



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0000536
(43) 공개일자 2008년01월02일

- | | |
|--|---|
| <p>(51) Int. Cl.
G06F 11/22 (2006.01) G06F 11/273 (2006.01)
G06F 11/26 (2006.01)</p> <p>(21) 출원번호 10-2007-0063728</p> <p>(22) 출원일자 2007년06월27일
심사청구일자 없음</p> <p>(30) 우선권주장
11/477,049 2006년06월27일 미국(US)</p> | <p>(71) 출원인
실리콘 이미지, 인크.(델라웨어주 법인)
미국 캘리포니아 94086 써니베일 이스트 아쿠에스
에비뉴 1060</p> <p>(72) 발명자
술, 친송
미국 94040 캘리포니아주 마운틴 뷰 아파트먼트
1628 콘티넨탈서클 707</p> <p>(74) 대리인
양영준, 백만기</p> |
|--|---|

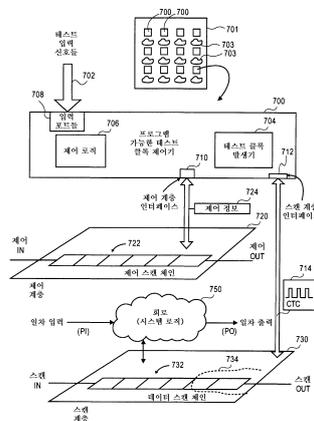
전체 청구항 수 : 총 17 항

(54) 프로그램 가능한 테스트 클럭 제어기들을 사용해 전자회로들의 스캔-기반 테스트를 위한 구성 가능한 테스트 클럭들을 생성하는 테스트 클럭 제어 구조들

(57) 요약

전자 회로들의 스캔-기반 테스트를 구현하기 위해 스캔 체인들을 위한 테스트 클럭을 생성하기 위한 시스템들, 구조들, 및 방법들이 개시된다. 일 실시예에서, 테스트 클럭 제어 구조는 프로그램 가능한 테스트 클럭 제어기를 포함한다. 프로그램 가능한 테스트 클럭 제어기는 구성 가능한 테스트 클럭을 생성하기 위한 테스트 클럭 생성기를 포함한다. 프로그램 가능한 테스트 클럭 제어기는 구성 가능한 테스트 클럭으로써 스캔 체인 일부분을 구동하기 위한 스캔 계층 인터페이스 및 스캔 체인 일부분을 제어하기 위한 제어 정보에 액세스하도록 구성된 제어 계층 인터페이스도 포함한다. 다른 실시예에서, 방법은 회로들의 스캔-기반 테스트를 실현한다. 이 방법은, LSL(last-shift-launch) 테스트 패턴들 및 브로드사이드 테스트 패턴들을 포함할 수 있는 동적인 장애 검출 테스트 패턴들을 사용해, 하나 이상의 도메인내 테스트를 수행하는 단계 및 하나 이상의 도메인간 테스트를 수행하는 단계를 포함한다.

대표도 - 도7



특허청구의 범위

청구항 1

전자 회로들의 스캔-기반 테스트를 구현하기 위해 테스트 클럭을 발생하기 위한 테스트 클럭 제어 구조로서,
회로를 테스트하기 위한 프로그램 가능한 테스트 클럭 제어기를 포함하고,
상기 프로그램 가능한 테스트 클럭 제어기는,

구성 가능한 테스트 클럭을 발생하도록 구성된 테스트 클럭 발생기;

상기 구성 가능한 테스트 클럭으로 스캔 체인 중의 스캔 체인 일부분을 구동하도록 구성된 스캔 계층 인터페이스; 및

상기 스캔 체인 일부분을 제어하기 위한 제어 정보를 액세스하도록 구성된 제어 계층 인터페이스를 포함하는 테스트 클럭 제어 구조.

청구항 2

제1항에 있어서,

상기 프로그램 가능한 테스트 클럭 제어기는 제어 체인의 일부분을 인터페이스하는 제어 로직을 포함하고, 상기 제어 로직은 상기 제어 정보에 기초하여 스캔 데이터를 상기 스캔 체인 일부분으로 선택적으로 로딩하도록 구성되는 테스트 클럭 제어 구조.

청구항 3

제2항에 있어서,

상기 제어 체인은 상기 스캔 체인과는 상이한 채널이고, 이에 의해, 상기 스캔 데이터의 전송과는 무관하게, 상기 프로그램 가능한 테스트 클럭 제어기로의 상기 제어 정보 전송을 용이하게 하는 테스트 클럭 제어 구조.

청구항 4

제1항에 있어서,

상기 제어 로직은 스캔 클럭과 기능 클럭 사이에서 멀티플렉싱하여, 상기 스캔 클럭 및 상기 기능 클럭의 실질적으로 정상 상태 일부분들 동안, 상기 구성 가능한 테스트 클럭을 형성하도록 구성되는 테스트 클럭 제어 구조.

청구항 5

제1항에 있어서,

상기 프로그램 가능한 테스트 클럭 제어기는 통상적인 단일 스캔 인에이블 신호를 집합적으로 대체하는 다수의 스캔 인에이블 제어 신호들을 수신하도록 구성된 다수 입력 포트들을 더 포함하는 테스트 클럭 제어 구조.

청구항 6

제1항에 있어서,

상기 프로그램 가능한 테스트 클럭 제어기는,

상기 클럭 펄스 제어기에 프로그램된 클럭 명령 정보에 따라 클럭 제어 신호를 발생하도록 구성된 클럭 펄스 제어기를 포함하고,

상기 구성 가능한 테스트 클럭은 상기 클럭 명령 정보의 일부분의 함수이며,

상기 클럭 명령 정보는 기능 클럭 신호를 스캔 체인에 인가할 것인지의 여부를 특정하는 테스트 클럭 제어 구조.

청구항 7

제6항에 있어서,

상기 프로그램 가능한 테스트 클록 제어기는,

복수개 제어 클록 신호들을 합성하도록 구성된 스캔 클록 디코더를 더 구비하고, 상기 복수개 제어 클록 신호들은,

적어도 상기 클록 펄스 제어기에 상기 클록 명령 정보를 공급하기 위한 적어도 제어 계층 클록 신호, 및

상기 스캔 체인을 구동하기 위한 스캔 계층 클록 신호를 포함하는 테스트 클록 제어 구조.

청구항 8

제6항에 있어서,

상기 클록 펄스 제어기는,

상기 클록 제어 신호를 위해 개개의 클록 예지들을 발생하도록 구성된 클록 제어 신호 발생기; 및

상기 개개의 클록 예지들 각각에 대한 클록 사이클 주기를 설정하도록 구성된 클록 속도 선택기를 포함하고,

상기 클록 속도 선택기는 앳-스피드(at-speed) 테스트를 위한 제1 클록 속도를 지시하는 제1 주기와 적어도 상기 스캔 체인을 구동하기 위한 제2 클록 속도를 지시하는 제2 주기 사이에서 선택하는 테스트 클록 제어 구조.

청구항 9

복수개 회로들을 포함하는 전자 디바이스의 스캔-기반 테스트를 구현하기 위해 스캔 체인들에 대한 테스트 클록 파형을 발생하기 위한 방법으로서,

프로그램 가능한 테스트 클록 제어기를, 클록 명령 정보가 프로그램 가능한 테스트 클록과 교환되는 데이터 전달 모드에서 동작하도록 구성하는 단계;

체인 클록을 상기 복수개 회로들 중의 회로와 연관된 스캔 체인 일부분에 인가할 것인지의 여부를 판정하는 단계; 및

상기 체인 클록을, 다른 회로들로의 다른 체인 클록들의 인가와 무관하게, 상기 회로를 테스트하기 위한 테스트 클록으로서 인가하는 단계를 포함하는 방법.

청구항 10

제9항에 있어서,

상기 프로그램 가능한 테스트 클록 제어기를, 스캔 데이터가 적어도 상기 스캔 체인 일부분과 교환되는 또 다른 데이터 전달 모드에서 동작하도록 구성하는 단계를 더 구비하고, 상기 체인 클록의 스피드는 시프팅 클록 속도인 방법.

청구항 11

제10항에 있어서,

상기 프로그램 가능한 테스트 클록 제어기를, 상기 클록 명령 정보에 따라 테스트 실행 모드에서 동작하도록 구성하는 단계를 더 구비하는 방법.

청구항 12

제10항에 있어서,

상기 프로그램 가능한 테스트 클록 제어기를 상기 테스트 실행 모드에서, 상기 체인 클록의 상기 스피드를 동적 테스트를 수행하기 위한 앳-스피드(at-speed) 클록 속도로 설정하도록 작동하는 단계를 더 구비하는 방법.

청구항 13

제9항에 있어서,

상기 스캔 데이터의 스캔 데이터 비트들의 제1 부분 및 상기 스캔 데이터의 스캔 데이터 비트들의 제2 부분을 실질적으로 동시에 상기 스캔 체인 일부분으로 로딩하는 단계 - 상기 제1 부분 및 상기 제2 부분은, 각각, N개 비트들 및 N-1개 비트들을 포함함 -;

상기 프로그램 가능한 테스트 클록 제어기를, 상기 N개 비트들에 대한 N개의 스캔 클록 에지들 이후에 브로드사이드 테스트(broadside test)를 수행하도록 구성하는 단계 - 상기 클록 명령 정보는 상기 N개 비트들 이후에 상기 앳-스피드(at-speed) 클록 속도에서 개시 클록 에지 및 포착 클록 에지를 제공함 -; 및

상기 프로그램 가능한 테스트 클록 제어기를, 상기 N-1개 비트들에 대한 N-1개의 스캔 클록 에지들 이후에 LSL(last shift launch) 테스트를 수행하도록 구성하는 단계 - 상기 클록 명령 정보는 상기 개시 클록 에지를 N번째 클록 에지로서 그리고 상기 포착 클록 에지를 N+1번째 클록 포착 클록 에지로서 제공함 -

를 더 포함하는 방법.

청구항 14

스캔-기반 테스트를 구현하기 위해 테스트 클록들을 발생하기 위한 테스트 클록 제어 구조를 포함하는 전자 디바이스로서,

복수의 회로들 - 상기 복수의 회로들은,

상기 전자 디바이스에 대한 기능 모드에서 동작 가능한 다수 도메인들을 포함하고, 상기 다수 도메인들 중 하나 이상의 도메인이 테스트 모드에서 각각을 독립적으로 테스트하기 위한 서브-도메인들로 세분됨 -;

상기 다수 도메인들에 대한 테스트 클록 신호들을 발생하기 위한 복수개의 프로그램 가능한 테스트 클록 제어기들 - 상기 복수개의 프로그램 가능한 테스트 클록 제어기들은, 상기 테스트 모드 동안 상기 서브-도메인들 중 하나 이상을 실질적으로 병렬로 테스트하도록 구성되는 상기 복수개의 프로그램 가능한 테스트 클록 제어기들의 서브세트를 포함함 -

을 포함하는 전자 디바이스.

청구항 15

제14항에 있어서,

상기 서브-도메인들 중 어떤 것이, 선택된 서브-도메인들로서 테스트될 것인지를 선택하기 위한 클록 명령 정보를 전달하기 위해, 상기 복수개의 프로그램 가능한 테스트 클록 제어기들에 연결된 하나 이상의 제어 체인들;

상기 선택된 서브-도메인들과 데이터를 교환하기 위해 상기 도메인들 및 상기 서브-도메인들에 연결된 하나 이상의 스캔 체인들; 및

상기 데이터의 교환을 제어하기 위해 상기 클록 명령 정보에 따라 상기 테스트 클록 신호들을 발생하도록 구성된 테스트 클록 발생기들을 더 포함하는 전자 디바이스.

청구항 16

제15항에 있어서,

하나의 상기 도메인을 상기 서브-도메인들로 세분하는 것은, 상기 테스트 클록 신호들의 제어하에 있는 상기 하나 이상의 스캔 체인들의 서브세트에 대하여 상기 데이터를 자극 또는 결과 데이터로서 선택적으로 로딩 또는 언로딩하는 것에 의해 테스트 패턴 볼륨 감소를 용이하게 하는 전자 디바이스.

청구항 17

제14항에 있어서,

상기 복수개 회로들 중 적어도 일부를 저전력 상태로 두기 위해 PWD(power down) 신호를 전달하기 위한 PWD 신호 발생기를 더 포함하고, 상기 PWD 신호 발생기는 상기 서브-도메인들 중 하나 이상의 동작을 정지시키기 위해 상기 테스트 클록 발생기들에 연결되는 전자 디바이스.

명세서

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

<39> 발명의 간단한 설명

<40> 이 발명은 일반적으로 회로들을 스캔 테스트하기 위한 전자 디바이스들 및 클록 아키텍처들(clock architectures)에 관한 것으로서, 좀더 구체적으로는, 예를 들어, 적어도 지연-관련 장애들을 검출하기 위한 앳-스피드(at-speed) 스캔-기반 테스트를 용이하게 하는 구성 가능한 테스트 클록 파형들을 발생시키기 위한 시스템들, 구조들, 및 방법들에 관한 것이다.

<41> 발명의 배경

<42> 전자 디바이스 제조자들은 나노미터-레벨 반도체 가공 프로세스들을 사용해 감소된 지오메트리 사이즈들의 집적 회로들을 제조함으로써, 단위 면적당 좀더 많은 트랜지스터들 및 상호 접속 리소스들을 제공한다. 그러나, 점차 작아지는 지오메트리들로 디바이스들을 제조하는 것은 컨덕터 관련 장애들 및 저항 유형 결함들 역시 증가시킨다. 컨덕터 관련 장애들은 일반적으로, 예를 들어, 일관성없는 접촉 저항들을 초래하는, 금속 상호 접속들의 품질을 열화시키는 프로세스 변화들로부터 발생한다. 저항 결함들은, 영향 받은 회로 노드들에서 신호 전이들이 좀더 느리게 상승하고 하강하게 한다. 그러나 전통적인 고착(stuck-at) 및 파라메트릭 누설 전류 테스트(예를 들어, IDDQ 테스트) 방법들은 장애들의 이 유형들을 효과적으로 검출할 수 없으므로, 전자 디바이스 제조자들은 그러한 장애 모드들을 배제하기 위해 흔히 지연 테스트 방법들에 의존한다. 그러나, 구조적 지연 테스트들은, 고착 테스트와 같은, 전통적인 테스트 접근 방법들에 비해 눈에 띄게 고비용이다. 예를 들어, 지연 테스트들을 구현하기 위한 데이터 볼륨이 전통적인 고착 테스트들을 위한 데이터 볼륨보다 훨씬 크다. 일부 경우들에서, 지연 테스트들은 고착 테스트들보다 3배 내지 5배 많은 데이터를 요구한다. 더 나아가, 고착 테스트들은, 1이나 0의 논리 값으로 "고착되는" 회로 노드와 같은, 정적 장애들(static faults)을 검출한다. 비교해 보면, 지연 테스트들은, 신호들이 수용 가능한 것보다 좀더 느리게 상승하거나 하강하게 하는 동적 장애들(dynamic faults)을 검출한다. 일반적으로, 지연 테스트들은, 테스트가 지연 관련 장애들을 검출하는 회로를 통해 전이 신호를 전파할 것을 요구한다. 지연 테스트들은 소정 타이밍 제약들내에서 전이 신호를 개시(launch) 및 포착(capture)할 것을 요구하므로, 지연 테스트 벡터들은 고착 테스트 벡터들보다 테스트 클록을 발생시키고 테스트 클록과 동기하기가 훨씬 더 어렵다.

<43> 도 1은 CUT(circuit under test: 테스트 대상 회로, 102)에서 관련된 동적 장애들을 검출하기 위한 전통적인 지연 테스트들을 구현하는데 흔히 사용되는 스캔 체인의 일부분(100)을 나타내는 도면이다. 전통적인 스캔 체인들은 일반적으로, 플립플롭들(108) 및 멀티플렉서들(106)을 포함한다. 스캔 인에이블(SE;scan enable) 신호(104)는 멀티플렉서들(106)을 제어하여, 회로(102)와 자극 및 결과 신호들을 교환하거나 스캔 데이터를 플립플롭들(108)로 또는 플립플롭들(108)로부터 시프팅한다. 전통적으로, SE(104)는, 단일 소스로부터, 테스트 클록(CLK;120)과 유사한 팬아웃 정렬을 통해 전파하는 전역적 신호이다. SI(scan in terminal;110)는, ATE(automatic test equipment)와 같은, 외부 소스로부터의 스캔 데이터를 수용하는 한편, SO(scan out terminal;112)는 자극 신호들에 의해 발생된 결과들을 시프팅한다. 스캔 체인 일부분(100)을 사용해 지연 테스트를 조정하기 위해, 테스트 클록 발생기들은 일반적으로, 스캔 체인을 통해 스캔 데이터를 구동하기 위한 CLK(120)를 발생시킨다. 멀티플렉서(130)는 저속 클록(140)이나 고속 클록(150)을 게이팅한다. 구체적으로, 스캔 체인 일부분(100)은 스캔 체인을 통해 스캔 데이터를 구동하기 위해 저속 클록(140)을 사용하고, 회로(102)에 대한 앳-스피드 기능 테스트를 수행하기 위해 고속 클록(150)을 사용한다. ATE들은 테스트 클록 발생기들로서 동작할 수 있는 한편, 온칩 PLL(phase-locked loop) 회로들과 같은, 온칩 기능 클록 회로들은 저비용으로 고속 테스트 클록 신호들을 제공할 수 있다. 그러나, 전통적인 테스트 클록 발생 회로들은 복잡하고 고비용인데, DUT(device under test)가, 20개 내지 100개의 클록 도메인들 또는 그 이상과 같은, 다수 클록 도메인들을 포함하는 경우에 특히 그러하다.

<44> 도 2는 단일 SE 신호를 사용해 전통적인 스캔 체인 구조들에서 전통적인 앳-스피드 지연 테스트들을 수행하는 것으로부터 발생하는 타이밍 불확실성들을 예시한다. 동적 장애들을 검출하는데 흔히 사용되는 앳-스피드 지연 테스트들의 일례는 "LSL(last shift launch)" 테스트이다. 이 기술에서는, 제1 테스트 패턴을 위해 스캔 체인으로 시프팅된 마지막 스캔 데이터 비트가 한번의 추가 시프팅 이후에 제2 테스트 패턴의 입력이 된다. 타이밍

도(200)는 전통적인 LSL 테스트를 수행하는 도 1의 CLK(120) 및 SE(104) 신호들을 나타낸다. 특히, 제1 테스트 패턴은 스캔 모드 동안 저속 클럭(140)을 사용해 스캔 체인으로 시프팅되는데, 마지막 스캔 데이터 비트는 개시 에지(210)로서 스캔 체인으로 시프팅된다. 앳-스피드 지연 테스트를 수행하기 위해, SE(104)는, 스캔 체인으로 고속 클럭(150)을 인가하여 기능 테스트 결과를 포착하는 것과 함께, 상태를 변경한다. 앳-스피드 지연 테스트들을 구현하기 위해 단일 SE 신호(104)를 사용하는 것의 단점은, 포착 에지(220)의 검출이 소정 시구간내에 위치해야 하고, 그것에 의해, 앳-스피드 타이밍 제약(202)을 부과한다는 것이다. 따라서, SE(104)는 포착 에지(220)를 적절하게 검출하기 위해 앳-스피드 타이밍 제약(202) 동안 일 상태에서 후속 상태로 전이해야 한다. 그러나, 좀더 작은 지오메트리들에 대한 지연 테스트들을 수용하기 위해, 앳-스피드 타이밍 제약(202)이 점차 좁아짐에 따라, SE(104)가 상태들을 적절하게 전이할 것으로 기대하기는 어려워진다.

<45> 도 3은 전통적인 테스트 클럭 제어 기술들을 사용해 앳-스피드 지연 테스트들을 실현하기 위한 통상적인 스캔 체인 구조(300)를 예시한다. 도시된 바와 같이, 스캔 체인 구조(300)는 스캔 입력(310) 및 스캔 출력(312)을 가진 스캔 체인(320) 뿐만 아니라 내부 클럭 발생기(330; 예를 들어, 하나 이상의 PLL 회로들) 및 내부 테스트 클럭 제어기(340)를 포함한다. 스캔 체인 구조(300)는 내부 테스트 클럭 제어기(340)를 사용해 CUT(302; circuits under test)에 대한 앳-스피드 지연 테스트들을 수행한다. 각각의 CUT(302)는 클럭 도메인(304)에 상주한다. 클럭 도메인은, 특정 클럭과 동기되는 회로 영역이다. 클럭 제어 비트들(350)이 내부 테스트 클럭 제어기(340)의 동작을 정의한다. 그러나, 내부 테스트 클럭 제어기(340)를 구성하기 위해, 전통적인 앳-스피드 테스트 기술들은, 스캔 데이터 비트들과 함께, 스캔 체인(320)에 클럭 제어 비트들(350)을 매립한다. 이 접근 방법의 단점은, 매 스캔 체인 로딩마다 하나의 클럭 도메인(304)을 테스트하기 위한 클럭 제어 비트들(350)이 로딩된다는 것이다. 이와 같이, 별도 도메인(304)이 테스트될 때마다, 전체 스캔 체인(320)이 로딩 및 언로딩된다. 클럭 제어 비트들(350)이 정적이라는 것에도 주의해야 하는데, 테스트 동안에 특히 그러하다. 구체적으로, 스캔 체인 구조(300)는 일반적으로, 내부 테스트 클럭 제어기(340)가 그 비트들에 따라 동작할 수 있도록 하기 위해, 클럭 제어 비트들(350)이 스캔 체인(320)에서 부동 상태를 유지할 것을 요구한다. 그에 따라, 그러한 체인 구조(300) 및 전통적인 유사한 다른 스캔 체인 구조들도 스캔 체인(320)에서의 비트들과 무관하게 내부 테스트 클럭 제어기(340)를 조작하기에는 그다지 적합하지 않은데, 특히, 클럭 도메인간(inter-clock domain) 테스트들(예를 들어, 개시 및 포착)을 구현하는 경우에 그러하다. 또 다른 단점은, 전통적인 스캔 체인 구조(300)는 일반적으로, 테스트 시간 및 데이터 볼륨을 감소시킬 목적으로 스캔 체인(320) 일부분들의 선택적인 로딩 및 언로딩을 제어하는 제어 시퀀스들 및/또는 프로그램들을 지원하기에 부적합하다. 예를 들어, 대다수 스캔 체인들(320)은 목표로 정한 CUT(302)만을 테스트하기 위해 스캔 체인(320)(또는 그것에 관한 하나 이상의 일부분들)을 선택적으로 리로딩(reloading)할 수 없다. 이것은, 스캔 체인(320)에 특정 테스트를 위해 불필요한 데이터가 로딩될 수 있고, 그에 따라, 스캔 체인(320)으로 로딩된 불필요한 데이터가 테스트 데이터 볼륨을 증가시키며, 이것은 다시 테스트 시간을 증가시킨다는 것을 의미한다. 목표로 정해진 회로들(302) 중 하나의 결과들을 조사하기 위해, 비교적 장황하게 긴 경향이 있는 전통적인 스캔 체인들(320)은, 그것의 조합이, 일반적으로 긴 테스트 시간들에 기여하는 불필요한 데이터와 결과 모두를 시프팅할 것을 요구한다. 스캔 체인 구조(300)에 대한 또 다른 단점은, 도메인간 로직(306)이, 일 클럭 도메인의 포착 클럭 펄스를 다른 클럭 도메인으로부터의 개시 클럭 펄스와 충분히 동기시키기에는 일반적으로 부적합한데, 특히, 양자의 클럭 도메인들이 상이한 클럭 주파수들을 갖는 경우에 그러하다.

<46> 도 4는 전통적인 테스트 클럭 제어 기술들을 사용하는 도 3의 스캔 체인 구조(300)를 사용하는 도메인간 로직(306; inter-domain logic)의 테스트를 예시한다. 통상적으로, 논리 0으로부터 1로의(또는 그 반대의) 전이는 제1 클럭 도메인("i"; 410)의 출력 레지스터("OutReg"; 402)로부터 제2 클럭 도메인("j"; 420)의 입력 레지스터("InReg"; 404)로 개시된다. 클럭("CLK[i]"; 412)은 출력 레지스터(402)로부터 도메인간 조합 로직(306)을 경유하여 입력 레지스터(404)로의 전이를 구동하는데, 입력 레지스터(404)는 클럭("CLK[j]"; 422)에서 동작하여 전이의 상태를 래치한다. 바람직한 소정의 개시 에지(450)는 테스트 응답을 적절하게 포착하기 위한 포착 에지(470)의 동기화를 제공한다. 그러나, 클럭(422)을 위한 일 클럭 주기가 클럭(412)의 5개 클럭 주기들에 대응된다는 것을 고려한다. 전이가, 전통적인 지연 테스트들에서 통상적으로 수행되는 바와 같이, (클럭 도메인(420)의) 클럭(422)을 위한 클럭 에지(460)와 동기되어 클럭 도메인(410)으로부터 개시되면, 클럭 도메인(420)내에서 테스트 응답이 포착되기 전에 5개 클럭 주기들이 경과할 수 있다. 이와 같이, 에지(410)에서의 비의도적인 개시는 에지(470)에서의 테스트 응답을 적절하게 포착할 수 없을 수도 있다. 전통적인 스캔 체인 구조들에서 클럭 도메인들 사이의 동기화를 관리하는 것은, 참여하는 클럭 도메인들의 수가 증가함에 따라, 점차 어려워진다. 또한, 전통적인 내부 테스트 클럭 제어기들에 고유한 지연이 유사한 원인들로 인해 클럭 도메인간 테스트를 복잡하게 할 수도 있다.

<47> 도 5는 옛-스피드 지연 테스트들을 구현하기 위해 전통적인 테스트 기능 클록 경로(530)를 갖춘 내부 테스트 클록 제어기(502)를 묘사하는 블록도(500)이다. 내부 테스트 클록 제어기(502)는 기능 클록("PLL Clk";510) 및 매립된 클록 제어 비트들(504)을 수신하여 지연된 포착 펄스들을 발생시킨다. 내부 테스트 클록 제어기(502)는 기능 클록(510)의 에지들을 카운트하기 위한 펄스 카운터(520) 및 매립된 클록 제어 비트들(504)의 값들에 응답하여 테스트 클록(550)을 발생시키기 위한 로직(522)도 포함한다. 동작시에, 펄스 카운터(520) 및 로직(522)은, 예를 들어, 도메인간 테스트를 수행하기 위해, 협력하여 개시 펄스(560) 이후의 포착 펄스(570)를 지연(562)만큼 지연시킨다. 이 접근 방법의 단점은, 내부 테스트 클록 제어기(502)가, 멀티플렉서(534) 이외의 추가 회로 소자들을 포함하는 테스트 기능 클록 경로(530)를 포함한다는 것이다. 이들 추가 소자들(536)은 옛-스피드 기능 클록 신호(510)를 유해하게 스큐잉(skewing)하여, 개시 및 포착 동작들을 수행할 때, 포착 펄스(570)의 타이밍에서 불확실성(580)을 초래할 수 있다. 일반적으로, 지연(562)은, 테스트 모드에서의 경로(530)상의 옛-스피드 기능 클록이 (예를 들어, 테스트 모드(TM)가 디스에이블인 경우에) 실행 모드에서 추가 소자들(536)을 포함하지 않는 경로(532)상의 기능 클록(510)을 모방할 수 있을 것을 보장하기 위해 클록 밸런싱을 요구한다.

