

(19)대한민국특허청(KR)
(12) 공개특허공보(A)

(51) Int. Cl.⁷

G06F 11/30
G06F 15/00
G21C 17/00

(11) 공개번호 10-2005-0086587

(43) 공개일자 2005년08월30일

(21) 출원번호 10-2005-7008534

(22) 출원일자 2005년05월12일

번역문 제출일자 2005년05월12일

(86) 국제출원번호 PCT/US2003/035336

(87) 국제공개번호 WO 2004/044749

국제출원일자 2003년11월05일

국제공개일자 2004년05월27일

(30) 우선권주장 10/293,734 2002년11월12일 미국(US)

(71) 출원인 소닉스, 인코퍼레이티드
미국 캘리포니아 94040 마운틴 뷰 수트 620 웨스트 엘 카미노 리얼 2440

(72) 발명자 에버트 제프리 앨런
미국 94019 캘리포니아주 하프 문 베이 세인트 요셉 애비뉴 415
베누고파란 라비
미국 95051 캘리포니아주 산타 클라라 홈스테드 로드 3455 넘버66
에반스 스콧 칼턴
미국 95051 캘리포니아주 산타 클라라 헤이스 애비뉴 252

(74) 대리인 특허법인코리아나

심사청구 : 없음

(54) 구성가능한 하드웨어를 분해 및 검증하는 방법 및 장치

요약

본 발명은 구성가능한 하드웨어를 분해 및 검증하는 방법 및 장치를 포함한다. 일 실시형태에서, 그 방법은 하드웨어 시스템을 하나 이상의 유닛들의 세트로 자동으로 분해하는 단계, 그 유닛들의 세트 각각에 대한 테스트벤치를 생성하는 단계, 및 하드웨어 시스템 설계를 검증하기 전에 그 유닛들의 세트 각각을 검증하는 단계를 포함한다.

명세서

발명의 기술분야

본 발명은 하드웨어 검증에 관한 것이다. 좀더 자세하게는, 본 발명은 구성가능한 하드웨어 (configurable hardware) 의 검증에 관한 것이다.

발명의 배경

"구성가능한 하드웨어" 또는 "파라미터화된 하드웨어"는 파라미터 또는 속성 세트에 대한 특정 값을 사용함으로써 설계 생성 시에 자동으로 맞춤제작되는 하드웨어 시스템을 나타낸다. 또한, 그러한 하드웨어는 파라미터 세팅에 의존하는 런-타임(run-time) 시의 변화를 지원한다. 통상적으로, 구성가능한 하드웨어 시스템은 범용 컴퓨터 시스템에서 구동하는 소프트웨어보다 더 우수한 성능을 제공하며, 회로 사이즈 및 비용을 증가시키지 않고도 종래의 주문형 집적회로(ASIC)보다 더 큰 융통성을 제공한다.

종래의 하드웨어 시스템에서는, 시스템 및 그 컴포넌트들을 테스트함으로써 시스템의 기능을 검증하는 것이 필요하다. 통상적으로, 시스템의 기능을 검증하는 것의 복잡도는 시스템을 구성하는 컴포넌트의 수에 따라 증가한다. 따라서, 종래의 방법은 각각의 유닛을 개별적으로 수동으로 검증한 후, "공지된 양호한 유닛"을 시스템에 조립하는 것이다. 만약 하드웨어가 계층적으로 배열되면, 그 계층의 각각의 레벨에 대하여 검증을 수행해야 한다. 시스템을 조립하기 전에 각 개별 유닛이 검증되었으면, 시스템 기능을 검증하는 것은 각 컴포넌트의 능력보다는 컴포넌트들 간의 상호작용에 대한 잠재적인 문제에 초점을 맞출 수 있다.

구성가능한 하드웨어 시스템은 이러한 타입의 종래의 계층적인 분해를 이용하여 검증될 수 있다. 그러나, 구성가능한 하드웨어 시스템의 각 인스턴스(instance)가 상이하기 때문에, 구성 파라미터가 변경될 때마다, 시스템 및 그 컴포넌트들은 수동으로 검증되어야 한다. 시스템 및 그 컴포넌트들을 반복적으로 수동으로 검증하는 것의 비용은, 종종, 구성가능한 하드웨어의 이점을 상쇄한다.

발명의 요약

본 발명은 구성가능한 하드웨어를 분해 및 검증하는 방법 및 장치를 포함한다. 일 실시형태에서, 그 방법은 하드웨어 시스템을 하나 이상의 유닛들의 세트에 자동으로 분해하는 단계, 그 유닛들의 세트 각각에 대한 테스트벤치(testbench)를 생성하는 단계, 및 하드웨어 시스템 설계를 검증하기 전에 그 유닛들의 세트 각각을 검증하는 단계를 포함한다.

도면의 간단한 설명

본 발명은 첨부 도면에 제한되지 않고 예로써 설명되며, 도면에서 동일한 도면부호는 동일한 대상을 나타낸다.

도 1은 본 발명의 일 실시형태에 따른 구성가능한 하드웨어를 분해 및 검증하기 위한 블록도이다.

도 2는 본 발명의 일 실시형태에 따른 구성가능한 하드웨어를 생성 및 검증하는 시스템을 나타낸 블록도이다.

도 3은 본 발명의 실시형태들에 따른 시스템의 생성을 나타낸 흐름도이다.

도 4는 본 발명의 실시형태들에 따른 시스템 설계의 개념적인 블록도이다.

도 5는 본 발명의 실시형태들에 따른 구성가능한 하드웨어 시스템을 분해 및 검증하기 위한 동작들을 나타낸 블록도이다.

도 6은 본 발명의 실시형태들에 따른 유닛을 검증하기 위한 테스트벤치를 나타낸 블록도이다.

도 7은 본 발명의 실시형태들에 따른 구성가능한 하드웨어를 분해 및 검증하기 위한 예시적인 시스템을 나타낸 것이다.

상세한 설명

구성가능한 하드웨어를 분해 및 검증하는 방법 및 장치가 설명된다. 이 명세서에서, "하나의 실시형태" 또는 "일 실시형태"에 대한 참조는, 참조되는 특성이 본 발명의 하나 이상의 실시형태에 포함된다는 것을 의미한다. 또한, 이 명세서에서 "일 실시형태"에 대한 별도의 참조는 반드시 동일한 실시형태를 말하는 것은 아니지만, 그렇게 말해지지 않고 당업자가 명백히 알 수 있는 것을 제외하면, 그러한 실시형태들이 상호 배타적이지는 않다. 따라서, 본 발명은 여기에서 설명되는 실시형태들의 임의의 다양한 조합 및/또는 통합을 포함할 수 있다.

여기에서, 블록도는 본 발명의 예시적인 실시형태들을 나타낸다. 또한, 여기서, 흐름도는 본 발명의 예시적인 실시형태들의 동작을 나타낸다. 흐름도의 동작은, 블록도에 도시되어 있는 예시적인 실시형태들을 참조하여 설명된다. 그러나, 흐름도의 동작은 블록도를 참조하여 설명되는 것과는 다른 본 발명의 실시형태들에 의해 수행될 수도 있으며, 블록도를 참조하여 설명되는 실시형태들은 흐름도를 참조하여 설명된 것과는 상이한 동작을 수행할 수도 있다.

