



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2013-0055181
(43) 공개일자 2013년05월28일

(51) 국제특허분류(Int. Cl.)

G06F 11/36 (2006.01)

(21) 출원번호 10-2011-0120791

(22) 출원일자 2011년11월18일

심사청구일자 2011년11월18일

(71) 출원인

고려대학교 산학협력단

서울 성북구 안암동5가 1

(72) 발명자

백두권

경기도 남양주시 조안면 북한강로626번길 69

이중현

서울특별시 성북구 안암로 145, 자연계캠퍼스 아산이학관 225호 (안암동5가, 고려대학교)

정현준

서울특별시 노원구 공릉동 태강아파트 1010동 1401호

(74) 대리인

특허법인엠에이피에스

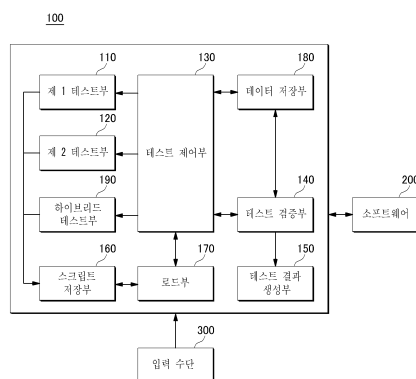
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 소프트웨어 테스트 장치 및 그 방법

(57) 요약

본 발명에 따른 소프트웨어 테스트 장치는 테스트 대상 소프트웨어에 대해 수행된 입력수단의 입력을 캡처(capture)하여 제 1 테스트 스크립트를 생성하고, 상기 생성된 제 1 테스트 스크립트를 재생하여 상기 소프트웨어에 대한 테스트를 수행하는 제 1 테스트부, 사용자에게 의하여 작성된 복수의 이벤트 정보를 포함하는 제 2 테스트 스크립트의 이벤트 정보를 기초로 상기 소프트웨어에 대한 테스트를 수행하는 제 2 테스트부, 상기 입력 수단의 입력을 캡처하여 생성한 제 1 테스트 스크립트 및 이벤트 정보를 포함하는 제 2 테스트 스크립트를 혼합한 하이브리드 테스트 스크립트에 기초하여 상기 소프트웨어에 대한 테스트를 수행하는 하이브리드 테스트부 및 사용자의 선택 또는 상기 소프트웨어에 대한 테스트 설정값에 따라 상기 제 1 테스트부, 제 2 테스트부 및 하이브리드 테스트부 중 어느 하나를 실행시켜 상기 소프트웨어에 대한 테스트를 수행하는 테스트 제어부를 포함한다.

대표도 - 도10



이 발명을 지원한 국가연구개발사업

과제고유번호	ITAB1100110100020001000200200
부처명	한국소프트웨어진흥원
연구사업명	SW공학 요소기술 개발과 전문인력 양성사업
연구과제명	융합소프트웨어 신뢰성 기술개발 연구
주관기관	고려대학교 산학협력단
연구기간	2011.01.01 ~ 2011.12.31

특허청구의 범위

청구항 1

소프트웨어 테스트 장치에 있어서,

테스트 대상 소프트웨어에 대해 수행된 입력수단의 입력을 캡처(capture)하여 제 1 테스트 스크립트를 생성하고, 상기 생성된 제 1 테스트 스크립트를 재생하여 상기 소프트웨어에 대한 테스트를 수행하는 제 1 테스트부,

사용자에 의하여 작성된 복수의 이벤트 정보를 포함하는 제 2 테스트 스크립트의 이벤트 정보를 기초로 상기 소프트웨어에 대한 테스트를 수행하는 제 2 테스트부,

상기 입력 수단의 입력을 캡처하여 생성한 제 1 테스트 스크립트 및 이벤트 정보를 포함하는 제 2 테스트 스크립트를 혼합한 하이브리드 테스트 스크립트에 기초하여 상기 소프트웨어에 대한 테스트를 수행하는 하이브리드 테스트부 및

사용자의 선택 또는 상기 소프트웨어에 대한 테스트 설정값에 따라 상기 제 1 테스트부, 제 2 테스트부 및 하이브리드 테스트부 중 어느 하나를 실행시켜 상기 소프트웨어에 대한 테스트를 수행하는 테스트 제어부

를 포함하는 소프트웨어 테스트 장치.

청구항 2

제 1 항에 있어서,

상기 하이브리드 테스트부는

상기 입력 수단의 입력을 캡처하여 제 1 테스트 스크립트를 생성하는 제 1 테스트 스크립트 생성부,

사용자에 의하여 작성된 복수의 이벤트 정보를 기초로 제 2 테스트 스크립트를 생성하는 제 2 테스트 스크립트 생성부 및

상기 제 1 테스트 스크립트 및 제 2 테스트 스크립트를 포함하는 하이브리드 테스트 스크립트를 실행하는 하이브리드 테스트 스크립트 실행부를 포함하는 소프트웨어 테스트 장치.