<48> 도 6a 및 도 6b는, 각각, 브로드사이드(broadside) 및 LSL(last-shift-launch) 테스트 프로토콜들을 위한 전통적인 구현들을 묘사한다. 도 6a는 다양한 레지스터들의 스테이지들(612)을 포함하는 스캔 체인(610)을 포함한다. 도표(600)에 도시된 바와 같은 브로드사이드 프로토콜 테스트를 위해, 소정 전이(602)가 선행 레지스터 스테이지(612b)로부터 개시되어 조합 회로(620b)를 통해 전파한 다음, 레지스터 스테이지(612c)의 레지스터(630)에서 포착된다. 이것은 조합 회로들(620a 및 620b)을 - 직접적으로 또는 간접적으로 - 테스트한다. 개시 에지 동안, 레지스터(632)는 논리 0를 래치하고, 그것에 의해, 조합 회로(620b)를 통해 1에서 0으로의 전이를 개시한다. 레지스터(630)는 포착 에지에서 논리 0 값을 포착한다. 논리 0의 존재는 레지스터(634)의 값으로부터의 선행 전이(601)에 의존하고, 그것에 의해, 조합 회로(620a)를 테스트한다는 것에 주의한다. 이 기술에서는, 전체 스캔 체인을, 기능 모드에서의 2개의 옛-스피드 클록들의 펄스들(예를 들어, 개시 및 포착 에지들)이 수반되는 테스트 모드에서의 느린 스피드들에서 시프팅하는 것에 의해, 전체 스캔 체인에 데이터가 로딩된다. 그 다음, 일단 값들이 포착되고 나면, 데이터는 테스트 모드에서 천천히 시프팅될 수 있다. 브로드사이드 프로토콜을 사용하는 지연 테스트가, 그렇지 않다면 검출 불가능한 지연 장애들을 검출할 수 있는 한편, 브로드사이드 프로토콜들을 위한 테스트 패턴들의 사이즈는 대체로 LSL 프로토콜들을 위한 패턴보다 크다. 더 나아가, 브로드사이드를 위한 테스트 패턴들은 본질적으로 순차적이므로, 발생시키기가 좀더 어렵다.

<49> 도 6b는 다양한 레지스터들의 스테이지들(672)을 포함하는 스캔 체인(660)을 포함한다. LSL 프로토콜 테스트를 위해, 레지스터 스테이지(672b)에서의 스캔 로딩 또는 언로딩 동안 마지막 시프트로부터 전이(662)가 개시된다. 그 다음, 그것은 후속 레지스터 스테이지(672c)의 레지스터(680)로 포착된다. 입력 전이(690)가 마지막 시프트(661)로부터 개시되므로, LSL 테스트들을 위한 패턴 사이즈들은 브로드사이드보다 좀더 작고 발생이 용이할 수 있는데, 순차적 테스트 패턴들은 조합 패턴들보다 압축하기가 좀더 어렵기 때문이다. 그러나, LSL 프로토콜은 상술된 단점들을 가진다. 브로드사이드 및 LSL 모두가 겉으로는 상호 배타적인 이점들 및 단점들을 가지므로, 대다수 설계자들은 둘 중 하나만을 구현한다.

발명이 이루고자 하는 기술적 과제

<50> 상기한 관점에서, 앞서 언급된 단점들을 최소화하는 시스템들, 구조들, 및 방법들을 제공하고 적어도 지연-관련 장애들을 검출하는 옛-스피드 스캔-기반 테스트를 제공하는 것이 바람직스러울 것이다.

발명의 구성 및 작용

<51> **발명의 요약**

<52> 전자 회로들의 스캔-기반 테스트를 구현하는 스캔 체인들을 위한 테스트 클록을 발생시키기 위한 시스템들, 구조들, 및 방법들이 개시된다. 일 실시예에서, 테스트 클록 제어 구조는 프로그램 가능한 테스트 클록 제어기를 포함한다. 프로그램 가능한 테스트 클록 제어기는 구성 가능한 테스트 클록을 발생시키기 위한 테스트 클록 발생기를 포함한다. 그것은 구성 가능한 테스트 클록으로써 스캔 체인 일부분을 구동하기 위한 스캔 계층 인터페이스 및 스캔 체인 일부분을 제어하기 위한 제어 정보에 액세스하도록 구성된 제어 계층 인터페이스도 포함한다. 프로그램 가능한 테스트 클록 제어기는 제어 체인의 일부분과 인터페이스하는 제어 로직도 포함한다. 제어 로직은 제어 정보에 기초해 스캔 데이터를 스캔 체인 일부분으로 선택적으로 로딩할 수 있다. 다양한 실시예들에서, 제어 체인은 스캔 체인과는 상이한 채널이므로, 스캔 데이터의 전송과는 무관하게, 프로

그럼 가능한 테스트 클록 제어기로의 제어 정보 전송을 용이하게 한다. 바람직스럽게도, 이것은 스캔 로드 및 언로드 시퀀스들을 감소시킴으로써, 테스트 시간을 감소시킨다. 일부 경우들에서, 제어 로직은, 스캔 체인으로 부터의 실질적으로 동일한 데이터를 사용해, 회로를 테스트하는 2 이상의 스캔 테스트 프로토콜들의 구현을 용이하게 할 수 있다.

<53> 다른 실시예에서, 프로그램 가능한 테스트 클록 제어기는 클록 펄스 제어기 및 테스트 클록 발생기를 포함한다. 클록 펄스 제어기는 클록 명령 정보에 따라 클록 제어 신호를 발생시킨다. 일부 경우들에서는, 클록 펄스 제어기가 클록 명령 정보를 보유할 수 있다. 테스트 클록 발생기는 클록 명령 정보의 일부분의 함수로서 테스트 클록을 발생시킨다. 클록 명령 정보는, 옛-스피드 클록 속도의 테스트 클록과 같은, 기능 클록 신호를 스캔 체인에 인가할 것인지의 여부를 특정한다.

<54> 또 다른 실시예에서, 방법은 회로들의 스캔-기반 테스트를 실시한다. 방법은 하나 이상의 도메인내 테스트 (intra-domain test)를 수행하는 단계 및, LSL 테스트 패턴들 및 브로드사이드 테스트 패턴들을 포함할 수 있는 동적인 장애 검출 테스트 패턴들을 사용해 하나 이상의 도메인간 테스트를 수행하는 단계를 포함한다. 도메인들은 상이한 전력 도메인들, 상이한 클록 도메인들, 또는 양자일 수 있다는 것에 주의한다. 일부 경우들에서, 방법은 상이한 클록 도메인들을 실질적으로 병렬로 테스트하도록 프로그램 가능한 테스트 클록 제어기들을 구성하는 단계를 더 포함한다. 바람직스럽게도, 이것은, 상이한 클록 도메인들을 테스트할 때, 하나 이상의 스캔 체인들이 로딩되는 횟수들을 감소시킬 수 있다. 다른 실시예들에서, 도메인이라는 용어는 일반적으로, 특징들이 클로킹(clocking), 전력 소비 등의 관점에서 상이한 것인지의 여부와 상관없이, 다른 일부분 또는 다른 회로와 비교하여 상이한 동작 특징들을 가진 임의 회로 또는 그것에 관한 일부분을 의미할 수 있다.

<55> 또 다른 실시예에서, 방법은 회로들의 스캔-기반 테스트를 실현하기 위한 테스트 클록 제어 구조를 구현하는데, 방법은, 스캔 체인들을 로딩하기 위한 프로그램 가능한 테스트 클록 제어기들의 제1 서브세트를 구성하는 단계 및 스캔 체인들의 로딩을 특정하는 클록 명령 정보에 따라 스캔 체인들로 스캔 데이터를 로딩하는 단계를 포함한다. 방법은, 장애들의 검출을 특정하는 추가적인 클록 명령 정보에 응답하여 테스트 클록을 발생시키는 프로그램 가능한 테스트 클록 제어기들의 제1 서브세트의 프로그램 가능한 테스트 클록 제어기들을 구성하는 것에 의해 계속된다. 더 나아가, 방법은 하나 이상의 회로들을 테스트하기 위해 추가적인 클록 명령 정보에 따라 테스트 클록을 발생시키는 단계를 포함할 수 있다.

<56> 일부 실시예들에 따르면, 프로그램 가능한 테스트 클록 제어기 및 수반되는 테스트 클록 제어 구조는 비교적 저렴한 옛-스피드 지연 테스트를 제공한다. 프로그램 가능한 테스트 클록 제어기는 실질적으로 클록 트리 네트워크의 (예를 들어, 기능적인 온칩 PLL 트리내의) 어디에든 삽입될 수 있고, ATE를 통해 프로그램될 수 있다. 테스트 클록 제어 구조는, 프로그램 가능한 테스트 클록 제어기들을 프로그램하는 것에 의해 비교적 다수의 클록 도메인들을 핸들링할 수 있다. 프로그램 가능한 테스트 클록 제어기는 동일 칩에서 LSL 및 브로드사이드 테스트 프로토콜들 양자를 사용할 수 있게 함으로써(예를 들어, 디바이스를 테스트하기 위한 스캔 로드/언로드 시퀀스들의 수를 감소시키는 것에 의해), 테스트 시간 및 테스트 데이터 볼륨을 감소시키면서 테스트 커버리지를 향상시킬 수 있다.

<57> 테스트 클록 제어 구조는, 발명의 다양한 실시예들에 따른, 테스트 시간 및 테스트 데이터 볼륨 양자를 감소시키기 위한 효과적인 테스트 개발 및 테스트 흐름들을 제공한다. 특히, 지연 테스트들은 단순한 고착 테스트들보다 구현에 약 3배 내지 5배로 비용이 많이 들기 때문에, 그러한 감소는 중요하다. 그리고 전통적인 지연 테스트에서, 테스트 시간 및 테스트 데이터 볼륨을 지배하는 것은 스캔 로드 및 언로드 동작들이다. 통상적으로, ATE의 1초 사용에는 약 5~10 센트의 비용이 든다. 이와 같이, 전통적으로 구현되는 지연 테스트들을 위한 증가된 ATE 시간은 저가의 상용 전자 디바이스들의 가격에 상당한 테스트 비용 오버헤드(test cost overhead)를 추가할 수 있다. 하나 이상의 실시예들은, 다른 것들 중에서도, 테스트 클록 구조에 프로그램 가능성을 도입하는 것에 의해, 동시에 또는 거의 동시에 다수의 테스트 프로토콜들을 구현하는 것에 의해, 그리고 다수 도메인들(예를 들어, 클록 도메인들)을 병렬로 테스트하는 것에 의해, 스캔 로드 및 언로드 동작들의 수를 감소시킨다.

<58> 첨부 도면들과 함께 고려되는 다음의 상세한 설명과 관련하여, 발명이 좀더 완전하게 이해될 수 있을 것이다.

<59> 유사한 참조 번호들은 도면들의 수개 장면들 전체에 걸쳐 대응되는 일부분들을 참조한다. 대다수 참조 번호들이, 일반적으로 그 참조 번호를 처음으로 도입하는 도면을 식별하는 하나 또는 2개의 최좌측 숫자들을 포함한다는 것에 주의한다.

<60> **실시예들의 일례들을 위한 상세한 설명**

- <61> 도 7은, 발명의 특정한 일 실시예에 따른, 하나 이상의 회로를 테스트하기 위한 테스트 클록 제어 구조를 예시하는 블록도이다. 테스트 클록 제어 구조(701)는 CUT(703)를 테스트하기 위한 하나 이상의 프로그램 가능한 테스트 클록 제어기들(700)을 포함한다. 도시된 바와 같이, 프로그램 가능한 테스트 클록 제어기들은 2개의 별도 계층들: 제어 계층(720) 및 스캔 계층(730)에서 데이터와 상호 작용한다. 프로그램 가능한 테스트 클록 제어기(700)는 구성 가능한 테스트 클록(CTC;714)을 발생시키기 위한 테스트 클록 발생기(704)를 포함한다. 프로그램 가능한 테스트 클록 제어기(700)는, 그것을 통해, CTC(714)로써 스캔 체인(732)의 적어도 스캔 체인 일부분을 구동하는 스캔 계층 인터페이스(712)도 포함할 수 있다. 도시된 바와 같이, 스캔 체인(732)은 스캔 계층(730)에 위치한다. 더 나아가, 프로그램 가능한 테스트 클록 제어기(700)는, 예를 들어, 스캔 체인(732) 또는 스캔 체인 일부분(734)을 제어하기 위해 제어 정보(724)에 액세스하는 제어 계층 인터페이스(710)를 포함할 수 있다.
- <62> 프로그램 가능한 테스트 클록 제어기(700)는, 예를 들어, 제어 계층 인터페이스(710)를 통해 제어 계층(720)에서의 제어 체인(722)의 일부분과 인터페이싱하는 제어 로직(706)을 포함할 수 있다. 그것에 의해, 제어 로직(706)은, 제어 정보(724)에 기초해 스캔 데이터를 스캔 체인(732)(또는 그것에 관한 일부분)으로 선택적으로 로딩할 것인지의 여부를 제어할 수 있다. 다양한 실시예들에서, 제어 체인(722)은 스캔 체인(734)과는 상이한 채널이므로, 스캔 데이터의 전송과 무관하게, 프로그램 가능한 테스트 클록 제어기(700)로의 제어 정보(724) 전송을 용이하게 한다. 바람직스럽게도, 이것은 스캔 로드 및 언로드 시퀀스들의 양을 감소시킴으로써, 테스트 시간을 감소시킨다. 일부 경우들에서, 제어 로직(706)은 스캔 체인으로부터의 실질적으로 동일한 데이터를 사용해 (예를 들어, 상이한 테스트 프로토콜들 사이에서의 스캔 체인의 전체적인 로딩 또는 언로딩없이) 테스트 회로(750)에 대한 2 이상의 스캔 테스트 프로토콜들의 구현을 용이하게 할 수 있다. 예를 들어, 프로그램 가능한 테스트 클록 제어기(700)는 동일한 테스트 흐름 동안 브로드사이드 테스트 및 LSL 테스트 프로토콜들의 사용을 제공할 수 있다. 프로그램 가능한 테스트 클록 제어기(700)는 단일 프레임워크에 상이한 테스트 프로토콜들을 수용할 수 있으므로, 그것은 바람직스럽게도, 단일 지연 테스트 프로토콜의 구현에 비해, 테스트 커버리지를 증가시킬 수 있다. 그래서, LSL 기술을 사용해 장애가 검출되지 않으면, 예를 들어, 브로드사이드 테스트가 장애를 검출하기 위해 구현될 수 있다. 일 실시예에서, 프로그램 가능한 테스트 클록 제어기(700)는, 전통적인 단일 스캔 인에이블 신호를 집합적으로 대체하는 다수 스캔 인에이블 제어 신호들을 수신하기 위한 입력 포트들(708)을 포함한다. 다른 실시예에서는, 테스트 클록 제어 구조(701)가 회로(750)를 포함하는 단일 기관상에 형성될 수 있다. 구성 가능한 테스트 클록(714)을 테스트 클록 뿐만 아니라 체인 클록이라고도 할 수 있다. 일반적으로, 테스트 클록 제어 구조는 프로그램 가능한 테스트 클록 제어기들, 제어 체인들, 스캔 체인들 뿐만 아니라 회로를 테스트하기 위한 다른 테스트-관련 구조들도 포함한다.
- <63> 도 8은, 발명의 하나 이상의 특정 실시예에 따른, 프로그램 가능한 테스트 클록 제어기를 예시하는 블록도이다. 프로그램 가능한 테스트 클록 제어기(800)는 비교적 낮은 스피드(즉, "스캔 스피드")의 스캔 클록을 사용하는 구조적 지연 테스트들을 위한 구성 가능한 테스트 클록들 및 다수의 지연 테스트 프로토콜들을 구현하기 위한 테스트 클록(830)을 구성하기 위한 비교적 높은 스피드(즉, "엡 스피드")의 고속 기능 클록을 발생시키도록 구성된다. 프로그램 가능한 테스트 클록 제어기(800)는 스캔 체인의 로드 및 언로드 동작들의 시퀀스들을 수행하기 위한 테스트 클록을 구성할 수 있다. 또는, 입력 전이를 개시하고 테스트 응답을 포착하기 위한 테스트 클록을 구성할 수도 있다. 이와 같이, 프로그램 가능한 테스트 클록 제어기(800)는 테스트 클록(830)을, 스캔 데이터를 스캔 체인으로 또는 스캔 체인으로부터 구동하기 위한 스캔-스피드 클록(SLCK;832) 또는 동작들의 개시 및/또는 포착을 수행하기 위한 엡-스피드 기능 클록(PCLK;834)으로서 구성할 수 있다. 하나 이상의 실시예에서, 멀티플렉서(840)(또는 멀티플렉싱 동작을 수행할 수 있는 임의의 등가 구조)는 스캔-스피드 클록(832) 또는 엡-스피드 기능 클록(834)을 테스트 클록(830)을 제공하기 위한 출력 터미널로 라우팅한다. 프로그램 가능한 테스트 클록 제어기(800)는, 내부 클록 구조들을 크게 변경하지 않으면서, 실질적으로 임의의 기능 클록 경로에 삽입될 수 있고, 온칩 PLL과 같은, 임의의 기능 클록 소스와 통합될 수 있다. 더 나아가, 프로그램 가능한 테스트 클록 제어기(800)는 임의의 클록 도메인을 디스에이블하도록 프로그램될 수 있고, 그것에 의해, 임의의 개시, 포착, 스캔 로드, 및 스캔 언로드 동작으로부터 하나 이상의 클록 도메인들을 선택적으로 포함하고 배제할 수 있다.
- <64> 프로그램 가능한 테스트 클록 제어기(800)는 기능 클록(PCLK[i];817) 및 외부적으로 구동되는 스캔 스피드 스캔 클록("Scan Clock";810)으로부터, 테스트 클록("CLK";830)과 같은, 구성 가능한 테스트 클록을 발생시키기 위한 스캔 클록 디코더(802) 및 클록 모듈(804)을 포함한다. 일부 경우들에서, PLL 회로(850)는 기능 클록(834)을 발생시키고 ATE는 스캔 클록(810)을 발생시킨다. 기준 클록("RefClk[k]";815)은 PLL(850)에 제공되어 기능 클록(817)을 발생시킬 수 있고, 예를 들어, 클록 모듈(804)의 동작을 제어할 수 있다. 일부 실시예들에서는, 엡-

스피드 기능 클록(834)이 실질적으로 기능 클록(817)이다. 스캔 클록 디코더(802)는 스캔 클록(810)을 디코딩하여, 제어 계층 클록 신호("TCLK";820) 및 스캔 계층 클록 신호("SCLK";822)를 포함하는 클록 제어 신호들을 합성한다. 제어 계층 클록 신호(820)는 적어도 (도시되지 않은) 제어 체인으로부터의 클록 명령 정보를 클록 모듈(804)로 공급하는데 사용된다. 스캔 계층 클록 신호("SCLK";822)는 스캔 체인 또는 (도시되지 않은) 그것에 관한 일부분을 구동하는데 사용된다. 제어 계층 클록 신호(820) 및 스캔 계층 클록 신호(822)는 본질적으로 개념적일 수 있으며 프로그램 가능한 테스트 클록 제어기(800) 이상으로 확장할 필요는 없다는 것에 주의한다.

<65> 전통적인 스캔 테스트 클록 제어 기술들은 단일 SE 신호를 사용해 스캔 경로와 기능 경로 사이를 멀티플렉싱하지만, 프로그램 가능한 테스트 클록 제어기(800)는 전통적인 단일 SE 신호를 2 이상의 스캔 인에이블 제어 신호들: SeC(Scan enable Clock) 제어 신호(812) 및 SeD(scan enable data) 제어 신호(814)로 대체한다. SeD 제어 신호(814)는 기능 경로와 스캔 경로 사이를 멀티플렉싱하도록 구성되는 한편, SeC 제어 신호(812)는 스캔 클록과 기능(즉, 앳-스피드) 클록 사이를 멀티플렉싱한다. 바람직스럽게도, 클록과 데이터 제어의 분리는 클록 발생 구조들을 위한 설계를 간략화하고, 스캔 시프팅으로부터 개시 및 포착 동작들로의 또는 그 반대의 동기화를 용이하게 한다. 더 나아가, 별도의 SeD 클록 제어 신호(814) 및 SeC 데이터 제어 신호(812)는 프로그램 가능한 테스트 클록 제어기(800)가, 브로드사이드 및 LSL과 같은, 상이한 테스트 프로토콜들을 동일한 칩상에 구현할 수 있는 능력도 제공한다.

<66> 도 9는 발명의 실시예에 따른 프로그램 가능한 테스트 클록 제어기에 의해 수행되는 기능들의 일례들을 예시하는 도면이다. 프로그램 가능한 테스트 클록 제어기(900)는 비교적 낮은 스피드의 스캔 클록("SCLK";902)과 비교적 높은 스피드의 기능 클록("PCLK";904)을 멀티플렉싱하여, 적어도 브로드사이드 및 LSL 지연 테스트 프로토콜들을 구현하기 위한 테스트 클록(930)을 발생시키도록 구성된다. 특히, 프로그램 가능한 테스트 클록 제어기(900)는 스캔 클록(902) 및 기능 클록(904)의 실질적으로 정상 상태 일부분들(908;steady-state portions) 동안 클록 소스들 사이를 멀티플렉싱한다. 정상 상태 위상(906)에서 동일한 정상 상태 값(예를 들어, 논리 로우)을 가진 클록 소스들 사이를 멀티플렉싱하는 것에 의해, 프로그램 가능한 테스트 클록 제어기(900)는 스캔 클록(902)과 기능 클록(904) 사이의 안전한 스위칭을 제공하고, 그에 따라, 예를 들어, 듀티 사이클들은 변경없이 유지되고 글리치들은 최소화된다. 일 실시예에서, 프로그램 가능한 테스트 클록 제어기(900)는 멀티플렉서(910)를 사용해, 앳-스피드 클록 발생 경로에 추가적인 논리 소자들을 요구하지 않으면서, 스캔 클록(902)과 기능 클록(904) 사이를 멀티플렉싱한다. 이와 같이, 프로그램 가능한 테스트 클록 제어기(900)는, 클록 게이팅 목적들을 위해 추가적인 논리 소자들을 사용하는 일부 전통적인 테스트 클록 발생기에서 흔히 볼 수 있는 클록 스크류를 방지할 수 있다. 프로그램 가능한 테스트 클록 제어기(900)는 테스트 클록(930)을 브로드사이드 클로킹을 제공하기 위한 클록(926)이나 LSL 클로킹을 제공하기 위한 클록(946)으로서 구성할 수 있다. 이와 같이, 프로그램 가능한 테스트 클록 제어기(900)는 바람직스럽게도 브로드사이드 및 LSL 프로토콜들 양자를 구현할 수 있다. 브로드사이드 클로킹(920)을 구현하기 위해, 프로그램 가능한 테스트 클록 제어기(900)는 스캔 모드 동안 N개의 스캔 데이터 비트들을 스캔 체인으로 시프팅하도록 동작한 다음, 기능 모드 동안, 앳-스피드의 개시 클록 예지(922) 및 포착 클록 예지(924) 양자를 제공한다. LSL 클로킹(940)을 구현하기 위해, 프로그램 가능한 테스트 클록 제어기(900)는 스캔 모드 동안 N-1개의 스캔 데이터 비트들을 시프팅하도록 동작한다. N-1번째 비트 이후에, 프로그램 가능한 테스트 클록 제어기(900)는 개시 클록 예지(924)를 스캔 모드에서의 N번째 클록 예지로서 제공하는데, 그 이후에는 포착 클록 예지(944)가 기능 모드 동안 N+1번째 클록의 포착 클록 예지로서 수반된다.

<67> 도 10은 발명의 실시예에 따른 프로그램 가능한 테스트 클록 제어기의 일례를 예시하는 도면이다. 프로그램 가능한 테스트 클록 제어기(1000)는 클록 펄스 제어기(1002)로 프로그램된 클록 명령 정보("CCI";1012)에 따라 클록 제어 신호("TC_En";1024)를 발생시키도록 구성된 클록 펄스 제어기("CPC";1002)를 포함한다. 클록 펄스 제어기(1002)는 제어 계층에서의 제어 체인(1012)의 일부분을 형성할 수 있다. 프로그램 가능한 테스트 클록 제어기(1000)는 클록 명령 정보(1012)의 일부분의 함수로서 테스트 클록("CLK";1052)을 발생시키도록 구성된 테스트 클록 발생기(1030)도 포함한다. 일부 경우들에서, 클록 명령 정보(1012)는, 기능 클록 신호를 스캔 체인에 대한 테스트 클록(1052)으로서 인가할 것인지의 여부를 특정한다. 프로그램 가능한 테스트 클록 제어기(1000)는, PLL 회로로 입력될 수 있는 기준 클록("Ref Clk";1046)을 수신한다. 기준 클록(1046)의 위상은 하나 이상의 PLL 회로들로부터의 하나 이상의 PLL 출력 클록들로 로킹될 수 있다.

<68> 일 실시예에서, 클록 펄스 제어기(1002)는 클록 제어 신호 발생기(1010) 및 클록 속도 선택기(1020)를 포함한다. 클록 제어 신호 발생기(1010)는 클록 명령 정보(1012)에 기초해 각각의 클록 예지에서 클록 제어 신호(1024)를 발생시키는 한편, 클록 속도 선택기(1020)는 개개 클록 예지들 각각을 위한 클록 사이클 주기를 설

정한다. 클록 속도 선택기(1020)는 옛-스피드 테스트를 위한 제1 클록 속도를 지시하는 제1 주기와 적어도 스캔 체인을 구동하기 위한 제2 클록 속도를 지시하는 제2 주기 사이에서 선택한다. 테스트 클록 발생기(1030)는 테스트 설정 및 실행 제어기(1040)와 테스트 클록 선택기(1050)를 포함할 수 있다. 테스트 설정 및 실행 제어기(1040)는 프로그램 가능한 테스트 클록 제어기(1000)를 데이터 전달 모드 또는 테스트 실행 모드로 구성하도록 동작한다. 또한, 테스트 설정 및 실행 제어기(1040)는, 개개 클록 예지들이, 예를 들어, 테스트 클록(1052)으로서 시프팅되는 클록 속도를 선택하기 위한 클록 속도 제어 신호("CPC_En";1022)를 제공한다. 테스트 클록 선택기(1050)는 클록 제어 신호(1024)에 기초해 기능 클록 신호(1054) 또는 개개 클록 예지들의 서브세트를 스캔 체인에 인가하도록 구성된다. 클록 제어 신호(1024)는 테스트 클록(1052)을 형성하는 기능 클록 신호(1054)를 위한 개개 클록 예지들의 프로그램 가능한 양을 특정할 수 있다는 것에 주의한다. 하나 이상의 실시예에서는, 테스트 클록 선택기(1050)도 스캔 클록("SCLK";1052)을, 스캔 데이터를 스캔 체인으로 그리고 스캔 체인으로부터 시프팅하기 위한 테스트 클록(1052)으로서 제공한다. 일 실시예에 따르면, 테스트 클록 발생기(1030)는 2 이상의 입력들을 포함한다. 제1 입력은 동작중인 프로그램 가능한 테스트 클록 제어기(1000)가 데이터 전달 모드인지 아니면 테스트 실행 모드인지를 지시하는 스캔 인에이블 데이터("SeD") 신호(1042)를 수신하도록 구성되는 한편, 제2 입력은 프로그램 가능한 테스트 클록 제어기(1000)를, 기능 클록 신호("PCLK";1054)를 아니면 개개 클록 예지들의 서브세트를 스캔 체인에 인가하도록 구성하는 것을 지시하는 스캔 인에이블 클록("SeC") 신호(1044)를 수신하도록 구성된다. 프로그램 가능한 테스트 클록 제어기(1000)는, 스캔 인에이블 데이터 신호(1042)가 데이터 전달 모드를 표현하면, 스캔-스피드 클록 속도를 사용해 스캔 체인이나 제어 체인으로 데이터를 시프팅하도록 동작한다. 또는, 프로그램 가능한 테스트 클록 제어기(1000)는, 스캔 인에이블 데이터 신호(1042)가 테스트 실행 모드를 표현하면, 스스로를 옛-스피드 클록 속도를 사용해 기능 테스트들을 수행하도록 구성하도록 동작한다. 그러나, 스캔 인에이블 클록 신호(1044)는 프로그램 가능한 테스트 클록 제어기(1000)가 (예를 들어, 누설 IDDQ 테스트들을 수행하기 위한) 사일런트 상태에서 아니면 옛-스피드 개시 및/또는 포착에서 동작하게 한다. 도 10의 프로그램 가능한 테스트 클록 제어기는 논의의 간략화를 위해 스캔 클록 디코더를 생략한다는 것에 주의한다.

