(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2007-0114306
(43) 공개일자 2007년11월30일

(51) Int. Cl.
G01R 31/319 (2006.01)
(21) 출원번호 10-2007-7022478
(22) 출원일자 2007년10월01일
심사청구일자 없음
번역문제출일자 2007년10월01일
(86) 국제출원번호 PCT/JP2006/304517
국제출원일자 2006년03월02일
(87) 국제공개번호 WO 2006/093327
국제공개일자 2006년09월08일
(30) 우선권주장
11/070,957 2005년03월02일 미국(US)

(71) 출원인
주식회사 아도반테스토
일본 도쿄, 네리마-쿠, 아사히-쵸 1-쵸메, 32-1
(72) 발명자
프라마닉 안칸
미국 캘리포니아 95318 산 호세 카필라노 드라이브 5839
엘스톤 마크
미국 캘리포니아 93906 살리나스 머롯 웨이 1777
아다치 토시아키
미국 캘리포니아 95120 산 호세 니케트 웨이 1196
(74) 대리인
오병석

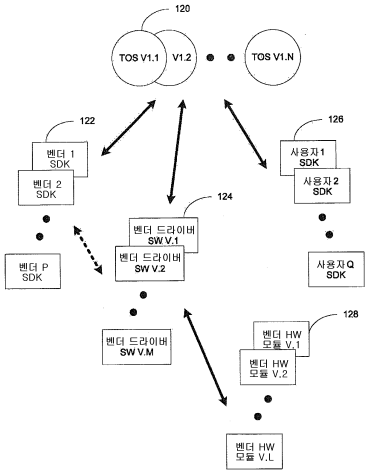
전체 청구항 수 : 총 25 항

(54) 개방형 구조 테스트 시스템에서의 벤더 하드웨어 모듈수정의 간결한 표시

(57) 요 약

개방형 구조 테스트 시스템에서, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트를 조합하는 방법 및 시스템이 개시된다. 상기 방법은, 벤더 하드웨어 모듈의 하드웨어 버전의 집합을 수신하는 단계, 상기 벤더 하드웨어 모듈에 의해 지원되는 소프트웨어 컴포넌트의 집합을 수신하는 단계, 마스크 값을 이용하여, 상기 하드웨어 버전의 집합이 하드웨어 버전 번호의 등가 클래스로서 표시되도록 상기 하드웨어 버전의 집합을 처리하는 단계, 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전을 취득하는 단계, 상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 단계, 및 상기 사용자 선택의 하드웨어 버전에 따라 시스템 프로파일을 생성하는 단계를 포함한다.

대표도 - 도1b



특허청구의 범위

청구항 1

개방형 구조 테스트 시스템에서 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트를 조합하는 방법에 있어서,

벤더 하드웨어 모듈의 하드웨어 버전의 집합을 수신하는 단계;

상기 벤더 하드웨어 모듈에 의해 지원되는 소프트웨어 컴포넌트의 집합을 수신하는 단계;

마스크(mask) 값을 이용하여, 상기 하드웨어 버전의 집합이 하드웨어 버전 번호의 등가 클래스(equivalence class)로서 표시되도록, 상기 하드웨어 버전의 집합을 처리하는 단계;

사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전을 취득하는 단계;

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 단계; 및

상기 사용자 선택의 하드웨어 버전에 따라 시스템 프로파일을 생성하는 단계를 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 2

제1항에 있어서,

상기 소프트웨어 컴포넌트의 집합은,

하나 또는 그 이상의 벤더 소프트웨어 개발 키트;

하나 또는 그 이상의 사용자 소프트웨어 개발 키트; 및

상기 벤더 하드웨어 모듈의 상기 하드웨어 버전의 집합에 대응하는 하나 또는 그 이상의 벤더 드라이버 소프트웨어 버전을 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 3

제1항에 있어서,

상기 하드웨어 버전의 집합을 처리하는 단계는,

마스크된 번호 및 마스크를 포함하는 마스크 값을 정의하는 단계 - 상기 마스크는 상기 하드웨어 버전의 집합의 범위 내의 데이터 패턴이고, 상기 데이터 패턴은 적어도 돈케어(don't care) 비트 및 마스크 되지 않는 비트를 포함함 -; 및

상기 마스크 및 상기 마스크된 번호에 따라 하드웨어 버전 번호의 등가 클래스를 규정하는 단계를 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 4

제3항에 있어서,

상기 하드웨어 버전 번호의 등가 클래스를 규정하는 단계는,

상기 마스크 내의 돈케어 비트의 각각에 대해, 이에 대응하는 상기 마스크된 번호 내의 비트를 마스크 하여, 상기 등가 클래스의 모든 멤버(member)가 상기 돈케어 비트에 대응하는 비트 위치 내에서 1 또는 0을 갖도록 지시하는 단계; 및

상기 마스크 내의 마스크 되지 않는 비트의 각각에 대해, 상기 등가 클래스 내의 상기 각 비트에 대응하는 상기 마스크된 번호의 비트를 이용하여, 상기 대응하는 마스크된 번호 내의 비트가 마스크 되지 않도록 지시하는 단계를 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 5

제3항에 있어서,

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 단계는,

만약 사용자 선택이 상기 마스크 값의 범위에 포함된다면, 상기 마스크 값에 대응하는 전체 등가 클래스를 선택된 것으로서 취급하는 단계를 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 6

제5항에 있어서,

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 단계는,

사용자 선택의 하드웨어 버전의 각각에 대해, 상기 사용자 선택의 하드웨어 버전과 상기 마스크의 비트마다의 논리곱(AND)이 상기 마스크된 번호와 동일한 경우에만, 상기 사용자 선택의 하드웨어 버전이 유효하다고 표시하는 단계를 더 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 7

제1항에 있어서,

상기 시스템 프로파일을 생성하는 단계는,

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 따라 호환성 있는 소프트웨어 컴포넌트의 집합을 결정하는 단계;

각각의 상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대해, 소프트웨어 컴포넌트의 버전을 선택하는 단계;

각각의 상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대해, 사용자에 의해 선택된 소프트웨어 컴포넌트의 목록을 생성하는 단계; 및

상기 사용자에 의해 선택된 소프트웨어 컴포넌트의 목록을 저장하는 단계를 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 8

제7항에 있어서,

상기 소프트웨어 컴포넌트의 버전을 선택하는 단계는,

사용자에게, 상기 호환성 있는 소프트웨어 컴포넌트의 집합으로부터 소프트웨어 컴포넌트의 버전을 선택하도록 질의하는 단계; 및

사용자 선택의 상기 소프트웨어 컴포넌트의 버전의 각각에 대해, 지원되는 하드웨어 버전의 목록을 산출하기 위해, 상기 사용자 선택의 하드웨어 버전들에 걸쳐 교집합을 연산하는 단계를 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 9

제7항에 있어서,

상기 교집합을 연산하는 단계는,

상기 사용자 선택의 하드웨어 버전을 이용하여 결과인 마스크를 연산하는 단계; 및

상기 사용자 선택의 상기 하드웨어 버전을 이용하여 결과인 마스크된 번호를 연산하는 단계를 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 10

제1항에 있어서,

시스템 제어기, 하나 또는 그 이상의 사이트 제어기, 하나 또는 그 이상의 벤더 하드웨어 모듈, 및 하나 또는 그 이상의 피시험 장치를 포함하는 개방형 구조 테스트 시스템을 제공하는 단계;

상기 하드웨어 버전 번호의 등가 클래스의 멤버인, 벤더 하드웨어 모듈의 새로운 하드웨어 버전을 수신하는 단계; 및

상기 시스템 프로파일을 이용하여 상기 벤더 하드웨어 모듈의 상기 새로운 하드웨어 버전을 시험하는 단계를 더 포함하는, 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트의 조합 방법.

청구항 11

개방형 구조 테스트 시스템에 있어서,

시스템 제어기;

상기 시스템 제어기에 연결된 사이트 제어기;

상기 사이트 제어기에 연결된 하나 또는 그 이상의 하드웨어 모듈;

하드웨어 모듈에 병렬로 연결된 하나 또는 그 이상의 피시험 장치;

벤더 하드웨어 모듈의 하드웨어 버전의 집합을 수신하는 수단;

상기 벤더 하드웨어 모듈에 의해 지원되는 소프트웨어 컴포넌트의 집합을 수신하는 수단;

마스크 값을 이용하여, 상기 하드웨어 버전의 집합이 하드웨어 버전 번호의 등가 클래스로서 표시되도록, 상기 하드웨어 버전의 집합을 처리하는 수단;

사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전을 취득하는 수단;

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 수단; 및

상기 사용자 선택의 하드웨어 버전에 따라 시스템 프로파일을 생성하는 수단을 포함하는 개방형 구조 테스트 시스템.

청구항 12

제11항에 있어서,

상기 소프트웨어 컴포넌트의 집합은,

하나 또는 그 이상의 벤더 소프트웨어 개발 키트;

하나 또는 그 이상의 사용자 소프트웨어 개발 키트; 및

상기 벤더 하드웨어 모듈의 상기 하드웨어 버전의 집합에 대응하는 하나 또는 그 이상의 벤더 드라이버 소프트웨어 버전을 포함하는 개방형 구조 테스트 시스템.