청구항 3

제 1 항에 있어서,

상기 생성된 제 1 테스트 스크립트, 제 2 테스트 스크립트 및 하이브리드 스크립트 중 하나 이상을 파일로 저장하는 스크립트 저장부 및

상기 테스트 제어부의 제어에 따라 상기 저장된 파일을 로드(load)하여, 상기 소프트웨어에 대한 테스트가 수행되도록 하는 로드부를 더 포함하는 소프트웨어 테스트 장치.

청구항 4

제 1 항에 있어서,

상기 테스트 제어부는

상기 테스트 설정값에 따라 상기 제 1 테스트부를 통해 상기 소프트웨어에 대한 테스트를 수행한 결과, 해당 소프트웨어의 특정 모듈이 비정상적으로 판단된 경우, 제 2 테스트부를 실행시켜 해당 모듈에 대한 보다 정밀한 테스트가 수행되도록 하는 것인 소프트웨어 테스트 장치.

청구항 5

제 1 항 내지 제 4 항 중 어느 하나의 항에 있어서,

상기 소프트웨어는 반도체 공정 제어 소프트웨어로서 FDC(Fault Detection and Classification) 및 R2R(Run to Run) 중 하나 이상을 포함하는 것인 소프트웨어 테스트 장치.

명세서

기술 분야

[0001] 본 발명은 소프트웨어 테스트 장치 및 그 방법에 관한 것이다.

배경 기술

[0002] 일반적으로 소프트웨어는 요구사항, 설계, 구현 및 테스트 과정을 통해 개발된다. 여기서, 개발의 최종 단계인 테스트에서 해당 소프트웨어의 에러가 발견된 경우, 요구 사항부터 구현에 이르는 과정의 산출물을 모두 변경해야 하므로 소프트웨어 개발 비용이 증가하게 된다.

[0003] 그리고, 소프트웨어의 일부 기능에 발생한 에러를 수정한 경우 복잡하게 서로 연동되어 있는 내부 구조를 가진 소프트웨어의 특징 때문에, 해당 수정 사항으로 인하여 또 다른 기능에 에러가 발생할 수 있다. 이를 해결하기 위한 방법의 하나로 종래에는 소프트웨어의 구현 전에 테스트 시나리오를 작성하고 테스트 시나리오를 통해 소프트웨어를 테스트하는 방법이 사용되었다.

[0004] 이러한 소프트웨어 테스트 방법은 소프트웨어를 개발하면서 테스트 시나리오에 따라 실행된 값이 미리 예측된 값과 다른 경우에 해당 소프트웨어가 정상적인 동작을 수행하고 있지 않다는 것으로 판단한다. 이 경우, 테스트 시나리오에 따라 미리 예측된 값이 나올 때까지 소프트웨어를 계속 수정하면서 테스트를 진행하게 된다.

[0005] 그러나, 테스트 계획을 수립하고 테스트 시나리오와 테스트 데이터를 만드는 작업에 많은 시간이 소요되었고, 개발 도중 사용자의 요구가 변경된 경우 테스트 계획부터 테스트 데이터까지 모두 변경되는 문제점이 있었다.

[0006] 따라서, 보다 효과적으로 소프트웨어에 대한 테스트를 수행할 필요성이 제기된다.

[0007] 이와 관련하여, 한국공개특허 제2005-0109305호(발명의 명칭: 공정장비의 상태를 모니터링하기 위한 시스템 및 방법)는 공정장비의 상태를 모니터링하기 위한 시스템 및 방법에 관한 것으로, 공정장비의 오류를 정확하게 감지할 수 있는 공정장비의 상태를 모니터링하기 위한 시스템 및 방법을 제공한다.

[0008] 또한, 한국공개특허 제 2002-0063270호(발명의 명칭: F D C 반도체 제조 프로세스로부터 트레이스 데이터리포트들을 요구하는 방법)는 오류 검출 제어를 채용하여 반도체 제조 프로세스에서 트레이스 데이터 리포트들을 동적으로 생성하는 방법을 개시한다.

발명의 내용

해결하려는 과제

[0009] 본 발명의 일 실시예는 사용자가 직접 수동으로 테스트 스크립트를 작성하기 어렵거나 GUI(Graphic User Interface)를 기초로 동작하는 소프트웨어를 테스트하거나, 소프트웨어의 특정 모듈, 복수의 모듈들이 통합되어 동작하도록 하는 인터페이스, 각 프로그램 코드의 변수 및 함수 정의 등에 대한 효과적인 테스트를 수행하는 소프트웨어 테스트 장치 및 그 방법을 제공한다.

과제의 해결 수단

[0010] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 소프트웨어 테스트 장치는, 테스트 대상 소프트웨어에 대해 수행된 입력수단의 입력을 캡처(capture)하여 제 1 테스트 스크립트를 생성하고, 상기 생성된 제 1 테스트 스크립트를 재생하여 상기 소프트웨어에 대한 테스트를 수행하는 제 1 테스트부, 사용자에게 의하여 작성된 복수의 이벤트 정보를 포함하는 제 2 테스트 스크립트의 이벤트 정보를 기초로 상기 소프트웨어에 대한

테스트를 수행하는 제 2 테스트부, 상기 입력 수단의 입력을 캡처하여 생성한 제 1 테스트 스크립트 및 이벤트 정보를 포함하는 제 2 테스트 스크립트를 혼합한 하이브리드 테스트 스크립트에 기초하여 상기 소프트웨어에 대한 테스트를 수행하는 하이브리드 테스트부 및 사용자의 선택 또는 상기 소프트웨어에 대한 테스트 설정값에 따라 상기 제 1 테스트부, 제 2 테스트부 및 하이브리드 테스트부 중 어느 하나를 실행시켜 상기 소프트웨어에 대한 테스트를 수행하는 테스트 제어부를 포함한다.