개관

본 발명의 일 실시형태에서, 생성 및 검증 유닛은 구성 데이터 및 구성가능한 하드웨어 라이브러리에 기초하여 구성가능한 하드웨어 시스템을 생성한다. 구성가능한 하드웨어 시스템은 다수의 유닛으로 이루어진다. 본 발명의 일 실시형태에서, 생성 및 검증 유닛은, 구성가능한 하드웨어 시스템을 그 시스템 설계를 구성하는 유닛들로 계층적으로 분해한다. 구성 데이터는 각각의 유닛에 적용되어, 유닛 자체를 변경하지 않고 시스템의 외부부를 제거 및 검증 또는 분석받을 수 있다. 생성 및 검증 유닛은 테스트 벤치, 테스트, 및 각 유닛에 대한 제어 스크립트를 생성한다.

예시적인 구조

도 1 은 본 발명의 일 실시형태에 따른 구성가능한 하드웨어를 분해 및 검증하기 위한 블록도이다. 도 1 은 생성 및 검증 유닛 (103) 을 포함하며, 이 생성 및 검증 유닛 (103) 은 생성 모듈 (107), 분해 모듈 (109) 및 검증 모듈 (111) 을 더 포함한다. 생성 및 검증 유닛 (103) 은 구성가능한 하드웨어 라이브러리 (101) 및 구성 데이터 저장 유닛 (105) 에 접속된다. 구성 데이터 저장 유닛 (105) 은 구성 데이터 (113) 를 포함한다. 생성 및 검증 유닛 (103) 은 구성가능한 하드웨어 시스템 (104) 및 그 구성 유닛 (유닛 (1 내지 N) 으로서 도시되어 있음) 을 생성한다.

구성가능한 하드웨어 시스템 설계 계층에서, "시스템" 이라는 용어는 특정 계층 레벨에서의 유닛들의 합성 (composition) 을 말하며, 여기서, 유닛들의 세부사항은 은닉된다. 따라서, 구성가능한 하드웨어 시스템 설계 계층의 특정 레벨에서, 유닛들은 분할할 수 없는 컴포넌트이다. 그러나, 하위 계층 레벨에서, 상위 레벨로부터의 유닛들은 그 세부사항 및 노출된 내부 컴포넌트들을 가진다. 예를 들어, 도 1 을 참조하면, 일 설계 계층 레벨에서, 시스템 (104) 은 더 큰 시스템의 "블랙 박스" 유닛으로서 간주되며, 여기서, 유닛들 (1 내지 N) 에 대한 세부사항은 은닉된다. 그러나, 하위 설계 계층 레벨에서, 시스템 (104) 은 유닛들 (1 내지 N) 을 포함하는 것으로서 간주되며, 여기서, 유닛 접속 세부사항은 노출된다. 구성가능한 하드웨어 시스템 설계 계층의 하위 레벨에서도, 유닛들 (1 내지 N) 의 내부 세부사항은 노출된다. 최하위 계층 레벨에서, 유닛은 분해될 수 없다. 시스템 (104) 및 유닛들 (1 내지 N) 의 생성은 아래의 도 4 에서 더 상세히 설명한다.

구성 데이터 저장 유닛 (105) 은, 구성가능한 하드웨어 시스템을 계층적으로 설명하는 구성 데이터 (113) 를 포함한다. 예를 들어, 구성 데이터 (113) 는 적절한 모든 계층 레벨에서의 시스템 및 유닛 파라미터들을 특정한다. 최종 사용자가 구성 데이터 (113) 내의 대부분의 파라미터들을 설정하지만, 하드웨어 통합 및/또는 분해 프로세스 동안에는 생성 및 검증 유닛 (103) 이 일부 파라미터를 설정한다. 본 발명의 실시형태들에 따라, 구성 데이터 (113) 는 임의의 적절한 전자 설계 자동 스크립팅 (scripting) 언어에 의해 표현될 수도 있다. 본 발명의 일 실시형태에서, 구성 데이터 (113) 는 톨 제어 언어 (TCL) 스크립팅 언어로 표현된다. 특히, 구성 데이터 (113) 는 시스템 설계 명칭을 정의하는 TCL 텍스트 파일, 시스템-레벨 파라미터, 유닛-레벨 명칭 및 파라미터, 유닛-레벨 접속 파라미터 (예를 들어, 단일 번들 (bundle) 에서의 배선의 수, 핸드셰이킹 (handshaking) 프로토콜, 파이프라이닝 동작 (pipelining behavior) 등), 및 유닛 인스턴스를 특정 접속부에 바인딩하기 위한 인터페이스문 (interface statements) 을 포함할 수도 있다. 본 발명의 또 다른 실시형태에서, 이러한 시스템 정보는 확장가능한 마크업 언어 (XML) 포맷 또는 관계 데이터베이스로 표현될 수도 있다.

임의의 특정 하드웨어 유닛의 다중의 인스턴스들이 하드웨어 시스템에 포함될 수 있기 때문에, 각각의 유닛 인스턴스는 구성 데이터 (113) 에서 고유하게 명명된다. 또한, 동일한 유닛의 상이한 인스턴스들은 상이하게 구성될 수 있다. 예를 들어, FIFO 의 일 인스턴스는 10바이트의 깊이를 갖도록 구성될 수도 있지만, FIFO 의 또 다른 인스턴스는 100바이트의 깊이를 갖도록 구성될 수도 있다.

구성가능한 하드웨어 라이브러리 (101) 는 시스템의 하드웨어 컴포넌트의 모든 가능한 구성을 나타낸다. 예를 들어, 구성 하드웨어 라이브러리 (101) 는 FIFO 의 깊이, 폭, 및 다른 구성가능한 파라미터를 포함하여 FIFO 의 가능한 모든 구성을 나타낼 수도 있다. 본 발명의 일 실시형태에서, 구성가능한 하드웨어 라이브러리는, 구성 데이터 (113) 를 해석하는 방법을 나타내는 전처리문 (preprocessing statements) 이 내장된 하드웨어 설명 언어 (HDL) 코드 (예를 들어, Verilog 또는 VHDL) 를 포함한다.

도 2 는 본 발명의 실시형태들에 따른 구성가능한 하드웨어 시스템의 생성, 분해, 및 검증을 나타낸 흐름도이다. 흐름도 (200) 의 동작들은 도 1 의 블록도를 참조하여 설명된다. 프로세스 블록 202 에서는, 구성 데이터를 판독한다. 예를 들어, 도 1 에 도시되어 있는 본 발명의 실시형태에 따라, 생성 및 검증 유닛 (103) 의 생성 모듈 (107) 은 구성 데이터 저장 유닛 (105) 로부터의 구성 데이터 (113) 를 판독한다. 상술한 바와 같이, 구성 데이터 (113) 는, 구성가능한 하드웨어 시스템을 계층적으로 정의하는 TCL 파일일 수도 있다. 제어는 블록 204 로 진행된다.

블록 204 에서는, 구성가능한 하드웨어 라이브러리를 분석한다. 예를 들어, 생성 모듈 (107) 은 구성가능한 하드웨어 라이브러리 (101) 를 분석하여, 구성 데이터 (113) 에 의해 정의되는 하드웨어 시스템을 생성하는데 필요한 하드웨어 컴포넌트들의 가능한 구성을 결정한다. 제어는 블록 206 으로 진행된다.