청구항 13

제11항에 있어서,

상기 하드웨어 버전의 집합을 처리하는 수단은,

마스크된 번호 및 마스크를 포함하는 마스크 값을 정의하는 수단 - 상기 마스크는 상기 하드웨어 버전의 집합의 범위 내의 데이터 패턴이고, 상기 데이터 패턴은 적어도 돈케어 비트 및 마스크 되지 않는 비트를 포함함 -; 및

상기 마스크 및 상기 마스크된 번호에 따라 하드웨어 버전 번호의 등가 클래스를 규정하는 수단을 포함하는 개방형 구조 테스트 시스템.

청구항 14

제13항에 있어서,

상기 하드웨어 버전 번호의 등가 클래스를 규정하는 수단은,

상기 마스크 내의 돈케어 비트의 각각에 대해, 이에 대응하는 상기 마스크된 번호 내의 비트를 마스크 하여, 상기 등가 클래스의 모든 멤버(member)가 상기 돈케어 비트에 대응하는 비트 위치 내에서 1 또는 0을 갖도록 지시

하는 수단; 및

상기 마스크 내의 마스크 되지 않는 비트의 각각에 대해, 상기 등가 클래스 내의 상기 각 비트에 대응하는 상기 마스크된 번호의 비트를 이용하여, 상기 대응하는 마스크된 번호 내의 비트가 마스크 되지 않도록 지시하는 수단을 포함하는 개방형 구조 테스트 시스템.

청구항 15

제13항에 있어서,

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 수단은,

만약 사용자 선택이 상기 마스크 값의 범위에 포함된다면, 상기 마스크 값에 대응하는 전체 등가 클래스를 선택된 것으로서 취급하는 수단을 포함하는 개방형 구조 테스트 시스템.

청구항 16

제15항에 있어서,

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 수단은,

사용자 선택의 하드웨어 버전의 각각에 대해, 상기 사용자 선택의 하드웨어 버전과 상기 마스크의 비트마다의 논리곱이 상기 마스크된 번호와 동일한 경우에만, 상기 사용자 선택의 하드웨어 버전이 유효하다고 표시하는 수단을 더 포함하는 개방형 구조 테스트 시스템.

청구항 17

제11항에 있어서,

상기 시스템 프로파일을 생성하는 수단은,

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 따라 호환성 있는 소프트웨어 컴포넌트의 집합을 결정하는 수단;

각각의 상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대해, 소프트웨어 컴포넌트의 버전을 선택하는 수단;

각각의 상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대해, 사용자에 의해 선택된 소프트웨어 컴포넌트의 목록을 생성하는 수단; 및

상기 사용자에 의해 선택된 소프트웨어 컴포넌트의 목록을 저장하는 수단을 포함하는 개방형 구조 테스트 시스템.

청구항 18

제17항에 있어서,

상기 소프트웨어 컴포넌트의 버전을 선택하는 수단은,

사용자에게, 상기 호환성 있는 소프트웨어 컴포넌트의 집합으로부터 소프트웨어 컴포넌트의 버전을 선택하도록 질의하는 수단; 및

사용자 선택의 상기 소프트웨어 컴포넌트의 버전의 각각에 대해, 지원되는 하드웨어 버전의 목록을 산출하기 위해, 상기 사용자 선택의 하드웨어 버전들에 걸쳐 교집합을 연산하는 수단을 포함하는 개방형 구조 테스트 시스템.

청구항 19

제17항에 있어서,

상기 교집합을 연산하는 수단은,

상기 사용자 선택의 하드웨어 버전을 이용하여 결과인 마스크를 연산하는 수단; 및

상기 사용자 선택의 상기 하드웨어 버전을 이용하여 결과인 마스크된 번호를 연산하는 수단을 포함하는 개방형 구조 테스트 시스템.

청구항 20

제11항에 있어서,

시스템 제어기, 하나 또는 그 이상의 사이트 제어기, 하나 또는 그 이상의 벤더 하드웨어 모듈, 및 하나 또는 그 이상의 피시험 장치를 포함하는 개방형 구조 테스트 시스템을 제공하는 수단;

상기 하드웨어 버전 번호의 등가 클래스의 멤버인, 벤더 하드웨어 모듈의 새로운 하드웨어 버전을 수신하는 수단; 및

상기 시스템 프로파일을 이용하여 상기 벤더 하드웨어 모듈의 상기 새로운 하드웨어 버전을 시험하는 수단을 더 포함하는 개방형 구조 테스트 시스템.

청구항 21

시스템 제어기, 사이트 제어기, 하나 또는 그 이상의 벤더 하드웨어 모듈 및 하나 또는 그 이상의 피시험 장치를 포함하는 개방형 구조 테스트 시스템에서의 벤더 하드웨어 모듈 버전을 표시하는 방법에 있어서,

벤더 하드웨어 모듈의 하드웨어 버전의 집합을 수신하는 단계; 및

마스크 번호 및 마스크를 포함하는 마스크 값을 이용하여, 상기 하드웨어 버전의 집합을 하드웨어 버전 번호의 등가 클래스로서 표시하는 단계를 포함하되,

상기 마스크는 상기 하드웨어 버전의 집합의 범위 내의 데이터 패턴이고,

상기 데이터 패턴은 적어도 돈케어 비트와 마스크 되지 않는 비트를 포함하는 벤더 하드웨어 모듈 버전 표시 방법.

청구항 22

제21항에 있어서,

상기 하드웨어 버전을 표시하는 단계는,

상기 마스크 내의 돈케어 비트의 각각에 대해, 이에 대응하는 상기 마스크된 번호 내의 비트를 마스크 하여, 상기 등가 클래스의 모든 멤버(member)가 상기 돈케어 비트에 대응하는 비트 위치 내에서 1 또는 0을 갖도록 지시하는 단계; 및

상기 마스크 내의 마스크 되지 않는 비트의 각각에 대해, 상기 등가 클래스 내의 상기 각 비트에 대응하는 상기 마스크된 번호의 비트를 이용하여, 상기 대응하는 마스크된 번호 내의 비트가 마스크 되지 않도록 지시하는 단계를 더 포함하는 벤더 하드웨어 모듈 버전 표시 방법.

청구항 23

제21항에 있어서,

사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전을 취득하는 단계; 및

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 단계를 더 포함하는 벤더 하드웨어 모듈 버전 표시 방법.

청구항 24

제23항에 있어서,

상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전의 유효성을 검증하는 단계는,

사용자 선택의 하드웨어 버전의 각각에 대해, 상기 사용자 선택의 하드웨어 버전과 상기 마스크의 비트마다의 논리곱(AND)이 상기 마스크된 번호와 동일한 경우에만, 상기 사용자 선택의 하드웨어 버전이 유효하다고 표시하는 단계; 및

사용자 선택이 유효하다고 결정된 경우, 상기 마스크 값에 대응하는 전체 등가 클래스를 선택된 것으로서 취급하는 단계를 더 포함하는 벤더 하드웨어 모듈 버전 표시 방법.

청구항 25

제21항에 있어서,

상기 하드웨어 버전 번호의 등가 클래스의 멤버인, 상기 벤더 하드웨어 모듈의 새로운 하드웨어 버전을 수신하는 단계; 및

상기 하드웨어 버전 번호의 등가 클래스를 이용하여 상기 벤더 하드웨어 모듈의 상기 새로운 하드웨어 버전을 시험하는 단계를 더 포함하는 벤더 하드웨어 모듈 버전 표시 방법.

명 세 서

기 술 분 야

<1> 본 발명은 반도체 시험을 위한 자동화된 시험 장비(automated test equipment; ATE}의 분야에 관한 것이다. 특히, 본 발명은 효율적인 필드(field) 배치를 위한, 개방형 구조 테스트 시스템에서의 벤더 하드웨어 모듈 버전의 간결한 표시에 관한 것이다.

배 경 기 술

<2> 개방형 구조 테스트 시스템에 있어서, 소프트웨어는, 수반된 모듈 소프트웨어의 각 버전과 함께 작동하는 것이 보장되는 벤더 하드웨어 모듈 버전의 목록과, 사용자 특정된 시스템을 구성하는 모든 벤더 하드웨어 모듈과, 그리고, 그러한 각각의 벤더 하드웨어 모듈에 대한 모든 조합된 서로 다른 모듈 소프트웨어 컴포넌트를 위한, 벤더 규정(specifications)에 접근할 필요가 있다. 그리고, 상기 시스템은, 상기 시스템의 소프트웨어 콘텐츠에 대한 후보로서 사용자에게 제공해야 할 소프트웨어 컴포넌트를 선택하기 위해, 이러한 복잡한 정보를 효율적으로 분석할 것이 요구된다.

<3> 이상적으로, 소프트웨어 컴포넌트에 대한 상기 제공된 선택지들은, 사용자가 주어진 선택지들로부터 고른 임의의 완성된 소프트웨어 집합이 서로 함께 작동할 것과, 사용자가 선택한 하드웨어 구성과도 함께 작동할 것이 보장된다는 총체적인 제한 조건 내에서, 가능한 한 큰 집합을 포함해야 한다. 이러한 문제의 맥락에서, 하나의 방법은, (인접하는 번호들을 나타내기 위한 범위 규정자(range specifier)와 함께) 상기 하드웨어 버전 번호를 표시하기 위해, 콤마로 구분되는 정수의 목록과 같은 각 소프트웨어 컴포넌트에 대해 호환성 있는 하드웨어 버전들을 열거하는 것이다. 이러한 방법은 많은 문제점을 지닌다.