발명의 효과

[0011] 전술한 본 발명의 과제 해결 수단 중 어느 하나에 의하면, 사용자가 직접 수동으로 테스트 스크립트를 작성하기 어렵거나 GUI(Graphic User Interface)를 기초로 동작하는 소프트웨어를 테스트하거나, 소프트웨어의 특정 모듈, 복수의 모듈들이 통합되어 동작하도록 하는 인터페이스, 각 프로그램 코드의 변수 및 함수 정의 등에 대한 빠르고 정밀한 테스트를 수행할 수 있다.

[0012] 또한, 전술한 본 발명의 과제 해결 수단 중 어느 하나에 의하면, 커스터마이징이 필요한 FDC(Fault Detection and Classification) 및 R2R(Run to Run)와 같은 반도체 공정제어 소프트웨어에 대한 테스트 효율성을 높일 수 있다.

도면의 간단한 설명

- [0013] 도 1은 본 발명의 일 실시예에 따른 소프트웨어 테스트 장치의 블록도이다.
- 도 2는 본 발명의 일 실시예에 따른 제 1 테스트부의 블록도이다.
- 도 3은 본 발명의 일 실시예에 따른 제 2 테스트부의 블록도이다.
- 도 4는 본 발명의 일 실시예에 따른 제 1 테스트부를 통한 소프트웨어 테스트 방법을 예시한 도면이다.
- 도 5는 본 발명의 일 실시예에 따른 소프트웨어 테스트 전 및 테스트 후의 테스트 결과 이미지를 예시한 도면이다.
- 도 6은 본 발명의 일 실시예에 따른 제 2 테스트부를 통한 소프트웨어 테스트 방법을 예시한 도면이다.
- 도 7은 본 발명의 일 실시예에 따른 소프트웨어 테스트 방법의 순서도이다.
- 도 8은 본 발명의 일 실시예에 따른 제 1 테스트부를 통한 소프트웨어 테스트 방법의 순서도이다.
- 도 9는 본 발명의 일 실시예에 따른 제 2 테스트부를 통한 소프트웨어 테스트 방법의 순서도이다.
- 도 10은 본 발명의 다른 실시예에 따른 소프트웨어 테스트 장치의 블록도이다.
- 도 11은 본 발명의 일 실시예에 따른 하이브리드 테스트부의 블록도이다.
- 도 12는 본원 발명의 다른 실시예에 따른 하이브리드 테스트부를 통한 소프트웨어 테스트 방법의 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0014] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0015] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0016] 도 1은 본 발명의 일 실시예에 따른 소프트웨어 테스트 장치의 블록도이다.

[0017] 도 1에 도시된 바와 같이, 본 발명의 소프트웨어 테스트 장치(100)는 제 1 테스트부(110), 제 2 테스트부(120), 테스트 제어부(130), 테스트 검증부(140), 테스트 결과 생성부(150), 스크립트 저장부(160), 로드부(170) 및 데이터 저장부(180)를 포함한다.