블록 206 에 나타낸 바와 같이, 구성가능한 하드웨어 시스템이 생성된다. 예를 들어, 생성 모듈 (107) 은 구성 데이터 (113) 및 구성가능한 하드웨어 라이브러리 (101) 에 기초하여 구성가능한 하드웨어 시스템을 생성한다. 블록 206 의 동작은 도 3 을 참조하여 아래에서 더 설명한다. 제어는 블록 208 로 진행된다.

블록 208 에서, 시스템은 분해되고, 그 시스템 및 그 컴포넌트들이 검증된다. 예를 들어, 분해 모듈 (109) 및 검증 모듈 (111) 은 시스템 컴포넌트들을 분해 및 검증한다. 블록 208 에서의 동작은 도 5 를 설명할 때에 아래에서 더 상세히 설명한다.

당업자는, 흐름도 (200) 에 설명되어 있는 동작들이 계층적인 시스템 설계의 임의의 레벨에서의 하드웨어를 생성 및 검증하기 위하여 반복될 수도 있음을 명백히 알 수 있다. 예를 들어, 특정한 계층 레벨에서의 시스템을 검증하기 위하여, 시스템의 모든 컴포넌트들은 검증되어야 한다. 이것은 하위 레벨 시스템의 검증을 요구할 수도 있으며, 차례로, 더 하위 레벨 시스템들도 검증할 것을 요구할 수 있다. 일단 최하위 레벨 시스템이 검증되면, 차례로, 상위 레벨 시스템들이 검증될 수도 있다. 따라서, 흐름도 (200) 에서 설명된 동작들은 임의의 설계 계층 레벨에서의 시스템 및/또는 컴포넌트들을 생성 및 검증하기 위해 반복될 수 있다.

도 3 은 본 발명의 실시형태들에 따른 시스템의 생성을 나타낸 흐름도이다. 흐름도 (300) 의 동작들은 도 1 에 도시되어 있는 예시적인 실시형태를 참조하여 설명된다. 판정 블록 302 에서는, 구성 데이터가 구문 (syntax) 및 어의 (semantics) 규칙에 부합하는지 여부를 결정한다. 예를 들어, 통합 모듈 (109) 은, 구성 데이터 저장 유닛 (105) 으로부터의 구성 데이터 (113) 가 구문 및 어의 규칙에 부합하는지 여부를 결정한다. 더 특정한 예로서, 구성 데이터 (113) 가 TCL 텍스트 파일에 의해 표현되는 실시형태에서, 통합 유닛 (109) 은, TCL 파일이 구성가능한 하드웨어 라이브러리 (101) 에 의해 이용되는 HDL 의 구문 및 어의 규칙에 부합하는지 여부를 결정한다. 일 실시형태에서, 통합 모듈 (109) 은 하이-레벨 (high-level) 언어 프로그램 (예를 들어, C++, 파이썬 (Python), 또는 자바 프로그램) 을 이용하여 구문 및 어의에 대한 TCL 파일을 분석한다. 만약 구성 데이터 파일이 구문 및 어의 규칙에 부합하면, 제어는 블록 304 로 진행된다. 그렇지 않으면, 흐름은 에러 리포트에 의해 종료한다. 제어는 블록 304 로 진행된다.

블록 304 에서는, 파라미터를 유도한다. 예를 들어, 통합 모듈 (109) 은 구성 데이터 (113) 로부터 시스템 파라미터를 유도한다. 더 특정한 예로서, 일 실시형태에서, 통합 모듈 (109) 은, 구성가능한 하드웨어 시스템을 정의하는 TCL 파일을 분석함으로써 시스템 파라미터를 유도한다. 예를 들어, 시스템 파라미터는 내부 통신 경로에 요구되는 최소 대역폭을 특정할 수도 있다. 이러한 세팅으로부터, 시스템 내의 다양한 접속 포인트에서 사용되는 배선의 수를 특정하는 파라미터들이 구성 데이터에서의 규칙에 따라 유도된다. 제어는 블록 306 으로 진행된다.

블록 306 에서 나타낸 바와 같이, 유도된 파라미터에 기초하여, 전처리문이 구성된다. 예를 들어, 본 발명의 일 실시형태에서, 통합 모듈 (109) 은, 특정되고 유도된 파라미터에 의해 영향을 받는 HDL 코드 전처리문 (구성가능한 하드웨어 라이브러리 (101) 에 저장됨) 을 구성한다. 이것을 수행할 시, 통합 모듈 (109) 은 HDL 코드에 내장된 전처리기 (preprocessor) 문에 특정한 값 또는 제어 구조를 부여할 수도 있다. 제어는 블록 308 로 진행된다.

블록 308 에 나타낸 바와 같이, HDL 원시 코드가 사전-처리된다. 예를 들어, 통합 모듈 (109) 은, 유도된 파라미터에 따라 구성된 HDL 원시 코드를 사전-처리한다. 본 발명의 일 실시형태에서, 통합 모듈 (109) 은, 내장된 HDL 원시 코드를 사전-처리하기 위한 매크로 언어 전처리기 (예를 들어, C 전처리기, M4 전처리기, 또는 SIMPLE 전처리기) 를 포함한다. 제어는 블록 310 으로 진행된다.

블록 310에서는, 구성 데이터에 특정된 구성가능한 하드웨어 시스템에 대한 HDL 코드를 생성한다. 예를 들어, 통합 모듈 (109)은, 구성가능한 하드웨어 라이브러리 (101)로부터의 HDL 코드를 이용하여 구성 데이터 (113)에 특정된 시스템에 대한 HDL 코드를 생성한다. 블록 310으로부터, 제어는 종료한다.

도 4는 본 발명의 실시형태들에 따른 시스템 설계의 개념적인 블록도이다. 상술한 바와 같이, 본 발명의 일 실시형태에 따라, 도 3의 동작들은 HDL 코드에 표현된 시스템 설계를 생성한다. 도 4는 그러한 시스템의 그래픽 표현을 제공한다. 도 4는 유닛 1, 유닛 2, 유닛 3, 및 유닛 4를 구비하는 시스템 (400)을 포함한다. 시스템 (400)에서, 유닛 1은 통신 경로 (410)를 통하여 시스템 (400) 외부의 시스템들과 통신한다. 유닛 1은 통신 경로 (401 및 403)에 의해 각각 유닛 2 및 유닛 3에 커플링된다. 유닛 2는 통신 경로 (411)를 통하여 시스템 (400) 외부의 시스템들과 통신한다. 유닛 3은 통신 경로 (405 및 408)에 의해 유닛 4에 커플링된다. 또한, 유닛 3은 통신 경로 (409)에 의해 유닛 2에 커플링된다.

과선으로 도시되어 있는 유닛 2는 시스템 (400)에서 옵션 유닛이다. 접속 경로 (401 및 409) 또한 옵션이다. 시스템 설계 계층의 소정 레벨에 대하여, 시스템 외부의 인자들이 옵션 유닛의 기능 (예를 들어, 설계 계층의 상위 레벨에서의 시스템)을 요구하는지 여부를 알 수 없을 경우에 유닛 (또는 접속 경로)은 옵션이다. 예를 들어, 시스템 (400)이 2개의 상이한 모드로 동작하도록 구성될 수 있을 경우, 그 기능이 제 1 모드에 의해 요구되지만 제 2 모드에 의해 요구되지 않으면 유닛 2는 옵션이다.