<69> 도 11은 발명의 실시예에 따른 다수의 스캔 인에이블 제어 신호들에 기초해 프로그램 가능한 테스트 클록 제어를 위한 동작의 상태들을 예시하는 도표이다. 도표(1100)는, 스캔 인에이블 데이터(SeD) 제어 신호(1101) 및 스캔 인에이블 클록(SeC) 제어 신호(1103)가, 발명의 다양한 실시예들에 따른, 프로그램 가능한 테스트 클록 제어기의 소정 기능들을 구현하는데 사용될 수 있는 신호 상태들(1105;또는 테스트 모드들)의 세트를 생성한다는 것을 나타낸다. 사일런트 상태(1102) 동안, CUT로의 모든 클록들은 차단되는 한편, 개시/포착 상태(1104)에서는, 옛-스피드 기능 클록이 스캔 체인으로 인가되어 지연 테스트를 실행한다. 스캔 로드/언로드 체인들의 상태(1106)에서는, 스캔 체인들이 로드되거나 언로드될 수 있고, 제어 로드/언로드 상태(1108)에서는, 프로그램 가능한 테스트 클록 제어기가 프로그램될 수 있다. 일반적으로, 스캔 인에이블 데이터 제어 신호(1101)의 값은, 프로그램 가능한 테스트 클록 제어기가 데이터 전달 모드에서 아니면 테스트 실행 모드에서 동작하는지의 여부를 판정한다. SeD 제어 신호(1101)가 1의 값을 갖는 경우, 프로그램 가능한 테스트 클록 제어기는, ATE 디바이스와 같은, 외부 소스로부터의 스캔 클록을 사용해 비교적 낮은 스피드에서 데이터 전달 동작들에 종사한다. 그 다음, SeC 제어 신호(1103)가 0의 값을 가지면, 프로그램 가능한 테스트 클록 제어기는 스캔 체인을 로드하거나 언로드하여 회로의 내부 레지스터들을 공지 상태로 초기화한다. 대조적으로, SeC 제어 신호(1103)를 위한 1의 값은, 프로그램 가능한 테스트 클록 제어기가 클록 명령 정보로써 프로그램되게 한다. 사일런트 상태(1102)에서, 프로그램 가능한 테스트 클록 제어기는, CUT의 도메인들에 전이들을 발생시키기 위한 클록 펄스들을 인가하지 않는다. 이 상태에서, ATE는 일차 입력들(PI들)을 강제하여 일차 출력들(PO들)을 측정하거나 전원 공급 장치로부터 그라운드에 이르는 IDDQ 누설 전류들을 측정할 수 있다. SeD 제어 신호(1101)가 0의 값을 갖는 경우, 프로그램 가능한 테스트 클록 제어기는, 기능 클록을 사용하는 비교적 높은 스피드들에서, 입력 전이를 개시하기 위한, 테스트 응답을 포착하기 위한, 또는 양자를 위한 옛-스피드 테스트들에 종사할 수 있다.

<70> 도 12a 및 도 12b는, 각각, 발명의 다양한 실시예들에 따른, 스캔 클록 디코더 및 프로그램 가능한 테스트 클록 제어기의 특정 구현들을 예시한다. 도 12에서, 스캔 클록 디코더(1200)는 스캔 체인을 구동하기 위한 스캔 클록(SCLK;1208) 및 프로그래밍 목적들을 위한 제어 계층 클록(TCLK;1210)을 발생시키기 위한 로직을 포함하는데, SCLK 및 TCLK 양자는 클록 제어 신호들이다. 로직 게이트들(1201)은 일차 스캔 클록(ScanClk;1206), 스캔 인에이블 클록 제어 신호(SeC;1204), 및 클록 제어 신호(TCEn;1202)를 사용해 클록 제어 신호들(SCLK, TCLK)을 형성한다. 일 실시예에서는, ATE가 일차 스캔 클록(1206) 및 스캔 인에이블 클록 제어 신호(1204)를 발생시켜 도 12b의 프로그램 가능한 테스트 클록 제어기에 인가할 수 있다. 일부 실시예들에서는, 스캔 인에이블 데이터 제어 신호(SeD;1211)가 로직 게이트들(1201a)의 동작을 제어하여, 일반적으로 외부 테스트 또는 ATE에 의해 제공

되는 일차 스캔 클록(ScanClk;1206)의 상태와 무관하게, 클록 제어 신호들(SCLK 및 TCLK)의 형성을 디스에이블할 수 있다(예를 들어, 논리 0의 SeD).

- <71> 도 12b에서, 프로그램 가능한 테스트 클록 제어기(1220)는 클록 펄스 제어기(CPC;1230) 및 테스트 클록 발생기(TCG;1250)를 포함한다. 클록 펄스 제어기(1230)는 클록 발생기(1250)를 테스트하기 위한 TCEn 신호(1202)의 값들로서 클록 에지들의 스트림을 제공하는데, 클록 발생기(1250)는 클록 에지들의 값들을 사용해 앳-스피드 테스트 클록을 형성한다. 일 실시예에서, 클록 펄스 제어기(1230)는, 각각, CSI(control scan in) 터미널(1236) 및 CSO(control scan out) 터미널(1238)을 통해, 직렬로 로딩 및 언로딩될 수 있는 2-비트 시프트 레지스터(SR;1234)를 포함한다. 동작시에, 2-비트 시프트 레지스터(1234)는 TCEn 신호(1202)에 개시 및/또는 포착 동작들 동안 기능 클록(PCLK;1252)에 기초해 2개까지의 앳-스피드 펄스들을 발생시키기 위한 제어 값들을 제공한다. 하나 이상의 실시예에서, 2-비트 시프트 레지스터(1234)는 테스트 클록 발생기(1250)로 시프팅될 개개 클록 에지들의 프로그램된 양을 홀딩한다. TCEn 신호(1202)는, 클록 펄스 제어기 인에이블(CPCEn) 신호(1240)가 논리 1의 값을 갖기만 한다면, 개시/포착 상태에서, 기능 클록(PCLK;1252)에 기초한 "스피드에서(at speed)" 시프팅한다. 그러나, 2-비트 시프트 레지스터(1234)는, 제어 체인 로드/언로드 동작들 동안, 도 12a의 비교적 느린 일차 스캔 클록(ScanClk;1206)에 기초한 "스캔-스피드(scan-speed)"에서 시프팅한다.
- <72> 2-비트 시프트 레지스터(1234)는 사일런트 상태 뿐만 아니라 스캔 로드/언로드 상태 동안 클록 펄스들로 클로킹되지 않으므로, 거기에 저장된 클록 명령 정보는 이 상태에서 변경없이 유지된다는 것에 주의한다. 일부 실시예들에서, 클록 펄스 제어기(1230)는 개시 및/또는 포착 동작들에 참여하기 위한 "i-번째" 클록 도메인을 선택적으로 인에이블하기 위한 클록 도메인 인에이블(CDEn[i]) 제어 신호(1244)를 발생시킨다. 다른 실시예들에서, 클록 펄스 제어기(1230)는, 예를 들어, 스트림 발생기 입력(SGenIn;1248)으로부터 보충 클록 명령 정보를 수신하기 위해 (도시되지 않은) 보충 클록 명령 소스에 ICLK 신호(1246)를 제공한다. 보충 클록 명령 소스는 2-비트 시프트 레지스터(1234)의 기능성을 2개 비트들 이상으로 확장한다.
- <73> 테스트 클록 발생기(1250)는 테스트 클록(CLK;1254)을 구성하여, 기능 클록(1252)의 일부분들 및 스캔 클록(SCLK;1208)의 일부분들로 이루어진 소정 파형을 제공하도록 설계된다. 특히, 테스트 클록 발생기(1250)는 스캔 클록(SCLK;1208)과 기능 클록(PCLK;1252) 사이를 멀티플렉싱한다. 일부 실시예들에서, 기능 클록(1252)은 TCEn 신호(1202)와 논리적으로 AND되어 테스트 클록(1254)을 발생시킬 수 있다. 테스트 클록 발생기(1250)는 SeD 신호(1258) 및 SeC 신호(1260)를 사용해 기능 클록(1252)이나 스캔 클록(1208)을 클록들의 소스들로서 선택한다. 더 나아가, 테스트 클록 발생기(1250)는 이 신호들을 사용해, 테스트 클록(CLK[i];1254)을 크래프팅하기 위한 TCEn 신호(1202)를 발생시킨다. 동작시에, 테스트 클록 발생기(1250)는, 개개 클록 에지들이 테스트 클록(1254)으로서 등장하는 클록 속도를 선택하는데 사용되는 클록 속도 제어 신호인 클록 펄스 제어기 인에이블 신호(1240)를 통해, 클록 펄스 제어기(CPC;1230)와 테스트 클록들을 동기한다.
- <74> 일부 경우들에서, 테스트 클록 발생기(1250)는 억제 능력(suppression capability)을 갖춘 스트림 멀티플렉서로서 간주될 수 있다. 다시 말해, 그것은 TCEn 신호 값들(즉, 개개 클록 에지들의 값들)의 스트림과 AND된 소스 클록 신호 파형들을 멀티플렉싱하여 테스트 클록(1254)을 위한 파형을 생성한다. 더 나아가, SeC 신호(1260)와 SeD 신호(1258) 양자의 상태들은, 소스 클록들 중 어떤 것이 TCEn 신호 값들의 스트림과 AND될 것인지를 선택한다. TCEn 신호(1202)가 0의 값을 가지면, 소스 클록 펄스는 억제된다. 그렇지 않으면, 소스 펄스는 계속 통과되어 테스트 클록(1254)의 일부를 형성한다. 더 나아가, 테스트 클록 발생기(1250)는, 개시 및/또는 포착 동작들을 개시할 때, 그것이 "차단되거나" 낮은 논리 값을 갖는 정상 상태 위상에서 스캔 클록(SCLK;1208)을 선택하는 것에 의해, 기능 클록 경로와 비트식-AND 연산을 수행하는 것으로부터 발생하는 가능한 클록 스큐를 억제한다. 정상 상태 위상의 일레가 도 9의 정상 상태 위상(908)이다.
- <75> 도 12b에 도시된 일례에서, 테스트 클록 발생기(1250)는 샘플 및 홀드 레지스터(SHR;1270)와 클록 선택 레지스터(CSR;1272)를 포함한다. 샘플 및 홀드 레지스터(1270)는 기준 클록(RefClk[k];1262)을 사용해 SeC 신호(1260)를 샘플링할 뿐만 아니라 테스트 클록 속도 선택을 위한 클록 펄스 제어기 인에이블(CPCEn) 신호(1240)를 클록 펄스 제어기(1230)에 제공한다. CPCEn 신호(1240)가 1의 값을 가지면, 멀티플렉서(1242)는 기능 클록(PCLK;1252)을 선택한다. 0의 값으로써, 멀티플렉서(1242)는 클록 명령 정보 비트들을 시프팅하기 위한 제어 계층 클록(TCLK;1210)을 선택한다. 샘플 및 홀드 레지스터(1270)는, SeD 신호(1258)가 0의 값을 가지며 SeC 신호(1260)가 1의 값을 가지면, 1의 논리 값을 발생시킨다. 그렇지 않으면, 샘플 및 홀드 레지스터(1270)의 출력력(1271)은 논리 0으로 유지된다. 일부 실시예들에서는, 샘플 및 홀드 레지스터(1270)가 클록 도메인간 테스트들을 수행하기 위한 동기화 타이밍 기준점도 제공할 수 있다.

<76> 제어 로직은 제어 계층 클록(TCLK;1210)에 의해 클로킹되므로, 샘플 및 홀드 레지스터(1270)는 제어 계층에서의 제어 로직에 대해 고착 테스트들을 수행하는 것도 지원한다. 고착 테스트들을 구현하기 위해, RefClk(1262)는 차단된다. 샘플 및 홀드 레지스터(1270)는 SeD 신호(1258)에 의해 리셋되고 정적 테스트 동안 리셋 상태를 유지한다. 샘플 및 홀드 레지스터(1270)의 리셋 값은 정적 테스트 동안 스캔 클록(SCLK;1208)을 선택한다. 샘플 및 홀드 레지스터(1270)는 SeC 신호(1260)가 클록 선택 레지스터(1272)로 전파하는 것을 방지하므로, SeC 신호(1260)는, 고착 테스트들을 수행하기 위해, 클록 제어 신호들(TCLK(1210) 및 SCLK(1208))을 멀티플렉싱하는데 사용될 수 있다. 이와 같이, 프로그램 가능한 테스트 클록 제어기의 개시/포착 상태는 제어 로직의 고착 테스트 동안의 포착에 사용될 수 있는 한편, 스캔 로드/언로드 상태는 데이터 스캔 체인들에 액세스하는데 사용될 수 있다. 마찬가지로, 사일런트 상태는 포착을 위해 사용될 수 있는 한편, 제어 로드/언로드 상태는 CUT의 고착 테스트 동안 제어 체인들에 액세스하는데 사용될 수 있다.

<77> 클록 선택 레지스터(1272)는, 테스트 클록(1254)의 구성 가능한 일부분들을 형성하기 위한 소스 클록을 선택하는 클록 선택 제어 정보를 저장하도록 구성된다. 클록 선택 레지스터(1272)의 클록 선택 제어 정보는 샘플 및 홀드 레지스터(1270)와 TCEn 신호(1202) 양자의 상태들에 의해 판정된다. 정상적인 기능 모드에서, 테스트 모드 신호(TM;1280)는 클록 선택 레지스터(1272)를 리셋한다. 이와 같이, 기능 클록(1252)이 테스트 클록(1254)에 접속된다. 클록 선택 레지스터(1272)의 상태는, PWD(power down signal;1282)가 인가되지 않는 한, 기능 모드 동안 변함없이 유지된다. PWD(1282)는 정상적인 기능 모드 동안 대응되는 클록 도메인의 동작을 정지시킬 수 있다. 클록 도메인의 전력을 감소시키는 것은 저전력 회로 애플리케이션들을 테스트하기에 바람직한 사양일 수 있다. 그런데, PWD(1282)는 다양한 실시예들을 위한 구조들을 테스트하기 위한 향상된 기능성을 제공할 수 있다. 특히, PWD(1282)는, 기능 클록 경로를 따라 간섭하는 추가적인 논리 게이트들을 삽입하지 않으면서, 클록 게이팅을 수행하기 위한 테스트 클록 제어기로 통합된다. PWD 신호(1282)가 테스트 클록(1254)을 위해 기능 클록을 차단하도록 설정되면, 멀티플렉서(1255)는, 정상적인 기능 모드 동안 비활성인 스캔 클록(SCLK;1208)을 선택하고, 그에 따라, 클록 펄스들이 테스트 클록(1254)으로서 발생되지 않는다. 더 나아가, PWD 신호(1282)는 기능 테스트들과 실리콘 디버그의 양자에서 클록-정지 회로로서 사용될 수도 있다. 이와 같이, 문제의 소지가 있는 클록 도메인은, (도시되지 않은) 매립된 내부 모니터 회로가 시스템 장애를 검출하는 경우, 바람직스럽게도 차단될 수 있다. 장애가 검출되면, 대응되는 클록 도메인은, 그것이 더 이상 전파되는 것을 방지하기 위해 정지될 수 있다. 이것은, 특히, 장애들을 재빨리 검출하기 위한 제어 메커니즘들이 결여된 비교적 긴 스캔 체인들을 포함하는 전통적인 스캔 체인 구조들을 사용하는 디버깅 회로들에 비해, 디버깅 프로세스를 간략화한다. 대체로, 수개 클록 사이클들을 초과하는 장애들이나 지연 검출을 불명료하게 하는 것은, 전통적인 스캔 체인들의 길이 및 간섭하는 로직으로부터의 출력들이다. 대조적으로, 발명의 다양한 실시예들은 PWD(1282)를 이용해, 예를 들어, 2개의 또는 더 적은 클록 사이클들내에서 장애들을 재빨리 검출하기 위해 테스트를 정지시킨다. PWD(1282)를 생성하기 위한 PWD 신호 발생기들은 저전력 회로 애플리케이션들을 제공하기 위한 회로 설계 업계에 널리 공지되어 있으므로 상세하게 논의될 필요가 없다는 것에 주의한다.

<78> 도 13은, 발명의 일 실시예에 따른, 특정 클록 도메인을 프로그램하기 위한 도 12b의 2-비트 시프트 레지스터의 일례를 위한 상태를 묘사한다. 2-비트 시프트 레지스터(1300)를 로딩하는 것에 의해, 개개 클록 도메인은 스캔 로드 및 언로드 동작들을 위해서 뿐만 아니라 개시 및 포착 동작들을 위해서도 프로그램될 수 있다. 상태도(1310)는, 도 12a 및 도 12b의 TCEn 신호(1202)가 2-비트 시프트 레지스터(1300)내에 저장되어 있는 프로그램된 상태로부터 발생하는 방법을 요약한다. 상태도(1310)는, 보충 클록 명령 소스로부터의 스트링 발생기 입력(SGenIn;1302)이 사용되지 않는다(즉, SGenIn 터미널이 0의 값으로 설정된다)고 가정한다. 상태(1312)로부터 상태(1318)에 이르는 상태 각각은, 전이가 "SGenIn/TCEn" 표기법으로써 레이블링되는, 2-비트 시프트 레지스터(1300)의 내용들을 표현한다. 보충 클록 명령 소스의 사용을 배제하는 입력 제한(즉, SGenIn=0)으로 인해, 점선들(1320)로서 지시되는 일부 전이들은 인가 불가능하다는 것에 주의한다. 2-비트 시프트 레지스터(1300)의 초기 상태들은 개시/포착 상태(1312)(SR의 내용들은 "11"), 개시만의 상태(1314)(SR의 내용들은 "10"), 포착만의 상태(1318)(SR의 내용들은 "01"), 및 디스에이블 상태(1316)(SR의 내용들은 "00")를 포함한다. 합산에서, 2-비트 시프트 레지스터(1300)에서의 1들의 수는 테스트 클록(CLK[i])으로서 통과될 수 있는 기능 클록 펄스들의 수에 대응된다.

<79> 예를 들어, 2-비트 시프트 레지스터(1300)가 "11"의 내용들을 포함하면, 2개의 연속적인 기능 클록 펄스들이 발생되어 테스트 클록을 형성하는데, 일 펄스 에지는 개시 동작을 위한 것이고 다른 펄스 에지는 포착 동작을 위한 것이다. 개시/포착 상태(1312)가 종료한 후, 2-비트 시프트 레지스터(1300)는 "00"의 내용들을 가진 디스에이블 상태(1316)에 도달함으로써, 테스트 클록(CLK)을 디스에이블한다. 구체적으로, 디스에이블 상태(1316)는

스캔 클록(SCLK)을 디스에이블한다. SCLK는 개시/포착 동안 부동일 수 있다는 것에 주의한다. 마찬가지로, 2-비트 시프트 레지스터(1300)가 "10"이나 "01"의 값들을 포함하면, 하나의 기능 클록 펄스가 개시만의 상태(1314) 및 포착만의 상태(1318)를 위해 발생한다. 개시 및 포착 클록 펄스 에지들 사이의 차이점은 그 클록 에지들의 시간적 변위(temporal displacement)이다. 예를 들어, 개시 클록 펄스는 일반적으로, 적어도 하나의 기능(PCLK) 클록 주기만큼 포착 클록 펄스를 선행한다.

<80> 스캔 로드 및 언로드 동작들을 구현하기 위해, 2-비트 시프트 레지스터(1300)의 초기 상태는, 테스트 클록이 스캔 클록 펄스들을 포함하는지의 여부를 판정하는 TCEn 신호 값들을 제공한다. 2-비트 시프트 레지스터(1300)가 "1x"의 내용들을 포함하면, 스캔 로드 및 언로드 동작들을 위해 스캔 클록이 인에이블되는데, 여기에서, x는 "상관없음(don't care)"을 지시한다. 그러나, 내용들이 "0x"라면, 스캔 클록은 디스에이블되고, 펄스 에지들이 스캔 체인으로 인가되지 않는다. TCEn 신호의 값은 일반적으로, 스캔 로드/언로드 동안 불변이라는 것에 주의한다. 일 실시예에서, 모든 스캔 체인들이 선택적으로가 아니라 한번에 로딩/언로드되어야 한다면, SCLK 경로의 AND 게이트는 제거될 수 있다.

<81> 도 14a 및 도 14b는, 각각, 발명의 다양한 실시예들에 따른, 개시 및 포착 동작들을 수행하기 위한 그리고 그것이 대응되는 클록 도메인을 디스에이블하기 위한 테스트 클록을 구성하도록 프로그램 가능한 테스트 클록 제어기를 구성하기 위한 신호들의 타이밍을 예시한다. 도 14a의 타이밍도(1400)는, 프로그램 가능한 테스트 클록 제어기가 스캔 클록(SCLK) 신호 및 기능 클록(PCLK) 신호(1402)를 사용해 개시 및/또는 포착 동작들을 위한 구성 가능한 테스트 클록 파형(CLK;1410)을 발생시키는 방법을 예시한다. 다른 신호들(1420)은 도시된 상태들로 초기화된다고 가정한다. CPCEn 신호(1406)는, 스캔 로드 또는 언로드 동작 이후에 논리 0의 값으로 설정된다. CPCEn 신호(1406)의 값은 출력 신호(Mux.SO;1408)의 값이 논리 1일 것을 강제한다. SeC 신호(1401) 및 SeD 신호(1430) 양자가 논리 0의 값으로 설정될 때, CPCEn 신호(1406)는 논리 0의 값을 유지함으로써, 클록 선택 레지스터(CSR) 출력 신호(Mux.SO;1408)를 논리 1의 값으로 홀딩한다. 이와 같이, CSR 출력 신호(1408)는, 멀티플렉서가 스캔 클록(SCLK[i];1412)을 선택하게 하고, 스캔 클록(SCLK[i];1412)은 스캔 모드 구간(1413) 동안 테스트 클록(1410)을 형성한다. 다음으로, 클록 펄스 제어기는, TCEn 신호(1404)가 인에이블될 때(즉, 1의 값일 때), 제어 체인으로부터의 로드 동작 동안 그것의 시프트 레지스터에 "11"을 포함하도록(즉, SR=11이도록) 프로그램된다고 가정한다. CSR의 상태는, SeC 신호(1401)가 1의 논리 값으로 전이할 때까지, CPCEn 신호(1406)를 위해 계속해서 논리 0의 값을 발생시킨다. 그 다음, 기능 클록 신호(1402)의 제1 하강 에지(1490)에서, SeC 신호(1401)가 샘플 및 홀더 레지스터로 샘플링된 후, CPCEn 신호(1406)를 위한 에지(1407)에서 논리 1의 값이 클록 선택 레지스터로 포착된다. 클록 선택 레지스터 내용들의 상태가 CSR 출력 신호(1408)를 1의 논리 값에서 0으로 변화시키면, 멀티플렉서는 테스트 클록(1410)으로서 기능 클록(PCLK[i];1414)을 선택한다. 기능 모드 구간(1415) 동안, 2개의 연속적인 기능 클록 펄스들(1419)이 TCEn 신호(1404)에 의해 개시 클록 펄스 및 포착 클록 펄스로서 인에이블된다. 각각의 기능 클록 펄스(1419)를 위해, 0의 논리 값이 (예를 들어, 논리 0으로 설정되어 있는 SGenIn 입력을 통해) 시프트 레지스터로 시프팅된다. 시프트 레지스터가 디스에이블 상태에 도달하면(즉, 그것의 내용들이 "00"의 값을 포함하면), TCEn 신호(1404)는 디스에이블되고 에지(1421)에서 0의 논리 값으로 떨어진다. 더 나아가, CSR 출력 신호(1408)는 에지(1423)에서 상태를 변경함으로써, 프로그램 가능한 테스트 클록 제어기를 구간(1417) 동안 스캔 코드로 복귀시켜 스캔 클록(SCLK[i];1416)을 테스트 클록(1410)으로서 출력한다.

<82> 스캔 클록(1412)은 일반적으로, SeD 신호가 논리 0의 값일 경우, 0의 정상 상태 값으로 유지된다는 것에 주의한다. 프로그램 가능한 테스트 클록 제어기는, 그것이 논리 0의 정상 상태 값을 가질 경우, 기능 클록(1402)의 듀티 사이클 대략 중간(1430)에서 멀티플렉서 출력(즉, 출력 신호(1408))의 전이가 발생하는 것을 목표로 삼는다. 바람직스럽게도, 정상 상태 위상(1432) 중간(1430)에서 테스트 클록(1410)을 스위칭하는 것은, 제조시에 발생할 수 있는 프로세스 변경들에 대한 허용 범위를 제공함으로써, 타이밍 불확실성들을 감소시킬 뿐만 아니라 기능 클록 듀티 사이클들을 변경없이 유지한다.

<83> 도 14b의 타이밍도(1450)는, 발명의 일 실시예에 따른, 프로그램 가능한 테스트 클록 제어기가 클록 도메인을 디스에이블하기 위한 구성 가능한 테스트 클록 파형을 발생시키는 방법을 예시한다. 다른 신호들(1470)은 도시된 상태들로 초기화되는 것으로 가정한다. 이와 같이, 시프트 레지스터(SR)에는 "00"의 내용들이 로딩되고, 그것에 의해, 논리 0의 TCEn 신호(1454) 값들을 발생시킨다. 결과적으로, 클록 선택 레지스터는 1의 논리 값을 가진 CSR 출력 신호(1458)를 발생시키고, 그것에 의해, 스캔 클록(SCLK[i];1462)을 테스트 클록(1460)으로서 선택한다. 따라서, 스캔 펄스들도 기능 펄스들도 테스트 클록을 위해 발생되지 않는다.