- [0018] 본 발명의 실시예에 따른 소프트웨어는 반도체 공정 제어 소프트웨어로서 FDC(Fault Detection and Classification) 및 R2R(Run to Run) 중 하나 이상을 포함할 수 있다. FDC 공정 제어 소프트웨어는 반도체 공정에서 반도체 장비와 각각의 공정을 제어하기 위한 것으로, 웨이브릿(wavelet)을 분석하고 결함여부를 검사하여 반도체 생성 수율 및 생산성을 향상시키고 불량률을 감소시킨다.
- [0019] 그리고, R2R 공정 제어 소프트웨어는 반도체 공정 제어 프로세스를 안정화시키고 방해 요소들의 영향을 최소화하기 위한 것으로, 유동적으로 엔지니어의 요구사항에 따라 커스터마이징(customizing)될 수 있다. 이때, 본 발명의 소프트웨어 테스트 장치(100)를 통해 커스터마이징이 필요한 반도체 공정 제어 소프트웨어에 대한 테스트 효율성을 높일 수 있다.
- [0020] 본 발명의 소프트웨어 테스트 장치(100)는 상술한 반도체 공정 제어 소프트웨어에 한정되지 않고, 휴대폰, 컴퓨터, 영상 기기, 가전 기기 등 각종 산업 분야 및 개인 휴대용 기기 등에서 동작하는 소프트웨어를 테스트하거나 구현 과정에 있는 소프트웨어를 테스트하는 데에 이용될 수 있다. 이하, 소프트웨어 테스트 장치(100)의 각 구성요소에 대해서 구체적으로 설명하기로 한다.
- [0021] 제 1 테스트부(110)는 테스트 대상 소프트웨어에 대해 수행된 입력수단의 입력을 캡처(capture)(또는 기록)하여 제 1 테스트 스크립트를 생성하고, 생성된 제 1 테스트 스크립트를 재생(playback or replay)하여 소프트웨어에 대한 테스트를 수행한다. 예컨대 제 1 테스트부(110)는 사용자가 직접 수동으로 테스트 스크립트를 작성하기 어렵거나 GUI(Graphic User Interface)를 기초로 동작하는 소프트웨어를 테스트하는 데에 사용될 수 있다.
- [0022] 상술한 입력수단은 마우스, 터치패드 및 키보드 중 하나 이상을 포함할 수 있다. 그리고, 사용자가 소프트웨어에 대해 입력수단을 통한 입력을 수행하도록 인터페이스가 제공될 수 있다. 제 1 테스트부(110)의 자세한 구성에 대해서는 도 2에서 설명하기로 한다.
- [0023] 제 2 테스트부(120)는 복수의 이벤트 정보를 포함하는 제 2 테스트 스크립트를 생성하고, 생성된 제 2 테스트 스크립트의 이벤트 정보를 기초로 소프트웨어에 대한 테스트를 수행한다. 여기서, 사용자가 이벤트 정보를 입력하거나 수정할 수 있도록 인터페이스가 제공될 수 있다.
- [0024] 이와 같은 제 2 테스트부(120)는 소프트웨어의 특정 모듈, 복수의 모듈들이 통합되어 동작하도록 하는 인터페이스, 각 프로그램 코드의 변수 및 함수 정의 등에 대한 빠르고 정밀한 테스트를 요구할 경우 사용될 수 있다. 제 2 테스트부(120)의 구성에 대한 보다 구체적인 설명은 도 3에서 설명하기로 한다.
- [0025] 테스트 제어부(130)는 사용자의 선택 또는 소프트웨어에 대한 테스트 설정값에 따라 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나를 실행시키고 소프트웨어에 대한 테스트를 제어한다. 여기서, 테스트 제어부(130)는 사용자가 선택한 사항에 따라 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나를 실행시킬 수 있다. 또한, 테스트 제어부(130)는 테스트 설정값에 따라 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나를 실행시킬 수도 있다.
- [0026] 이때, 테스트 제어부(130)는 테스트 설정값에 따라 제 1 테스트부(110) 및 제 2 테스트부(120)를 하나씩 순차적으로 실행시킬 수 있다. 예컨대, 테스트 제어부(130)는 제 1 테스트부(110)에 대한 테스트가 완료된 이후 검증 결과에 따라 이어서 제 2 테스트부(120)를 실행시킬 수 있다.
- [0027] 보다 구체적으로, 테스트 제어부(130)는 테스트 설정값에 따라 제 1 테스트부(110)를 통해 소프트웨어에 대한 테스트를 수행한 결과, 하기 테스트 검증부(140)를 통해 해당 소프트웨어의 특정 모듈이 비정상적으로 판단된 경우, 제 2 테스트부(120)를 실행시켜 해당 모듈에 대한 보다 정밀한 테스트가 수행되도록 할 수 있다.
- [0028] 이때, 테스트 제어부(130)는 에러가 발생한 소프트웨어의 특정 모듈에 대한 테스트를 수행하기 위해 필요한 이벤트 정보를 체크 리스트 형태로 사용자에게 제공하거나, 자동으로 제 2 테스트부(120)로 전달하여 제 2 테스트 스크립트가 생성되도록 할 수 있다.
- [0029] 또한, 테스트 제어부(130)는 테스트 설정값에 따라 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나를 정해진 테스트 속도에 따라 일정 횟수 반복 실행시켜 소프트웨어에 대한 테스트를 수행하도록 하여, 소프트웨어 테스트의 신뢰성을 높일 수 있다. 이와 같이, 각각 특성을 가지는 제 1 테스트부(110) 및 제 2 테스트부(120)를 사용자의 선택 또는 테스트 설정값에 따라 실행시킴으로써, 보다 정밀하고 신뢰성 있는 소프트웨어에 대한 자동화된 테스트를 수행할 수 있다.
- [0030] 테스트 검증부(140)는 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나의 실행에 따라 테스트가 수행된 소프트웨어의 테스트 결과 데이터를 기초로, 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다. 여기

서, 테스트 검증부(140)는 하기 도 2의 이미지 검증부(118) 및 도 3의 이벤트 검증부(126)를 포함할 수 있다. 또한, 테스트 검증부(140)는 이미지 검증부(118) 및 이벤트 검증부(126)로 각각 구현되어 제 1 테스트부(110) 및 제 2 테스트부(120)에 포함될 수 있다.