도 5 및 6은 시스템 (400)이 본 발명의 실시형태들에 따라 분해 및 검증되는 방법을 나타낸 것이다. 도 5는 본 발명의 실시형태들에 따른 구성가능한 하드웨어 시스템을 분해 및 검증하기 위한 동작들을 나타낸 흐름도이다. 도 6은 본 발명의 실시형태들에 따른 유닛을 검증하기 위한 테스트벤치를 나타낸 블록도이다. 도 5 및 6은 도 4의 예시적인 시스템 및 도 1의 예시적인 실시형태를 참조하여 설명한다.

도 5의 흐름도를 참조하면, 블록 502에서는, 선택된 유닛의 파라미터에 구성 데이터 (113)를 매핑한다. 예를 들어, 분해 모듈 (109)은 시스템 (400)을 정의하는 구성 데이터 (113)를 시스템 (400)의 유닛 (예를 들어, 유닛 1)의 파라미터에 매핑한다. 더 특정한 예로서, 분해 모듈 (109)은 구성 데이터 (113)를 분석하여, 어떻게 유닛 1의 파라미터가 시스템 (400)의 요건을 만족시키도록 구성되어야 하는지를 결정한다. 제어는 블록 504로 진행한다.

블록 504에서는, 다른 유닛으로의 각 접속부에 대하여, 인터페이스 모델, 모니터, 및/또는 프로토콜 체커를 그 유닛에 부가한다. 예를 들어, 검증 모듈 (111)은 구성 데이터 (113)를 분석하여, 선택된 유닛 (예를 들어, 유닛 1)에 대한 접속부를 결정한다. 각 접속부 (예를 들어, 통신 경로)에 대하여, 분해 모듈 (109)은, 인터페이스 구동기 모델, 인터페이스 모니터, 및/또는 프로토콜 체커를 포함할 수도 있는 유닛에 일 모델을 커플링한다. 이러한 동작은 도 6에 개념적으로 도시되어 있다. 도 6에서, 모델 A는 통신 경로 (403)를 통하여 유닛 1에 접속되지만, 모델 B는 통신 경로 (401)를 통하여 유닛 1에 접속된다. 모델 C는 통신 경로 (410)를 통하여 유닛 1에 접속된다. 테스트 스티물러스 (test stimulus) 및 응답 체크 유닛 (415)은 모델 A, B 및 C에 접속된다. 테스트 스티물러스 및 응답 체크 유닛 (415)은 테스트 동작을 모니터링하고 그 동작을 용이하게 한다. 테스트벤치에서, 모델들은, 검증된 유닛에 대한 정보를 송신 및 수신하기 위해 이용된다. 예를 들어, 모델 A 및 B는 유닛 1의 파라미터에 따라 (예를 들어, 특정한 통신 경로에 대하여 정의되는 특정한 통신 프로토콜에 따라) 유닛 1로부터 데이터 스트림을 수신한다. 이와 유사하게, 모델 C는 유닛 1의 파라미터에 따라 데이터를 유닛 1로 송신한다. 유닛으로 및 유닛으로부터 송신될 특정한 데이터는, 유닛을 검증하는데 이용되는 테스트에 의해 결정된다. 이들 테스트는 아래에서 더 상세히 설명한다. 블록 504로부터, 제어는 블록 506으로 진행한다.

블록 506에서는, 구성 데이터를 생성한다. 예를 들어, 분해 모듈 (109)은, 선택된 유닛의 파라미터를 특정하는 구성 데이터 (113)를 생성한다. 본 발명의 일 실시형태에 따라, 도 1의 설명에서 상술된 바와 같이, 분해 모듈 (109)은 구성 데이터 (113)를 TCL 파일의 형태로 생성한다. 또 다른 실시형태에 따라, 분해 모듈 (109)은 구성 데이터 (113)를 XML 파일의 형태로 생성한다.

블록 508에서는, 구성 데이터 (113)에 기초하여 설계를 생성한다. 예를 들어, 생성 모듈 (107)은, 구성 데이터 (113)에 기초하여 구성가능한 하드웨어 시스템 설계를 생성하기 위해 구성가능한 하드웨어 라이브러리 (101)를 이용한다. 도 3의 설명에서, 이 동작은 더 상세히 설명되어 있다. 일 실시형태에서, 그 설계는 HDL 코드에 의해 표현된다. 제어는 블록 510으로 진행한다.

블록 510에서는, 테스트 및 스크립트 및/또는 분석 툴로의 입력을 생성한다. 예를 들어, 검증 모듈 (111)은 테스트 및/또는 분석 툴로의 입력을 구동시키기 위한 테스트 및 스크립트를 생성한다. 블록 510으로부터, 제어는 블록 512로 진행한다. 테스트를 생성할 시, 검증 모듈 (111)은 특정한 유닛의 기능을 검증하기 위해 공지되어 있는 종래의 테스트를 이용

할 수도 있으며, 또는, 유닛 구성의 분석에 기초하여 맞춤형 제작된 테스트를 생성할 수도 있다. 이들 테스트는 테스트되는 구성 유닛의 기능을 수행하고 검증한다. 본 발명의 일 실시형태에 따라, 검증 모듈 (111) 은, 유닛의 임의의 구성을 검증할 수 있는 테스트를 생성한다. 이 실시형태에서, 그 테스트는 구성 데이터 (113) 를 관독하며, 이에 따라, 테스트가 시작되기 이전보다는 테스트가 구동하고 있는 중에 그 스티플러스를 변경한다. 또한, 검증 모듈 (111) 은 그 테스트를 자동으로 수행하기 위한 스크립트를 생성할 수도 있다.

테스팅의 추가적인 또는 또 다른 형태로서, 검증 모듈 (111) 은, 설계의 정적 분석을 수행하는 분석 툴에게 그 설계를 제공한다. 예를 들어, 본 발명의 일 실시형태에 따라, 검증 모듈 (111) 은, 에러에 대한 HDL 코드를 분석하는 정적 검증 툴에게 HDL 코드에 의해 표현된 유닛 설계를 제공한다. 일 실시형태에서, 정적 검증 툴은 HDL 코드의 분석에 기초하여 경고 또는 에러 메시지를 생성한다.

블록 512 에 나타낸 바와 같이, 스크립트가 실행된다. 예를 들어, 검증 모듈 (111) 은, 선택된 유닛을 자동으로 테스트하고 검증하는 스크립트를 실행한다.

당업자는, 도 5 의 흐름도에서 나타낸 동작들이 설계 계층의 임의의 레벨에서의 임의의 유닛/시스템을 검증하도록 반복될 수 있음을 명백히 알 수 있다.

