 데이터가 일부 예상된 데이터와 일치해야만 한다는 것을 지정한다. 일례에서, CARE 비트는 예상된 데이터와 캡처된 데이터 사이에 일치가 있어야만 한다는 것을 지정하도록 설정된다. 캡처된 데이터가 예상된 데이터와 일치하지 않는 경우에, 실패 조건이 발생하고, 디코드 오류 또는 불일치 오류와 같은 오류가 발생한다. 이러한 예에서, Rx Patgen(304)은 실패 조건을 잘못된 심볼, 디코드 오류 또는 불일치 오류로서 기록한다.

[0046] Rx Patgen(304)에 의해 사용되는 연산 코드에 포함된 명령어는, ELOOP, WAIT, SET_CAP_ON, SEP_CAP_OFF 및 RESET 명령을 제외하고는, Tx Patgen(302)에 의해 사용되는 연산 코드에 포함된 명령어와 동일할 수 있다.

[0047] ELOOP 명령 및 WAIT 명령의 경우, 정렬 및 일치 조건을 기다리기 위해 부가의 선택 비트가 추가되어 있다. 비트 선택은 다음과 같다:

[0048] 비트 47 = TX 플래그를 기다림

[0049] 비트 46 = 호스트 플래그를 기다림

[0050] 비트 45 = ISL 명령을 기다림

[0051] 비트 44 = 정렬을 기다림

[0052] 비트 43 = 정렬 이어서 Match A를 기다림

[0053] 비트 42 = 정렬 이어서 Match B를 기다림

[0054] 비트 41 = 정렬 이어서 Match A+B를 기다림

[0055] 비트 40 = 일치 이전에 캡처

[0056] 비트 39 = 일치 이후에 캡처

[0057] 비트 40:16 = 예약됨

[0058] 비트 15:0 = 대기 또는 루프 카운트(루프 카운트 전용 10b)

[0059] 캡처 비트는 WAIT 명령이 데이터 패킷의 시작을 지정하는 데이터를 기다리게 한다. WAIT 명령어의 실행을

통해, Rx Patgen(304)은 이어서 데이터 패킷의 끝에서 일치가 발생할 때까지 기다리고 데이터를 수신한다.

[0060] SET_CAP_ON 및 SET_CAP_OFF 명령은 캡처된 데이터가 캡처 메모리(316) 내에 저장되어 있는지를 판정하는 모드 비트를 설정하고 소거한다. SET_CAP_ON 명령은 그 다음 사이클에서 데이터를 캡처(즉, 연산 코드의 실행)하기 시작하는 반면, SET_CAP_OFF 명령은 현재의 사이클에서 데이터를 캡처하는 것을 중단한다.

[0061] RESET 명령은 정렬 또는 일치 조건을 리셋하거나 일치 값을 변경하는 데 사용되고, 여기서 A=정렬이고, M=일치이며 V=값이다.

[0062] 도 9를 참조하면, Rx Patgen(304)은 대역외(out of band, "OOB") 시퀀스(350), 유휴 시퀀스(idle sequence)(352), 수신 패킷 시퀀스(354) 및 전송 패킷 시퀀스(356)와 같은 다양한 연산 코드 패턴으로 프로그램될 수 있다. 예를 들어, Rx Patgen(304)의 전송 패킷 시퀀스(356)에서, Rx Patgen(304)은 DUT(18)로부터 데이터를 수신하기 위해 기다리는데, 그 이유는 Tx Patgen(302)의 전송 패킷 시퀀스가 데이터를 DUT(18)로 전송하기 때문이다. 도 10을 참조하면, Tx Patgen(302)은 또한 대역외("OOB") 시퀀스(360), 유휴 시퀀스(362), 수신 패킷 시퀀스(364) 및 전송 패킷 시퀀스(366)와 같은 다양한 연산 코드 패턴으로 프로그램된다.