<4> 첫째, 하드웨어 버전 번호는 종종 커서, 32비트나 필요로 하는데, 상기 번호 내의 서로 다른 비트-필드(bit-field)는 상기 하드웨어 벤더에 대해 서로 다른 독점적인 중요성을 갖는다. 만약 벤더가, 상기 번호 내의 비트들의 특정 패턴에 대하여 가능한 모든 값들을 갖는 모든 버전들이, 단일 소프트웨어 버전과 호환성이 있음을 증명하고자 한다면, 이는 다음과 같은 문제점을 야기할 수 있다: a) 상기 범위 규정자를 통해 효율적으로 표시될 수 없는 버전 번호들의 인접하지 않는 수열; 및/또는 b) 매우 긴 목록을 산출하는 매우 많은 수의 버전 번호들 - 이는 저장 및/또는 처리 목적으로는 매우 비실용적일 수 있음 -. 둘째, 상기 방법은, 특히 매우 빈번한 하드웨어 제품 갱신의 경우에 매우 제한적인데, 이는 상기 목록에의 빈번한 추가를 요구하고, 결과적으로 상기 추가를 통합하기 위해, 사용자의 사이트에 장황하고 빈번한 시스템 프로파일의 재생성 처리를 요구할 수 있다. 따라서, 효율적인 필드 배치를 위한, 개방형 구조 테스트 시스템에서의 벤더 하드웨어 모듈 버전의 간략한 표시가 요구된다.

발명의 상세한 설명

<5> 하드웨어 모듈 벤더가, 개방형 구조 테스트 시스템에 있어서, 조합된 모듈 소프트웨어의 주어진 버전과 함께 작동할 것이 보장되는 하드웨어 버전 번호의 임의의 수열을, 간결한 방식으로 규정할 수 있는 방법이 개시된다. 또한, 상기 하드웨어 버전 번호는 상기 벤더 하드웨어 모듈을 위한 소프트웨어 컴포넌트 선택지의 최적의 집합을 사용자에게 제공하도록 처리되는데, 이는 필드 내에서의 배치를 위한 유효한 시스템 구성을 이끌어낸다.

<6> 일 실시예에서, 개방형 구조 테스트 시스템에서 벤더 하드웨어 모듈 버전과 소프트웨어 컴포넌트를 조합하는 방법은, 벤더 하드웨어 모듈의 하드웨어 버전의 집합을 수신하는 단계, 상기 벤더 하드웨어 모듈에 의해 지원되는

소프트웨어 컴포넌트의 집합을 수신하는 단계, 마스크(mask) 값을 이용하여, 상기 하드웨어 버전의 집합이 하드웨어 버전 번호의 등가 클래스(equivalence class)로서 표시되도록, 상기 하드웨어 버전의 집합을 처리하는 단계, 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전을 취득하는 단계, 상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 단계, 및 상기 사용자 선택의 하드웨어 버전에 따라 시스템 프로파일을 생성하는 단계를 포함한다.

<7> 다른 실시예에서, 개방형 구조 테스트 시스템은 시스템 제어기, 상기 시스템 제어기에 연결된 사이트 제어기, 상기 사이트 제어기에 연결된 하나 또는 그 이상의 하드웨어 모듈, 하드웨어 모듈의 그룹에 의해 병렬로 원조되는 하나 또는 그 이상의 피시험 장치를 포함한다. 상기 시스템은, 벤더 하드웨어 모듈의 하드웨어 버전의 집합을 수신하는 수단, 상기 벤더 하드웨어 모듈에 의해 지원되는 소프트웨어 컴포넌트의 집합을 수신하는 수단, 마스크 값을 이용하여, 상기 하드웨어 버전의 집합이 하드웨어 버전 번호의 등가 클래스로서 표시되도록 상기 하드웨어 버전의 집합을 처리하는 수단, 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전을 취득하는 수단, 상기 사용자 선택의 상기 벤더 하드웨어 모듈의 하드웨어 버전에 대한 유효성을 검증하는 수단, 및 상기 사용자 선택의 하드웨어 버전에 따라 시스템 프로파일을 생성하는 수단을 포함한다.

<8> 또 다른 실시예에서, 개방형 구조 테스트 시스템에서의 벤더 하드웨어 모듈 버전을 표시하는 방법에 있어서, 벤더 하드웨어 모듈의 하드웨어 버전의 집합을 수신하는 단계, 및 마스크 번호 및 마스크를 포함하는 마스크 값을 이용하여, 상기 하드웨어 버전의 집합을 상기 하드웨어 버전 번호의 등가 클래스로서 표시하는 단계를 포함한다. 상기 마스크는 상기 하드웨어 버전의 집합의 범위 내의 데이터 패턴이고, 상기 데이터 패턴은 적어도 돈케어(don't care) 비트와 마스크 되지 않는 비트를 포함한다.

실 시 예

<15> 개방형 구조 테스트 시스템에서의 벤더 하드웨어 모듈 버전의 간결한 표시를 위한 방법과 시스템이 제공된다. 이하의 기재는 본 발명이 속하는 기술 분야에서 통상의 지식을 지닌 자라면 누구라도 본 발명을 만들고 사용할 수 있도록 기술된다. 특정 실시예 및 응용에 관한 기재는 단지 예로서 제공된다. 본 명세서에 기재된 예시들의 다양한 수정 및 조합이, 본 발명이 속하는 기술 분야에서 통상의 지식을 지닌 자에게 명확할 것이며, 본 명세서에서 정의된 전반적인 원리는 다른 예시와 응용에 적용될 수 있다. 따라서, 본 발명은 기재되고 도시된 예시들에 한정되도록 의도된 것이 아니며, 본 명세서에 개시된 원리들 및 특징들에 부합되는 가장 넓은 권리 범위가 부여되어야 할 것이다.

<16> 재구성 가능한(re-configurable) 개방형 구조 테스트 시스템은, 제3자 소프트웨어 및 하드웨어(벤더 하드웨어 모듈이라고도 함)가 개발되고, 개별적으로 증명되며, 상기 개방형 구조 테스트 시스템으로 신뢰성 있게 통합되는 것을 지원하는 솔루션을 제공한다. 도 1a는 본 발명의 일 실시예에 따른 하나 또는 그 이상의 SiteC로 복수의 DUT를 병렬 시험하는 개방형 구조 테스트 시스템을 도시한다. 상기 시스템은 시스템 제어기 102, 복수의 사이트 제어기 104, 복수의 벤더 하드웨어 모듈 106, 및 그에 대응하는 DUT의 그룹 110을 포함하며, 상기 DUT의 각 그룹은 하나 또는 그 이상의 개별 DUT 112를 포함할 수 있다. 시험 헤드 108이 상기 DUT의 그룹 110을 보유한다. 일 실시예에서, 상기 시스템 제어기 및 사이트 제어기(들)은 복수의 지리적인 위치에 있는 복수의 컴퓨터를 갖는 분산 컴퓨팅 시스템에 의해 구현될 수 있다. 다른 실시예에서, 상기 시스템 제어기 및 사이트 제어기는 단일 컴퓨터에 의해 구현될 수 있다. 개방형 구조 테스트 시스템에 관한 상세한 기재가, 미국 특허 출원 제60/447,839호의 "반도체 집적 회로에 대한 시험 프로그램 개발을 위한 방법 및 구조(Method and Structure to Develop a Test Program for Semiconductor Integrated Circuits)"와, 미국 특허 출원 제60/573,577호의 "개방형 구조 테스트 시스템에서의 소프트웨어 개발(Software Development in an Open Architecture Test System)"에 의해 제공된다. 또한, 개방형 구조 테스트 시스템은, 2004년 10월 발간된 '2004 IEEE 국제시험회의(*2004 IEEE International Test Conference*)'의 413페이지에서 422페이지에 게재된, 프라마닉(Pramanick) 등에 의한 "모듈식 개방형 구조 테스트 시스템에서의 시험 프로그래밍 환경(Testing Programming Environment in a Modular Open Architecture Test System)"에 또한 기재되어 있다.

<17> 상기 개방형 구조 테스트 시스템에서, 소프트웨어와 하드웨어 양쪽 모두의 각 모듈식 유닛은 제3자 솔루션이 통합 체제(framework)의 요건을 따르는 한, 제3자 유닛에 의해 대체될 수 있다. 따라서, 제3자 하드웨어는 디지털 핀 카드, 아날로그 카드, 디바이스 전원 장치 등과 같은 소정의 기능 유닛이 될 수 있다. 벤더 하드웨어 모듈(또는 간략히 모듈)이라는 용어는 그러한 하드웨어 기능 유닛을 일컫는 한편, 상기 벤더 하드웨어 모듈을 지원하는, 조합된 벤더 소프트웨어 컴포넌트는 모듈 소프트웨어라고 한다. 후자는, 모듈 제어 드라이버 소프트웨어, 교정(calibration) 및 진단(diagnostic) 소프트웨어, 에뮬레이션(emulation) 소프트웨어, 벤더에 특유한

패턴 컴파일러 등과 같은 다수의 별개의 소프트웨어 컴포넌트를 포함한다.