도 1 을 참조하면, 생성 및 검증 유닛 (103), 구성가능한 하드웨어 라이브러리 (101), 및 구성 데이터 저장 유닛 (105) 는 하나 이상의 처리기 (processors), 주문형 집적회로 (ASIC), 메모리, 및/또는 본 발명의 실시형태들에 따른 동작들을 수행하기 위한 명령들이 저장되는 머신-관독가능 매체를 포함하여 종래의 컴퓨팅 플랫폼의 형태로 구현될 수도 있다. 머신-관독가능 매체는 머신 (예를 들어, 컴퓨터) 에 의해 관독가능한 형태로 정보를 제공 (즉, 저장 및/또는 송신) 하는 임의의 메카니즘을 포함한다. 예를 들어, 머신-관독가능 매체는 ROM; RAM; 자성 디스크 저장 매체; 광학 저장 매체; 플래시 메모리 디바이스; 전파되는 신호들 (예를 들어, 반송파, 적외선 신호, 디지털 신호 등) 의 전기, 광학, 음향 또는 다른 형태 등을 포함한다. 일 실시형태에서, 도 1 에 도시되어 있는 유닛들은, 여기에서 설명되어 있는 동작들을 수행하기 위하여 처리기 상에서 실행하는 머신-관독가능 매체이다. 그러나, 또 다른 실시형태에서, 도 1 의 유닛들은 여기에서 설명되어 있는 동작들을 실행하기 위한 다른 타입의 로직 (예를 들어, 디지털 로직) 이다. 다른 방법으로, 본 발명의 일 실시형태에 따라, 생성 및 검증 유닛 (103), 구성가능한 하드웨어 라이브러리 (101), 및 구성 데이터 저장 유닛 (105) 은 하나 이상의 별도의 컴퓨터 시스템을 포함할 수 있다. 또한, 본 발명의 실시형태들에 따라, 도 1 에 도시되어 있는 컴포넌트들은 다수의 네트워크된 컴퓨터를 통하여 분배될 수도 있으며, 여기서, 이들 다수의 네트워크된 컴퓨터는 원격적으로 제어 및 동작할 수 있다.

도 7 은 본 발명의 실시형태들에 따른 구성가능한 하드웨어를 분해 및 검증하기 위한 예시적인 시스템을 도시한 것이다. 도 7 에 도시된 바와 같이, 컴퓨터 시스템 (700) 은 처리기(들) (702) 를 구비한다. 또한, 컴퓨터 시스템 (700) 은 메모리 (732), 처리기 버스 (710) 및 입력/출력 제어기 허브 (ICH; 740) 를 구비한다. 처리기(들) (702), 메모리 (732) 및 ICH (740) 는 처리기 버스 (710) 에 커플링된다. 처리기(들) (702) 는 임의의 적절한 처리기 구조체를 포함할 수도 있다. 본 발명의 다른 실시형태의 경우, 컴퓨터 시스템 (700) 은, 본 발명의 실시형태들을 따르는 명령 세트를 실행할 수도 있는 하나, 2개, 3개, 또는 그 이상의 처리기를 구비할 수도 있다.

메모리 (732) 는 데이터 및/또는 명령을 저장하며, 예를 들어, DRAM 과 같은 임의의 적절한 메모리를 포함할 수도 있다. 본 발명의 일 실시형태에서, 구성 하드웨어 라이브러리 (101), 생성 및 검증 유닛 (103), 및 구성 데이터 저장 유닛 (105) 은 메모리 (732) 에 저장된다. 그러나, 이들은 임의의 또는 모든 IDE 드라이브(들) (742), 메모리 (732), 및/또는 다른 적절한 저장 디바이스에 저장될 수도 있다. 본 발명의 실시형태들에 따라, 그래픽 제어기 (734) 는 디스플레이 디바이스 (737) 상으로의 정보의 디스플레이를 제어한다.

입력/출력 제어기 허브 (ICH; 740) 는 컴퓨터 시스템 (700) 에 대한 I/O 디바이스 또는 주변 컴포넌트로의 인터페이스를 제공한다. ICH (740) 는 처리기(들) (702), 메모리 (732), 및/또는 ICH (740) 와 통신하는 임의의 적절한 디바이스 또는 컴포넌트와의 임의의 적절한 통신 링크용으로 제공하기 위한 임의의 적절한 인터페이스 제어기를 구비할 수도 있다. 본 발명의 일 실시형태의 경우, ICH (740) 는 적절한 조정 및 각 인터페이스에 대한 버퍼링을 제공한다.

본 발명의 일 실시형태의 경우, ICH (740) 는, 예를 들어, 데이터 및/또는 명령을 저장하기 위한, 예를 들어, 하드 디스크 드라이브 (HDD) 또는 CD-ROM 과 같은 하나 이상의 적절한 통합형 드라이브 전자장치 (IDE) 드라이브 (742) 로의 인터페이스, 및 하나 이상의 범용 직렬 버스 (USB) 포트 (744) 를 통한 하나 이상의 USB 디바이스로의 인터페이스를 제공한다. 또한, 본 발명의 일 실시형태의 경우, ICH (740) 은 키보드 (751), 마우스 (752), 플로피 디스크 드라이브 (755), 하나 이상의 병렬 포트 (753; 예를 들어, 프린터) 를 통한 하나 이상의 적절한 디바이스, 및 하나 이상의 직렬 포트 (754) 를 통

한 하나 이상의 적절한 디바이스와 인터페이스를 제공한다. 또한, 본 발명의 일 실시형태의 경우, ICH (740) 는 네트워크 인터페이스 (756) 를 제공하며, 이 네트워크 인터페이스 (756) 를 통하여, 컴퓨터 시스템 (700) 은 다른 컴퓨터 및/또는 디바이스와 통신할 수 있다.

따라서, 컴퓨터 시스템 (700) 은, 여기에서 설명된 임의의 일 방법 또는 모든 방법들을 구현하는 명령 세트 (즉, 소프트웨어) 를 저장하는 머신-판독가능 매체를 포함한다. 예를 들어, 소프트웨어는 메모리 (732) 및/또는 처리기(들) (702) 내에 완전히 또는 적어도 부분적으로 상주할 수 있다.

본 발명은 수개의 실시형태들의 관점에서 설명되었지만, 당업자는 본 발명이 설명된 실시형태들에 제한되지 않음을 알 수 있다. 본 발명의 방법 및 장치는, 첨부된 특허청구범위의 사상 및 범위 내에서 변형 및 변경에 의해 실시될 수 있다. 따라서, 본 명세서는 본 발명을 제한하는 것이 아닌 예시적인 것으로서 간주되어야 한다.

(57) 청구의 범위

청구항 1.

하드웨어 시스템을 하나 이상의 유닛들의 세트로 자동으로 분해하는 단계;

유닛들의 상기 세트 각각에 대한 테스트벤치를 생성하는 단계; 및

상기 하드웨어 시스템을 검증하기 전에 유닛들의 상기 세트 각각을 검증하는 단계를 포함하는, 컴퓨터 구현 방법.

청구항 2.

제 1 항에 있어서,

유닛들의 상기 세트는 구성가능한 하드웨어 라이브러리에서 정의되며,

상기 시스템은 구성 데이터 저장 유닛에서 특정되는, 컴퓨터 구현 방법.

청구항 3.

제 2 항에 있어서,

상기 구성 데이터는 계층적인 언어로 표현되며,

상기 구성가능한 하드웨어 라이브러리는 하드웨어 설계 언어 (HDL) 로 표현되는, 컴퓨터 구현 방법.

청구항 4.

제 1 항에 있어서,

상기 테스트벤치는 각 유닛 접속부에 접속되는 모델들을 포함하며,

상기 모델들은 상기 유닛의 파라미터에 따라 상기 유닛으로 데이터를 송신하고 상기 유닛으로부터 데이터를 수신하는, 컴퓨터 구현 방법.

청구항 5.