[0063] 도 6을 참조하면, Tx Patgen(302) 및 Rx Patgen(304)은 DUT(18)을 테스트하는 데 다양한 동작(400)을 수행한다. Tx Patgen(302)은 심볼 데이터가 DUT(18)로 전송되게 하는 NEXT 명령어를 비롯한 연산 코드와 같은 연산 코드를 실행한다(402). Tx Patgen(302)은 또한 Tx 플래그(310)를 Rx Patgen(304)으로 전송한다(404). Tx 플래그(310)는 Rx Patgen(304)이 DUT(18)로부터 전송되는 데이터가 있는지 리스닝(listening)하기 시작해야 한다는 것을 Rx Patgen(304)에 통지한다. DUT(18)는 데이터를 Rx Patgen(304)으로 전송한다(406). Rx Patgen(304)은 특정의 데이터를 수신했는지를 검증하기 위해 수신된 데이터에 대해 그의 연산 코드를 실행한다(408). Rx Patgen(304)은 Rx 플래그(312)를 Tx Patgen(302)으로 전송한다(410).

[0064] 도 7을 참조하면, 일부 실시예들에서, 전용 제어기(502)는 프로그램가능 하드웨어(104)의 일부이고, Tx Patgen(302) 및 Rx Patgen(304)에 의한 연산 코드의 실행을 제어하는 데 사용된다. DUT(18)가 수많은 유형의 데이터를 다시 전송하고 Rx Patgen(304)이 다양한 형태의 수신된 데이터를 어떻게 처리할지를 모를 때 전용 제어기(502)가 사용된다. 이 경우에, Rx Patgen(304)은 수신된 데이터를 통신 링크(504)를 통해 전용 제어기(502)로 전송한다. 다른 경우에, Rx Patgen(304)은 수신된 데이터를 캡처 메모리 장치(316)로 전송하고, 전용 제어기(502)는 캡처 메모리 장치(316)에 액세스한다. Rx Patgen(304)은 Rx Patgen(304)이 데이터를 수신했다는 것을 전용 제어기(502)에 경고하는 메시지를 전용 제어기(502)로 전송한다. 전용 제어기(502)는 이어서 캡처 메모리 장치(316)로부터 수신된 데이터에 액세스한다. 전용 제어기(502)는 Tx Patgen(302)에 의해 실행되어야 하는 연산 코드를 결정하기 위해 수신된 데이터를 분석한다. 전용 제어기(502)는 이어서 그 다음에 실행할 연산 코드에 관해 Tx Patgen(302)에 알려준다.

[0065] 일부 예들에서, Tx Patgen(302) 및 Rx Patgen(304)은 Tx 플래그(310) 및 Rx 플래그(312)를 통해 서로 직접 통신하지 않는다. 그 대신에, 전용 제어기(502)는 Rx Patgen(304)으로부터 수신된 데이터를 처리하고 데이터를 다시 Rx Patgen(304)으로 전송하여, 실행해야만 하는 임의의 부가적인 연산 코드, 예컨대 추가의 데이터를 수신하기 위해 기다리는 연산 코드 또는 수신된 데이터와 Rx Patgen(304)이 수신할 것으로 예상된 데이터 사이의 비교 일치를 수행하는 연산 코드를 Rx Patgen(304)에 알려준다. 전용 제어기(502)가 Rx Patgen(304)으로부터 수신된 데이터를 처리한 후에, 전용 제어기(502)는 그 다음에 어느 연산 코드를 실행할지를 Tx Patgen(302)에 알려주는 명령을 Tx Patgen(302)으로 전송한다.

[0066] 일부 예들에서, Tx Patgen(302) 및 Rx Patgen(304)은 실행할 해당 연산 코드를 결정할 시에 통신 링크들(306, 312) 및 Tx 및 Rx 플래그들(310, 312)과 관련하여 전용 제어기(502)를 사용한다. 일례에서, Rx Patgen(304)은 Rx Patgen(304)이 직접 처리할 수 있는 일부 데이터를 DUT(18)로부터 수신하고 실행할 해당 연산 코드를 결정한다. 그러나, Rx Patgen(304)은 또한 처리할 수 없는 다른 데이터를 수신할 수 있다. 이러한 경우에, Rx Patgen(304)은 데이터를 전용 제어기(502)로 전달한다. 전용 제어기(502)는 데이터를 처리하고, 그 다음에 어느 연산 코드를 실행할지에 관한 명령어를 통신 링크(504)를 통해 Rx Patgen(304)으로 전송한다.