<18> 개방형 구조 테스트 시스템은 상기 테스트 시스템의 동작뿐만 아니라, 시험 환경의 제어 및 새로운 벤더 하드웨어 모듈과 그에 수반되는 모든 소프트웨어의 도입에 대비한다. 도 1b는 개방형 구조 테스트 시스템에서의 복수의 벤더 하드웨어 모듈 버전 128 및 그에 대응하는 소프트웨어 컴포넌트 버전의 예를 도시한다. 상기 소프트웨어 컴포넌트는, 벤더 소프트웨어 드라이버 124의 복수의 버전들뿐만 아니라, 테스터 운영 체제(Tester Operating System; TOS) 120의 복수의 버전들과 그와 조합된 벤더 소프트웨어 개발 키트(software development kit; SDK)들 122, 사용자 SDK들 126의 버전들을 포함한다. 개방형 구조 테스트 시스템에서 복수의 벤더 하드웨어 모듈 버전과 TOS들, 벤더 SDK들 및 사용자 SDK들과 같은 복수의 소프트웨어 컴포넌트 버전들을 관리하기 위한 방법이, 미국 특허 출원 제60/635,094호의 "테스터 도구 모듈의 설치 및 구성 관리 수행을 위한 방법 및 시스템(Method and System for Performing Installation and Configuration Management of Tester Instrument Modules)"에 의해 제공되고, 이는 본 출원의 양수인에게 양도되었다.

<19> 상기 테스트 시스템은, 벤더 하드웨어 모듈 M에 대한 모듈 소프트웨어의 특정 버전 n이 특정의 시스템 구성 내에서 공존할 수 있는 벤더 하드웨어 모듈들의 소정의 서로 다른 버전들과 호환성이 있음을 모듈 M의 벤더가 증명하는 한, 상기 소프트웨어의 버전 n이 상기 특정의 시스템 구성과 호환성이 있는 것으로 상기 테스트 시스템에 의해 검증될 수 있도록, 충분히 융통성이 있을 것이 요구된다. 상기 벤더는 모듈 M에 대해, 모듈 소프트웨어의 버전 n과 함께 작동할 것이 보장되는 상기 M을 위한 하드웨어 버전들의 목록을 공개하도록 요구된다.

<20> 사용자가 벤더 하드웨어 모듈들의 특정 버전들과 그와 조합된 모듈 소프트웨어로 개방형 구조 테스트 시스템을 구성하는 때, 상기 사용자의 선택의 결과로서 실제로 시스템이 작동한다고 검증하는 것은 시스템 소프트웨어에 달려있다. 이러한 이유로, 상기 개방형 구조 테스트 시스템은, 대형 시스템에는 조합된 복수의 소프트웨어 컴포넌트를 갖는 서로 다른 많은 벤더 하드웨어 모듈들이 포함될 수 있다는 점을 기초로, 상기 사용자로 하여금 하드웨어 및 소프트웨어 모두의 시스템 구성에 대해 유효한 선택만을 하도록 하는, 적절한 안내를 받은 사용자에 의해 생성되는 시스템 프로파일에 의존할 필요가 있다 - 상기 안내는, 프로파일 생성 동안 상기 시스템으로부터 사용자에게 제공됨 -. 따라서, 시스템 프로파일을 생성하기 위해 사용자에게 적절한 안내를 제공하는 동시에 상기 사용자를 불필요하게 제한하지 않는 작업은 매우 복잡하다.

<21> 일반적으로, 효율적인 필드 배치를 위해, 개방형 구조 테스트 시스템에서의 벤더 하드웨어 모듈 버전의 간결한 표시를 제공하는 방법은, 두 가지의 주요한 부분을 포함한다: 1) 하드웨어 버전을 규정하는 단계; 및 2) 하드웨어 버전 정보를 처리하고, 사용자가 유효한 시스템 프로파일을 생성할 수 있도록 하는 단계. 본 발명이 속하는 기술 분야에서 통상의 지식을 지닌 자는, 상기 방법이, 개방형 구조 테스트 시스템에서의 TOS들, 벤더 SDK들 및 사용자 SDK들의 버전들과 같은, 소프트웨어 컴포넌트 버전을 표시하기 위해 적용될 수 있다는 점을 인식할 것이다.

<22> 일 실시예에서, 호환성 있는 하드웨어 버전의 규정은, 상기 하드웨어 버전 번호 내의 (벤더에 특유한) 비트들의 임의의 집합을 마스크 함으로써 수행된다. 예를 들면, 하드웨어 버전 번호의 구문(syntax)은 규정 파일(specification file)에서 이하와 같이 표시될 수 있다: 1095188784, 0x01403500 / 0x0ff0ff00, 8800000111, 여기서 번호 "1095188784" 및 "8800000111"는 마스크 되지 않는 단일 하드웨어 버전을 나타내는 반면, 번호 "0x01403500 / 0x0ff0ff00"는, 등가인 하드웨어 버전 번호들의 전체 집단(family)(클래스)을 표시하는 마스크된 번호를 나타낸다.

<23> 본 예에서, 마스크된 값은 구문 "n/m" 내에서 16진법의 형식으로 표시되는데, 여기서 'n'은 마스크된 번호를 표시하고 'm'은 마스크를 표시한다. "n/m"이라는 표현에서, 마스크된 번호 n에 대한 상기 마스크 m은, n에 기초하여 "EQC(n/m)"으로 표현된 번호들의 등가 클래스(equivalence class; EQC)를 규정하는 비트 패턴으로서 정의되는데, 1) m 내의 모든 돈케어(don't care) 비트(본 예에서는 "0" 비트로 표시됨)에 대해, 그에 대응하는 n 내의 비트가 마스크 되어, 상기 등가 클래스의 모든 멤버(member)가 그 비트 위치 내에서 1 또는 0 중 하나의 값을 가질 수 있음을 나타내고, 2) m 내의 마스크 되지 않는 모든 비트(본 예에서는 "1" 비트로 표시됨)에 대해, 그에 대응하는 상기 등가 클래스 내의 임의의 번호 내의 비트는 n 내의 대응하는 비트와 동일하다.

<24> 도 2는 본 발명의 실시예에 따른 마스크된 번호에의 마스크의 적용을 도시한다. 도 2에서의 "010/110"으로 된 요소(설명의 용이함을 위해 16진법 형식 대신 2진법 형식으로 표현됨)를 살펴본다. 마스크 m 202가 버전 번호의 등가 클래스 206을 생성하기 위해 번호 n 204에 적용된다.

<25> 상기 주어진 마스크의 정의를 적용하면, EQC(010/110)의 멤버인 번호 x는 비트 위치 2 및 3에서 각각 '1'과 '

0'을 갖는다(오른쪽 즉 최하위로부터, 왼쪽 즉 최상위로 3개의 비트를 카운트함). 이것은 마스크 "110" 내의 비트 위치 2와 3이 1(즉, 마스크 되지 않음)이기 때문인데, 상기 '마스크 되지 않음'은, x 내에서, 이러한 두 개의 비트가 n 내의 대응하는 비트의 바로 그 값을 갖는다는 것을 의미한다. 그러나, x는 비트 위치 1에서는 1 또는 0 중 하나를 가질 수 있는데, 이는 상기 마스크 "110" 내의 비트 1의 값이 '0'(즉, 마스크 됨)이기 때문이다. 따라서, EQC(010/110)은 {010,011}의 집합, 즉 집합 {2, 3}이고, 따라서 x는 2 또는 3일 수 있다.

<26> 개방형 구조 테스트 시스템에서, 공통으로 수행되는 동작은 사용자로부터 제공받은 하드웨어 버전 번호 x가 마스크된 등가 클래스의 멤버인지를 결정하는 것이다. 마스크된 등가 클래스 내의 멤버 자격(membership)을 결정하는 한가지 방식은, x & m = n - 여기서, "&"은 비트 단위의 논리곱(AND) 연산자임 - 인 경우에만, 번호 x가 EQC(n/m)의 멤버라고 나타내는 것이다. 이러한 속성은, 요구되는 결정을 편리하게 수행하는 방식을 제공한다.

<27> 다음으로, 인접하지 않은 다수의 버전 번호들의 등가 클래스의 마스크된 표시를 이용하여 시스템 프로파일을 산출하는 방법에 대해 기재한다. 일반적으로, 상기 방법은 1) 마스크된 하드웨어 버전의 집합을 처리하는 단계; 2) 사용자가 선택한 벤더 하드웨어 모듈의 버전을 구동하기 위한, 호환성 있는 소프트웨어 컴포넌트 버전의 선택지를 사용자에게 제공하는 단계; 3) 상기 모듈에 대한 상기 사용자 선택의 호환성 있는 소프트웨어 컴포넌트 버전으로부터 산출되는, 각 벤더 하드웨어 모듈에 대한 하드웨어 버전 집합의 간결한 표시를 연산하는 단계; 및 4) 상기 사용자의 시스템 프로파일 내에 상기 간결한 표시를 저장하는 단계를 포함한다.

<28> 상기 마스크된 하드웨어 버전의 집합을 처리하기 위해, 벤더 하드웨어 모듈들과 그와 조합된 소프트웨어 컴포넌트들의 버전들을 표시하기 위해 사용되는 기호가 이하에서 기술된다. 우선, 특정 벤더 하드웨어 모듈을 M으로 나타내고, ATE 내에서 상기 하드웨어 모듈 M을 제어하기 위해 요구되는 소프트웨어 컴포넌트를 집합 {MX}로 나타내도록 하는데, 여기서, X는 A, B, C 등이 될 수 있다. 따라서, 예를 들면, M이 총 4개의 소프트웨어 컴포넌트를 포함한다면, 상기 컴포넌트들은 MA, MB, MC, MD로 표시된다. 다음으로, 상기 벤더 하드웨어 모듈 M에 대한 소프트웨어 컴포넌트 X의 특정 버전 k는 MX.k로 나타내도록 한다. 예를 들면, 모듈 M에 대한 컴포넌트 A의 버전 1은 MA.1로 나타내어진다. 나아가, 벤더 하드웨어 모듈 M에 대한 소프트웨어 컴포넌트 X의 k번째 버전에 의해 지원되는 하드웨어 버전들의 집합은, HWR(MX.k)로 나타내어진다. 예를 들면, HWR(MA.1)은 집합 {3, 5, 010/011}을 표시한다. 단순화를 위해, 마스크된 번호들은 2진법 형식으로 표시된다는 점을 주의하여야 한다. 본 발명이 속하는 기술 분야에서 통상의 지식을 지닌 자는 마스크된 번호들이 16진법 또는 다른 형식으로 표시될 수 있음을 인식할 것이다.