구성가능한 하드웨어 시스템 설계 계층의 제 1 레벨에서의 하나 이상의 유닛들의 세트를 상기 하드웨어 시스템 설계 계층의 최하위 레벨의 하나 이상의 유닛들의 세트로 자동으로 분해하는 단계로서, 상기 구성가능한 하드웨어 시스템 설계 계층은 하나 이상의 계층 레벨들의 세트를 포함하는, 상기 자동 분해 단계; 및

상기 최하위 레벨로부터, 연속적으로, 각각의 연속적인 레벨의 각 유닛에 대하여 동적으로 형성되는 테스트벤치를 갖는 상기 제 1 레벨로의 상기 하드웨어 시스템 설계 계층의 각 계층 레벨의 유닛들을 개별적으로 검증하는 단계를 포함하는, 컴퓨터 구현 방법.

청구항 6.

제 5 항에 있어서,

상기 자동으로 분해하는 단계는 구성가능한 하드웨어 라이브러리의 콘텐츠 및 구성 데이터에 기초하는, 컴퓨터 구현 방법.

청구항 7.

제 6 항에 있어서,

상기 구성 데이터는 구성가능한 하드웨어 시스템 설계 레벨들의 상기 세트 각각의 상기 유닛들에 대한 파라미터를 특정하는, 컴퓨터 구현 방법.

청구항 8.

제 6 항에 있어서,

상기 구성가능한 하드웨어 라이브러리는 계층 레벨들의 상기 세트 각각의 상기 유닛들을 정의하는, 컴퓨터 구현 방법.

청구항 9.

구성가능한 하드웨어 라이브러리로부터 선택된 대응하는 구성가능한 유닛 정의에 구성 데이터의 세트를 매핑하여 하나 이상의 구성가능한 하드웨어 유닛들의 세트를 생성하는 단계;

상기 구성 데이터에 기초하여 구성가능한 하드웨어 유닛들의 상기 세트 각각에 대한 테스트벤치를 동적으로 생성하는 단계;

대응하는 테스트벤치를 갖는 구성가능한 하드웨어 유닛들의 상기 세트 각각을 검증하는 단계;

구성가능한 하드웨어 유닛들의 검증된 상기 세트를 구성가능한 하드웨어 시스템으로 통합하는 단계; 및

상기 구성가능한 하드웨어 시스템을 검증하는 단계를 포함하는, 컴퓨터 구현 방법.

청구항 10.

제 9 항에 있어서,

상기 테스트벤치 상에서 구동될 테스트를 생성하는 단계;
상기 테스트를 실행하기 위한 스크립트를 생성하는 단계; 및
분석 툴로의 입력을 생성하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 11.

제 9 항에 있어서,
상기 대응하는 구성가능한 유닛 정의는 하드웨어 설계 언어 (HDL) 로 표현되는, 컴퓨터 구현 방법.

청구항 12.

제 9 항에 있어서,
상기 구성 데이터는 툴 제어 언어 (TCL) 로 표현되며,
상기 구성 데이터는 구성가능한 하드웨어 유닛들의 상기 세트 각각에 대한 파라미터를 정의하는, 컴퓨터 구현 방법.

청구항 13.

제 9 항에 있어서,
상기 테스트벤치는 상기 유닛의 각 통신 경로에 접속되는 모델들을 포함하는, 컴퓨터 구현 방법.

청구항 14.

구성 데이터 및 구성가능한 하드웨어 라이브러리에 기초하여 시스템으로부터 분해되는 개별 유닛들에 대한 하나 이상의 테스트벤치들의 세트를 형성하는 단계로서, 상기 개별 유닛들은 구성가능한 하드웨어 설계 계층의 하위 레벨이며 상기 시스템은 구성가능한 하드웨어 설계 계층의 상위 레벨인, 상기 형성 단계;

시스템 테스트벤치를 형성하는 단계; 및
상기 개별 유닛들을 검증한 후에 상기 시스템을 검증하는 단계를 포함하는, 컴퓨터 구현 방법.

청구항 15.

제 14 항에 있어서,
상기 시스템은 상기 시스템 테스트벤치를 이용하여 검증되며,
상기 개별 유닛들은 테스트벤치들의 상기 세트를 이용하여 검증되는, 컴퓨터 구현 방법.

청구항 16.

제 14 항에 있어서,

테스트벤치들의 상기 세트 및 상기 시스템 테스트벤치 상에서 구동될 테스트를 생성하는 단계;

상기 테스트를 실행하기 위한 스크립트를 생성하는 단계; 및

분석 툴로의 입력을 생성하는 단계를 더 포함하는, 컴퓨터 구현 방법.

청구항 17.

제 14 항에 있어서,

상기 구성가능한 하드웨어 라이브러리는 상기 시스템에 포함된 상기 유닛들을 정의하며,

상기 구성 데이터는 상기 파라미터에게 상기 시스템을 특정하는, 컴퓨터 구현 방법.

청구항 18.

하드웨어 시스템을 자동으로 생성, 분해, 및 검증하는 생성 및 검증 유닛;

상기 생성 및 검증 유닛에 커플링되며, 하드웨어 시스템 설계를 정의하기 위한 구성 데이터를 포함하는 구성 데이터 저장 유닛; 및

상기 생성 및 검증 유닛에 커플링되며, 하드웨어 시스템들을 통합 및 분해하기 위한 구성가능한 하드웨어 유닛들의 정의를 저장하는 구성가능한 하드웨어 라이브러리를 구비하는, 장치.

청구항 19.

제 18 항에 있어서,

상기 구성가능한 하드웨어 유닛들의 정의는 하드웨어 설계 언어 (HDL) 로 표현되는, 장치.

청구항 20.

제 18 항에 있어서,

상기 구성 데이터는 계층적인 언어로 표현되는, 장치.

청구항 21.

제 18 항에 있어서,

상기 생성 및 검증 유닛은, 하드웨어 설계 계층 내의 유닛들 및 시스템들을 생성하기 위한 생성 모듈, 상기 유닛들 및 상기 시스템들을 분해하기 위한 분해 모듈, 및 상기 유닛들 및 상기 시스템들에 대한 테스트벤치를 형성하고 상기 유닛들 및 상기 시스템들을 검증하기 위한 검증 모듈을 포함하는, 장치.

청구항 22.

제 21 항에 있어서,

상기 유닛들 및 시스템들은 HDL 로 표현되는, 장치.

청구항 23.

머신에 의해 실행될 때, 상기 머신으로 하여금,

하드웨어 시스템을 하나 이상의 유닛들의 세트로 자동으로 분해하는 단계;

유닛들의 상기 세트 각각에 대한 테스트벤치를 생성하는 단계; 및

상기 하드웨어 시스템 설계를 검증하기 전에 유닛들의 상기 세트 각각을 검증하는 단계를 포함하는 동작들을 수행하게 하는 명령들을 제공하는, 머신-판독가능 매체.

청구항 24.

제 23 항에 있어서,

유닛들의 상기 세트는 구성가능한 하드웨어 라이브러리에서 정의되며,

상기 시스템은 구성 데이터 저장 유닛에서 특정되는, 머신-판독가능 매체.

청구항 25.

제 24 항에 있어서,

상기 구성 데이터는 계층적인 언어로 표현되며,

상기 구성가능한 하드웨어 라이브러리는 하드웨어 설계 언어 (HDL) 로 표현되는, 머신-판독가능 매체.