[0067] 도 8을 참조하면, Rx Patgen(304), Tx Patgen(302) 및 전용 제어기(502)는 DUT(18)로 데이터를 전송하고 그로부터 데이터를 수신하는 프로세스에서 다양한 동작(600)을 수행한다. Rx Patgen(304)은 DUT(18)로부터의 데이터를 수신한다(602). Rx Patgen(302)은 수신된 데이터를 전용 제어기(502)로 전송한다(604). 전용 제어기(502)는 Rx Patgen(304)으로부터 수신된 데이터를 처리한다(606). 전용 제어기(502)는 Rx Patgen(304)이 DUT(18)로부터 추가의 데이터를 수신하기 위해 기다릴 필요가 있는지를 결정한다(608). 전용 제어기(502)가 Rx

Patgen(304)이 추가의 데이터를 수신할 필요가 있거나 부가의 연산 코드를 실행할 필요가 있는 것으로 결정하는 경우, 전용 제어기(502)는 실행할 연산 코드를 지정하는 부가의 명령어를 Rx Patgen(304)으로 전송한다(610). Rx Patgen(304)이 DUT(18)로부터 데이터를 수신하고 수신된 데이터를 전용 제어기(502)로 전달하는 사이클(614)은 전용 제어기(502)가 Rx Patgen(304)이 필요로 하는 모든 데이터를 수신한 것으로 판정할 때까지 계속된다. 전용 제어기(502)가 Rx Patgen(304)이 필요로 하는 모든 데이터를 수신한 것으로 판정하는 경우, 전용 제어기(502)는 명령을 Tx Patgen(302)으로 전달한다(612). Tx Patgen(302)으로 전달되는 명령은 Tx Patgen(302)이 DUT(18)로 전송해야 하는 연산 코드를 지정할 수 있다. Tx Patgen(302)은 연산 코드를 DUT(18)로 전송할 수 있다.

[0068] 프로그램가능 하드웨어(104)는 표준 회로 요소들을 포함할 수 있다. 일례에서, Tx Patgen(302)으로부터 DUT(18)로 전송되는 심볼 데이터는 병렬-직렬 변환기(506) 또는 대역외 검출 장치(508)를 통해 전송된다. 심볼 데이터는 이어서 논리 게이트(510)를 통해 전달되고, 그의 출력이 DUT(18)로 전송된다. 수신단에서, DUT(18)는 데이터를 다시 테스터(12) 및 프로그램가능 하드웨어(104)로 전달한다. 수신된 데이터는 테스터(12)의 일부인 논리 게이트(512)를 통해 전달된다. 일부 예들에서, 수신된 데이터는 또한 수신된 데이터가 이전에 설정된 통신 채널을 벗어나 있는지를 판정하기 위해 대역외 검출 장치(514)를 통해 전달된다. 수신된 데이터는 이어서 직렬-병렬 변환기(514)로 전달되고, Rx Patgen(304)으로 전송된다. 10b 정렬은 비트 스트림을 조사하고, 10b 정렬이 찾고 있는 정렬 문자에 기초하여, 10b 심볼 경계를 어디에 배치할지를 결정한다.

[0069] 일부 예들에서, 수신된 데이터는 추적기(518)로 전송되고 지연 장치(520)를 통해 피드된다. 추적기(518)는 데이터 천이를 검사하고, 클록이 데이터 아이(data eye)의 중심에 있도록 클록 비교기 스트로브(우리의 클록)의 위상을 이동시킨다. 일부 예들에서, 마스터 발진기(master oscillator, "MOSC")는 온보드 클록으로서 기능한다.

[0070] 일부 예들에서, Tx Patgen(302)은 Tx Patgen(302)과 Rx Patgen(304) 사이에서 이벤트를 동기화시키기 위해 "HRAM"(hierarchical random access memory)(522) 및 "ISL"(instrument sync link)(524)에 연결된다. Rx Patgen(304)은 또한 HRAM(526) 및 ISL(528)에 연결된다. 일부 예들에서, 전용 제어기(502)는 서버 또는 기타 컴퓨팅 장치와 같은 호스트 장치(530)와 인터페이스된다.

[0071] 다수의 실시예가 기술되었다. 그럼에도 불구하고, 다양한 수정이 이루어질 수 있다는 것을 이해할 것이다. 따라서, 다른 실시예들이 이하의 특허청구범위의 범주 내에 속한다.