- [0031] 테스트 결과 생성부(150)는 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나의 실행에 따라 테스트가 수행된 소프트웨어의 테스트 결과 데이터 및 이를 기초로 한 소프트웨어의 검증 결과를 평가 보고서로 생성한다. 예컨대 테스트 결과 생성부(150)는 XML(extensible markup language), HTML(HyperText Markup Language) 등의 형태로 평가 보고서를 작성할 수 있다. 여기서, 테스트 일시, 테스트 소요 시간, 테스트를 패스한 객체, 응답시간, 처리량, 처리속도, 활용도 등에 대한 정보가 평가 보고서에 포함될 수 있다.
- [0032] 스크립트 저장부(160)는 제 1 테스트 스크립트 및 제 2 테스트 스크립트 중 하나 이상을 파일로 저장한다. 여기서, 제 1 테스트 스크립트 및 제 2 테스트 스크립트는 마크업 언어(markup language)로 작성될 수 있다. 예컨대, 제 1 테스트 스크립트 및 제 2 테스트 스크립트 중 하나 이상은 마크업 언어 중 XML(extensible markup language) 언어로 작성될 수 있다.
- [0033] 로드부(170)는 테스트 제어부(130)의 제어에 따라 제 1 테스트 스크립트 및 제 2 테스트 스크립트의 저장 파일을 로드(load)하여, 소프트웨어에 대한 테스트가 수행되도록 한다. 예컨대, 제 1 테스트부(110)가 실행될 경우 기 작성된 제 1 테스트 스크립트가 로드될 수 있다. 이때, 기 작성된 테스트 스크립트를 재 사용하여 소프트웨어를 테스트할 수 있어, 테스트 스크립트 작성으로 인한 시간 비용을 줄일 수 있다.
- [0034] 데이터 저장부(180)는 소프트웨어 테스트 수행 및 검증을 위한 알고리즘, 설정값 및 기타 각종 정보를 저장한다. 예컨대 테스트 설정값, 이벤트 정보, 테스트 데이터, 테스트 속도, 테스트 반복 횟수, 테스트 결과 데이터, 검증 결과 데이터 등에 대한 정보가 저장될 수 있다.
- [0035] 도 2는 본 발명의 일 실시예에 따른 제 1 테스트부의 블록도이다.
- [0036] 도 2에 도시된 바와 같이, 제 1 테스트부(110)는 제 1 테스트 스크립트 생성부(112), 제 1 테스트 스크립트 실행부(114), 이미지 생성부(116) 및 이미지 검증부(118)를 포함한다.
- [0037] 제 1 테스트 스크립트 생성부(112)는 소프트웨어에 대해 수행된 입력수단의 입력을 캡처하여 제 1 테스트 스크립트로 생성한다. 여기서, 마우스, 키보드 및 터치 패드 등을 통한 입력이 캡처되어 제 1 테스트 스크립트로 생성될 수 있다. 예컨대 사용자가 소프트웨어의 GUI 컴포넌트(component)(또는 컨트롤)에 대해 마우스 또는 키보드를 통한 입력(클릭, 드래그, 등)을 수행하였을 경우, 이에 대한 캡처가 수행되어 해당 캡처된 값을 기초로 한 제 1 테스트 스크립트가 생성될 수 있다.
- [0038] 이때, 사용자가 선택한 GUI 컴포넌트의 속성값에 따라 테스트 데이터가 함께 생성될 수 있다. 예컨대, GUI 컴포넌트가 반도체 공정 제어 과정에서 기기의 회전속도 및 각도를 조절하는 모듈인 경우, 회전속도 및 각도 제어와 관련된 테스트 데이터가 해당 모듈에 대응되도록 자동 생성될 수 있다.
- [0039] 제 1 테스트 스크립트 실행부(114)는 제 1 테스트 스크립트를 재생(playback or replay)시켜 소프트웨어에 대한 테스트를 수행한다. 이때, 설정값에 따라 반복해서 테스트를 수행할 수 있다.
- [0040] 이미지 생성부(116)는 입력수단의 입력을 캡처하기 전의 소프트웨어에 대한 테스트 결과 이미지와 입력수단의 입력을 캡처할 때의 소프트웨어에 대한 테스트 결과 이미지를 비교한 차이점을 기초로 제 1 이미지를 생성한다. 또한, 이미지 생성부(116)는 제 1 테스트 스크립트를 재생하기 전의 소프트웨어에 대한 테스트 결과 이미지와 제 1 테스트 스크립트를 재생한 후의 소프트웨어에 대한 테스트 결과 이미지를 비교한 차이점을 기초로 제 2 이미지를 생성한다.
- [0041] 여기서, 이미지 생성부(116)는 사용자가 캡처한 화면을 통해 제 1 이미지 및 제 2 이미지를 생성하거나, 자동으로 해당 소프트웨어에 대한 테스트 결과 데이터를 이미지 처리하여 이미지 파일로 생성할 수 있다. 예컨대, 그래프 형태로 테스트 결과 데이터를 출력하는 반도체 공정제어를 수행하는 소프트웨어의 경우에는 테스트 결과 데이터로서 그래프 이미지가 생성될 수 있다.
- [0042] 이미지 검증부(118)는 제 1 이미지와 제 2 이미지의 차이점을 비교하여 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다. 예컨대, 상술한 반도체 공정제어를 수행하는 소프트웨어의 경우, 테스트 결과 데이터로서 그래프 이미지가 생성되므로, 이미지 검증부(118)는 입력수단의 입력을 캡처할 때의 제 1 이미지와 제 1 테스트 스크립트의 재생에 따라 생성된 제 2 이미지의 차이점을 비교하여 소프트웨어의 정상 여부를 판단하기 위한 검

증을 수행한다.