청구항 26.

제 23 항에 있어서,

상기 테스트벤치는 각 유닛 접속부에 접속되는 모델들을 포함하며,

상기 모델들은 상기 유닛의 파라미터에 따라 상기 유닛으로 데이터를 송신하고 상기 유닛으로부터 데이터를 수신하는, 머신-판독가능 매체.

청구항 27.

머신에 의해 실행될 때, 상기 머신으로 하여금,

구성가능한 하드웨어 시스템 설계 계층의 제 1 레벨에서의 하나 이상의 유닛들의 세트를 상기 하드웨어 시스템 설계 계층의 최하위 레벨의 하나 이상의 유닛들의 세트로 자동으로 분해하는 단계로서, 상기 구성가능한 하드웨어 시스템 설계 계층은 하나 이상의 계층 레벨들의 세트를 포함하는, 상기 자동 분해 단계; 및

상기 최하위 레벨로부터, 연속적으로, 각각의 연속적인 레벨의 각 유닛에 대하여 동적으로 형성되는 테스트벤치를 갖는 상기 제 1 레벨의 상기 하드웨어 시스템 설계 계층의 각 계층 레벨의 유닛들을 개별적으로 검증하는 단계를 포함하는 동작들을 수행하게 하는 명령들을 제공하는, 머신-판독가능 매체.

청구항 28.

제 27 에 있어서,

상기 자동으로 분해하는 단계는 구성가능한 하드웨어 라이브러리의 콘텐츠 및 구성 데이터에 기초하는, 머신-판독가능 매체.

청구항 29.

제 28 항에 있어서,

상기 구성 데이터는 구성가능한 하드웨어 시스템 설계 레벨들의 상기 세트 각각의 상기 유닛들에 대한 파라미터를 특정하는, 머신-판독가능 매체.

청구항 30.

제 28 항에 있어서,

상기 구성가능한 하드웨어 라이브러리는 계층 레벨들의 상기 세트 각각의 상기 유닛들을 정의하는, 머신-판독가능 매체.

청구항 31.

머신에 의해 실행될 때, 상기 머신으로 하여금,

구성가능한 하드웨어 라이브러리로부터 선택된 대응하는 구성가능한 유닛 정의에 구성 데이터의 세트를 매핑하여 하나 이상의 구성가능한 하드웨어 유닛들의 세트를 생성하는 단계;

상기 구성 데이터에 기초하여 구성가능한 하드웨어 유닛들의 상기 세트 각각에 대한 테스트벤치를 동적으로 생성하는 단계;

대응하는 테스트벤치를 갖는 구성가능한 하드웨어 유닛들의 상기 세트 각각을 검증하는 단계;

구성가능한 하드웨어 유닛들의 검증된 상기 세트를 구성가능한 하드웨어 시스템으로 통합하는 단계; 및

상기 구성가능한 하드웨어 시스템을 검증하는 단계를 포함하는 동작들을 수행하게 하는 명령들을 제공하는, 머신-판독가능 매체.

청구항 32.

제 31 항에 있어서,
상기 테스트벤치 상에서 구동될 테스트를 생성하는 단계;
상기 테스트를 실행하기 위한 스크립트를 생성하는 단계; 및
분석 툴로의 입력을 생성하는 단계를 더 포함하는, 머신-판독가능 매체.

청구항 33.

제 31 항에 있어서,
상기 대응하는 구성가능한 유닛 정의는 하드웨어 설계 언어 (HDL) 로 표현되는, 머신-판독가능 매체.

청구항 34.

제 31 항에 있어서,
상기 구성 데이터는 툴 제어 언어 (TCL) 로 표현되며,
상기 구성 데이터는 구성가능한 하드웨어 유닛들의 상기 세트 각각에 대한 파라미터를 정의하는, 머신-판독가능 매체.

청구항 35.

제 31 항에 있어서,
상기 테스트벤치는 상기 유닛의 각 통신 경로에 접속되는 모델들을 포함하는, 머신-판독가능 매체.

청구항 36.

머신에 의해 실행될 때, 상기 머신으로 하여금,

구성 데이터 및 구성가능한 하드웨어 라이브러리에 기초하여 시스템으로부터 분해되는 개별 유닛들에 대한 하나 이상의 테스트벤치들의 세트를 형성하는 단계로서, 상기 개별 유닛들은 구성가능한 하드웨어 설계 계층의 하위 레벨이며 상기 시스템은 구성가능한 하드웨어 설계 계층의 상위 레벨인, 상기 형성 단계;

시스템 테스트벤치를 형성하는 단계; 및

상기 개별 유닛들을 검증한 후에 상기 시스템을 검증하는 단계를 포함하는 동작들을 수행하게 하는 명령들을 제공하는, 머신-판독가능 매체.

청구항 37.

제 36 항에 있어서,
상기 시스템은 상기 시스템 테스트벤치를 이용하여 검증되며,

상기 개별 유닛들은 테스트벤치들의 상기 세트를 이용하여 검증되는, 머신-판독가능 매체.

청구항 38.

제 36 항에 있어서,

테스트벤치들의 상기 세트 및 상기 시스템 테스트벤치 상에서 구동될 테스트를 생성하는 단계;

상기 테스트를 실행하기 위한 스크립트를 생성하는 단계; 및

분석 툴로의 입력을 생성하는 단계를 더 포함하는, 머신-판독가능 매체.

청구항 39.