<84> 도 15는, 발명의 일 실시예에 따른, 간략화된 클록 펄스 제어기 및 간략화된 테스트 클록 발생기를 포함하는 프

로그래밍 가능한 테스트 클록 제어기를 묘사한다. 프로그램 가능한 테스트 클록 제어기(1500)는 간략화된 클록 펄스 제어기(1502) 및 간략화된 테스트 클록 발생기(1520)를 포함하는데, 이들 양자는 도 12b에서 설명된 유사하게 명명된 신호들과 상호 작용한다. 프로그램 가능한 테스트 클록 제어기(1500)가 도 12b의 프로그램 가능한 테스트 클록 제어기와 유사하게 거동하지만, 그것은 구체적으로 개시 및/또는 포착 동작들 및 순차적 테스트 능력들을 제공하도록 구성된다. 이 경우, 간략화된 클록 펄스 제어기(1502)는 단일 비트를 저장하고 제공하기 위한 1-비트 시프트 레지스터(CPCR; 1504)를 포함한다. 예를 들어, 프로그램 가능한 테스트 클록 제어기(1500)는, $1[1*00]$ 또는 $0[x*]$ 와 같은, TCEn 신호(1506) 값들의 대응되는 스트림을 제공할 수 있는데, 이 경우, 1-비트 시프트 레지스터(1504)의 초기 상태는 제1 비트만을 포함한다(즉, 괄호들 사이의 비트들은 외부적으로 공급된다). 이 경우, 클록 속도 제어 신호(예를 들어, CPCEn 신호)가 SeD(1521)로부터 유도된다. 신호 SeD(1521)는, 스캔 로드/언로드 이후에 CLK[i]을 선택할 것을 MUX(1508)에 강제한다. 그러나, CLK[i]는 일반적으로, CSR(1524)의 출력이 CLK[i]를 선택하도록 설정될 때까지 부동이다. 이것은, 신호 SeC(1523) 값에서의 논리 0로부터 논리 1로의 변화에 의해 초래된다. 프로그램 가능한 테스트 클록 제어기(1500)는 1-비트 시프트 레지스터(1504)와 클록 선택 레지스터(1526) 양자를, 개시 및 포착 동작들을 위해 TCEn 신호(1506) 값들을 수용하기 위한 효과적인 2-비트 시프트 레지스터(SR)로서 구현할 수 있다. 클록 선택 레지스터(1526)가 클록 명령 정보 로딩 및 앳-스피드 테스트 양자를 위한 테스트 클록을 판정한다는 것에 주의한다. 동작시에, 1-비트 시프트 레지스터(1504)의 내용은 개시 또는 포착 동작들 이전에 클록 선택 레지스터(1526)로 복사된다. 1-비트 시프트 레지스터(1504)를 시프팅하는 것에 의해, TCEn 신호(1506)의 값은 출력(1524)을 거쳐 멀티플렉서(1528)에 인가된다. 다른 실시예에서, 간략화된 테스트 클록 발생기(1520)는, 동기식 샘플-및-홀드가 정확하거나 안전한 테스트 동작들을 보장하기에 충분하지 않을 수도 있는 경우들을 수용하기 위해 2-비트 동기화 장치(1522)를 포함한다. 도시되지는 않았지만, 기준 클록(RefClk[k]) 역시, 기능 클록(PCLK[i]; 1530) 대신, 동기화 장치(1522)를 구동하는데 사용될 수 있다.

<85> 도 16은, 발명의 일 실시예에 따른, 보충 클록 명령 소스를 포함하는 프로그램 가능한 테스트 클록 제어기를 묘사한다. 도시된 바와 같이, 프로그램 가능한 테스트 클록 제어기(1600)는 발생된 기능 클록 펄스를 프로그램 가능한 시간량만큼 억제 또는 지연하기 위한 보충 클록 명령 소스(SCCS; 1602)에 연결된다. 보충 클록 명령 소스(1602)는, 예를 들어, 개개 클록 에지들을 위한 클록 제어(TCEn) 신호 값들로서 소정의 이진 비트 스트림을 발생시키는 유한 상태 머신(finite state machine)일 수 있다. 다시 도 12b를 참조하여, SGenIn 신호(1248)가 한정되지 않으면(즉, SGenIn 신호(1248)가, 그렇지 않다면, 2개 비트들을 시프팅한 이후에 프로그램 가능한 테스트 클록 제어기를 디스에이블하는 정상 상태의 낮은 값으로 홀딩되지 않으면), 보충 클록 명령 소스(1602)에 의해 발생하는 비트 스트림이 다중-사이클 경로 테스트를 용이하게 할 수 있다. 이와 같이, 포착 클록 에지는 특정한 개시 클록 에지 이후에 다수의 기능 클록 펄스들을 억제하는 것에 의해 다수 사이클들만큼 지연될 수 있다. 도 13을 참조하여, 다중-사이클 클로킹 방식은, 2-비트 시프트 레지스터(1300)를 위한 초기 상태로써 먼저 개시만의 (즉, "10"으로 프로그램된 내용들을 가진) 상태(1314)로부터 시작하는 것에 의해 구현될 수 있다고 가정한다. 그 다음, $10[0*100*]$ 의 도 16의 TCEn 신호 값들(1604)이 지연 테스트를 위한 다중 사이클 경로를 지원할 수 있다. 보충 클록 명령 정보($[0*100*]$; 즉, "10"의 처음 2개 비트들 이후의 나머지 시퀀스)가 보충 클록 명령 소스(1602)에 의해 공급될 수 있다.

<86> 바람직스럽게도, 보충 클록 명령 소스(1602)는 다중 사이클 경로 테스트 동작들을 위한 $10*100*$, 순차적 테스트 동작들을 위한 $1*00*$, 및 개시만의 또는 포착만의 동작들을 위한 $0*100*$ 의 TCEn 값들을 발생시켜, 대응되는 다중 사이클 경로 테스트 동작들, 순차적 테스트 동작들, 및 지연된 개시만의 동작들 또는 포착만의 동작들을 구현한다. TCEn 값들(1604)의 처음 2개 비트들은 클록 펄스 제어기에 저장될 수 있다. "x*"의 제1 발생에는 "x"의 보수(complement)가 수반되고, 여기에는 다시 "00*"가 수반된다는 것에 주의한다. 마지막 스트링 "00*"은 테스트 액션들의 끝에서 기능 클록을 디스에이블하고, 그것에 의해, 프로그램 가능한 테스트 클록 제어기(1600)를 디스에이블 상태화한다. 일부 실시예들에서, 보충 클록 명령 소스(1602)는 TCEn 신호(1604)를 위해 k-비트에 이르는 값들(또는 개개 클록 에지들)을 제공하기 위한 k-비트 시프트 레지스터일 수 있다. 그러나, k가 비교적 크다면, 하드웨어 오버헤드를 감소시키기 위해 스트링 발생기가 이용될 수도 있다.

<87> 도 17a 및 도 17b는, 각각, 발명의 다양한 실시예들에 따른, 스트링 발생기로서 구현된 보충 클록 명령 소스 및 대응되는 상태도를 예시한다. 일 실시예에서, 스트링 발생기(1700)는 보충 클록 명령 정보를 표현하는 데이터 비트들의 스트링을 발생시키는데, 스트링은 일반적으로, 예를 들어, k 비트 레지스터들에 k개 데이터 비트들을 저장하는데 요구되는 메모리 소자들의 양보다 긴 비트 길이를 가진다. 스트링 발생기(1700)는 제1 펄스 값 및 제2 펄스 값을 저장하도록 구성된 비트-홀딩 유닛(1720) 및, 제1 펄스 값과 제2 펄스 값 사이에서 다수의 간섭 클록 사이클들이 시프팅된 이후에 특수 링 카운터에서의 MSB(most significant bit)의 논리 전이를 실현하기 위

한 특수 링 카운터(1720)를 포함한다. 예를 들어, 비트-홀딩 유닛(1710)은 TCEn 신호 값들의 스트림에서 0이나 1의 논리 값을 "n"회 발생시킬 수 있다. 특수 링 카운터(1720)는 발생될 펄스들의 수를 추적한다. 특수 링 카운터(1720)의 초기 내용은 CSI(1711) 및 CSO(1712)를 통해 직렬로 로딩 또는 언로딩될 수 있다. 개시 및 포착 동작들 동안, 특수 링 카운터(1720)는, 제로 상태(zero state)에 도달될 때까지 카운팅을 계속한다.

<88> 동작시에, 비트-홀딩 유닛(1710)은, 논리 0의 값을 가진 Zero 신호(1716)가 터미널(1714)에 도달할 때까지, 초기화된 값을 홀딩한다. 이와 같이, Zero 신호(1716)는, 특수 링 카운터(1720)가 제로 상태에 도달하였는지의 여부를 지시한다. 레지스터(G1;1702) 및 레지스터(G0;1704)는, 예를 들어, 개시 및/또는 포착 동작이 진행하기 전에, 상보적인 논리 값들로써 초기화된다. 상보적인 논리 값은 카운트의 끝에서 (도시되지 않은) 클록 펄스 제어기로 출력된다. 특수 링 카운터(1720)는, 그것이 리로딩될 때까지 제로 상태를 유지한다. 예시를 위해, Zero 신호(1716)는 논리 1의 값을 가지며 레지스터들(1702 및 1704)은 상보적인 논리 값들을 포함한다고 가정한다. 더 나아가, XOR-게이트(1718)는 레지스터(1704)의 내용들을 계속해서 변경하기 위해 추가적인 반전을 도입한다. 다음으로, Zero 신호(1716)는 논리 0의 값을 가지며, 그것에 의해, 비트-홀딩 유닛(1710)을 2-비트 시프트 레지스터로서 구성한다고 가정한다. Zero 신호(1716)가 논리 1의 값을 갖는 경우에는 XOR-게이트(1718)가 인버터로서 기능하지만, 그렇지 않을 경우에는, 버퍼로서 동작한다는 것에 주의한다. 이와 같이, Zero 신호(1716)가 0의 값에 도달한 이후의 2개의 기능 클록 사이클들 동안, 레지스터들(1702 및 1704) 각각은 0의 논리 값을 포함한다. 양자의 레지스터들(1702 및 1704)로부터의 내용들에 기초해, SGenIn 신호(1719)가 발생되어, 스트링 발생기(1700)가 인에이블되는지의 여부를 지시한다. 레지스터들(1702 및 1704)이 각각 0 및 0의 논리 값들을 포함하면, SGenIn 신호(1719)는, 스트링 발생기(1700)가 디스에이블된다(그리고 그것이 리로딩될 때까지 디스에이블 상태를 유지한다)는 것을 지시한다. 그렇지 않으면, 스트링 발생기(1700)는 개시 및/또는 포착 동작들을 위해 인에이블된다. 도 17b의 상태도(1750)는 비트-홀딩 유닛(1710)의 동작을 설명하기 위한 상태도(1760)를 포함한다.

<89> 다시 도 17a를 참조하면, 특수 링 카운터(1720)는 레지스터(G1;1704)가 홀딩할 다수의 기능 클록 사이클들(예를 들어, 개시 클록 에지와 포착 클록 에지 사이의 다수의 개개 클록 에지들)을 포함하도록 프로그램 가능하다. 스트링 발생기(1700)는 제로 검출 능력을 갖춘 특수 링 카운터(1720)를 이용한다. 도시된 일례에서, 특수 링 카운터(1720)는 3-비트 링 카운터(1722)를 구현한다. 바람직스럽게도, 3-비트 링 카운터(1722)는 개시 및 포착 동작들 동안의 Zero 출력을 위한 단일 OR-게이트 지연의 최악 경우 지연을 보유한다. 3-비트 링 카운터(1722)는, 특수 링 카운터(1720)의 MSB를 인에이블하여 출력 신호를 논리 1-대-0의 전이를 가진 Zero 신호(1716)로 설정하는 것에 의해, 지연을 1개 OR-게이트 지연으로 제한한다. OR-게이트들의 체인(1724)은, Zero 신호(1716)의 제로 값을 직접적으로 판정하기 위한 LSB(least significant bit)로부터 전이가 전파될 수 없다는 점에서, 잘못된 경로를 제공한다. 특수 링 카운터(1720)는 2N까지 카운팅할 수 있는데, 여기에서, N은 발명의 다른 실시예들에서 M-비트 링 카운터를 구현하는데 사용되는 플립플롭들의 수라는 것에 주의한다. 다른 카운터들도 스트링 발생기에 통합될 수 있다. 도 17b의 상태도(1750)는 3-비트 링 카운터(1722)의 거동을 설명하기 위한 상태도(1770)를 포함한다.

<90> 도 18은, 발명의 실시예에 따른, 임의 갯수의 프로그램 가능한 테스트 클록 제어기들을 구현하는 개념적인 테스트 클록 제어 구조를 나타내는 블록도를 묘사한다. 테스트 클록 제어 구조(1800)는 임의 갯수(N개)의 프로그램 가능한 테스트 클록 제어기들(PTCC)을 포함하는데, 그것들 중 하나가 PTCC(1810)로서 묘사된다. 테스트 클록 제어 구조(1800)의 PTCC 각각은, 스캔 클록(SCLK;1816)과 기능 클록들(PCLK[1..N];1806) 중 하나와 같은, 소스 클록들 사이를 멀티플렉싱하기 위한 멀티플렉서(MUX;1812)를 포함한다. 이 일례에서, 하나 이상의 PLL 회로들(1802)은 PLL 출력 클록들(1806)로 로킹된 위상들을 가진 하나 이상의 기준 클록들(RefClk[1..M];1801)로부터 기능 클록들(1806)을 발생시킨다. 더 나아가, 각각의 PTCC는 클록 명령 정보(CCI)를 인가하여 스캔 체인들(1820)에서의 하나 이상의 플립플롭들 및/또는 스캔 체인들을 테스트 클록들(CLK[1..N];1818)로부터의 하나 이상의 테스트 클록으로써 구동하기 위한 로직(1814)을 포함한다. 스캔 체인(1820)은, 그것들 중 하나가 도 18에 도시되어 있는 임의 갯수의 순차적 셀들(1820a)을 포함한다. 일 실시예에서, 순차적 셀(1820a)은 멀티플렉서(MUX;1870) 및 저장 소자(1872)를 포함한다. 멀티플렉서(1870)는, 스캔 인에이블 데이터(SeD) 제어 신호(1840)의 재조정된 버전인 SE2FF 신호(1871)에 의해 (예를 들어, CUT로부터의 결과 데이터로서의) 입력(1880)이나 (순차적 선행 셀로부터의) 스캔 체인 입력(SI;1882)을 저장 소자(1872)로 스위칭하도록 제어된다. 저장 소자는 데이터를 출력(1884; 예를 들어, CUT로의 자극 데이터)으로서 그리고/또는 스캔 출력(SO;1886)을 통해 순차적 후속 셀에 제공하도록 구성된다.

<91> 이 일례에서, PTCC(1810)는 제어 체인(1830)의 적어도 일부분으로부터 클록 명령 정보를 수신한다. 일부 실시

예들에서, 스캔 체인들(1820) 및/또는 제어 체인(1830)은 각각 (예를 들어, 데이터를 직렬로 시프팅하기 위한) 단일 체인이나 (예를 들어, 데이터를 병렬로 시프팅하기 위한) 임의 갯수의 스캔 체인들의 일부분들을 구비할 수 있다. 일부 실시예들에서는, (도시되지 않은) ATE가 SeD 신호(1840), SeC 신호(1842), 일차 스캔 클럭 (ScanClk) 신호(1844), 및 기준 클럭들(1801)을 발생시킨다. 하나 이상의 실시예에서, 이들 ATE-발생 신호들은 전역적 신호들이다. 다시 말해, 이 신호들은 테스트 클럭 제어 구조(1800)의 PTCC들에 인가된다. 일부 실시예들에서는, 선택적인 DFT(design-for-test) 블록(1850)이 추가된다. DFT 블록(1850)은 SeD 신호(1840) 및 클럭 도메인 인에이블(CDEn) 신호(1852)에 의해 제어되는 다수 플립플롭들(SeFF들;1860)을 포함한다. 일반적으로, SeD 신호(1840)의 타이밍 제한들을 보장하기 위한 SeFF들(1860)은, 예를 들어, 거대 시스템들(예를 들어, 칩상의 시스템들)에서의 LSL 테스트 프로토콜들을 위해 합류된다. 로직(1814)이 개시 및/또는 포착 동작들을 실현하기 위한 클럭 명령 정보를 포함하는 경우, CDEn 신호(1852)는 대응되는 PTCC로 프로그램된 내용에 기초해 SeFF들(1860)의 내용을 조정한다. CDEn 신호(1852)의 상태는 인에이블된 클럭 도메인들에서의 SeFF들(1860)의 내용들을 보유하고, 디스에이블된 클럭 도메인들에서의 SeFF들(1860)을 리셋한다. 따라서, 인에이블된 클럭 도메인들만이 LSL 테스트에 참여할 수 있는 한편, 다른 클럭 도메인들은 디스에이블되거나 기능 모드에서 동작할 수 있다.

<92> 도 19는, 발명의 실시예에 따른, 상이한 도메인들의 회로들을 테스트하기 위한 프로그램 가능한 테스트 클럭 제어기들을 구현하는 테스트 클럭 제어 구조를 나타내는 도면이다. 테스트 클럭 제어 구조(1900;또는 그것에 관한 일부분)은, 각각, 클럭 도메인들(1902a, 1902b, 및 1902c)과 상호 작용하기 위한 프로그램 가능한 테스트 클럭 제어기들(PTCC;1910, 1920, 및 1930)을 포함한다. 각각의 클럭 도메인(1902)은 하나 이상의 스캔 체인들(1904;또는 그것에 관한 일부분들) 및 CUT(1906)를 포함한다. 이 일례에서, 클럭 도메인들(1902)의 스캔 체인들(1904)은, 하나 이상의 스캔 체인 일부분들(1904)이, 예를 들어, 스캔 언로드/로드 동작에 종속될 수 있는 한편, 다른 것들은 그렇지 않다는 점에서 병렬이다. 더 나아가, 하나 이상의 클럭 도메인들이 테스트 프로토콜의 동일하거나 상이한 유형들로써 실시될 수 있는 한편, 다른 것들은 휴지 상태를 유지하거나 다른 테스트들에 종사할 수 있다. 예를 들어, PTCC들(1910 및 1920)로 로딩된 클럭 명령 정보(CCI;1912 및 1922)는, 각각, 클럭 도메인(1920a)에서 개시 동작을 그리고 클럭 도메인(1920b)에서 포착 동작을 발생시킴으로써, 도메인간 테스트를 구현한다고 가정한다. 이것은, 클럭 도메인(1902c)의 스캔 체인 일부분들(1904)이 로드/언로드 동작에 종속되거나, 예를 들어, CCI(1924)에 응답하여 도메인(1902c)을 디스에이블하는 PTCC(1930)의 결과로서, (예를 들어, 정적 테스트를 수행하기 위해) 비활성인 동안 발생할 수 있다. 테스트 클럭 구조(1900)는 하나 이상의 클럭 도메인들(1902)을 병렬로 테스트하기 위한 융통성을 제공하므로, 그것은 클럭 도메인들 사이의 순차적 의존도를 감소시키고, 그것은 다시, 자동화된 테스트 패턴 발생의 복잡도를 감소시킨다. 바람직스럽게도, 테스트 클럭 제어 구조(1900)에서의 병렬 도메인들의 구현은, 스캔 로드/언로드 동작들의 수를 감소시키는 것에 의해 전반적인 테스트 시간 및 테스트 데이터 볼륨을 감소시킬 수 있다. 도 19는 도메인들(1902)을 클럭 도메인들로서 묘사하지만, 이들은, 전력 도메인들과 같은, 도메인들의 임의의 다른 유형으로서 특징지어질 수도 있다. 테스트 클럭 제어 구조(1900)에 관한 상기 내용은 다음의 도 23 및 도 24에도 적용될 수 있다.

<93> 도 20은, 발명의 실시예에 따른, 테스트 접근점들을 감소시키기 위해 일차 입력들(2020) 및 일차 출력들(2022)을 공유하는 제어 체인들(2002) 및 스캔 체인들(2004)을 나타내는 도면이다. 이 일례에서, SeC 신호(2001)는, 멀티플렉서(2006)가 제어 체인(2002)이나 스캔 체인(2004)으로부터 시프팅되는 데이터를 선택하게 하는 한편, SeD 신호(2003)는, 멀티플렉서(2008)가 체인들로부터 또는 기능 출력으로부터 시프팅되는 데이터를 선택하게 한다.

<94> 도 21은, 발명의 일 실시예에 따른, 도메인간 테스트를 수행하도록 구성된 테스트 클럭 제어 구조를 예시한다. 테스트 클럭 제어 구조(2100)는 2개의 클럭 도메인들을 위한 지연 테스트들을 조정하기 위한 프로그램 가능한 테스트 클럭 제어기들(PTCC들;2111 및 2121)을 포함한다. 클럭 도메인(2102) 및 클럭 도메인(2104)은, 각각, 회로(2103 및 2105)를 포함하는데, 이들 양자는 별도의 클럭 주파수들에서 동작하도록 설계된다. 이와 같이, PTCC(2111)는 기능 클럭(PCLK[i];2110)을 사용해 제1 클럭 속도를 발생시킬 수 있는 한편, PTCC(2121)는 기능 클럭(PCLK[j];2120)을 사용해 제2 클럭 속도를 발생시킬 수 있다. 발명의 다양한 실시예들에 따르면, PTCC들(2111 및 2121)의 클럭 펄스 제어기들(2130)은 제어 체인(2140)을 통해 - 비교적 동기된 상태에서 - 개시만의 그리고 포착만의 동작들을 각각 실행하여 도메인간 영역(2150)에서의 인터페이스 회로들 등을 테스트하도록 프로그램될 수 있다. 개시만의 그리고 포착만의 동작들을 포함하는, 다중-사이클 경로 테스트 동작들을 조정하기 위해, 자동화된 테스트 패턴 발생기(ATP 발생기)로서 기능하도록 구성된 컴퓨팅 디바이스는 도메인간 테스트들을 실현하여 도메인간 영역(2150)에서의 장애들을 검출하기 위해 PTCC들(2111 및 2121)의 동작을 동기화하는 테스트 패턴들을 발생시킬 수 있다. 예를 들어, ATP 발생기는 공통적인 기준 클럭(RefClk;2160)을 사용해 타이밍

기준점을 확립할 수 있다. 다른 실시예들에서, ATP 발생기는, 도메인간 테스트를 수행하기 위해, 서로에 대하여 비동기적인 상이한 기준 클록들도 사용할 수 있다. 일부 실시예들에서, "다중-사이클 경로 테스트(multi-cycle path test)"라는 용어는 일반적으로, 도메인간 테스트, 도메인내 테스트, 또는 양자를 의미할 수 있다.

<95> 도 22a 및 도 22b는, 발명의 다양한 실시예들에 따른, 도메인간 테스트를 구현하기 위한 하나 이상의 타이밍 기준점들을 사용해 시간적 거리들을 계산하기 위한 다양한 기술들을 예시한다. 도 22a는, 공통적인 기준 클록 (RefClk[k];2216)이, 하나 이상의 프로그램 가능한 테스트 클록 제어기들(PTCC)에 의해 수행되는 동작을 지연시키기 위한 클록 사이클들의 수를 판정하기 위한 타이밍 기준점(2220)을 제공하는데 사용되는 방법을 나타낸다. PTCC는, SeC 신호(2218)가 샘플링된 이후에 개시 동작 또는 포착 동작을 지연시킬 수 있다. PTCC는 그것의 특정 클록 도메인에서의 테스트 클록과 관련하여 동작을 지연시킨다. 도메인간(예를 들어, 클록 도메인간) 테스트는 클록 도메인[i]으로부터 포착이 발생할 다른 클록 도메인[j]으로의 개시를 실현하도록 설계될 수 있다고 가정한다. 이와 같이, 클록 도메인[i]의 개시 클록 펄스는, SeC 신호(2218)가 샘플링된 이후에 4개의 PCLK[i] 클록 사이클들만큼 지연될 수 있다. 더 나아가, 클록 도메인[j]을 위한 포착 동작은 1개의 PCLK[j] 클록 사이클만큼 지연될 수 있다. RefClk[k] 클록(2216)의 위상은 적어도 PCLK[i](2212) 및 PCLK[j](2214)로써 로킹된다는 것에 주의한다. "시간적 거리"는, PCLK[i](2212)나 PCLK[j](2214)와 같은, 소정 클록에 대한 사이클들 수의 지연(delay in a number of cycles)이다. 이와 같이, ATP 발생기는, 기준 클록 에지(2220)로부터 시작해 지정된 개시(또는 포착) 클록 에지까지의 PCLK[i](2212)의 하강 에지들의 수를 카운트하는 것에 의해 시간적 거리를 계산할 수 있다. ATP 발생기는 시간적 거리들에 기초해 테스트 패턴들을 형성할 수 있다. 그 다음, ATE는 시간적 거리들을 PTCC들의 클록 펄스 제어기들 뿐만 아니라 다중-사이클 테스트를 제공하기 위한 스텝 발생기들로도 프로그램할 수 있다.

<96> 도 22a의 다이어그램(2230)은 클록 도메인[i]으로부터 전이를 개시하고 그것을 클록 도메인[j]에서 포착하기 위해 계산된 시간적 거리들(2232 및 2234)을 나타낸다. 그래서, 입력 전이가 RefClk[k](2216)의 상승 에지(2220)로부터 - 4개 하강 에지들의 지연(예를 들어, 4의 시간적 거리(2234)) 이후에 - PCLK[i](2212)의 5번째 상승 에지(2217)에서 개시될 수 있다. 전이는, 1개 하강 에지의 지연(예를 들어, 1의 시간적 거리(2234)) 이후에, 클록 PCLK[j](2214)의 2번째 상승 에지(2219)에서 포착된다. 다이어그램(2240)은, 전이를 개시하는 클록 도메인[j]으로부터 클록 도메인[i]으로(다이어그램(2230)을 위한 것과는 반대 방향으로) 전이를 포착하기 위해 계산된 시간적 거리들(2242 및 2244)을 나타낸다. 이와 같이, 전이는 PCLK[j](2214)의 2번째 상승 에지(2219)에서(예를 들어, 1의 시간적 거리(2244) 이후에) 개시될 수 있고, 전이는, 6개 하강 에지들의 지연(예를 들어, 6의 시간적 거리(2242)) 이후에, PCLK[i](2212)의 7번째 상승 에지(2221)에서 포착될 수 있다.