- [0043] 만약, 제 1 이미지와 제 2 이미지의 차이점이 설정된 기준치를 초과하지 않을 경우, 이미지 검증부(118)는 해당 소프트웨어를 정상으로 판단하고, 그렇지 않을 경우에는 비정상으로 판단할 수 있다. 판단한 결과 데이터는 테스트 결과 생성부(150)로 전달되어 평가 보고서 작성에 이용될 수 있다.
- [0044] 도 3은 본 발명의 일 실시예에 따른 제 2 테스트부의 블록도이다.
- [0045] 도 3에 도시된 바와 같이, 제 2 테스트부(120)는 제 2 테스트 스크립트 생성부(122), 제 2 테스트 스크립트 실행부(124) 및 이벤트 검증부(126)를 포함한다.
- [0046] 제 2 테스트 스크립트 생성부(122)는 사용자에게 의하여 작성된 복수의 이벤트 정보를 포함하는 제 2 테스트 스크립트를 생성한다. 예컨대, 이벤트 정보는 소프트웨어의 특정 모듈을 테스트하기 위한 정보를 포함할 수 있다. 여기서, 사용자는 인터페이스를 통해 이벤트 정보를 직접 입력하거나, 테스트하고자 하는 이벤트 정보를 선택하여 복수의 이벤트 정보를 작성할 수 있다.
- [0047] 또한, 기본적인 이벤트 정보가 미리 설정되어 있는 경우, 사용자는 해당 인터페이스를 통해 이벤트 정보를 추가하거나 삭제할 수 있다. 이후, 제 2 테스트 스크립트 생성부(122)는 사용자에게 의해 입력 또는 선택된 이벤트 정보를 파싱(parsing)하여 제 2 테스트 스크립트로 생성할 수 있다.
- [0048] 제 2 테스트 스크립트 실행부(124)는 제 2 테스트 스크립트를 기초로 소프트웨어에 대한 테스트를 수행한다. 이때, 제 2 테스트 스크립트의 이벤트 정보에 따라 소프트웨어에 대한 테스트가 수행될 때, 해당 이벤트 정보에 대응되는 테스트 데이터가 함께 제공되어 특정 테스트 데이터에 대한 테스트가 이루어질 수 있다. 이러한 테스트 데이터는 사용자가 직접 입력하거나, 인터페이스를 통해 선택된 이벤트 정보에 따라 자동 생성될 수 있다.
- [0049] 이벤트 검증부(126)는 제 2 테스트 스크립트에 포함된 이벤트 정보에 따라 테스트가 수행된 소프트웨어로부터 테스트 결과 데이터를 수신하고, 테스트 결과 데이터를 해당 이벤트 정보를 기초로 생성된 예상 결과 데이터와 비교하여 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다.
- [0050] 도 4는 본 발명의 일 실시예에 따른 제 1 테스트부를 통한 소프트웨어 테스트 방법을 예시한 도면이다.
- [0051] 도 4에 도시된 바와 같이, 사용자는 인터페이스(400)를 통해 제 1 테스트부(110)를 통한 소프트웨어 테스트를 수행할 수 있다. 이때, 사용자는 테스트 수행을 위한 각각의 값을 설정할 수 있으며, 기본적인 값들이 미리 설정되어 있는 경우 이를 수정하여 소프트웨어 테스트를 진행할 수 있다. 이하 보다 구체적으로 설명하기로 한다.
- [0052] 먼저, 사용자는 캡처(capture) 버튼(412)을 클릭하여 테스트 대상 소프트웨어에 대해 수행된 입력수단의 입력을 캡처할 수 있다. 이때, 마우스를 통한 입력이 캡처되어 마우스 캡처 영역(402)에 표시되고, 키보드를 통한 입력이 캡처되어 키보드 캡처 영역(404)에 표시된다.
- [0053] 다음으로, 사용자는 스크립트 저장 버튼(413)을 클릭하여 입력수단의 입력을 캡처한 값을 기초로 제 1 테스트 스크립트를 생성 및 저장한다. 이때, 스크립트 로드 버튼(414)을 클릭하여 기 저장된 제 1 테스트 스크립트를 로드할 수 있다. 여기서, 해당 소프트웨어를 빌드하고, 입력수단의 입력을 통한 캡처가 수행되었을 때 제 1 테스트 스크립트가 자동으로 생성될 수 있으며, 기 저장된 제 1 테스트 스크립트가 있을 경우 이를 자동으로 체크하여 해당 기 저장된 제 1 테스트 스크립트를 로드할 수도 있다. 그리고, 생성된 제 1 테스트 스크립트와 관련된 내용이 스크립트 표시 영역(426)에 표시된다.
- [0054] 다음으로, 해당 소프트웨어에 대한 테스트 반복 횟수를 테스트 반복 횟수 설정 버튼(409)을 클릭하여 설정하고, 테스트 속도를 테스트 속도 설정 버튼(410)을 클릭하여 설정한다. 다음으로, 재생 버튼(416)을 클릭하면, 제 1 테스트 스크립트를 재생하여 소프트웨어에 대한 테스트를 수행한다.
- [0055] 그리고, 사용자는 캡처 버튼(418)을 클릭하여 소프트웨어에 대한 테스트 결과 이미지를 생성한다. 여기서, 소프트웨어에 대한 테스트가 완료되면 자동으로 테스트 결과 이미지가 생성될 수 있다.
- [0056] 도 5는 본 발명의 일 실시예에 따른 소프트웨어 테스트 전 및 테스트 후의 테스트 결과 이미지를 예시한 도면이다.
- [0057] 이어서 설명하면, 도 5의 (a)는 입력수단의 입력을 캡처하기 전의 해당 소프트웨어에 대한 테스트 결과 이미지를 나타낸다. 이때, 입력수단의 입력을 캡처하기 전이므로, 테스트 결과 이미지가 나타나지 않게 된다. 그리