부호의 설명

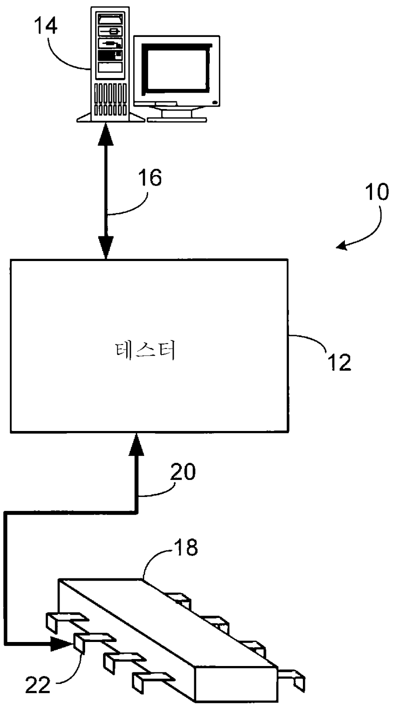
[0072] 104: 프로그램가능 하드웨어

302: 전송 패턴 발생기

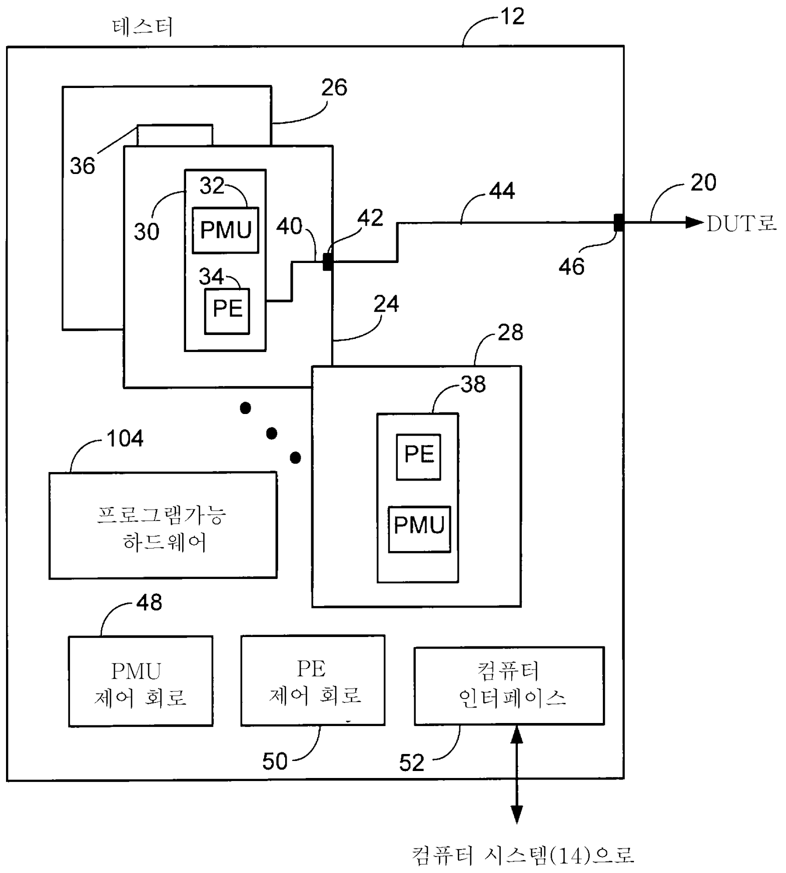
304: 수신 패턴 발생기

도면

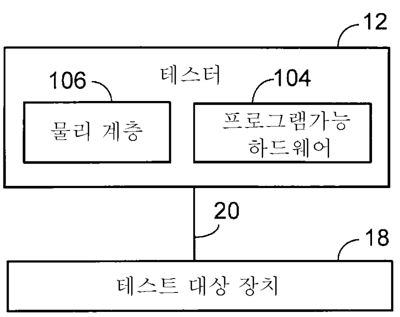
도면1



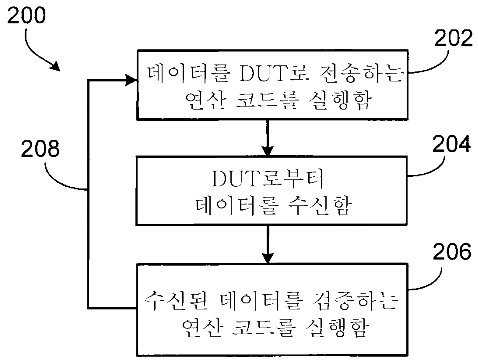
도면2


테스터
12
26
36
32
30
PMU
40
42
44
20
DUT로
46
34
PE
24
28
38
PE
PMU
104
프로그램가능
하드웨어
48
PMU
제어 회로
PE
제어 회로
컴퓨터
인터페이스
50
52
컴퓨터 시스템(14)으로