<97> 도 22b는, 기준 클록들도 테스트 클록들도 동위상(in phase)이 아닌 경우들에서의 시간적 거리들의 계산을 예시한다. 이 도면은, 클록 도메인들이, 동위상이 아니며 동기되지 않은 기능 클록들에 대응되는 클록 도메인간 테스트를 위한 타이밍을 묘사한다. 이 일례에서, 클록들(PCLK[i] 2262 및 PCLK[j] 2264)은, 각각, 비동기 상태인 기준 클록들(RefClk[k] 2266 및 RefClk[l] 2268)로부터 유도된다. 이와 같이, 시간적 거리 계산들은 다수 타이밍 기준점들의 관점에서 이루어진다. 예를 들어, 기준점들(2267 및 2269)은, 시간적 거리들을 판정하기 위한 상이한 베이스들을 제공한다. 도 22b의 다이어그램(2280)은, 각각, 클록 도메인[i]으로부터 전이를 개시하고 그것을 클록 도메인[j]에서 포착하기 위해 계산된 시간적 거리들(2282 및 2284)을 나타낸다. 그래서, 입력 전이는 RefClk[k](2266)의 상승 에지(2267)로부터 - 4개 하강 에지들의 지연(예를 들어, 4의 시간적 거리(2282)) 이후에 - PCLK[i](2262)의 5번째 상승 에지(2277)에서 개시될 수 있다. 전이는, 1개 하강 에지의 지연(예를 들어, 1의 시간적 거리(2284)) 이후에, 클록 PCLK[j](2264)의 2번째 상승 에지(2279)에서 포착된다. 다이어그램(2290)은, 전이를 개시하는 클록 도메인[j]으로부터 클록 도메인[i]으로 전이를 포착하기 위해 계산된 시간적 거리들(2292 및 2294)을 나타낸다. 이와 같이, 전이는 PCLK[j](2264)의 2번째 상승 에지(2279)에서(예를 들어, 1의 시간적 거리(2294) 이후에) 개시될 수 있고, 전이는, 7개 하강 에지들의 지연(예를 들어, 7의 시간적 거리(2292)) 이후에, PCLK[i](2262)의 8번째 상승 에지(2281)에서 포착될 수 있다. 일부 실시예들에서는, 클록 도메인간 테스트의 융통성있는 구현을 허용하기 위해 다수 SeC 신호들(2270)이 통합될 수 있다. (도시되지 않은) 다수 SeC 신호들(2270)은 기준 클록들을 위한 에지 선택의 조종 가능성(controllability)을 증가시킬 수 있고, 그것에 의해, 기준 클록들의 소정 시작 에지들을 선택함으로써 하나 이상의 스트링 발생기들의 사이즈를 감소시킬 수 있다. 예를 들어, SeC[m] 신호 및 SeC[m+1] 신호는, 예를 들어, 스트링 발생기들의 사이즈를 최소화하기 위한 (t)번째 주기 및 (t+h)번째의 하강 또는 상승 에지일 시작 에지를 선택할 수 있다.

<98> 도 23은, 발명의 일 실시예에 따른, 도메인내 테스트를 수행하도록 구성된 테스트 클록 제어 구조를 예시한다. 테스트 클록 제어 구조(2300)는, 각각이 대응되는 서브-도메인(2302)을 제어하기 위한 프로그램 가능한 테스트

클록 제어기(PTCC;2304)를 포함한다. 도메인(2301)은 다수의 좀더 작은 도메인들 또는 서브-도메인들(2302)로 분할된다. 도메인(2301)이 클록 도메인이라면, PTCC들(2304)은 집합적으로, 서브-도메인들(2302)을 테스트하기 위한 공통적인 기능 클록(PCLK;2306)을 서브-클록(clock) 도메인들로서 사용한다.

<99> 도 24는, 발명의 일 실시예에 따른, 간략화된 프로그램 가능한 테스트 클록 제어기들을 사용해 도메인내 테스트를 수행하도록 구성된 테스트 클록 제어 구조를 예시한다. 이 일례에서, 테스트 클록 제어 구조(2400)는 다수 서브-클록 도메인들(2403)의 스캔 테스트를 제어하기 위한 간략화된 프로그램 가능한 테스트 클록 제어기들의 정렬(2401)을 포함한다. 여기에서 사용되는 바와 같이, 일부 실시예들에서의 "클록 도메인 일반화(clock domain generalization)"라는 용어는 클록 도메인을 한 세트의 서브-클록 도메인들(2403)로 분할하는 프로세스를 의미하고, 그것에 의해, 간략화된 프로그램 가능한 테스트 클록 제어기들의 정렬(2401)은 선행 분할된 원래의 클록 도메인을 위한 구조에 의한 것보다 일반화된 클록 도메인에 의해 좀더 큰 융통성을 가질 수 있다. 예를 들어, 간략화된 프로그램 가능한 테스트 클록 제어기들의 정렬(2401)은, 회로에 매립된 전력 도메인들이 서브-도메인들(2403)로 분할되어 서브-도메인들(2403)로서 테스트될 수 있을 때에도 유용할 수 있다. 전력 회로의 성능이 전력 도메인내의 시스템 지연에 영향을 미칠 수 있으므로, 각각의 전력 도메인을 별도의 서브-도메인(2403)으로 형성하고, 지연 테스트 동안, 그것을 별도로 제어하는 것이 바람직스러울 수도 있다. 클록 도메인 일반화는 실리콘 디버그시에, 다른 서브-클록 도메인들의 테스트와 간섭하지 않으면서, 예를 들어, 서브-클록 도메인에서 장애 플립플롭을 검출하는데도 유용할 수 있다는 것에 주의한다. 정렬(2401)은, 사용자가 임의 클록 도메인에서의 스캔 로드/언로드를 프로그램할 수 있게 하므로, 진단 동안, 서브-클록 도메인들(2403)을 포함하는 클록 도메인들에서의 선택 스캔 체인들만이 로딩/언로드되면 된다.

<100> 정렬(2401)은 기능 클록(PCLK[i];2405)을 사용해 "m"개의 서브-클록 도메인들(2403)을 구동하도록 구성된다. 정렬(2401)은 부모 PTCC(2402) 및 자식 PTCC들(2406 및 2408)을 포함한다. 부모 PTCC(2402)는 프로그램 가능한 갯수의 테스트 클록 펄스들(2407)을 자식 PTCC들(2406 및 2408)로 브로드캐스팅하도록 동작하는데, 자식 PTCC들(2406 및 2408)은, 그들 중 어느 하나가 테스트 클록 펄스들(2407)을 사용하거나 사용하지 않도록 프로그램될 수 있다. PTCC들(2406 및 2408)이 부모 PTCC(2402)와 동일한 구조 및/또는 기능성을 가질 수도 있지만, 이 일례에서, PTCC들(2406 및 2408)은 간략화된 프로그램 가능한 테스트 클록 제어기들(SPTCC들)로 이루어진다. 바람직스럽게도, SPTCC들로서의 PTCC들(2406 및 2408)은 구현을 위해 좀더 적은 하드웨어 리소스들을 요구한다. 도시된 바와 같이, SPTCC는 간략화된 클록 펄스 제어기(SCPC) 및 클록 도메인 일반화(CDG) 셀을 포함한다. 여기에서, PTCC(2406)는 SCPC(2410) 및 CDG 셀(2420)을 포함하는 한편, PTCC(2408)는 SCPC(2430) 및 CDG 셀(2440)을 포함한다. 일부 실시예들에서, SCPC(2430) 및 CDG 셀(2440)은, 각각, 부모 PTCC(2402)의 (어느 것도 도시되어 있지 않은) CPC 및 TCG의 간략화된 버전들로서 설명될 수 있다. 동작시에, SCPC(2410) 및 SCPC(2430)의 출력들은 게이트들(2470)에서 테스트 클록 펄스들(2407)과 AND되어, 예를 들어, CLK[i,1], CLK[i,j],...CLK[i,m]를 발생시킨다. 부모 PTCC(2402)를 디스에이블하는 것은 모든 서브-클록 도메인들을 디스에이블하는 결과를 초래한다는 것에 주의한다. 또한, 게이트(2490) 및 게이트들(2492)은, 각각, 클록 도메인들(2403a 및 2403c)의 동작을 인에이블하기 위한 클록 도메인 인에이블 신호들(CEDN[i,1] 2496 및 CEDN[i,m] 2498)을 발생시킨다는 것에도 주의한다.

<101> 더 나아가, SPTCC의 구조는 서브-클록 도메인들 각각의 소정 테스트 요구 사항들을 위해 최적화될 수 있다는 것에 주의한다. 예를 들어, 테스트 클록 펄스들(2407)이 통과되거나 차단되어야 한다면, SPTCC는, 클록(CLK[i,1])을 제어하기 위한 단일 플립플롭(C0;2472)을 포함하는 SCPC(2410)를 포함할 수 있다. 이 경우, SCPC(2410)는 ICLK[i](2411)를 사용해 로딩될 수 있고, 개시 및/또는 포착 동작들 동안, 변경되지 않는다. 일부 실시예들에서, ICLK[i](2411)는, 도 12에 도시된 바와 같이, 신호들(CPCen(1240), PCLK(1252), 및 TCLK(1210))을 사용해 발생될 수 있다. 개시 및/또는 포착 동작들을 위해, SCPC(2410)에는, (예를 들어, 스트리밍 발생기 등으로부터) CLK[i]를 발생시키기 위한 제어 패턴이 로딩되는, 부모 PTCC(2402)의 (도시되지 않은) 부모 CPC를 따라 1의 논리 값이 (예를 들어, 소자(C0;2472)로) 로딩될 수 있다. 국지적 테스트 클록(CLK[i, j])이 부모 테스트 클록(CLK[i];2407)과 동일하다면, PTCC들(2406 및 2408)은 인에이블된다. 서브-클록 도메인들(2403)의 테스트가, 예를 들어, 지연된 개시만을 또는 포착만을 상이한 클록 사이클들에서 수행할 것을 요구하면, SPTCC는, 소자들(C0(2476) 및 C1(2433))로 이루어진 (부모 PTCC(2402)와 유사한) 2-비트의 국지적 시프트 레지스터를 가진 SCPC(2430)를 포함할 수 있다. 바람직스럽게도, SCPC(2430)와 같은, SCPC의 2-비트 구현은 테스트 시간 및 테스트 볼륨 감소를 용이하게 한다.

<102> 도 25a 내지 도 25d는, 발명의 일 실시예에 따른, 프로그램 가능한 테스트 클록 제어기들을 사용하는 도메인간 테스트 및 도메인내 테스트를 용이하게 하도록 구성된 테스트 클록 제어 구조를 예시한다. 도 25a에서, 테스트

클록 제어 구조(2500)는, 도메인들 또는 서브-도메인들의 일부분들(2510 및 2530)이 회로들(2520)을 테스트하기 위해 병렬로 또는 직렬로 실시되는지와 무관하게, 회로들의 도메인간 테스트, 도메인내 테스트, 또는 양자를 용이하게 한다. 프로그램 가능한 테스트 클록 제어기들은 각 클록 도메인 경계들내의 도메인들에 대한 내부 상호작용들을 한정하도록 프로그램될 수 있으므로, 도메인을 위한 인터페이스들이 나머지 도메인으로부터 독립적으로 제어될 수 있다. 결과적으로, 하나 이상의 도메인 인터페이스들(및 서브-도메인 인터페이스들)이 스캔-기반 테스트에서 도메인(또는 서브-도메인)의 다른 일부분들에 대한 테스트와 병렬로 또는 직렬로 구현될 수 있다. 따라서, 개개 PTCC들이, 스캔 체인 일부분, 회로 등과 같은, 서브 도메인의 선택 일부분들을 제어하는데 사용될 수 있다.

<103> 예시를 위해, 서브-도메인(2510)을 위한 PTCC들(2512 및 2514)은, 서브-도메인(2530)을 위한 PTCC들(2522 및 2524)에 의해 사용되는 기능 클록과는 상이한 클록 속도의 동일한 기능 클록으로써 각각 동작한다고 가정한다. 먼저, PTCC들(2514 및 2524)은 테스트 클록 제어 구조(2500)의 일부분(2502)을 소정 시점(T1)에서 활성화하도록 프로그램된다고 가정한다. 예를 들어, PTCC들(2514 및 2524)은 개시 및 포착 동작 또는, 스캔 체인 로드 또는 언로드 동작과 같은, 다른 소정 동작을 수행하도록 프로그램될 수 있다. 이와 같이, 일부분(2502)은 클록 제어 구조(2500)의 다른 일부분들로부터 독립적으로 제어될 수 있다. 더 나아가, 이 일례를 위해, PTCC들(2512 및 2522)은 도메인내(또는 서브-도메인내) 동적 및/또는 정적 테스트를 일부분(2502)의 테스트와 병렬로 또는 직렬로 수행하도록 프로그램될 수 있다고 가정한다. 다음으로, 시점(T2)에서, PTCC들(2514 및 2524)은 디스에이블될 수 있는 한편, 나머지 PTCC들은 일부분들(2504)을 테스트하기 위해 동적(예를 들어, 브로드사이드, LSL, 도메인간, 도메인내 등) 및/또는 정적 테스트에 종사하도록 인에이블된다고 가정한다. 일부 실시예들에서, 테스트 클록 제어 구조(2500) 및 수반되는 회로는 일반적으로, PTCC들(2512, 2514, 2522, 및 2524)로 이루어진 구조를 정의한다는 것에 주의한다.

<104> 도 25b 내지 도 25d는, 발명의 다른 실시예에 따른, 프로그램 가능한 테스트 클록 제어기들을 사용해 도메인간 테스트 및 도메인내 테스트를 용이하게 하도록 구성된 테스트 클록 제어 구조의 또 다른 일례를 예시한다. 테스트 클록 제어 구조(2550)는 도메인간 테스트, 도메인내 테스트, 또는 양자를 용이하게 한다. 프로그램 가능한 테스트 클록 제어기들은 내부 상호작용들을 각 클록 도메인의 경계들내에 한정하도록 프로그램될 수 있으므로, 도메인을 위한 인터페이스들은 도메인의 나머지로 부터 독립적으로 제어될 수 있다. 그러한 인터페이스들의 일례들은 입력 레지스터들 및/또는 출력 레지스터들을 포함한다. 결과적으로, 하나 이상의 도메인 인터페이스들(및 서브-도메인 인터페이스들)이 스캔-기반 테스트에서 도메인(또는 서브-도메인)의 다른 일부분들에 대한 테스트와 병렬로 그리고/또는 직렬로 구현될 수 있다. 따라서, 개개 SPCP들은, 스캔 체인 일부분, 회로 등과 같은, 서브-도메인의 선택 일부분들을 제어하는데 사용될 수 있다.

<105> 예를 들어, 도 25b는, 서브-도메인(2560)을 위한 SPCP들(2564 및 2566)이, 양자가, 각각, 서브-도메인들(2520 및 2580)을 위한 SPCP들(2571 및 2581)에 의해 사용되는 기능 클록과는 상이한 클록 속도를 가질 수 있는 동일한 기능 클록으로써 각각 동작할 수 있다는 것을 나타낸다. 먼저, SPCP들(2571 및 2564)은 회로 일부분(2530)에서의 로직(2572) 및/또는 로직(2474)을 테스트하기 위한 클록 도메인간 테스트를 활성화하도록 프로그램되는 한편, SPCP들(2566 및 2581)은 회로 일부분(2531)에서의 로직(2582) 및/또는 로직(2484)을 테스트하기 위한 클록 도메인내 테스트를 활성화하도록 프로그램된다고 가정한다. 모든 SPCP들(2571, 2564, 2566, 및 2581)은 소정 시점(T1)에서 동작하도록 프로그램될 수 있다. 레지스터(2559)가 서브-도메인(도메인 r;2580)을 위한 출력 레지스터를 표현할 수 있다는 것에 주의한다. 따라서, SPCP들(2571 및 2564)은, 각각, 예를 들어, (예를 들어, 서브-도메인(2570)으로부터의) 개시 및 (예를 들어, 서브-도메인(도메인 q;2560)으로의) 포착 동작 또는, 스캔 체인 로드 또는 언로드 동작과 같은, 소정의 다른 동작을 수행하도록 프로그램될 수 있다. 레지스터(2589)는 서브-도메인(도메인 p;2570)을 위한 입력 레지스터를 표현할 수 있다는 것에 주의한다. 마찬가지로, SPCP들(2566 및 2581)은, 각각, 예를 들어, (예를 들어, 서브-도메인(도메인 q;2560)로부터의) 개시 및 (서브-도메인(도메인 r;2580)으로의) 포착 동작 또는, 스캔 체인 로드 또는 언로드 동작과 같은, 소정의 다른 동작을 수행하도록 프로그램될 수 있다. 이와 같이, 회로 일부분들(2530 및 2531)은 (예를 들어, 서브-도메인(2560)의 로직(2561), 로직(2562), 및 로직(2563)과 무관한) 클록 제어 구조(2550)의 다른 일부분들로부터 독립적으로 제어될 수 있다. 더 나아가, 이 일례를 위해, SPCP들(2564, 2565, 및 2566)은 도 25b의 회로 일부분들(2530 및 2531)의 테스트와 병렬로 또는 직렬로 도메인내(또는 서브-도메인내) 동적 및/또는 정적 테스트를 수행하도록 프로그램될 수 있다고 가정한다. 도 25c는, 시점(T2)에서, SPCP들(2571 및 2581)은 (예를 들어, 레지스터들(2559 및 2589)의 내용들을 보존하도록) 디스에이블될 수 있는 한편, SPCP들(2564 및 2565)은 시스템 조합 로직(2561)의 일부분을 포함하는 회로 일부분(2590)과 내부 레지스터들 및 로직을 포함할 수 있는 다른 일부분(2562)을 테스트하기 위한 동적(예를 들어, 브로드사이드, LSL, 도메인간, 도메인내 테스트 등) 및/또는 정적 테스트에

중사하도록 구성된다. 또한, 도 25d는, 회로의 다른 일부(2562) 및 로직(2563)을 포함하는 회로 일부(2591)를 테스트하기 위해, 시점(T3)에서, SPCP들(2565 및 2566)이 동적 및 정적 테스트들에 중사하도록 인에이블될 수 있다는 것을 나타낸다.

<106> 도 26은, 발명의 실시예에 따른, 프로그램 가능한 테스트 클럭 제어기들(PTCC들)을 구현하는 테스트 클럭 제어기 구조를 사용해 회로들을 테스트하기 위한 마이크로-레벨 흐름의 일례를 나타낸다. 흐름(2600)은 일반적으로, 2-비트 시프트 레지스터들은 갖추었지만 보충 클럭 명령 소스는 갖추지 않은 PTCC들에 적용된다. 일부 경우들에서, SeFF들은, 흐름(2600)에서 스캔 로드/엔로드 동작들이 시작되기 전에, 초기화된다. SeFF들의 일례들은 도 18에서의 SeFF들(1860)로서 참조된다. 흐름(2600)은 2601에서 테스트 모드 활성화 신호를 어서팅(asserting)함으로써 스캔 테스트 모드(TM)를 인에이블하는 것에 의해 시작된다. 2602에서, 제어 로드/엔로드 동작은 제어 체인에 액세스하도록 초기화된다. 그 다음, PTCC는 2604에서 테스트 클럭의 스캔 클럭(SCLK)으로서의 전파를 인에이블 또는 디스에이블하도록 프로그램된다. 다음으로, 스캔 로드/엔로드 동작은, 스캔 경로가 인에이블될 때 2606에서 시작된다. 2608에서, 스캔 데이터는 스캔 체인 일부(2608)로 로딩되거나 스캔 체인 일부(2608)로부터 언로딩되고, 그 후, PTCC는 2610에서 프로그램 모드로 진입한다. 선택적으로, SeFF들은, 하나의 SCLK 클럭 펄스를 2612 이전에 모든 SeFF들에 인가하는 것에 의해 초기화될 수 있다. 일단 프로그램 모드에서는, 2612에서 그것으로 로딩된 클럭 명령 정보에 따라 PTCC가 구성될 수 있다. 예를 들어, PTCC는 개시 및/또는 포착 동작, 개시만의 동작, 포착만의 동작, 사일런트 동작(silent operation) 등의 하나 이상에 중사하도록 구성될 수 있다. 2614에서, PTCC는 사일런트 상태 동작으로 진입할 수 있다. 선택적으로, 그것은, IDDQ 누설 테스트와 같은, 정적 테스트를 수행할 수 있다. 그렇다면, ATE는 2618에서 입력을 인가(또는 강제)할 수 있고, 거기에는, 전원 공급 장치 및 그라운드 터미널들에서의 파라메트릭 IDDQ 측정이 수반된다. 다음으로, PTCC는 2620에서 기능 테스트 모드로 진입하여 개시 및/또는 포착과 연관된 동적 테스트를 수행한다. PTCC는, 2622에서, 프로그램된 개시 및/또는 포착을 수행한다. 2624에서, ATE는, 예를 들어, 모든 도메인들이 테스트되었는지의 여부를 판정하고, 그렇지 않다면, 액션의 후속 코스는 무엇이어서야 하는지를 판정한다. 하나 이상의 추가적인 클럭 도메인들이 테스트를 요구하는 경우, PTCC가, 수반되는 스캔 로드/엔로드 동작없이 그것의 기능을 변경하여 회로를 추가적으로 테스트하도록 프로그램될 수 있다면, 흐름(2600)은 2612에서 계속된다. 그러나, 스캔 로드/엔로드 동작이 요구된다면, 흐름(2600)은 2604에서 계속된다. 2630에서, ATE는, 패턴들이 구현될 때까지, 테스트를 계속한다.

<107> 흐름(2600)은 제어 체인을 사용해 PTCC를 위한 2개의 프로그래밍 단계들을 기술하지만, 2604 및 2610에서의 동작들과 연관된 테스트 시간 및 테스트 데이터 볼륨은, 스캔 로드 및 언로드 동작에 비해, 전체 테스트 시간 및 데이터 볼륨에 미치는 영향이 훨씬 적다. 더 나아가, 클럭 도메인들이 실시될 때까지, 전체 클럭 도메인들이 스캔 로드 및 언로드 동작들에 중사할 필요는 없다. 일반적으로, 테스트 클럭 제어 구조는, 실시된 클럭 도메인들에서의 스캔 체인들에 대해서만 스캔 로드 및 언로드 동작들을 수행하도록 설계된다. 바람직스럽게도, 흐름(2600)은, 좀더 빈번하게 스캔 체인들을 로딩 및 언로딩하는 전통적인 스캔-기반 테스트에 비해, 전체 테스트 시간의 감소를 제공한다.

<108> 흐름(2600)은, 테스트 클럭 제어 구조를 사용해 매-구성에 대한 또는 설정 레벨에 대한 테스트들을 구현하기 위한 방법의 일례를 설명하지만, 발명의 다른 실시예들은 여기에서 기술된 테스트 클럭 제어 구조들을 사용하는 매크로-레벨 테스트 흐름에 관한 것이다. 일 실시예에 따르면, 스캔-기반 테스트를 위해 테스트 클럭 제어 구조를 구현하기 위한 방법이 다음과 같이 설명될 수 있다. 이 방법은 도메인내 테스트를 수행하여, 동적인 장애 검출 테스트 패턴들을 구현하는 복수개 회로들의 도메인들의 제1 서브세트를 연습시킬 수 있다. 이 방법은 계속해서, 동적인 장애 검출 테스트 패턴들을 구현하는 복수개 회로들의 도메인들의 제2 서브세트를 연습시키기 위한 도메인간 테스트와 같은, 다른 테스트를 추가적으로 수행한다. 이러한 동적인 장애 검출 테스트 패턴들은 LSL 테스트 패턴들 및 브로드사이드 테스트 패턴들 뿐만 아니라 임의의 다른 동적 테스트 패턴들도 포함할 수 있다. 일부 실시예들에서, 이들 도메인들은 클럭 도메인들이다. 일부 경우들에서, 이 방법은 도메인간 테스트를 겪고 있는 회로들과는 구별되는 회로들의 일부(2608)들에 대해, 고착 테스트들 및 동적인 테스트들과 같은, 도메인내 테스트를 수행하는 단계를 더 포함한다.

<109> 다음으로, 이 방법은 상이한 클럭 도메인들을 실질적으로 병렬로 테스트하기 위한 프로그램 가능한 상이한 테스트 클럭 제어기들을 구성할 수 있다. 일반적으로, 이것은, 상이한 클럭 도메인들을 테스트할 때, 하나 이상의 스캔 체인들이 로딩되는 횟수를 감소시킨다. 프로그램 가능한 테스트 클럭 제어기들이 상이한 서브-클럭 도메인들을 테스트하기 위해 상이한 서브-클럭 도메인들을 포함할 수 있다는 것에 주의한다. 일부 실시예들에서, 도메인간 테스트의 수행은, 고착 테스트들으로써 구현될 수 있는 클럭 도메인간 테스트의 수행과 인터리빙될 수

있다. 예를 들어, 클록 도메인내 및 클록 도메인간 테스트 패턴은 단일 테스트 패턴으로 조합될 수 있고, 그것에 의해, 바람직스럽게도, 매 스캔 체인 동작당 테스트 커버리지를 증가시킬 수 있다. 클록 도메인내 테스트는 도메인간 테스트에 수반하여 수행될 수 있다는 것에 주의한다. 도메인간 테스트의 수행은, 정적인 장애 검출 테스트 패턴들을 구현하는 단계를 포함할 수도 있다. 예를 들어, 장애 검출 테스트 패턴들은 도메인간 테스트에 사용되는 동적인 장애 검출 테스트 패턴들과 실질적으로 동시에 스캔 체인들로 로딩될 수 있다. ATE 역시, 도메인간 테스트 동안, 정적인 장애 테스트와 실질적으로 동시에 LSL 테스트를 실행할 수 있다.

<110> 일부 실시예들에서, 이 방법을 위한 도메인간 테스트의 수행은 시간적 거리를 계산하는 단계 및, 시간적 거리에 기초해, 2개 도메인들 사이의 개시 및 포착 시퀀스를 개시하는 단계를 포함한다. 하나 이상의 실시예에서, 이 방법은 하나 이상의 제어 체인들 및 하나 이상의 스캔 체인들을 포함하는 특정한 테스트 클록 제어 구조를 위한 자동적인 테스트 패턴 발생(ATPG) 흐름에 따라 동적인 장애 검출 테스트 패턴들을 발생시키는 단계를 포함한다. ATPG 흐름의 일례가 다음에서 설명된다.

<111> 도 27은, 발명의 실시예에 따른, 프로그램 가능한 테스트 클록 제어기들(PTCC들)을 구현하는 테스트 클록 제어기 구조를 사용해 회로들을 테스트하기 위한 매크로-레벨 흐름의 일례를 나타낸다. 구체적으로, 흐름(2700)은 테스트 클록 제어 구조들을 구현하기 위한 테스트 패턴들을 발생시키기 위한 ATPG 흐름이다. 흐름(2700)에서, 클록 도메인내 및 클록 도메인간 테스트들은 스캔 로드/언로드 동작들의 수를 감소시키기 위해 분리된다. 클록 도메인내 및 클록 도메인간 테스트들내에서, LSL 테스트를 위한 옛-스피드 테스트가 처음으로 수행되고 브로드사이드가 수반된다. 이와 같이 브로드사이드 이전에 LSL을 배치하는 것은 전반적인 테스트 커버리지, 테스트 데이터 볼륨, 및 ATPG 런타임을 최적화하기 위한 시도이다. 더 나아가, 브로드사이드를 사용하는 테스트 패턴들을 LSL을 사용하는 패턴들에 첨부하는 것에 의해, 테스트 패턴들이 획득될 수 있다. 고착 테스트들을 위한 흐름(2700)에서의 ATPG가 클록 도메인간 테스트에서 구현될 수 있다는 것에 주의한다. 이것은, 지연 테스트 패턴들에 포함된 고착 테스트 패턴들이 지연 테스트 동안 실질적으로 영향을 미치지 않는 않지만, 그것들이, 발생될 고착 테스트 패턴들의 수를 감소시킬 수 있기 때문이다.