고, 도 5의 (b)는 입력수단의 입력을 캡처한 후의 소프트웨어에 대한 테스트 결과 이미지를 나타낸다.

- [0058] 이때, 도 5의 (a) 및 (b)의 이미지는 테스트 결과 데이터에 대해 캡처 버튼(418)을 클릭하였을 때에 화면 캡처를 통해 생성되거나, 테스트가 완료되었을 때에 자동으로 생성될 수 있다. 그리고, 도 5의 (c)는 (a)의 그래프 이미지와 (b)의 그래프 이미지를 비교한 차이점을 기초로 생성된 제 1 이미지를 나타낸다.
- [0059] 이와 같이, 입력수단의 입력을 캡처하기 전의 소프트웨어에 대한 테스트 결과 이미지와 입력수단의 입력을 캡처한 후의 소프트웨어에 대한 테스트 결과 이미지를 비교한 차이점을 기초로 제 1 이미지를 생성할 수 있다.
- [0060] 또한, 동일한 방법으로 제 1 테스트 스크립트를 재생하기 전의 소프트웨어에 대한 테스트 결과 이미지와 제 1 테스트 스크립트를 재생한 후의 소프트웨어에 대한 테스트 결과 이미지를 비교한 차이점을 기초로 제 2 이미지를 생성할 수 있다.
- [0061] 다시 도 4로 돌아가서 설명하면, 사용자는 이미지 로드 버튼(422)을 클릭하여 소프트웨어의 테스트 전/후로 생성된 테스트 결과 이미지(도 5의 (a) 및 (b)의 이미지 참조)를 각각 로드한다. 그리고, 비교 버튼(424)을 클릭하면 로드된 이미지간의 차이점 비교가 이루어지고, 이를 통해 제 1 이미지(도 5의 (c) 참조) 또는 제 2 이미지가 생성된다. 여기서, 각각의 로드된 이미지에 대한 명칭이 이미지 생성 영역(406)에 표시될 수 있다.
- [0062] 그리고, 이미지 로드 버튼(422)과 비교 버튼(424)을 통해 최종적으로 제 1 이미지와 제 2 이미지간에 비교가 이루어지면, 해당 비교 결과 값을 기초로 소프트웨어의 검증 결과가 테스트 결과 영역(408)에 표시될 수 있다. 구체적으로 테스트 결과 영역(408)에는 테스트 단계, 테스트 이미지, 테스트 결과, 예러로 예상되는 픽셀 등에 대한 정보가 포함될 수 있다.
- [0063] 상술한 과정에서 화면을 여러 번 캡처하여 테스트하고자 하는 내용이 변경된 부분만을 추출함으로써, 캡처를 통해 미리 생성해 놓은 이미지와 테스트 스크립트를 재생하여 생성한 이미지가 특정 부분이(GUI) 변경되었기 때문에 발생하는 차이로 인해 검증 결과가 오류로 나타나는 것을 방지할 수 있다.
- [0064] 도 6a 및 도 6b는 본 발명의 일 실시예에 따른 제 2 테스트부를 통한 소프트웨어 테스트 방법을 예시한 도면이다.
- [0065] 도 6a에 도시된 바와 같이, 사용자가 복수의 이벤트 정보를 입력하여 제 2 테스트 스크립트(502)를 생성할 수 있다. 여기서, 사용자는 인터페이스를 통해 소프트웨어를 테스트하기 위한 이벤트 정보를 입력하거나 기 입력된 이벤트 정보를 수정할 수 있다. 이때, 이벤트 정보 중 필요한 이벤트 정보를 선택하는 체크 리스트 형태로 인터페이스가 제공될 수 있다. 이후, 제 2 테스트 스크립트(502)에 포함된 테스트 시나리오의 각 단계(step)에 따라 이벤트 정보가 소프트웨어에 전달되어 테스트가 진행된다.
- [0066] 도 6b에 도시된 바와 같이, 제 2 테스트 스크립트(502)에 포함된 이벤트 정보는 순차적으로 테스트 시나리오(504)에 따라 소프트웨어에 전달되어 해당 소프트웨어의 특정 모듈의 기능에 대한 정상 여부를 테스트할 수 있다. 그리고, 제 2 테스트 스크립트(502)의 이벤트 정보에 따라 소프트웨어에 대한 테스트가 완료되면, 상술한 테스트 결과 영역(408)에 테스트 결과 데이터가 표시된다.
- [0067] 도 7은 본 발명의 일 실시예에 따른 소프트웨어 테스트 방법의 순서도이다.
- [0068] 도 7에 도시된 바와 같이, 먼저 사용자의 선택 또는 테스트 대상 소프트웨어에 대한 테스트 설정값에 따라 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나를 실행시킨다(S701). 예컨대, 사용자는 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나를 선택할 수 있는 인터페이스를 통해 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나를 선택하여 실행시킬 수 있다. 이때, 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나에 대한 테스트 설정값이 설정되어 있는 경우, 그에 따라 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나가 실행될 수 있다.
- [0069] 다음으로, 제 1 테스트부(110) 및 제 2 테스트부(120) 중 어느 하나의 실행에 따라 소프트웨어에 대한 테스트를 수행한다(S711). 여기서, 제 1 테스트부(110)의 실행에 따라 소프트웨어에 대한 테스트를 수행하는 과정은 도 8에서 구체적으로 설명하기로 한다. 그리고, 제 2 테스트부(120)의 실행에 따라 소프트웨어에 대한 테스트를 수행하는 과정은 도 9에서 구체적으로 설명하기로 한다.
- [0070] 다음으로, 테스트가 수행된 소프트웨어의 테스트 결과 데이터를 기초로, 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다(S721). 이때, 평가 보고서가 작성되어 검증 결과를 사용자에게 알릴 수 있다.
- [0071] 도 8은 본 발명의 일 실시예에 따른 제 1 테스트부를 통한 소프트웨어 테스트 방법의 순서도이다.