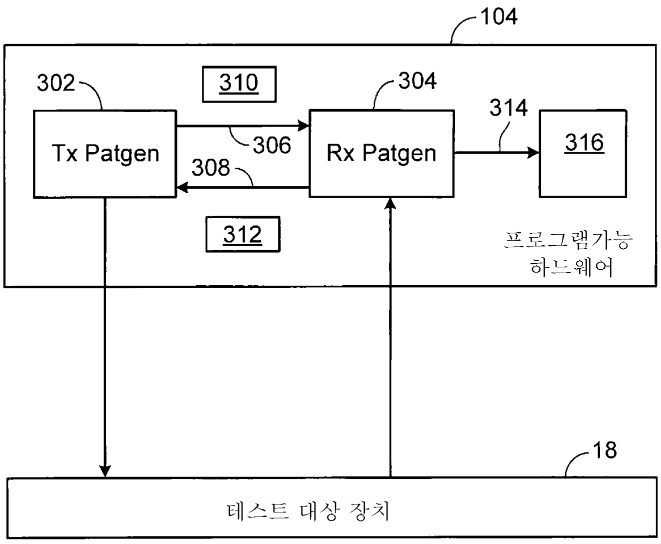
도면3


12
106
테스터
104
물리 계층
프로그램가능
하드웨어
20
18
테스트 대상 장치

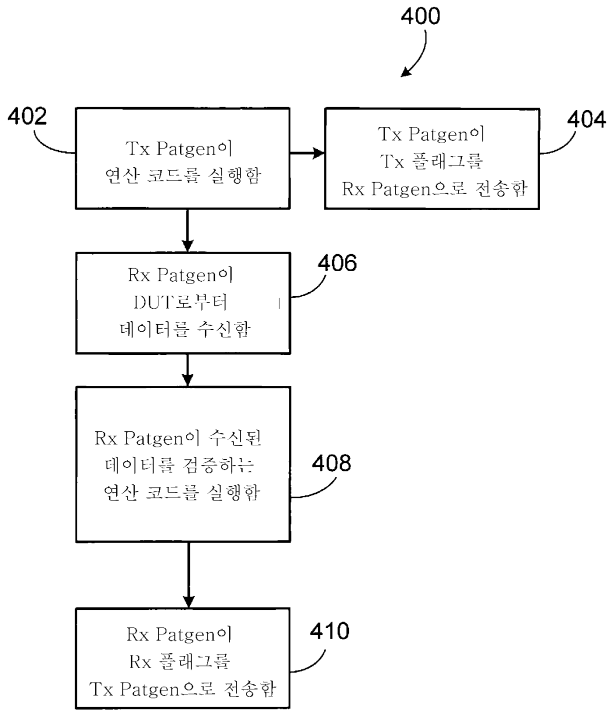
도면4


200
202
데이터를 DUT로 전송하는 연산 코드를 실행함
204
DUT로부터 데이터를 수신함
206
수신된 데이터를 검증하는 연산 코드를 실행함
208