<112> 이와 같이, 흐름(2700)은 2710에서 클록 도메인내 로직을 위한 지연 테스트들을 구현하기 위한 패턴들을 발생시키는 것에 의해 자동적인 테스트 패턴 발생을 시작한다. 이것은, 2720에서 LSL을 사용해 지연 장애들을 위한 테스트 패턴들을 발생시키는 단계 및 2730에서 브로드사이드를 사용해 2720에서의 미검출 장애들을 위한 테스트 패턴들을 발생시키는 단계를 포함할 수 있다. 일부 실시예들에서, 클록 도메인간 테스트 패턴들은 LSL 테스트를 위한 2722에서의 테스트 패턴 발생에 포함될 수 있을 뿐만 아니라 브로드사이드 테스트를 위한 2732에서의 테스트 패턴 발생에도 포함될 수 있다는 것에 주의한다. 흐름(2700)은, 흐름이 클록 도메인간 로직을 위한 지연 테스트 패턴들을 발생시키는 2740에서 계속된다. 2750에서, ATPG 흐름은 LSL을 사용해 지연 장애들을 위한 테스트 패턴들을 발생시키는데, 여기에는, 2750에서의 미검출 장애들을 위한 2760에서의 브로드사이드를 사용하는 테스트 패턴들의 발생이 수반된다. 예를 들어, 일부 경우들에서에서의 LSL을 사용하는 지연 테스트 패턴 발생에는, LSL을 사용하는 테스트로부터 탈출한 장애들을 검출하기 위해, 브로드사이드 테스트 프로토콜을 사용하는 동일한 테스트가 수반될 수 있다. 일부 실시예들에서는, 클록 도메인내 로직을 테스트하기 위한 고착 패턴들이 LSL 테스트를 위한 2752에서의 테스트 패턴 발생 뿐만 아니라 브로드사이드 테스트를 위한 2762에서의 테스트 패턴 발생에도 포함될 수 있다. 이와 같이, 흐름(2700)은 동적 테스트들과 병렬로 정적 테스트들을 수행하기 위한 고착 테스트 패턴들을 발생시킬 수 있다. 2780에서, ATPG 흐름은 고착 장애들을 검출하기 위해 2710과 2740에서 발생된 테스트 패턴들을 위한 장애 자극을 수행할 수 있다. 이것에는, 2780의 단계 3에서의 미검출 고착 장애들을 위한 테스트 패턴들의 발생이 수반된다. 이것은 2792에서 클록 도메인내 로직을 위해 그리고 2794에서 클록 도메인간 로직을 위해 테스트 패턴들을 발생시키는 단계를 포함한다. 기능 스피드에서 고착 테스트 패턴들을 실행하는 것은 테스트의 품질을 향상시킬 수 있을 뿐만 아니라, 고착 테스트 프로토콜이 고려될 필요가 없기 때문에, 테스트 프로세스를 간략화할 수도 있다. 2796에서, 흐름(2700)은 2792에서 발생된 고착 테스트 패턴들을 2720에서 발생된 지연 테스트 패턴들과 통합한다. 2798에서, 흐름(2700)은 2794에서 발생된 고착 테스트 패턴들을 2750에서 발생된 지연 테스트 패턴들과 통합하는 것에 의해 계속된다. 2799에서, 결과적인 브로드사이드 테스트 패턴들은 LSL 패턴들에 첨부되어 최종적인 테스트 패턴들을 형성할 수 있다.

<113> 본 발명의 실시예는, 다양한 컴퓨터 구현 동작들을 수행하기 위한 컴퓨터 코드를 갖춘 컴퓨터 판독 가능 매체와 컴퓨터 저장 제품을 관련짓는다. 매체들 및 컴퓨터 코드는 발명의 목적들을 위해 특별히 설계되고 구성된 것들일 수 있거나, 컴퓨터 소프트웨어 업계의 당업자들이라면 입수 가능한 주지의 종류일 수도 있다. 컴퓨터 판독 가능 매체들의 일례들로는 하드 디스크들, 플로피 디스크들, 및 자기 테이프와 같은 자기 매체들; CD-ROM들 및 홀로그래픽 디바이스들과 같은 광학 매체들; 플롭티컬(floptical) 디스크들과 같은 자기-광학 매체들, 및 ASIC

들(application-specific integrated circuits), PLD들(programmable logic devices), 및 ROM과 RAM 디바이스들과 같은, 프로그램 코드를 저장하고 실행하도록 특수하게 구성된 하드웨어 디바이스들을 들 수 있지만, 그것들로 제한되는 것은 아니다. 컴퓨터 코드의 일례들로는, 컴파일러에 의해 발생하는 것과 같은, 머신 코드 및 인터프리터를 사용해 컴퓨터에 의해 실행되는 하이레벨 코드를 포함하는 파일들을 들 수 있다. 예를 들어, 발명의 실시예는 자바, C++, 또는 다른 객체 지향 프로그래밍 언어 및 개발 도구들을 사용해 구현될 수 있다. 발명의 다른 실시예는, 머신-실행 가능 소프트웨어 명령어들 대신에 또는 머신-실행 가능 소프트웨어 명령어들과 조합하여, 하드와이어링된 회로로 구현될 수 있다.

<114> 상기 기술은, 설명의 목적들을 위해, 본 발명에 대한 완전한 이해를 제공하고자 특정한 명명법을 사용하였다. 그러나, 당업자라면, 발명을 실시하기 위해 구체적인 세부 사항들은 불필요하다는 것을 분명히 알 수 있을 것이다. 실질적으로, 이 기술은 발명의 임의 사양 또는 태양을 임의 실시예로 제한하는 것으로 판독되어서는 안되며; 오히려 실시예의 사양들 및 태양들은 다른 실시예들과 용이하게 교체될 수도 있다. 예를 들어, 도메인간 및 도메인내 테스트가 일반적으로 클록 도메인들의 관점에서 설명되었지만, 당업자라면, 그러한 테스트들이, 전력 도메인들과 같은, 도메인들의 임의의 다른 유형에 적용될 수도 있다는 것을 알 수 있어야 한다.

<115> 이와 같이, 발명의 특정 실시예들에 대한 상기 설명들은 예시 및 설명의 목적들을 위해 제시된다. 그것들은 총망라적이거나 발명을 개시된 정확한 형태들로 제한하려는 것이 아니고; 명백하게도, 상기 교수들의 관점에서 다수 변경들 및 변형들이 가능하다. 실시예들은 발명의 원리들 및 그것의 실제적인 애플리케이션들을 최선으로 설명하기 위해 선택되고 설명되었으며; 그것에 의해, 당업자들이라면, 발명 및 다양한 실시예들을, 예상되는 특정 사용에 적합한 다양한 변경들로서 이용할 수 있다. 특히, 여기에서 설명된 모든 이점이 발명의 실시예 각각에 의해 실현될 필요는 없으며; 오히려 임의의 특정 실시예가 앞서 논의된 이점들 중 하나 이상을 제공할 수도 있다. 다음의 청구항들 및 그것들의 등가물들이 발명의 범위를 정의한다.

발명의 효과

<116> 이와 같이, 본 발명에 따르면, 앞서 언급된 단점들을 최소화하는 시스템들, 구조들, 및 방법들이 제공되며, 적어도 지연-관련 장애들을 검출하는 옛-스피드 스캔-기반 테스트가 제공된다.

도면의 간단한 설명

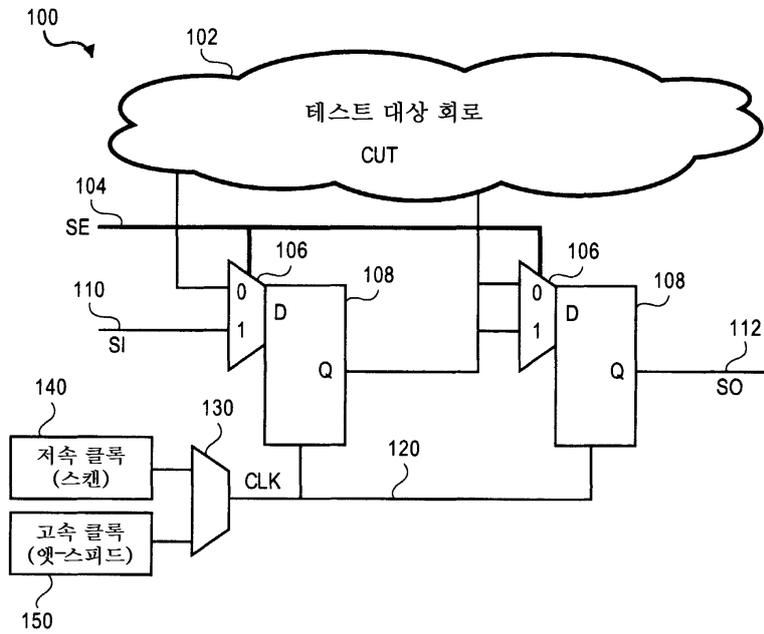
- <1> 도 1은 CUT(circuit under test)를 위한 전통적인 지연 테스트들을 구현하는데 흔히 사용되는 스캔 체인의 일부분을 나타내는 도면이다.
- <2> 도 2는 단일 스캔 인에이블(SE) 신호를 사용해 전통적인 스캔 체인 구조들에서 옛-스피드 지연 테스트를 수행하는 경우의 타이밍 불확실성을 예시한다.
- <3> 도 3은 전통적인 테스트 클록 제어 기술들을 사용해 옛-스피드 지연 테스트들을 구현하기 위한 통상적인 스캔 체인 구조를 예시한다.
- <4> 도 4는 도 3의 스캔 체인 구조를 사용하는 도메인간 로직의 테스트를 예시한다.
- <5> 도 5는 옛-스피드 클록 신호들을 스큐잉(skewing)하는 테스트 기능 클록 경로를 가진 전통적인 내부 테스트 클록 제어를 묘사하는 블록도이다.
- <6> 도 6a 및 도 6b는, 각각, 브로드사이드 및 LSL 테스트 프로토콜들을 위한 전통적인 구현들을 묘사한다.
- <7> 도 7은, 발명의 하나 이상의 특정 실시예에 따른, 회로를 테스트하기 위한 테스트 클록 제어 구조를 예시하는 블록도이다.
- <8> 도 8은, 발명의 하나 이상의 특정 실시예에 따른, 프로그램 가능한 테스트 클록 제어를 예시하는 블록도이다.
- <9> 도 9는, 발명의 실시예에 따른, 프로그램 가능한 테스트 클록 제어기의 일 기능의 일례를 예시하는 도면이다.
- <10> 도 10은 발명의 실시예에 따른 프로그램 가능한 테스트 클록 제어기의 일례를 예시하는 도면이다.
- <11> 도 11은, 발명의 실시예에 따른, 다수의 스캔 인에이블 제어 신호들에 기초한 프로그램 가능한 테스트 클록 제어기를 위한 동작의 상태들을 예시하는 도표이다.
- <12> 도 12a 및 도 12b는, 각각, 발명의 다양한 실시예들에 따른, 스캔 클록 디코더 및 프로그램 가능한 테스트 클록 제어기의 특정 구현들을 예시한다.

- <13> 도 13은, 발명의 일 실시예에 따른, 특정 클록 도메인을 프로그램하기 위해 도 12b의 프로그램 가능한 테스트 클록 제어기를 실시하기에 적합한 클록 펄스 제어기를 위한 상태도를 묘사한다.
- <14> 도 14a 및 도 14b는, 각각, 발명의 다양한 실시예들에 따른, 개시 및 포착 동작들을 위한 테스트 클록을 구성하도록 그리고 그것이 반응하는 클록 도메인을 디스에이블하도록, 프로그램 가능한 테스트 클록 제어기를 구성하기 위한 신호들의 타이밍을 예시한다.
- <15> 도 15는, 발명의 일 실시예에 따른, 간략화된 클록 펄스 제어기 및 간략화된 테스트 클록 발생기를 포함하는 프로그램 가능한 테스트 클록 제어기를 묘사한다.
- <16> 도 16은, 발명의 일 실시예에 따른, 보충 클록 명령 소스(supplemental clock command source)를 포함하는 프로그램 가능한 테스트 클록 제어기를 묘사한다.
- <17> 도 17a 및 도 17b는, 각각, 발명의 다양한 실시예들에 따른, 스트링 발생기로서 구현되는 보충 클록 명령 소스 및 대응되는 상태도를 예시한다.
- <18> 도 18은, 발명의 실시예에 따른, 전자 디바이스 및 그것의 회로들을 테스트하기 위한 임의 갯수의 프로그램 가능한 테스트 클록 제어기들을 구현하는 개념적인 테스트 클록 제어 구조를 나타내는 블록도를 묘사한다.
- <19> 도 19는, 발명의 실시예에 따른, 상이한 도메인들의 회로들을 테스트하기 위한 프로그램 가능한 테스트 클록 제어기들을 구현하는 테스트 클록 제어 구조를 나타내는 도면이다.
- <20> 도 20은, 발명의 실시예에 따른, 테스트 접근점들(test access points)을 감소시키기 위해 일차 입력들 및 일차 출력들을 공유하는 제어 체인들 및 스캔 체인들을 나타내는 도면이다.
- <21> 도 21은, 발명의 일 실시예에 따른, 도메인간 테스트를 수행하도록 구성된 테스트 클록 제어 구조를 예시한다.
- <22> 도 22a 및 도 22b는, 발명의 다양한 실시예들에 따른, 도메인간 테스트를 구현하기 위한 하나 이상의 타이밍 기준점들을 사용해 시간적 거리들(temporal distances)을 계산하기 위한 다양한 기술들을 예시한다.
- <23> 도 23은, 발명의 일 실시예에 따른, 도메인간 테스트를 수행하도록 구성된 테스트 클록 제어 구조를 예시한다.
- <24> 도 24는, 발명의 일 실시예에 따른, 간략화된 프로그램 가능한 테스트 클록 제어기들을 사용해 도메인간 테스트를 수행하도록 구성된 테스트 클록 제어 구조를 예시한다.
- <25> 도 25a 내지 도 25d는, 발명의 일 실시예에 따른, 프로그램 가능한 테스트 클록 제어기들을 사용해 도메인간 테스트 및 도메인내 테스트를 용이하게 하도록 구성된 테스트 클록 제어 구조들을 예시한다.
- <26> 도 26은, 본 발명의 실시예에 따른, 프로그램 가능한 테스트 클록 제어기들(PTCC들)을 구현하는 테스트 클록 제어 구조들을 사용해 회로들을 테스트하기 위한 마이크로-레벨 흐름의 일례를 나타낸다.
- <27> 도 27은, 발명의 실시예에 따른, 테스트 클록 제어기 구조들 및 PTCC들을 사용해 회로들을 테스트하기 위한 매크로-레벨 흐름의 일례를 나타낸다.
- <28> <도면의 주요 일부분에 대한 부호의 설명>
- <29> 700 : 프로그램 가능한 테스트 클록 제어기
- <30> 701 : 테스트 클록 제어 구조
- <31> 703 : CUT(circuit under test)
- <32> 704 : 테스트 클록 발생기
- <33> 714 : CTC(구성 가능한 테스트 클록)
- <34> 720 : 제어 계층
- <35> 722 : 제어 체인
- <36> 724 : 제어 정보
- <37> 730 : 스캔 계층
- <38> 732 : 스캔 체인

도면

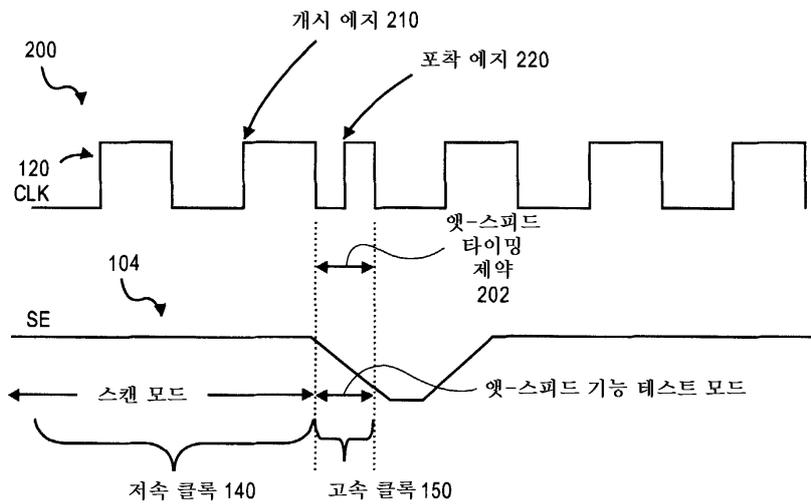
도면1

(종래 기술)



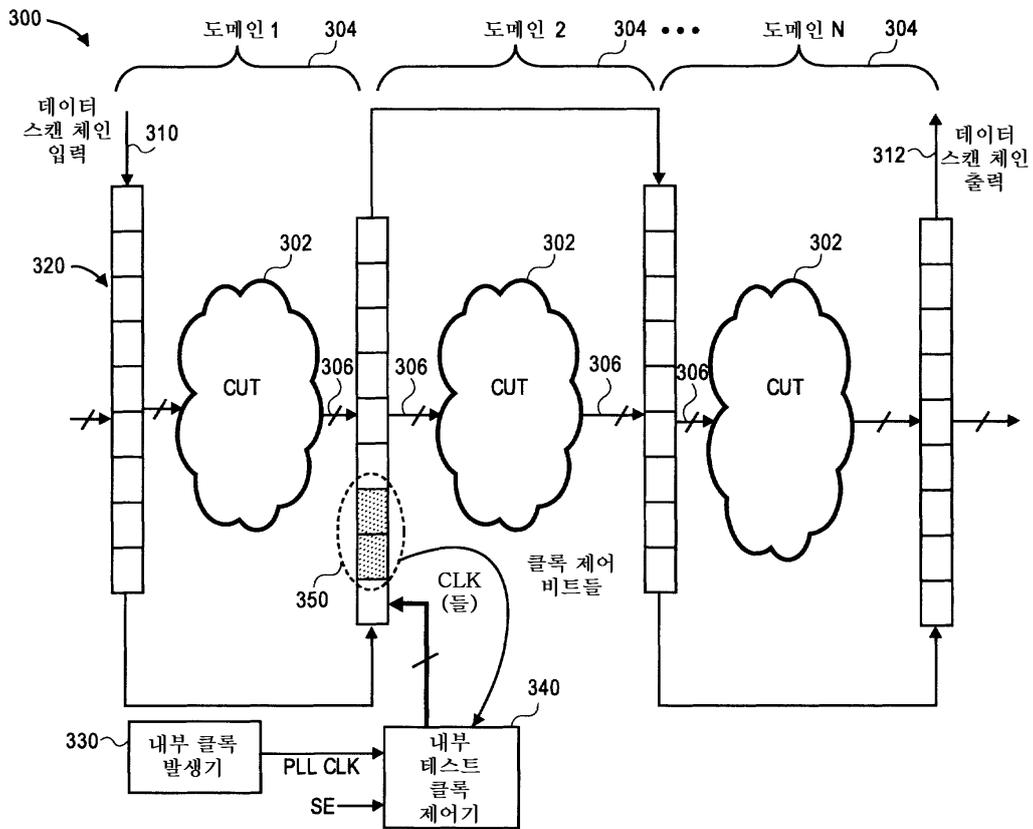
도면2

(종래 기술)



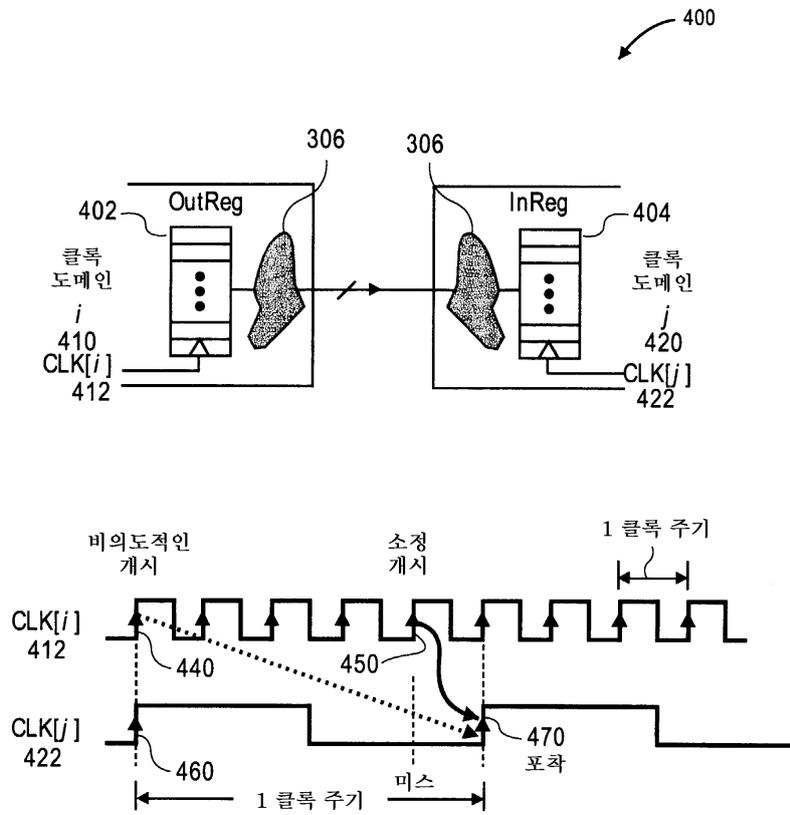
도면3

(종래 기술)



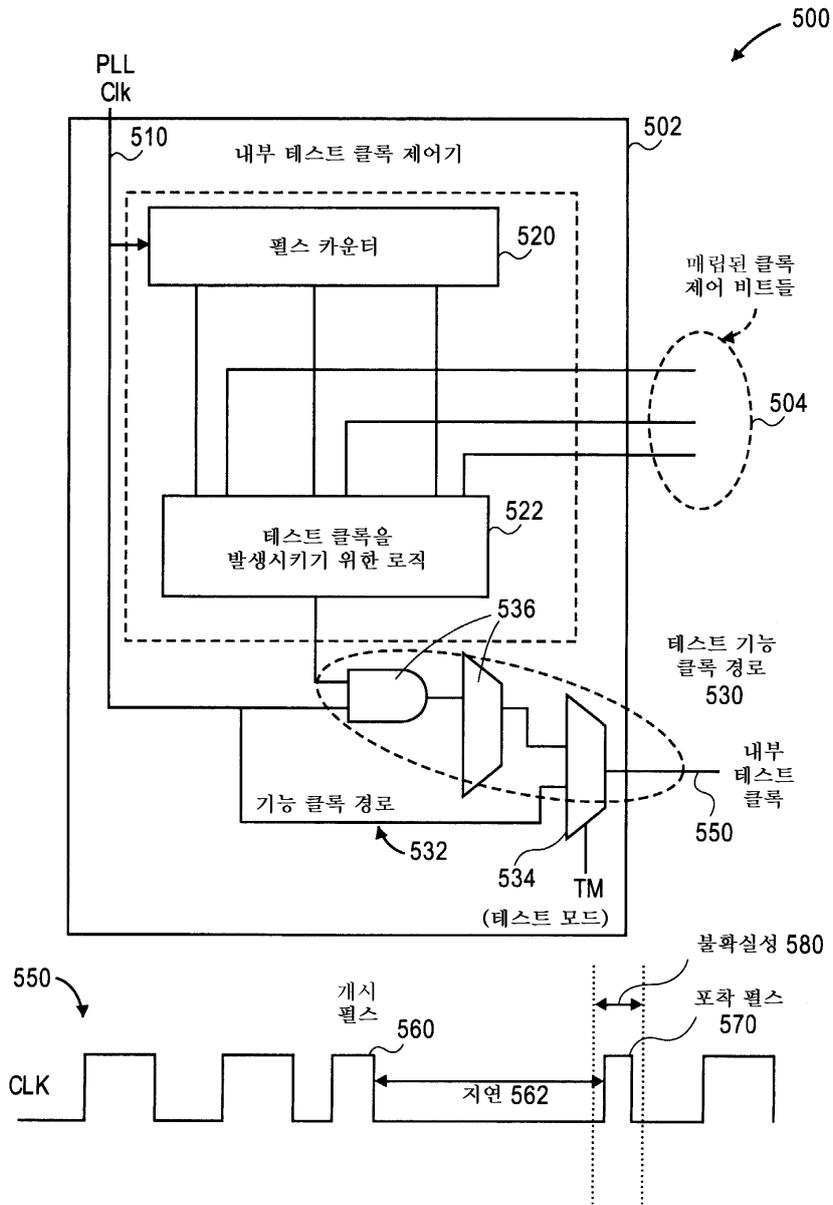
도면4

(종래 기술)

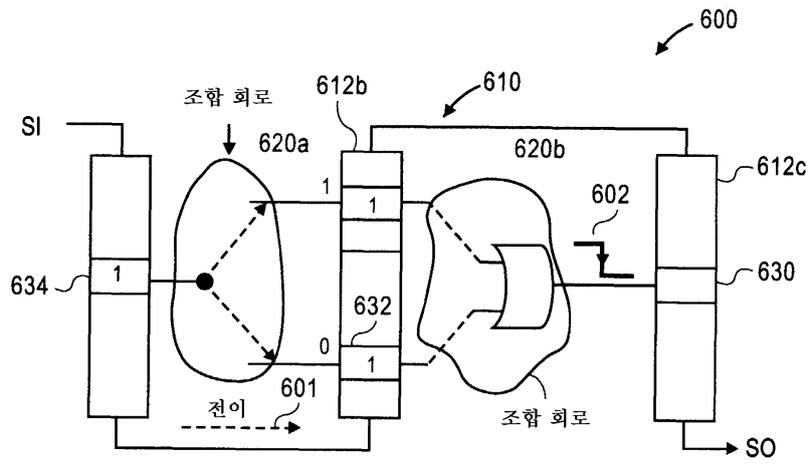


도면5

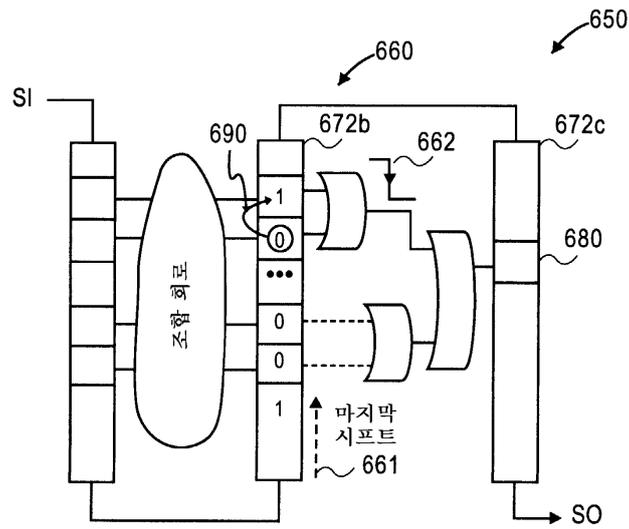
(종래 기술)