<29> 도 3은 본 발명의 일 실시예에 따른, 개방형 구조 테스트 시스템에서 서로 다른 벤더 하드웨어 모듈 버전을 테스트하기 위한 시스템 프로파일을 생성하는 방법을 도시한다. 상기 방법은 "시작" 상태(302)에서 개시되고, 다음으로 단계 1(304)로 이동하는데, 단계 1에서 상기 방법은, 상기 테스트 시스템 내의 모듈 M에 대한 하드웨어 버전 번호들의 선택지(들)에 대해 사용자들에게 질의한다. 상기한 바와 같이, 하드웨어 버전에 대해 임의적으로 마스크된 값과 그 결과인 하드웨어 버전 번호들의 집단(families)의 사용은, 사용자에게 등가 클래스 EQC(n/m)를 표시함에 있어서 장애가 된다. EQC(n/m)의 멤버 중 모든 가능한 값 x를 전부 열거하는 것(그리고, 그 후의 보수 및 유지)이 기하급수적으로 복잡해질 수 있는데, m 내에 규정된 마스크된 비트의 수의 관점에서 특히 그러하다. 심지어 적당하게 큰 m의 값들(예를 들면, m이 32보다 작거나 같은 경우)에 대해서도, 전부 열거하는 상기 방법은 명확하게 비효율적이고 비실용적이다. 따라서, 사용자에게 하드웨어 버전 값(들)에 관한 그의 선택을 입력하도록 요구한다.

<30> 단계 2(306)에서, 상기 방법은 각 사용자 입력(entry)이 유효한 선택인지를 결정한다. 상기 결정은, 하드웨어 버전 번호가 상기 EQC의 멤버인지를 결정함에 있어서 상기한 방법을 이용하여 수행된다. 만약 사용자 입력(들)이 유효하다면(306_예), 상기 방법은 단계 3으로 넘어간다. 만약 그렇지 않다면(306_아니오), 상기 방법은 단계 1을 반복하여 사용자에게 하드웨어 버전 번호를 재입력하도록 질의한다.

<31> 단계 3(308)에서, 하드웨어 버전 번호(들)에 관한 성공적인 사용자 입력에 기초하여, 상기 방법은 HWR(MX.k)(이는, M에 대한 사용자 선택의 하드웨어 버전(들)임)에 호환 가능한 소프트웨어 컴포넌트 X의 호환성 있는 버전 k의 집합을 결정한다. 각 단계에서, 마스크된 값들에 대해서 하드웨어 버전 번호(들)를 테스트하기 위해 상기한 방법이 사용된다. 사용자의 선택의 각각에 대해, 그것이 만약 마스크된 값의 범위에 포함된다면, 상기 마스크된 값에 대응하는 전체 등가 클래스가 선택된 것으로 간주된다. 이러한 속성은 하드웨어 버전 번호들의 효율적인 간결한 표시를 얻기 위해 중요하다.

<32> 단계 4(310)에서, 상기 방법은, 상기 결정된 호환성 있는 소프트웨어 컴포넌트의 집합으로부터 소프트웨어 컴포넌트의 하나의 버전을 선택하도록 사용자에게 질의한다.

<33> 단계 5(312)에서, 상기 방법은 단계 1에서의 하드웨어 버전 선택의 각각에 대해, 모든 X에 걸쳐서, HWR(MX.p)의 교집합을 연산하는데, 여기서 p는 X로부터의, 상기 선택된 소프트웨어 컴포넌트의 버전을 나타낸다. 교집합의 연산에 관한 상세한 설명은 도 4에 관한 논의와 관련하여 이하에서 기재된다. 본 단계는, 사용자에 의해 선택된 소프트웨어 컴포넌트 X의 각각의 버전 p에 대해, 모든 HWR(MX.p) 집합들의 교집합을 연산함으로써, 호환성 있는 모든 하드웨어 버전들의 전체적인 목록을 구한다. 본 단계는, 본 연산을 사용자가 선택한 하드웨어 버전의 각각에 대해 반복하고(각 사용자 선택은 단일 값 또는 단일 EQC가 됨) 그 결과를 수집한다.

<34> 단계 6(314)에서, 상기 방법은 M에 대해, 선택된 모든 하드웨어 버전에 걸쳐서 상기 단계 5로부터 수집된 결과들을 이용하여, 사용자의 소프트웨어 버전 선택들에 대해 호환성이 있는 하드웨어 버전의 최종 목록을 연산한다.

<35> 단계 7(316)에서, 상기 방법은 사용자의 프로파일에 상기 연산된 최종 목록을 저장한다. 상기 방법은 "끝" 상태(318)에서 종료한다.

<36> 도 4는, 본 발명의 일 실시예에 따른 마스크된 표시의 교집합을 연산하는 방법을 도시한다. 도 4에 도시된 바와 같이, 두 개의 컴포넌트 m_I, m_J의 교집합은 각 선택이 마스크된 것(즉, 완전한 등가 클래스)임을 가정하여 연산된다. 본 발명이 속하는 기술 분야에서 통상의 지식을 지닌 자는, 교집합을 연산하는 상기 방법이, 하드웨어 버전 번호에 관한 마스크 되지 않는 선택 대(versus) 마스크된 선택, 또는 마스크 되지 않는 선택 대 마스크 되지 않는 선택으로 확장될 수 있음을 인식할 것이다. 도 4에 도시된 바와 같이, m_I에 대한 사용자 선택은 EQC(n_I/m_I)로 표시되고, m_J에 대한 사용자의 선택은 EQC(n_J/m_J)로 표시된다. 상기 교집합의 결과를 EQC(n/m)이라고 하자.

<37> 상기 방법은 "시작" 상태(402)에서 개시되고, 다음으로 단계 1(404)로 이동하는데, 상기 단계 1에서 상기 방법은 결과인 마스크를 연산한다. "|"라는 기호는 비트마다의 논리합(OR) 연산을 표시한다. m_I 및 m_J의 비트마다의 논리합의 결과가 실제로 상기 결과인 마스크인데, 왜냐하면 상기 결과가, 상기 비트마다의 논리합 연산에 대한 정의와 마스크에 대한 상기 정의에 따라, m_I와 m_J 모두에 의해 만족되는 유일한 마스크이기 때문이다. 만약 m이 비어있다면(즉, 모두 1임) EQC(n/m)은 단일 값 n을 포함한다. 단계 2(406)에서 상기 방법은 결과인 마스크된 번호 n을 연산한다. 상기 방법은 "끝" 상태(408)에서 종료된다. 본 발명이 속하는 기술 분야에서 통상의 지식을 지닌 자는, 상기 교집합의 연산을 둘 이상의 컴포넌트로 확장하기 위하여, 상기 연산에서의 다음 차례의 반복에 대한 시작 값으로서, 단계 2에서 연산된 n 및 m의 값을 이용할 수 있다는 점 등을 인식할 것이다.

<38> 이하의 예시는 상기 도 3과 도 4에서 기재된 방법에 적용된다. 특정 벤더 하드웨어 모듈 M이 세 개의 소프트웨어 컴포넌트, 즉 컴포넌트 MA, 컴포넌트 MB, 컴포넌트 MC를 갖는다고 가정한다. 또한, 상기 컴포넌트들의 이용 가능한 소프트웨어 버전들과 그에 대응하는 지원되는 하드웨어 버전들에 대해 특정 사용자가 갖는 선택들의 집합이 표 1에 주어져 있다고 가정하고, 상기 표 1에서는 "MX.n"이라는 기호가 소프트웨어 컴포넌트 MX의 n번째 버전을 의미한다. 설명의 용이함을 위해, 마스크된 번호들은 16진법 대신 2진법으로 표시된다. 또한, 모든 가능한 하드웨어 버전들의 집합이 3개의 비트에 의해 표시될 수 있다고 가정된다.

표 1

<39>

모듈 M에 대한 소프트웨어 컴포넌트	소프트웨어 컴포넌트에 의해 지원되는 하드웨어 버전
MA.1	3, 5, 010/011
MA.2	5, 000/100
MA.3	000/110
MB.1	010/010, 5
MB.2	000/000
MC.1	5, 010/110

<40> 궁극적인 목적은, 모듈 M에 대해, 선택된 하드웨어 버전(들)과 함께 작동하도록 이용될 수 있는 *각* 소프트웨어 컴포넌트의 *하나의* 버전(즉, MA, MB 및 MC 각각의 하나의 버전)을 사용자로 하여금 *정확하게* 선택할 수 있도록 하는 것이라는 점을 유의하여야 한다.

<41> 도 3의 방법을 상기 예시에 적용하면:

<42> 1. 사용자 선택지들이 전체 *MX.n* 집합, 즉, {3, 5, 010/011, 000/100, 000/110, 010/010, 000/000, 010/110}에 대해 시험됨.

<43> 사용자가 M에 대해 하드웨어 버전 '2' 및 '5'를 선택한다고 가정한다. 이는 다음과 같은 조합을 이끌어낸다.