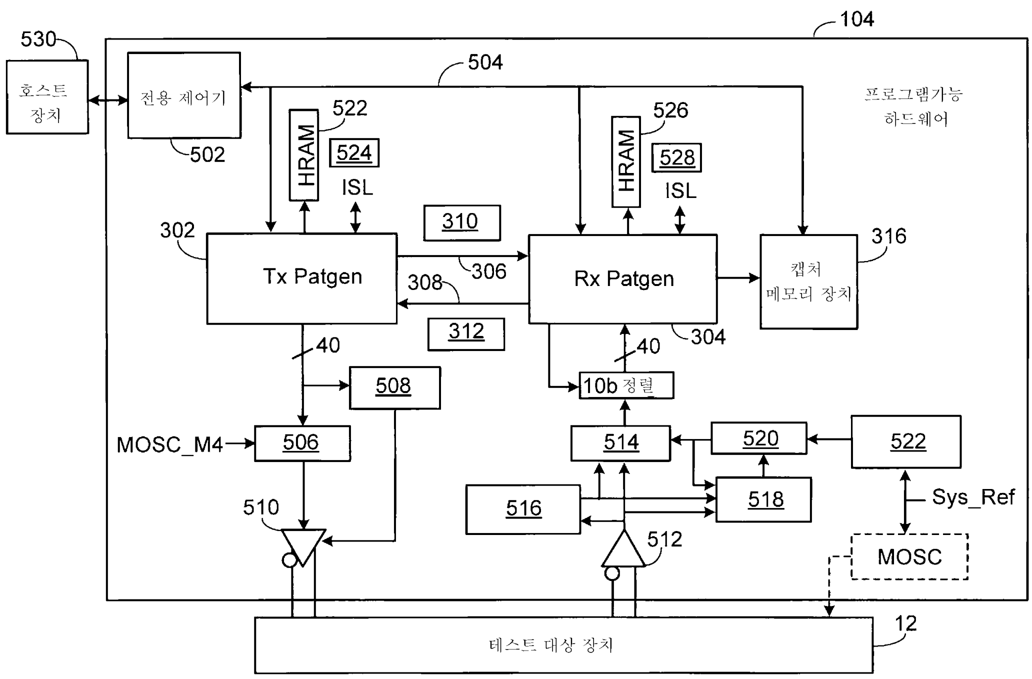
도면5


104
302
310
304
314
316
Tx Patgen
306
308
Rx Patgen
312
프로그램가능 하드웨어
18
테스트 대상 장치

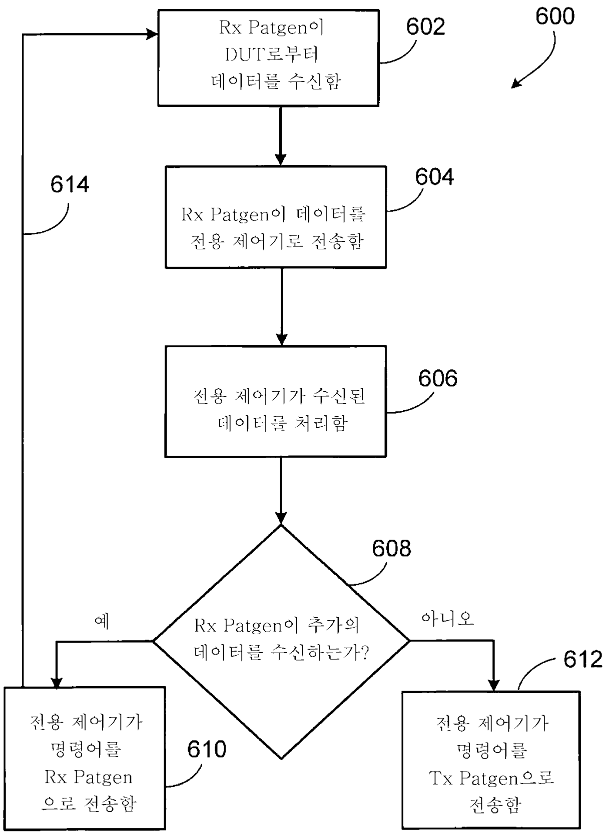
도면6


400
402
Tx Patgen이
연산 코드를 실행함
Tx Patgen이
Tx 플래그를
Rx Patgen으로 전송함
404
Rx Patgen이
DUT로부터
데이터를 수신함
406
Rx Patgen이 수신된
데이터를 검증하는
연산 코드를 실행함
408
Rx Patgen이
Rx 플래그를
Tx Patgen으로 전송함
410

도면7


530
호스트
장치
전용 제어기
502
504
104
프로그램가능
하드웨어
522
HRAM
524
ISL
526
HRAM
528
ISL
302
Tx Patgen
310
306
308
Rx Patgen
312
304
캡처
메모리 장치
316
40
40
508
10b 정렬
MOSC_M4
506
514
520
522
510
516
518
Sys_Ref
512
MOSC
테스트 대상 장치
12