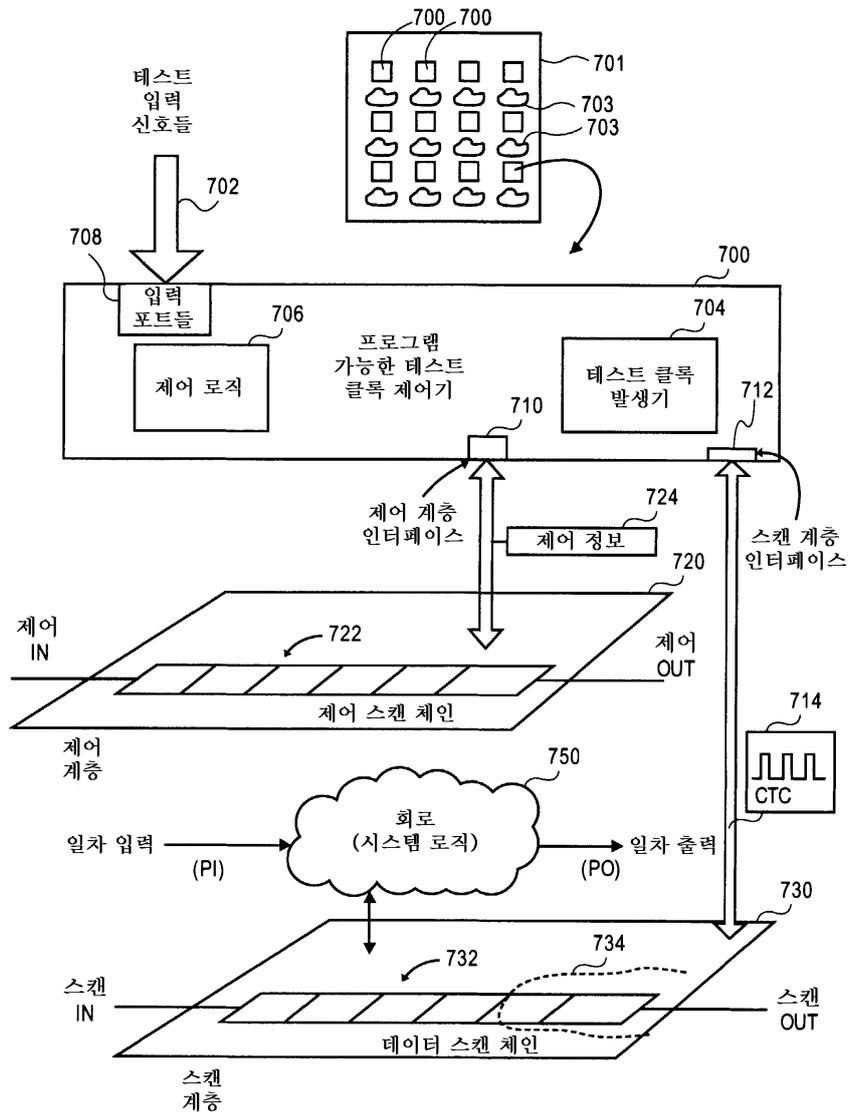
도면6a



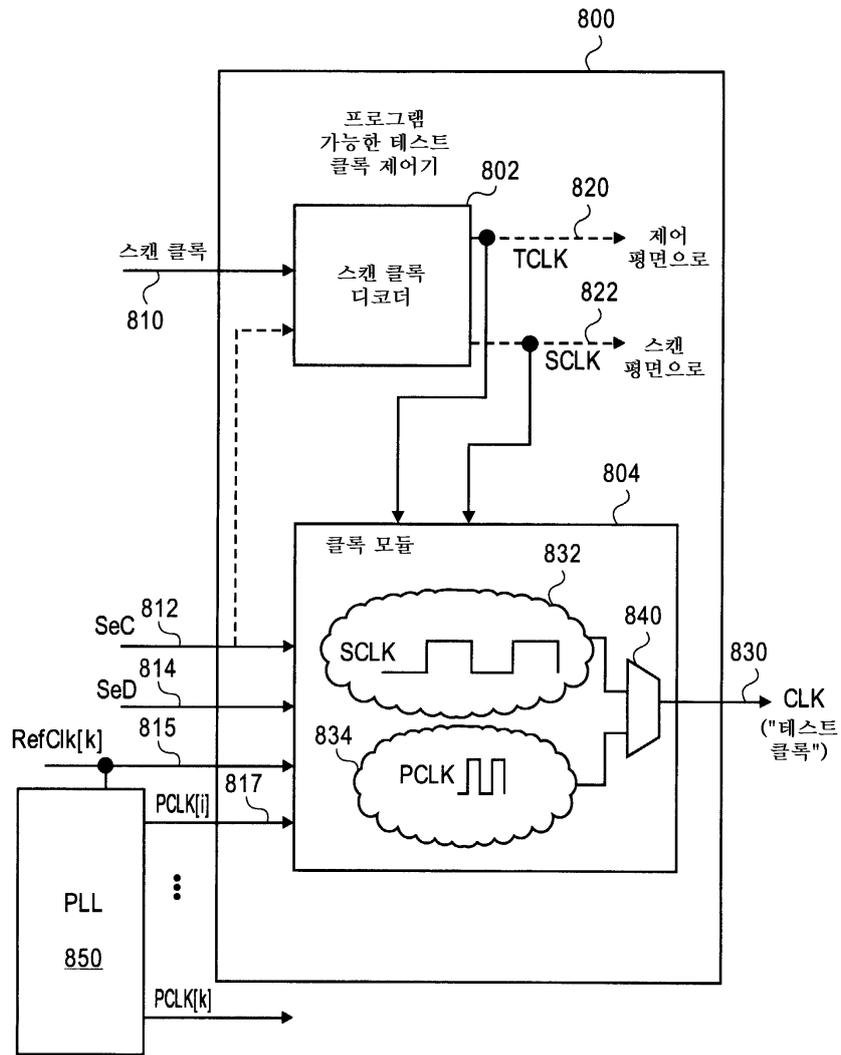
도면6b



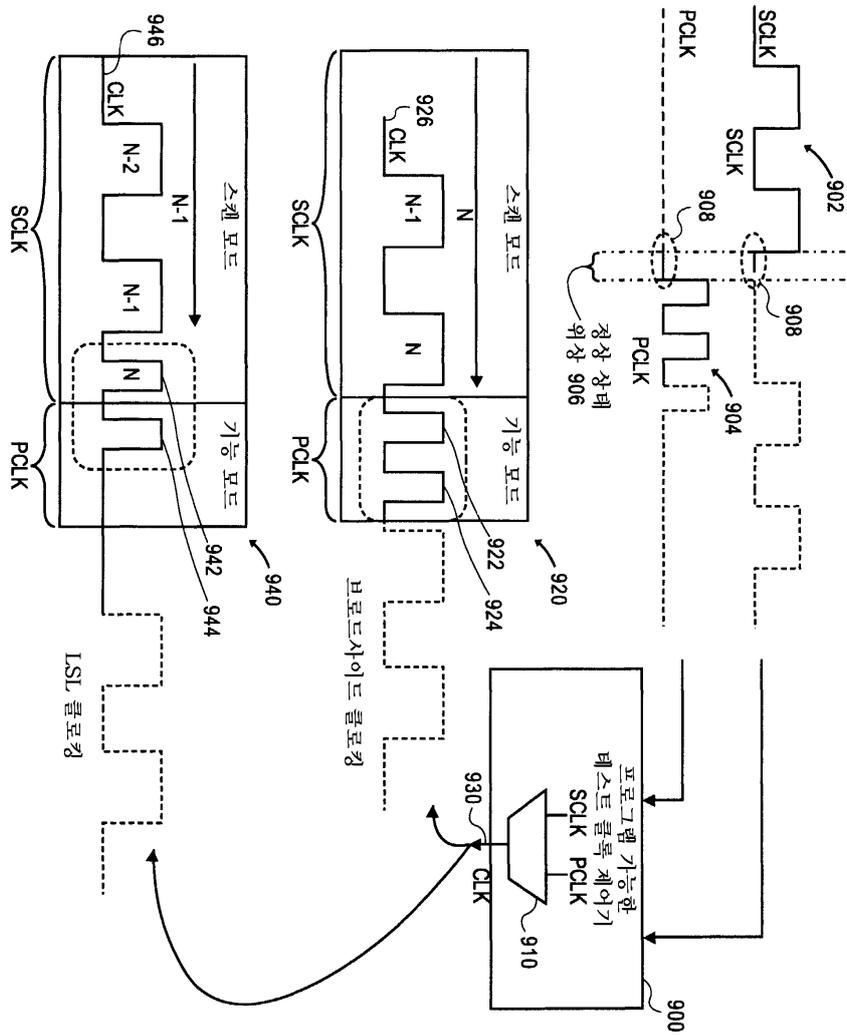
도면7



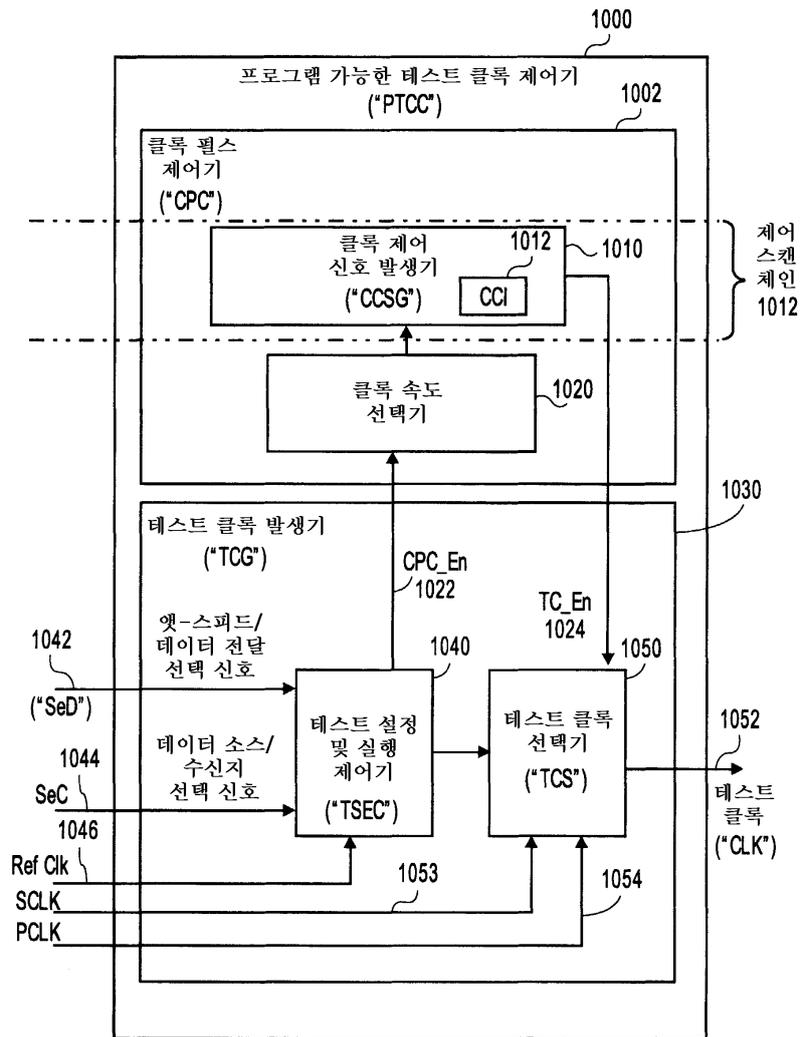
도면8



도면9



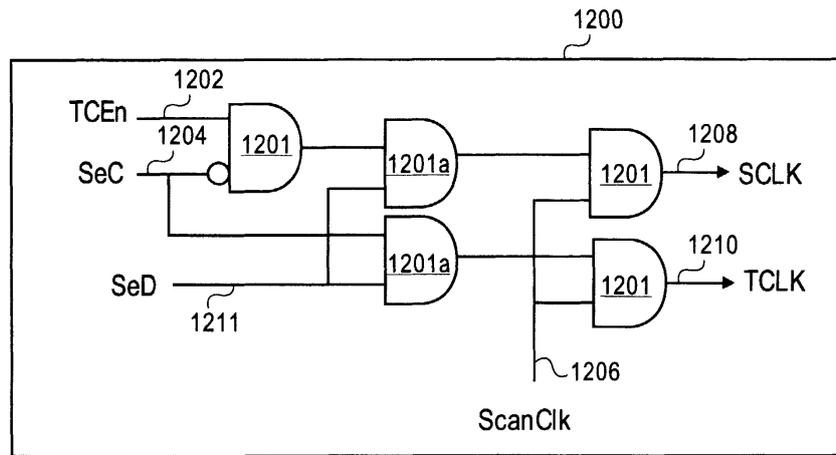
도면10



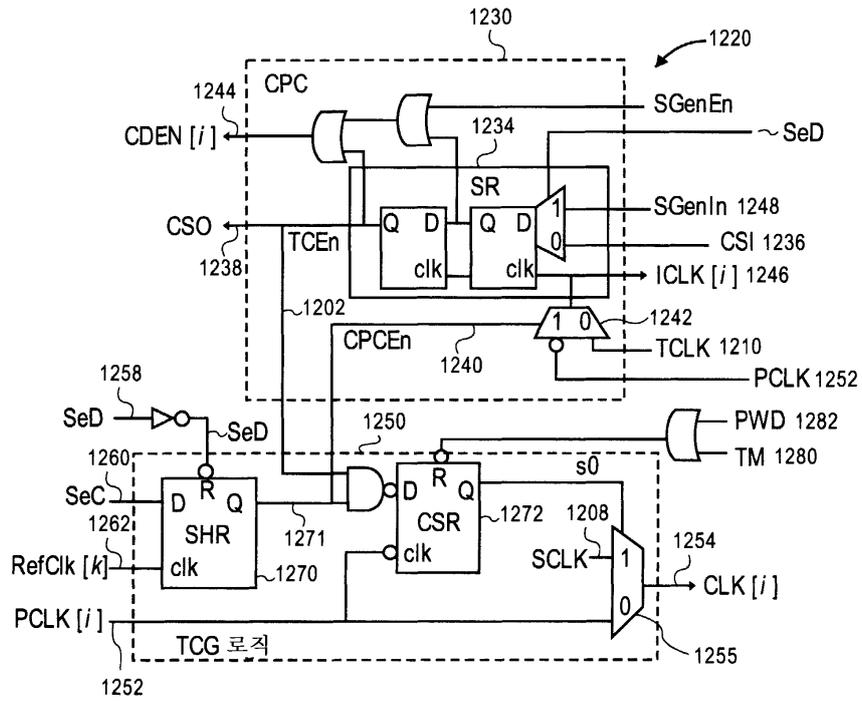
도면11

SeD	SeC	상태
0	0	사일런트
0	1	개시/포착
1	0	스캔 로드/언로드
1	1	제어 로드/언로드

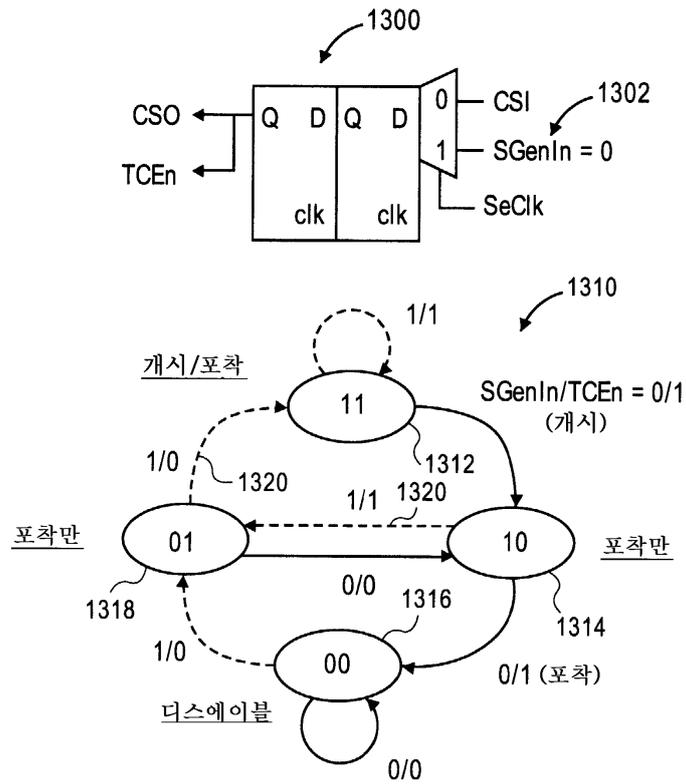
도면12a



도면12b

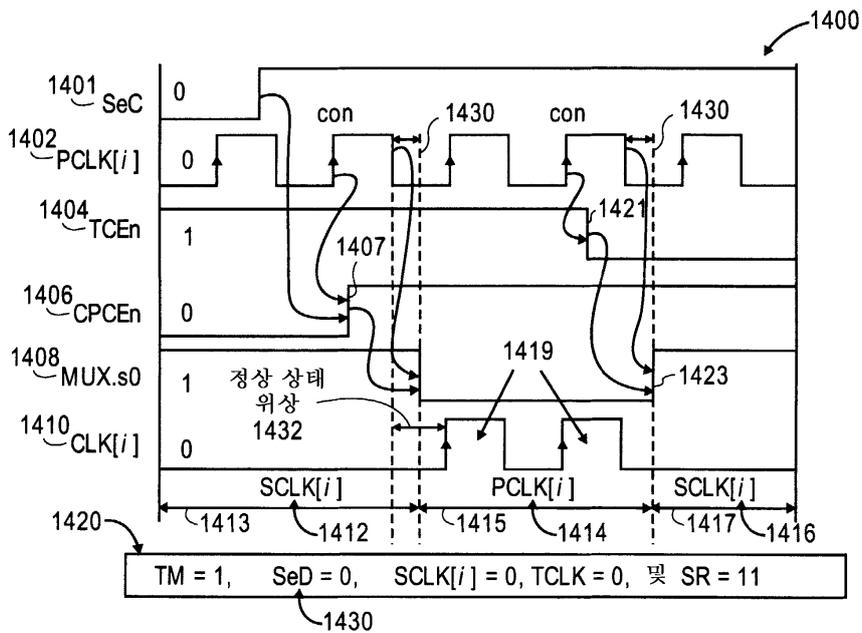


도면13

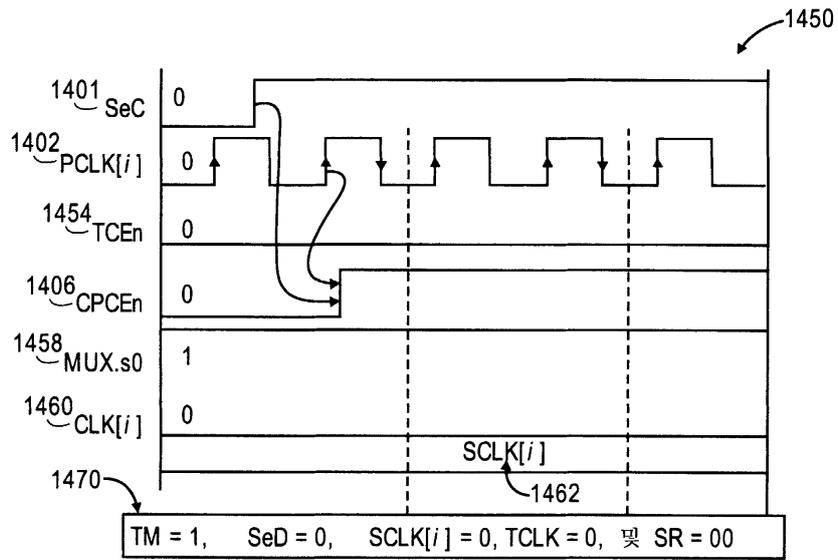


개시/포착을 위한 SR의 상태도

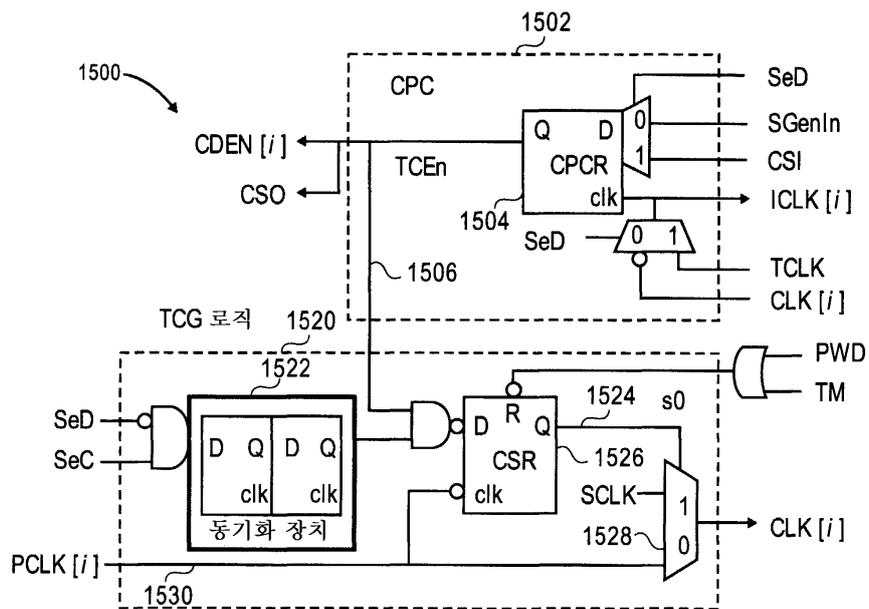
도면14a



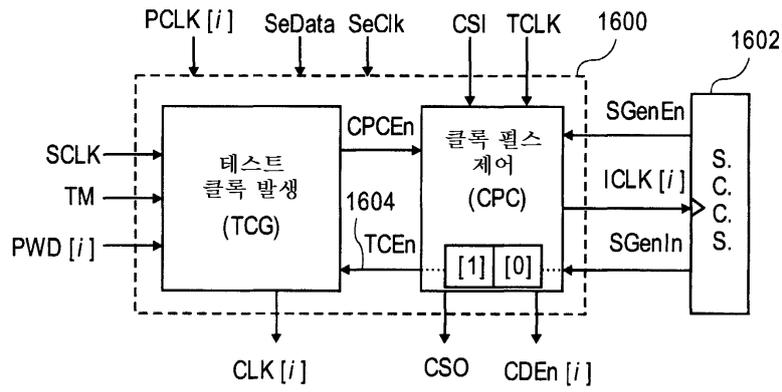
도면14b



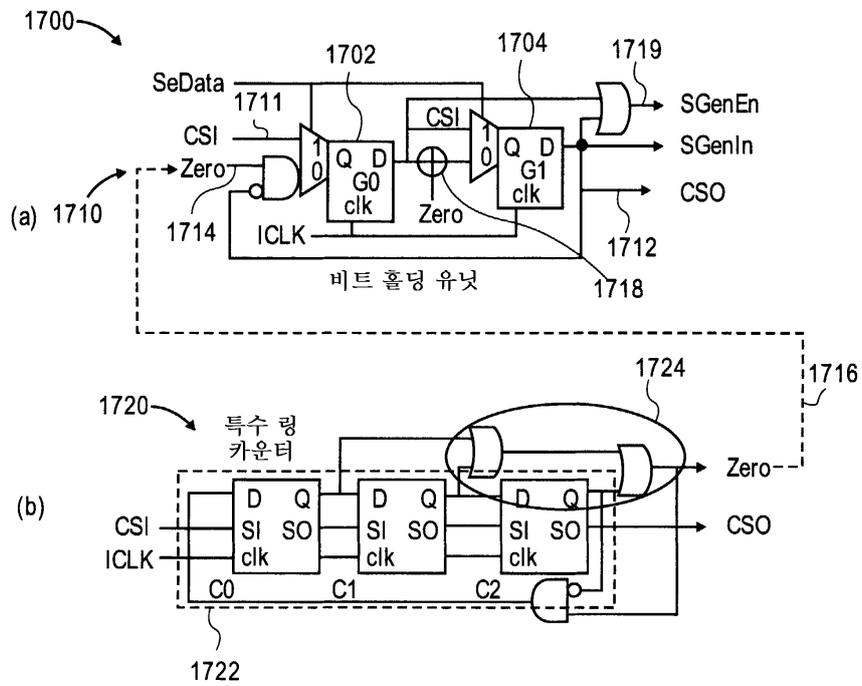
도면15



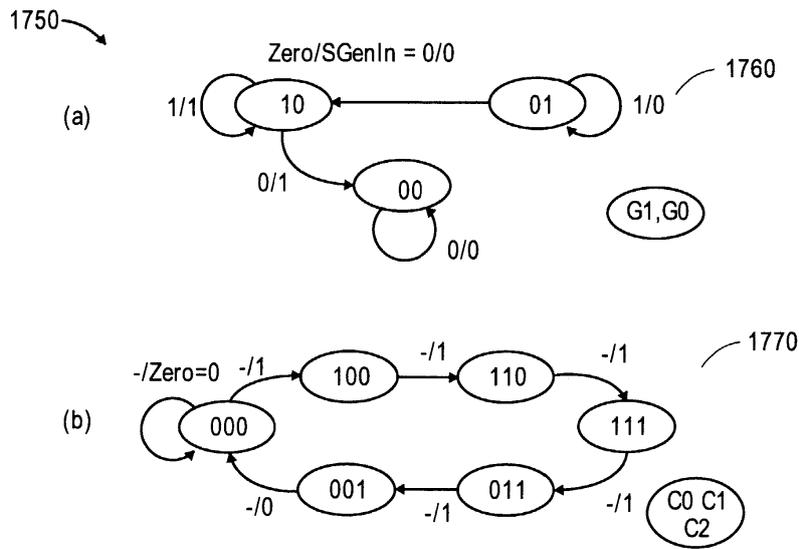
도면16