<44> a. 하드웨어 버전 '2'에 대해, 상기 조합은, (즉, 집합$HWR(MX.n)_2$는):

<45> *MA*.1 ⇔ 010/011

<46> *MA*.2 ⇔ 000/100

<47> *MA*.3 ⇔ Ø('2'는 *MA*.3에 의해 지원되지 않음)

<48> *MB*.1 ⇔ 010/010

<49> *MB*.2 ⇔ 000/000

<50> *MC*.1 ⇔ 010/110

<51> b. 하드웨어 버전 '5'에 대해, 상기 조합은, (즉, 집합$HWR(MX.n)_5$는):

<52> *MA*.1 ⇔ 5

<53> *MA*.2 ⇔ 5

<54> *MA*.3 ⇔ Ø('5'는 *MA*.3에 의해 지원되지 않음)

<55> *MB*.1 ⇔ 5

<56> *MB*.2 ⇔ 000/000

<57> *MC*.1 ⇔ 5

<58> 2. 상기 조합은 상기 사용자를 위한 이용 가능한 소프트웨어 컴포넌트 선택들이 다음과 같음을 나타낸다.

<59> {*MA*.1, *MA*.2} 중 하나,

<60> {*MB*.1, *MB*.2} 중 하나, 및

<61> {*MC*.1} 중 하나.

<62> 3. 사용자(상기 2에서, 각 컴포넌트 그룹으로부터 하나를 선택함)가 *MA*.2, *MB*.2 및 *MC*.1을 선택한다고 가정한다. 그러면,

<63> a. 하드웨어 버전 선택 '2'에 대하여, 지원되는 하드웨어 버전의 목록은 $HWR(MA.2)_2$, $HWR(MB.2)_2$ 및 $HWR(MC.1)_2$의 집합들의 교집합에 의해, 다음 단계에 따라 주어진다:

<64> ⅰ. 도 4의 상기 방법을 이용하여, $HWR(MA.2)_2$와 $HWR(MB.2)_2$의 교집합을 다음과 같이 연산한다: 000/100 ∩ 000/000 ⇒ 000/100 (여기서, 000 & 100 = 000이고, 000 & 000 = 000임).

<65> ⅱ. $HWR(MC.1)_2$와 000/100의 교집합을 연산한다: 000/100 ∩ 010/110 ⇒ 010/110 (여기서, 010 & 100 = 000이고, 010 & 110 = 010임).

<66> ⅲ. 따라서, 하드웨어 버전 '2'의 선택에 대한 최종 등가 클래스는 010/110, 즉, {2, 3}이다.

<67> b. 하드웨어 버전 선택 '5'에 대하여, 지원되는 하드웨어 버전의 목록은 $HWR(MA.2)_5$, $HWR(MB.2)_5$ 및 $HWR(MC.1)_5$의 집합들의 교집합에 의해 주어진다:

<68> ⅰ. 우선, 도 3의 상기 방법을 이용하여, $HWR(MA.2)_5$와 $HWR(MB.2)_5$의 교집합을 다음과 같이 연산한다: 5 ∩ 000/000 ⇒ 101/111 ∩ 000/000 ⇒ 101/111.

<69> ⅱ. 다음으로, $HWR(MC.1)_5$와 101/111의 교집합을 연산한다: 101/111 ∩ 5 ⇒ 101/111 ∩ 101/111 ⇒ 101/111

⇒ 5.

<70> iii. 따라서, 하드웨어 버전 '5'의 선택에 대한 최종 값은 5이다.

<71> 4. 상기 3a 및 3b로부터, 도 3의 단계 6을 적용하면, 소프트웨어 컴포넌트 *MA*.2, *MB*.2, *MC*.1에 관한 선택에 대해, 사용자의 프로파일 내에 포함될 지원되는 하드웨어 버전의 목록은 {010/110, 5}, 즉 {2, 3, 5}이다.

<72> 따라서, 하드웨어 모듈 벤더는, 개방형 구조 테스트 시스템에서, 조합된 모듈 소프트웨어의 주어진 버전과 함께 작동할 것이 보장되는 하드웨어 버전 번호의 임의의 수열을 규정할 수 있다. 또한, 상기 하드웨어 버전 번호는 벤더 하드웨어 모듈에 대한 소프트웨어 컴포넌트 선택지의 최적의 집합을 사용자에게 제공하도록 처리되며, 이는 필드 내에서의 배치를 위한 유효한 시스템 구성을 이끌어낸다.

<73> 본 발명은 적어도 이하의 이점을 제공한다. 첫째, 지원되는 모듈 하드웨어 버전의 간략한 표시를 사용하여, 벤더 하드웨어 모듈에 대한 지원되는 실제의 제품 버전의 전체 집단을 표시한다. 둘째, 상기 개시된 방법은 벤더로 하여금, 규정된 값에 의해 함축적으로 지시되는 모든 하드웨어 버전을 동시에 지원하는 호환성 있는 소프트웨어 컴포넌트에 대한 하드웨어 버전 규정을 공개할 수 있도록 한다. 셋째, 임의의 비트 패턴이 벤더에 의해 규정될 수 있고, 따라서 벤더는 인접하는 버전 번호만을 선택하도록 제한되지 않는다. 넷째, 벤더에 특유한 값이, 집합을 구성하는 개별적인 하드웨어 버전 번호 모두를 간결하게 수집하는 방식으로 개방형 구조 테스트 시스템 소프트웨어에 의해 처리될 수 있다. 다섯째, 상기 개시된 방법은, 사용자가 선택한 벤더 하드웨어 모듈과 조합되는 수많은 소프트웨어 모듈에 대한, 복수의 호환성 있는 하드웨어 버전 목록을 효율적으로 분석하고, 그리고 올바른 시스템 동작을 보장하면서 선택을 위한 호환성 있는 소프트웨어 컴포넌트의 최적의 집합을 사용자에게 제공한다. 마지막으로, 상기 개시된 방법은, 선택된 각 모듈에 대한 등가의 하드웨어 버전의 전체 집단의, 최적으로 간결한 표시를 연산하고, 상기 사용자의 시스템 프로파일에 저장한다. 따라서, 벤더 하드웨어 모듈을 동일한 집단으로부터의 다른 모듈과 교체하는 것이 하드웨어 버전 목록을 다시 릴리스(re-release)하도록 요구하지 않고, 상기 시스템 프로파일을 재생성하도록 요구하지도 않는다.

<74> 상기의 기재는 명확성을 위하여, 서로 다른 기능적인 유닛들 및 프로세서들을 참조하여 본 발명의 실시예에 대해 기술하였음을 이해하여야 할 것이다. 그러나, 본 발명으로부터 이탈되지 아니한 채, 서로 다른 기능적인 유닛들 또는 프로세서들 간에 기능의 적절한 분산이 이루어질 수 있음은 명백할 것이다. 예를 들면, 서로 구분되는 프로세서들 또는 제어기들에 의해 수행되도록 기술된 기능은, 하나의 프로세서 또는 제어기에 의해 수행될 수 있다. 따라서, 특정한 기능적 유닛에 대한 참조는, 한정된 논리적 또는 물리적 구조나 체계를 나타내는 것이라기보다는, 단지 상기 기술된 기능을 제공하기 위한 적절한 수단에 대한 참조에 지나지 아니한다.

<75> 본 발명은, 하드웨어, 소프트웨어, 펌웨어, 또는 이들의 조합을 포함하는 적절한 형태로 구현될 수 있다. 본 발명은 선택적으로, 하나 또는 그 이상의 데이터 프로세서 및/또는 디지털 시그널 프로세서에서 구동되는 컴퓨터 소프트웨어로서 부분적으로 구현될 수도 있다. 본 발명의 일 실시예의 구성 요소 및 컴포넌트는 물리적으로, 기능적으로, 그리고 논리적으로 소정의 적절한 방법으로 구현될 수 있다. 실제로 기능성은 단일 유닛으로, 복수의 유닛으로, 또는 다른 기능적인 유닛의 일부로서 구현될 수도 있다. 유사하게, 본 발명은 단일 유닛으로 구현될 수도 있고, 또는 서로 다른 유닛들과 프로세서들 간에 물리적으로 그리고 기능적으로 분산될 수도 있다.

<76> 본 발명이 속하는 기술 분야에서 통상의 지식을 지닌 자는, 근원이 되는 동일한 메커니즘과 방법론을 채택하면서, 개시된 실시예의 많은 가능한 변경과 조합이 사용될 수 있다는 것을 인식할 것이다. 설명의 목적으로, 상기의 기재는 특정 실시예를 참조로 하여 기술되었다. 그러나, 상기의 예시적인 논의는 모든 것을 남김없이 규명하거나 또는 본 발명을 개시된 정확한 형태로 제한하고자 의도된 것은 아니다. 많은 변경과 변형이 상기 시사(示唆)의 관점에서 가능하다. 상기 실시예들은 본 발명의 원리와 그 실제 응용을 설명하기 위해서, 그리고 본 발명이 속하는 기술 분야에서 통상의 지식을 지닌 자가 본 발명, 그리고, 고려될 수 있는 특정 사용에 적합한 다양한 변형을 갖는 다양한 실시예를 이용할 수 있게 하도록 선택되고 기술되었다.

도면의 간단한 설명

<9> 본 발명의 상기한 특징 및 이점뿐만 아니라 그로부터의 추가적인 특징과 이점이, 이하의 도면들과 관련된 본 발명의 실시에 관한 상세한 설명으로부터 더욱 명확하게 이해될 것이다.