도면8



Rx Patgen이 DUT로부터 데이터를 수신함
602
600
614
604
Rx Patgen이 데이터를 전용 제어기로 전송함
606
전용 제어기가 수신된 데이터를 처리함
608
예
Rx Patgen이 추가의 데이터를 수신하는가?
아니오
612
전용 제어기가 명령어를 Rx Patgen으로 전송함
610
전용 제어기가 명령어를 Tx Patgen으로 전송함

도면9

IN ROV	연산 코드	피연산자	워드3	워드2	워드1	워드0	
	RPT	1280	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2.	D10.2	K28.5	
	RPT	12	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	
	RPT	12	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	
	RPT	12	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	
	RPT	12	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	
	RPT	12	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	350
14	Wait_RX		XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	
	RPT	4	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	OOB 시퀀스
	RPT	4	D27.3	D10.2	D10.2	K28.5	
	RPT	4	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	
	RPT	4	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	
	RPT	4	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	
	RPT	4	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
	RPT	4	D27.3	D10.2	D10.2	K28.5	
27	Wait_RX		XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	
30	Wait_RX		D10.2	D10.2	D10.2	D10.2	
32	Wait_RX		D27.3	D10.2	D10.2	K28.5	
	RPT	280	D21.5	D21.5	D21.4	K28.3	
	RPT	2	D27.3	D10.2	D10.2	K28.5	유휴 시퀀스
	RPT	522	D21.5	D21.5	D21.4	K28.3	
	RPT	2	D27.3	D10.2	D10.2	K28.5	352
	RPT	12	D21.5	D21.5	D21.4	K28.3	
			D25.4	D25.4	D10.5	K28.3	
36	Wait_RX						
	RPT	2	D10.2	D10.2	D21.4	K28.3	
			D25.4	D25.4	D10.5	K28.3	
	Wait_RX						
	RPT	2	D21.2	D21.2	D21.5	K28.3	354
			D25.4	D25.4	D10.5	D28.3	
42	Wait_RX						수신 패킷
	RPT	2	D21.1	D21.1	D21.5	K28.3	
			D25.4	D25.4	D10.5	K28.3	
44	Wait_RX						
		2	D21.5	D21.5	D21.5	K28.3	
		2	D23.2	D23.2	D23.2	K28.3	
			D25.4	D25.4	D25.4	K28.3	356
47	Wait_RX						
			D23.1	D23.1	D23.1	K28.3	전송 패킷
			D1.6-C1	D29.1+3D	D22.7-F6	D10.5 AA	
			D31.5+	D6.1	D19.5	D9.3+	
			D5.5	D8.0+	D19.0	D19.3-	
			20.1	D18.2	D19.6	D21.0+	
			10.4+	D21.2	D21.4+	D2.0-	
			D15.7+EF	D25.4+	D18.0+	D2.5-A2	
			D21.6	D21.6	D21.5	K28.3	
		2	D24.2	D24.2	D21.5	K28.3	
			D25.4	D25.4	D10.5	K28.3	
52	Wait_RX						
			D21.5	D21.5	D21.4	K26.3	

도면10

연산 코드	Care	워드3	워드2	워드1	워드0
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Set_TX	1111	0	0	0	0
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Reset	M			D10.2	K28.5
Wait_MCH					
Set_TX	1111	0	0	0	0
Reset	M			D10.2	K28.5
Wait_MCH					
Set_TX					
Reset	M			D21.5	D21.5
Wait_MCH					
Set_TX					
Reset	M			D21.5	D21.5
Wait_MCH					
Set_TX					
Reset	M			D23.5	D23.1
Wait_MCH					
SET_CAP_OFF					
RESET	M			D21.0	D21.5
SET_CAP_OFF					
Reset	M			D21.4	D21.5
Wait_MCH					
Set_TX					
Reset	M			D10.2	D21.4
Wait_MCH					
Set_TX					
Reset	M			D21.2	D21.2
Wait_MCH					
Reset				D21.1	D21.1
Wait_MCH	M				
Set_TX					

360 OOB 시퀀스

362 유휴 시퀀스

364 수신 패킷

366 전송 패킷