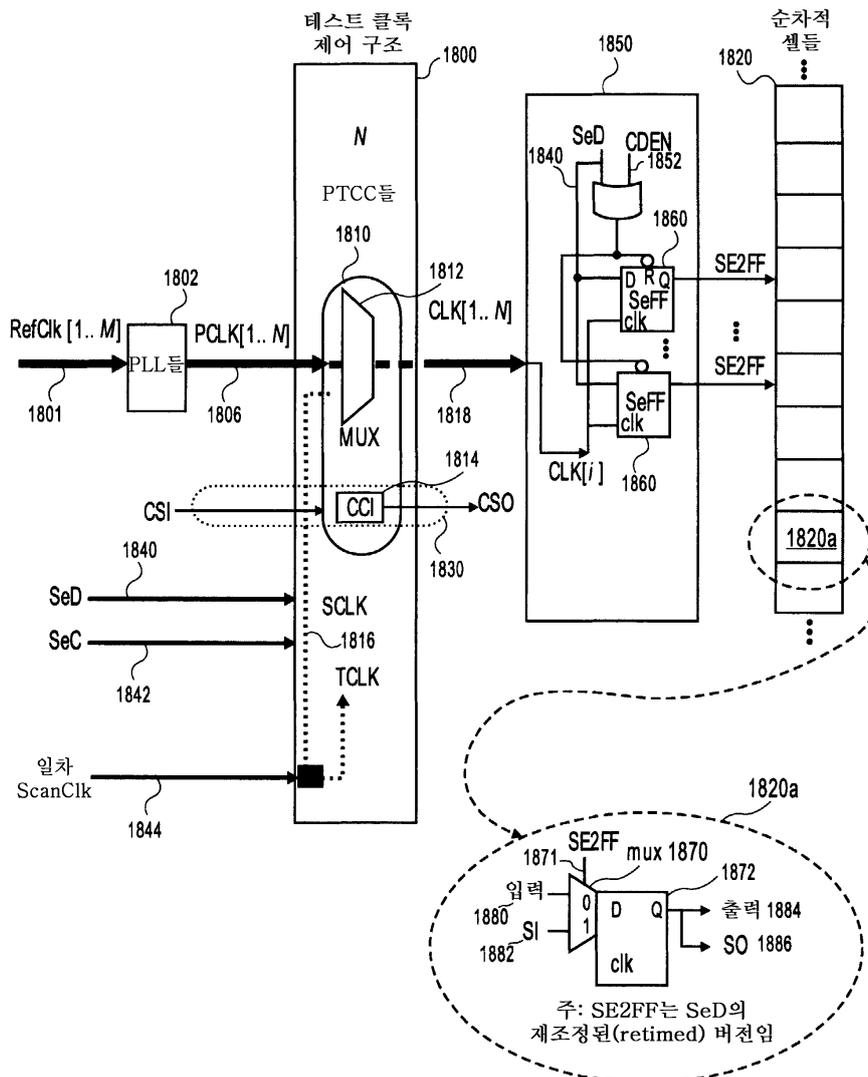
도면17a



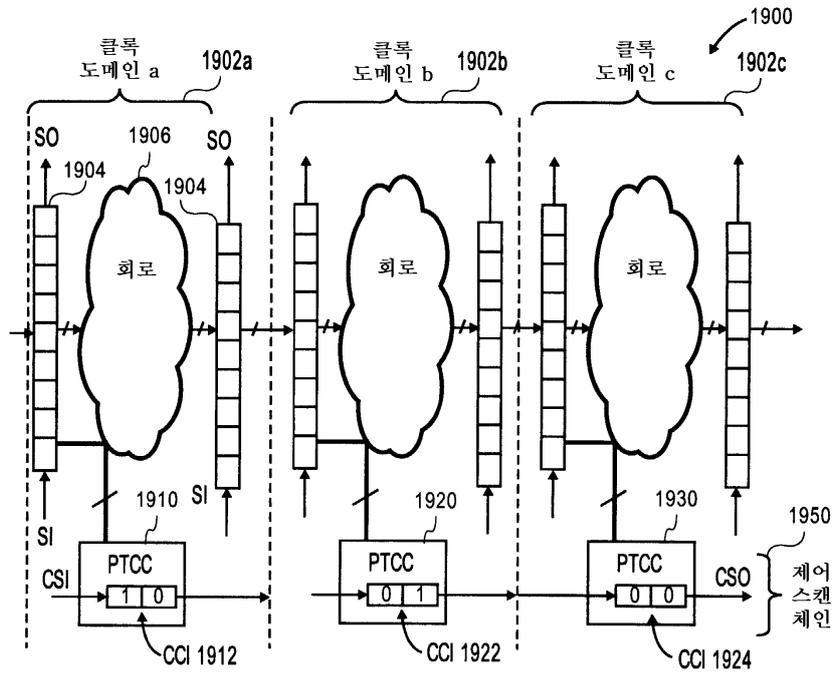
도면17b



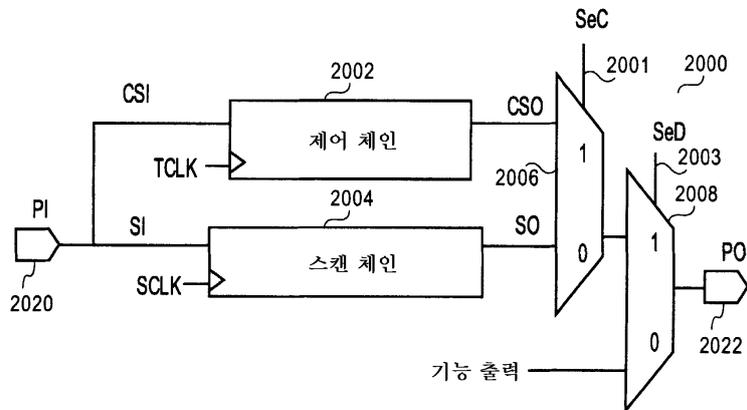
도면18



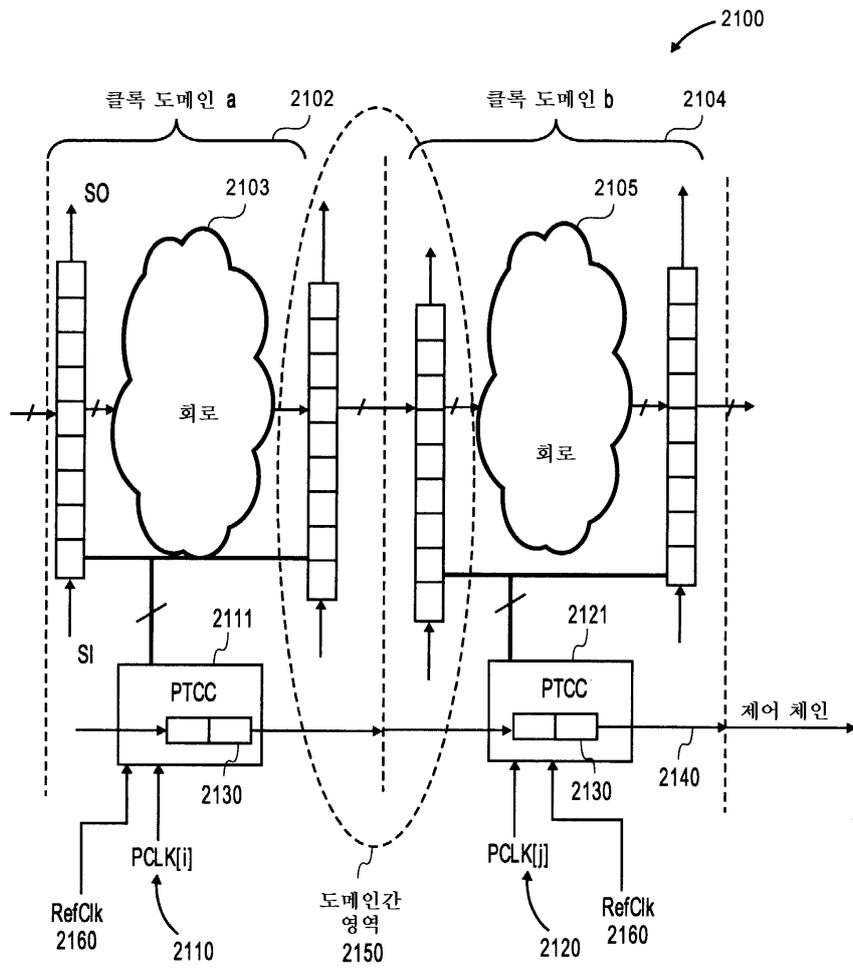
도면19



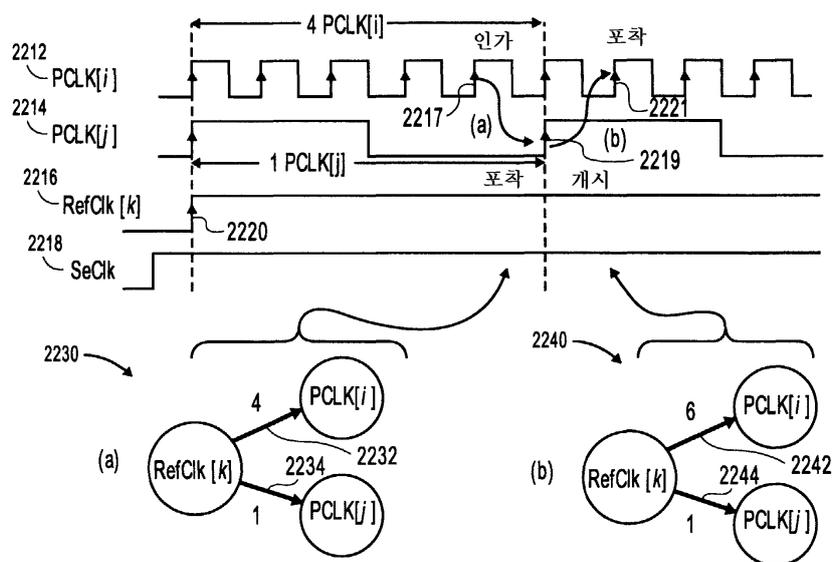
도면20



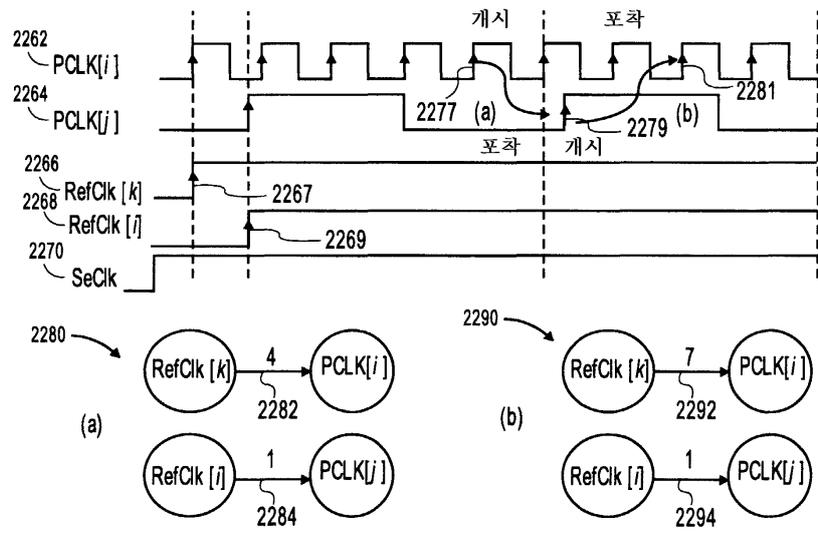
도면21



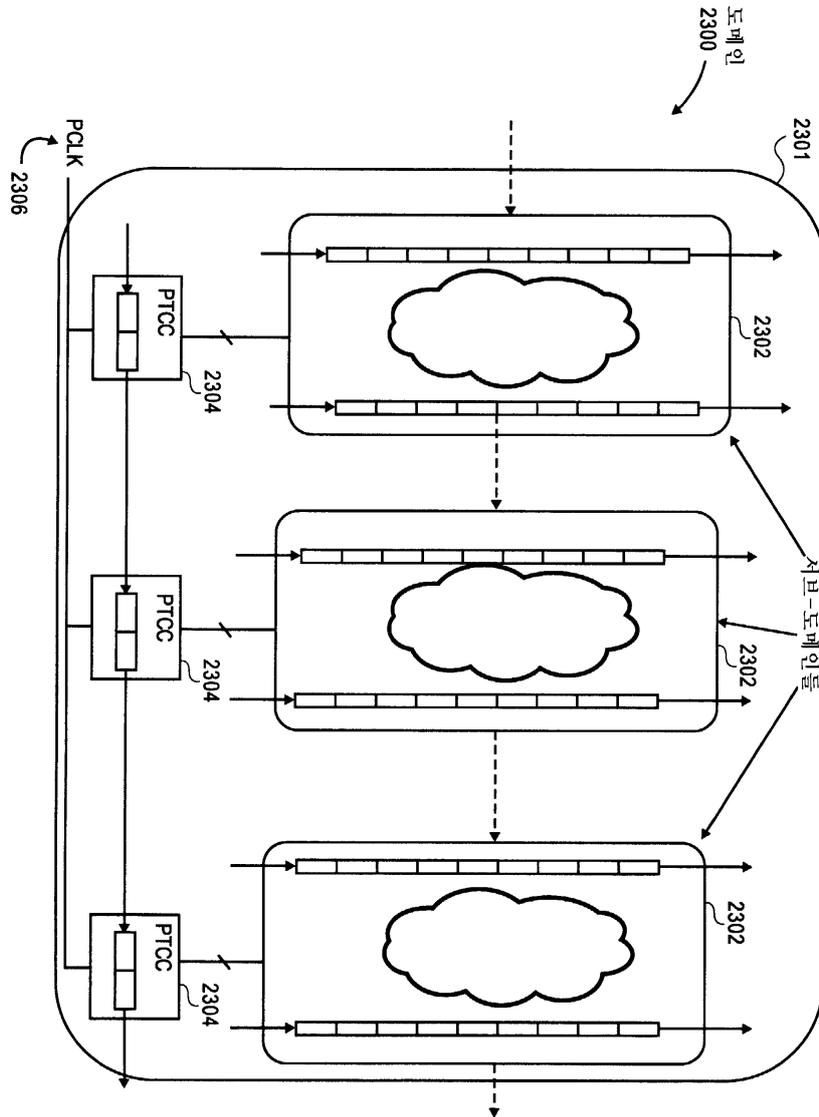
도면22a



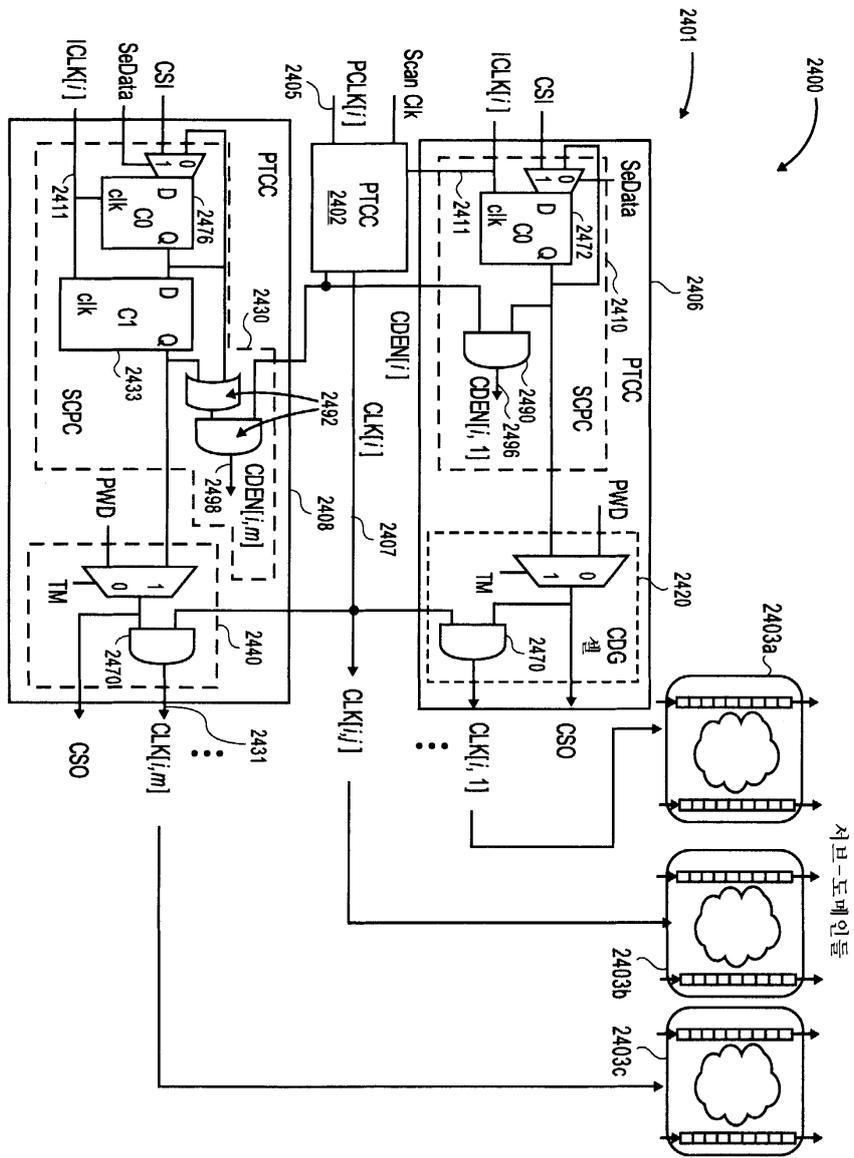
도면22b



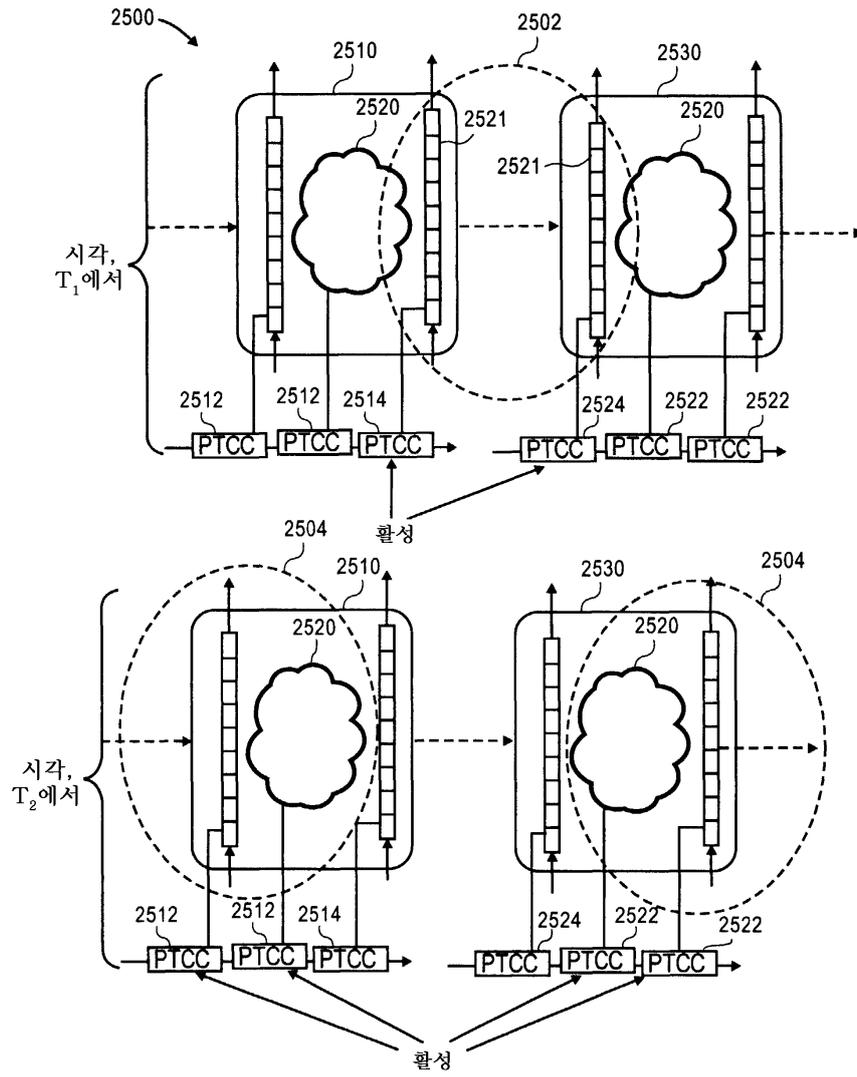
도면23



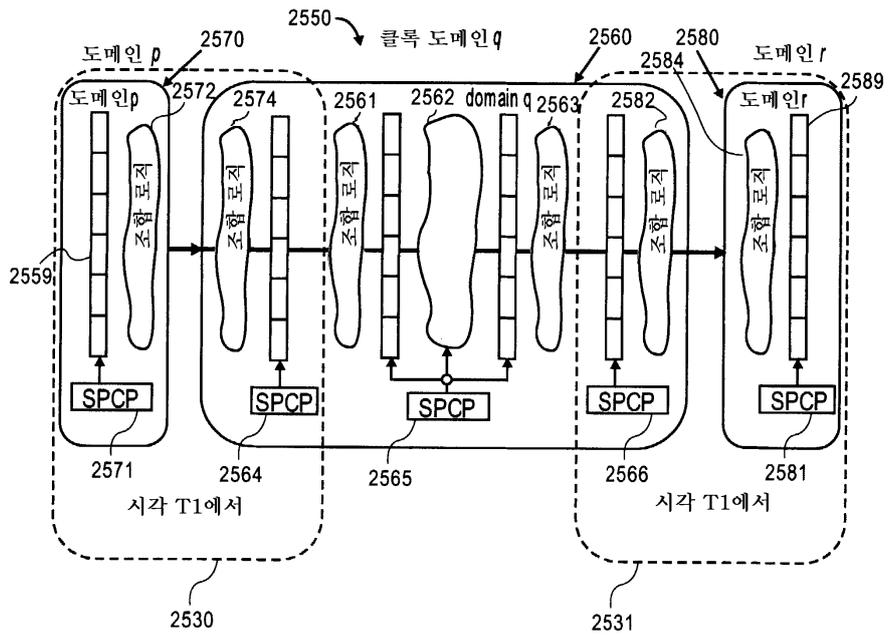
도면24



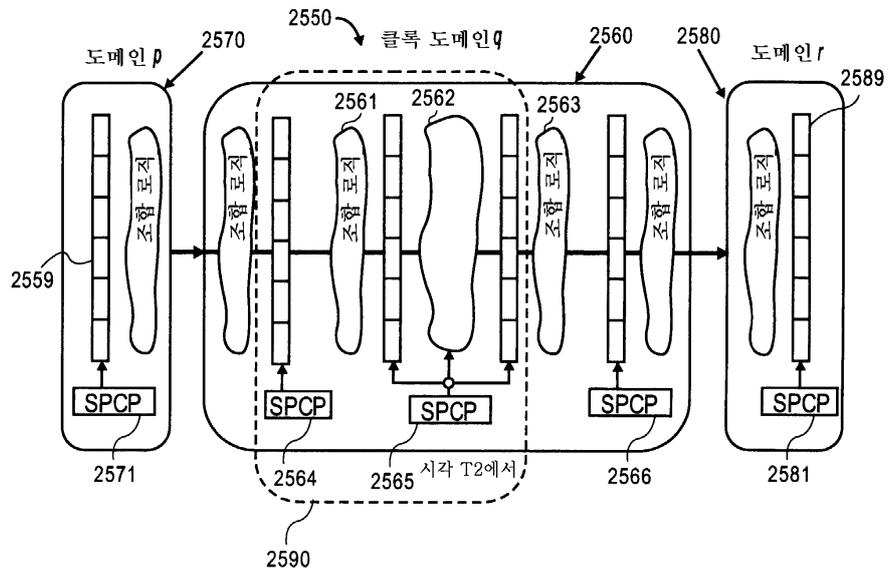
도면25a



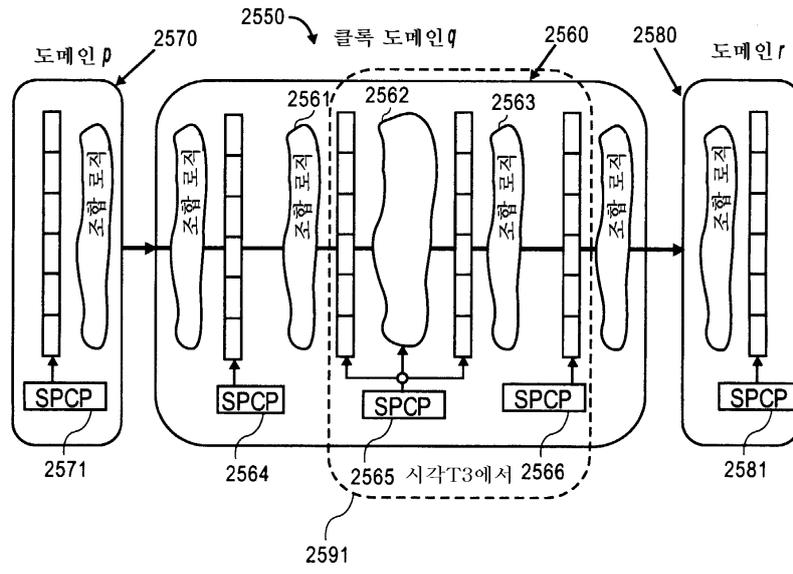
도면25b



도면25c

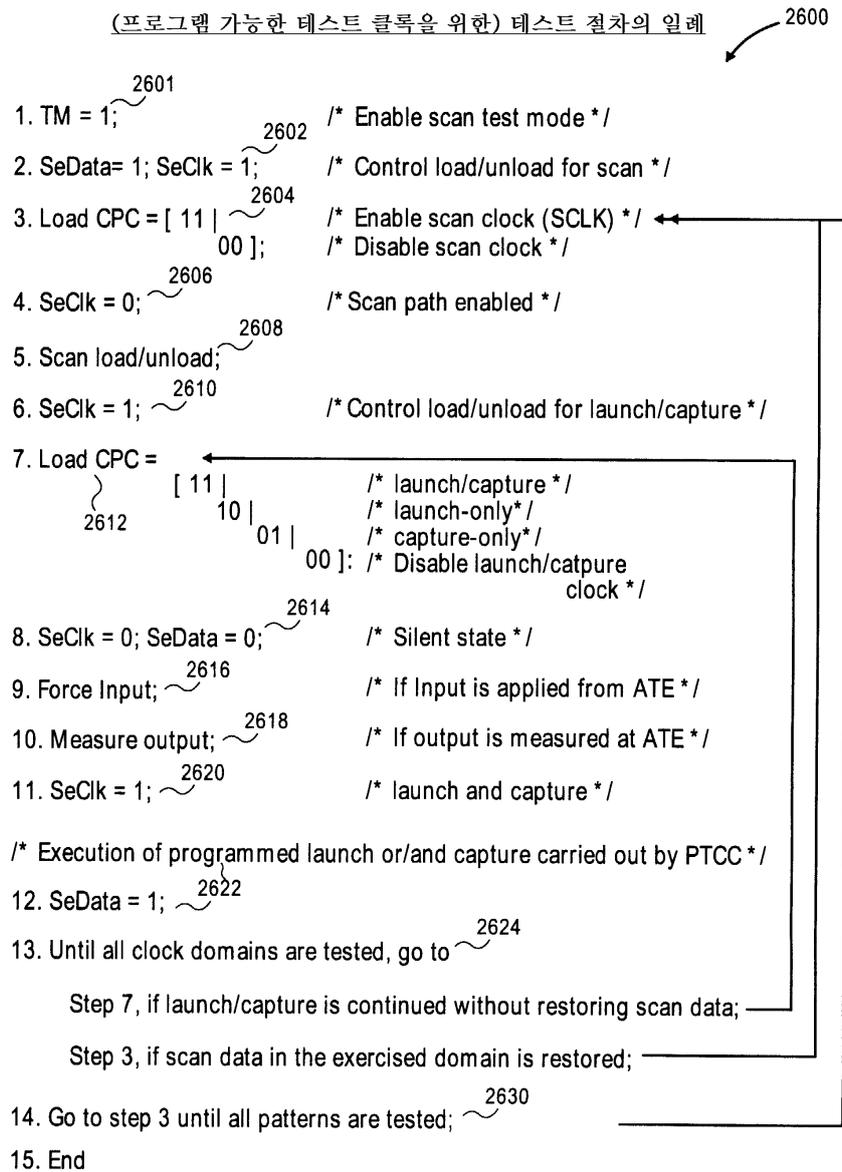


도면25d



도면26

(프로그램 가능한 테스트 클록을 위한) 테스트 절차의 일례



도면27

권장되는 ATPG 흐름