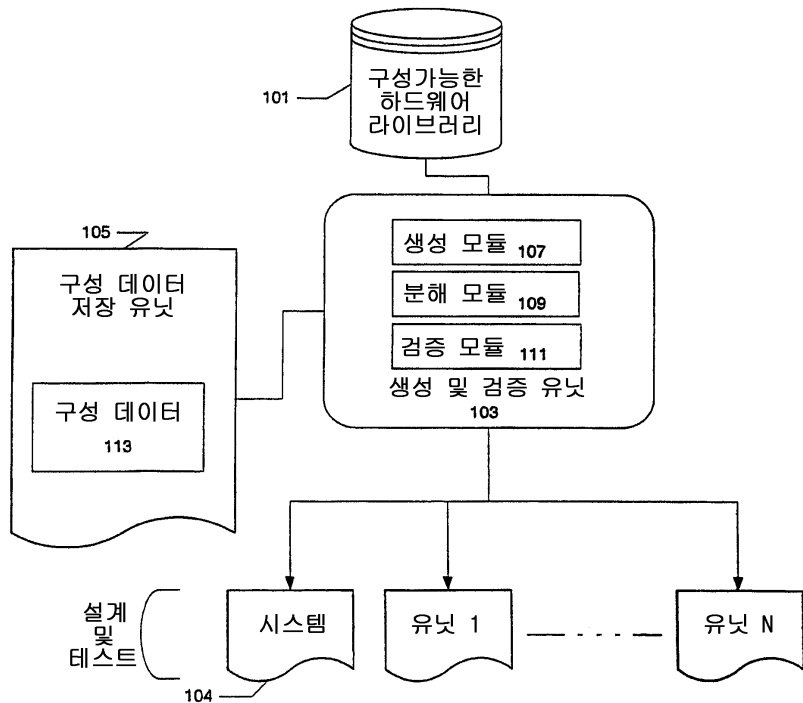
제 36 항에 있어서,

상기 구성가능한 하드웨어 라이브러리는 상기 시스템에 포함된 상기 유닛들을 정의하며,

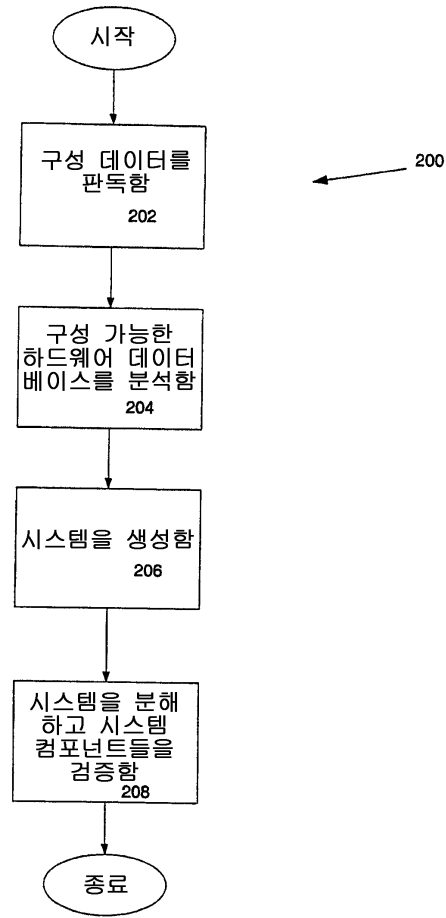
상기 구성 데이터는 상기 파라미터에게 상기 시스템을 특정하는, 머신-판독가능 매체.

도면

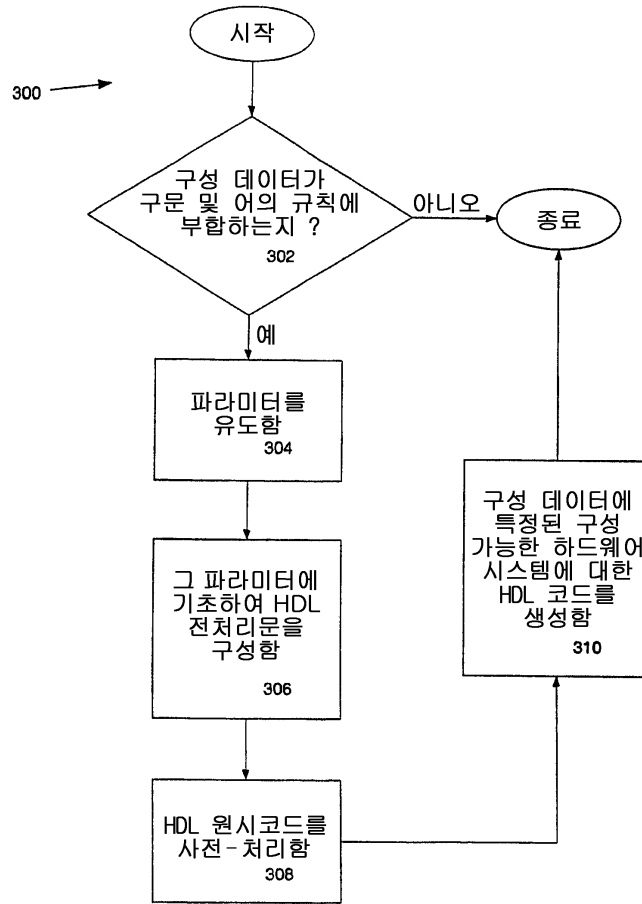
도면1



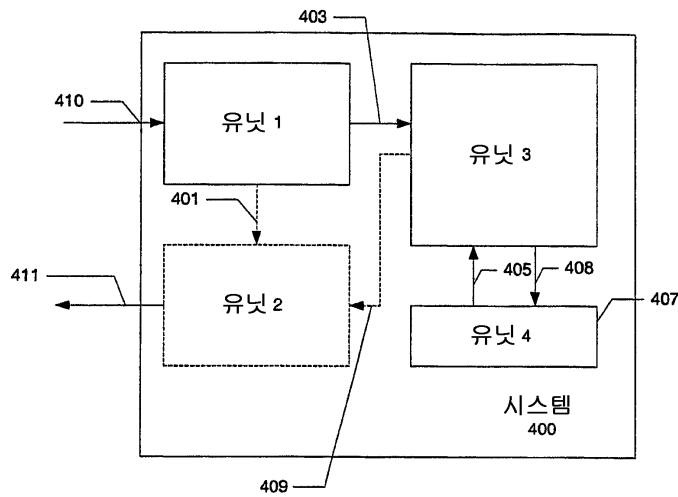
도면2



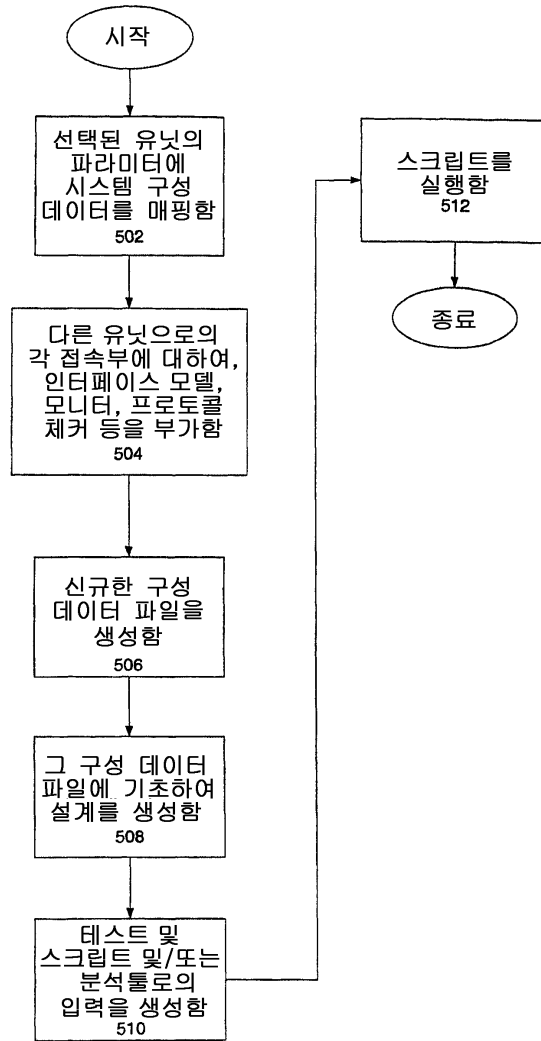
도면3



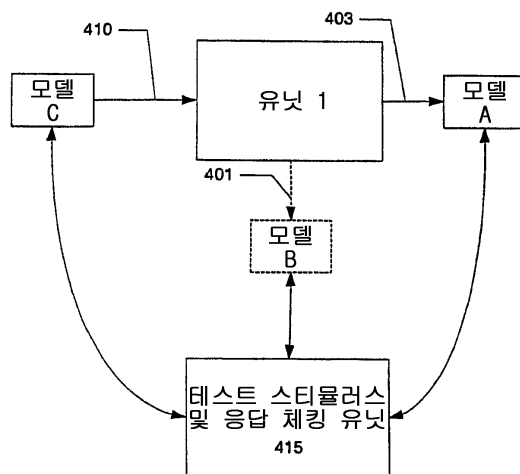
도면4



도면5



도면6



도면7