<10> 도 1a는 본 발명의 일 실시예에 따른, 하나 또는 그 이상의 SiteC로 복수의 DUT를 병렬 시험하는 개방형 구조 테스트 시스템을 도시한다.

<11> 도 1b는 개방형 구조 테스트 시스템에서의 복수의 벤더 하드웨어 모듈 버전과 그에 대응하는 소프트웨어 컴포넌트 버전의 일례를 도시한다.

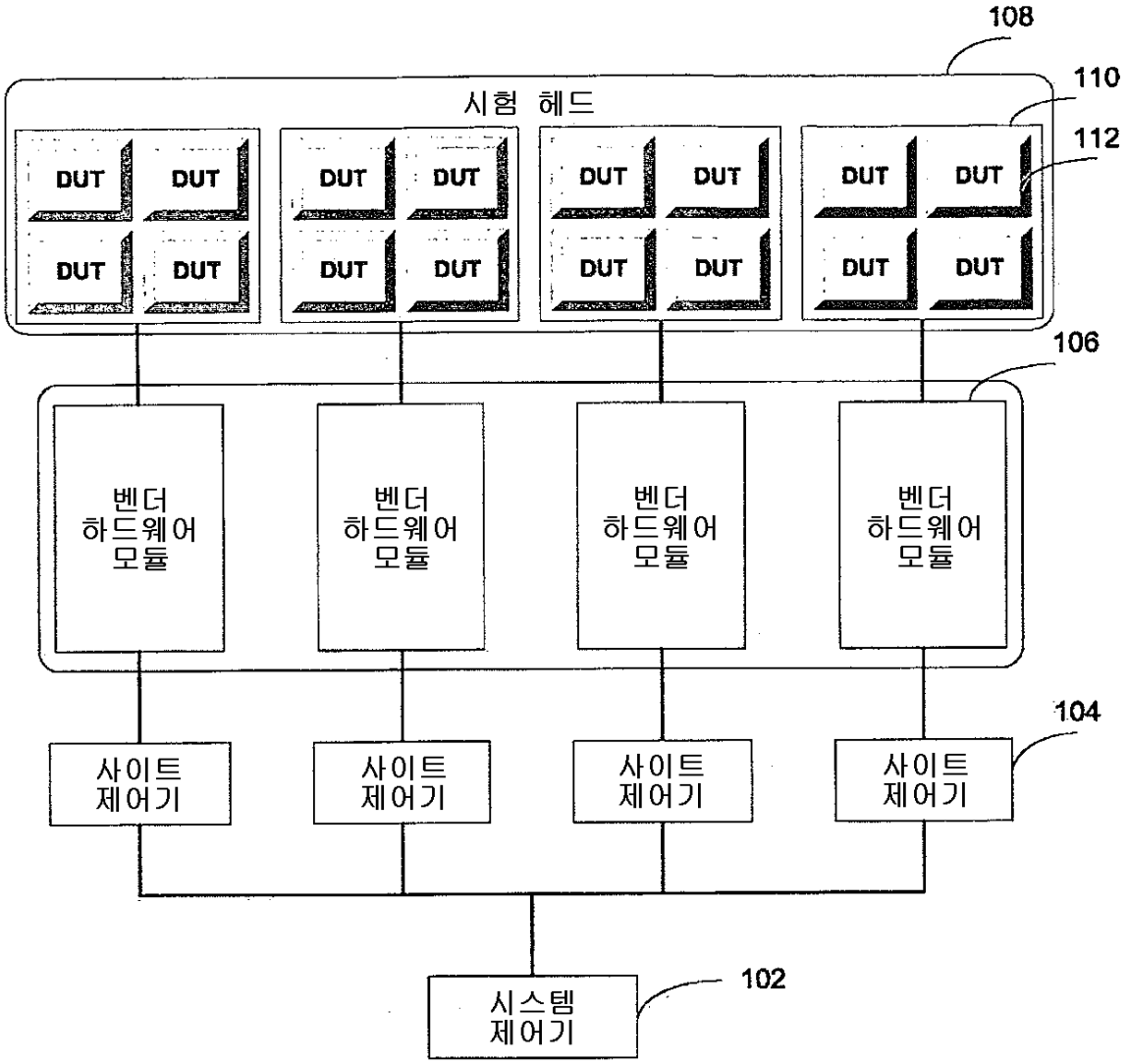
<12> 도 2는 본 발명의 실시예에 따른 마스크된 번호에의 마스크의 적용을 도시한다.

<13> 도 3은 본 발명의 일 실시예에 따른, 개방형 구조 테스트 시스템에서의 서로 다른 벤더 하드웨어 모듈 버전을 시험하기 위한 시스템 프로파일을 생성하는 방법을 도시한다.

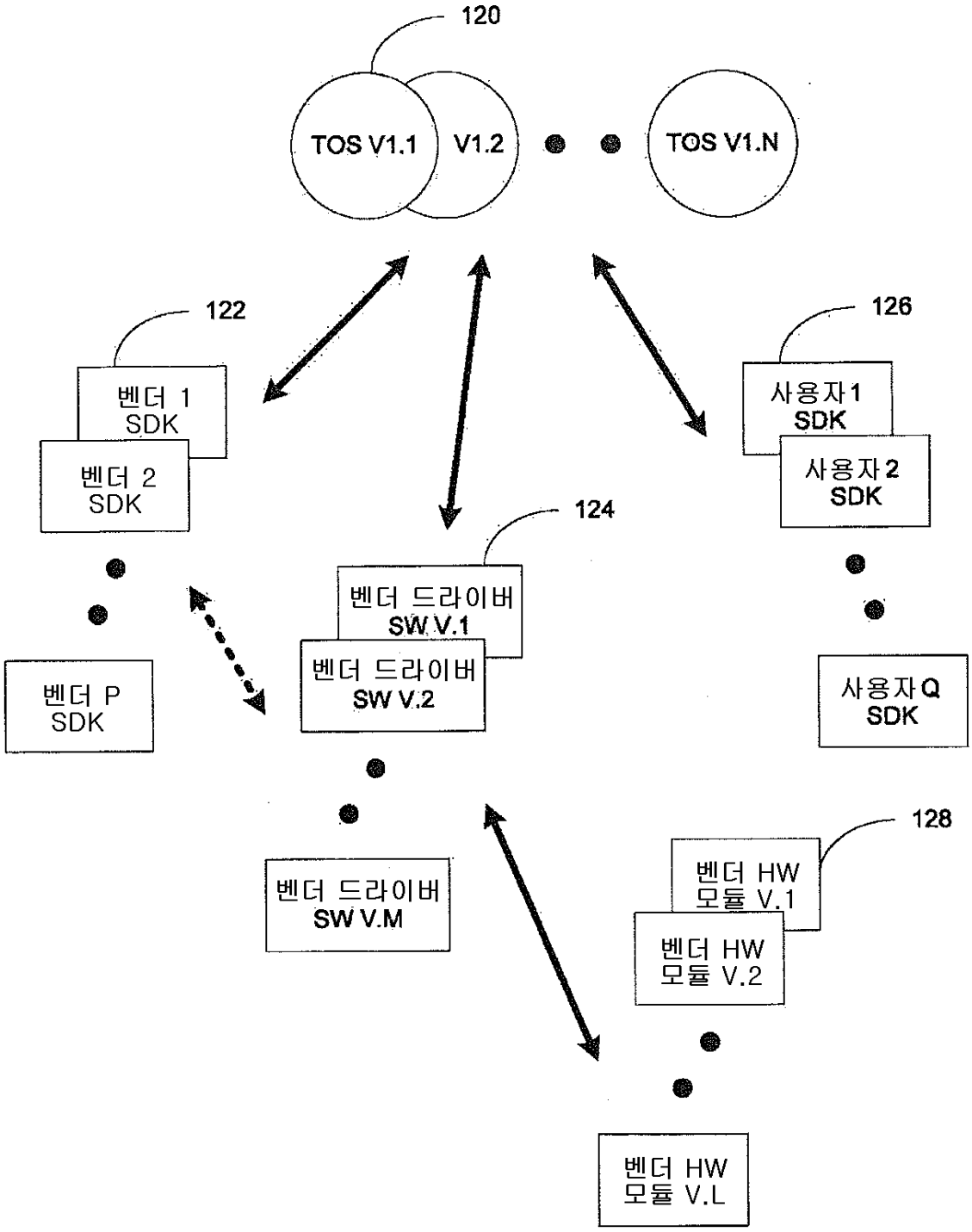
<14> 도 4는 본 발명의 일 실시예에 따른 마스크된 표시의 교집합을 연산하는 방법을 도시한다.

도면

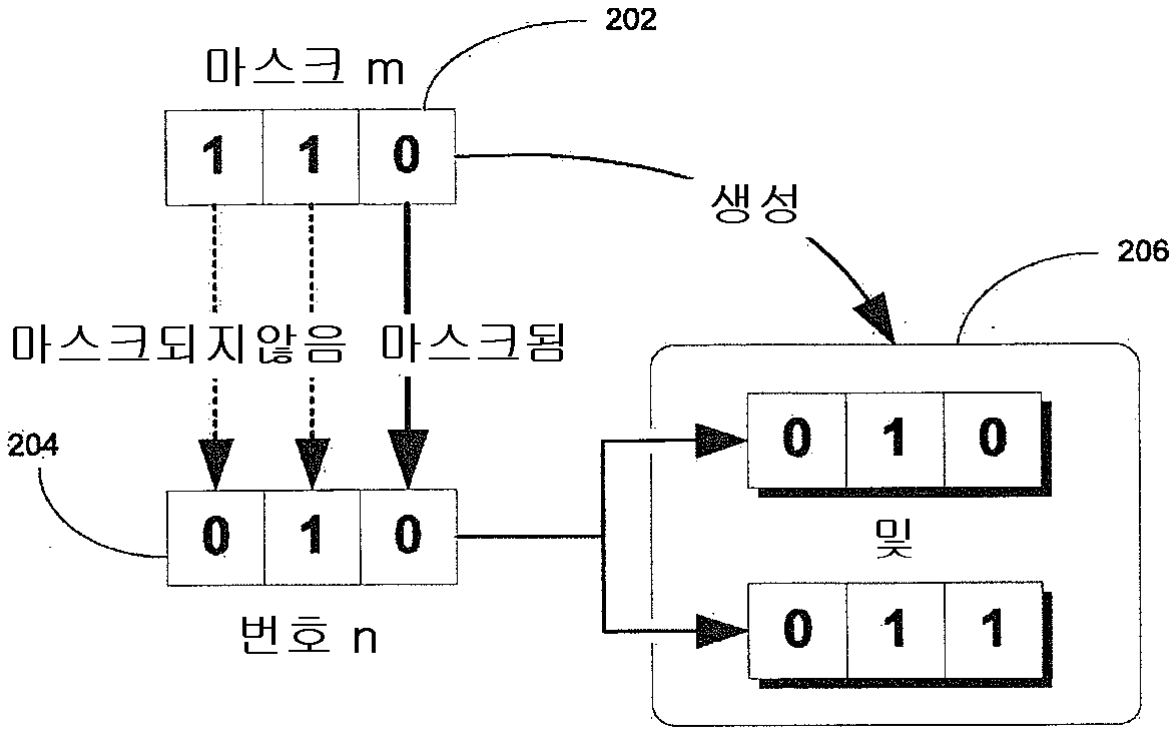
도면1a



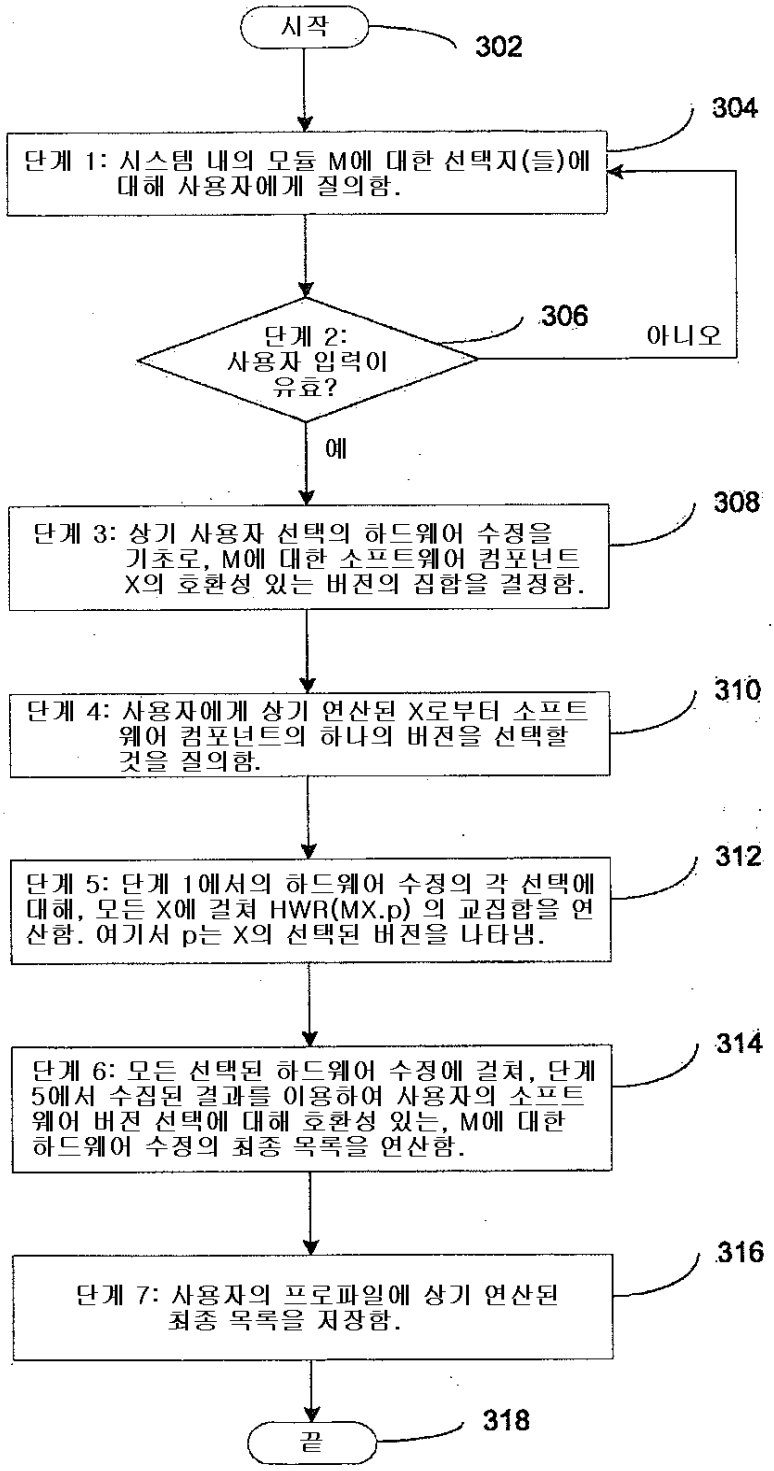
도면1b


120
TOS V1.1
V1.2
TOS V1.N
122
벤더 1 SDK
벤더 2 SDK
벤더 P SDK
124
벤더 드라이버 SW V.1
벤더 드라이버 SW V.2
벤더 드라이버 SW V.M
126
사용자1 SDK
사용자2 SDK
사용자Q SDK
128
벤더 HW 모듈 V.1
벤더 HW 모듈 V.2
벤더 HW 모듈 V.L

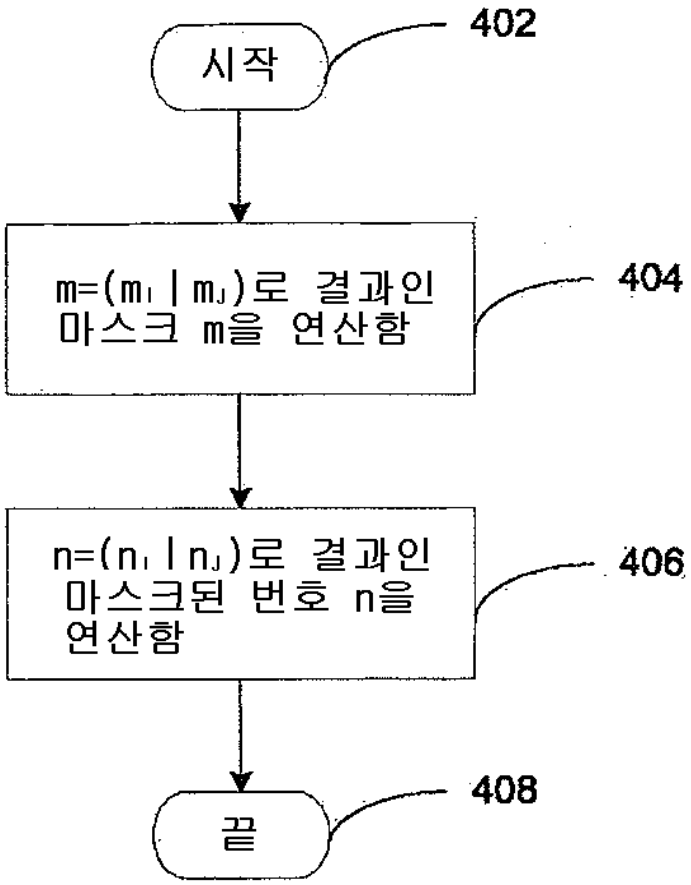
도면2



도면3


시작
302
304
단계 1: 시스템 내의 모듈 M에 대한 선택지(들)에 대해 사용자에게 질의함.
306
단계 2: 사용자 입력이 유효?
아니오
예
308
단계 3: 상기 사용자 선택의 하드웨어 수정을 기초로, M에 대한 소프트웨어 컴포넌트 X의 호환성 있는 버전의 집합을 결정함.
310
단계 4: 사용자에게 상기 연산된 X로부터 소프트웨어 컴포넌트의 하나의 버전을 선택할 것을 질의함.
312
단계 5: 단계 1에서의 하드웨어 수정의 각 선택에 대해, 모든 X에 걸쳐 HWR(MX.p) 의 교집합을 연산함. 여기서 p는 X의 선택된 버전을 나타냄.
314
단계 6: 모든 선택된 하드웨어 수정에 걸쳐, 단계 5에서 수집된 결과를 이용하여 사용자의 소프트웨어 버전 선택에 대해 호환성 있는, M에 대한 하드웨어 수정의 최종 목록을 연산함.
316
단계 7: 사용자의 프로파일에 상기 연산된 최종 목록을 저장함.
끝
318

도면4


시작
402
m=(m1 | mJ)로 결과인 마스크 m을 연산함
404
n=(n1 | nJ)로 결과인 마스크된 번호 n을 연산함
406
끝
408