- [0072] 도 8에 도시된 바와 같이, 먼저 테스트 대상 소프트웨어에 대해 수행된 입력수단의 입력을 캡처한다(S801).
- [0073] 다음으로, 입력수단의 입력을 캡처하기 전의 소프트웨어에 대한 테스트 결과 이미지와 입력수단의 입력을 캡처한 후의 소프트웨어에 대한 테스트 결과 이미지를 비교한 차이점을 기초로 제 1 이미지를 생성한다(S811). 본 단계(S811)는 소프트웨어의 정상 여부를 판단하기 위한 것으로, 단순히 소프트웨어에 대한 테스트를 수행할 경우에는 생략될 수 있다. 그리고 제 1 이미지는 사용자가 화면 캡처 버튼을 클릭하는 과정을 통해 생성되거나, 사용자에게 의한 입력수단의 입력을 캡처한 후에 완료 버튼 동작과 함께 생성될 수 있다.
- [0074] 다음으로, 입력수단의 입력을 캡처한 값을 기초로 제 1 테스트 스크립트를 생성한다(S821). 여기서, 기 저장된 제 1 테스트 스크립트가 있으면, 해당 제 1 테스트 스크립트를 로드할 수 있다.
- [0075] 다음으로, 제 1 테스트 스크립트를 재생하여 소프트웨어에 대한 테스트를 수행한다(S831).
- [0076] 다음으로, 제 1 테스트 스크립트를 재생하기 전의 소프트웨어에 대한 테스트 결과 이미지와 제 1 테스트 스크립트를 재생한 후의 소프트웨어에 대한 테스트 결과 이미지를 비교한 차이점을 기초로 제 2 이미지를 생성한다(S841).
- [0077] 다음으로, 제 1 이미지와 제 2 이미지의 차이점을 비교하여, 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다(S851). 이때, 제 1 이미지와 제 2 이미지의 차이가 기준치 범위안에 있을 경우에는 해당 소프트웨어의 테스트 결과를 정상으로 판단한다.
- [0078] 도 9는 본 발명의 일 실시예에 따른 제 2 테스트부를 통한 소프트웨어 테스트 방법의 순서도이다.
- [0079] 도 9에 도시된 바와 같이, 먼저 사용자에게 의해 입력된 복수의 이벤트 정보를 포함하는 제 2 테스트 스크립트를 생성한다(S901). 여기서, 기 저장된 제 2 테스트 스크립트가 있으면, 해당 제 2 테스트 스크립트를 로드할 수 있다.
- [0080] 다음으로, 제 2 테스트 스크립트의 이벤트 정보를 기초로 소프트웨어에 대한 테스트를 수행한다(S911).
- [0081] 다음으로, 제 2 테스트 스크립트에 포함된 이벤트 정보에 따라 테스트가 수행된 소프트웨어로부터 테스트 결과 데이터를 수신한다(S921).
- [0082] 다음으로, 테스트 결과 데이터를 이벤트 정보에 따라 기 생성된 예상 결과 데이터와 비교하여 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다(S931). 여기서, 테스트 결과 데이터와 기 생성된 예상 결과 데이터를 비교한 결과 값이 기준치 범위안에 있을 경우에는 해당 소프트웨어의 테스트 결과를 정상으로 판단한다.
- [0083] 한편, 도 1 내지 도 3에서 도시된 각각의 구성요소는 일종의 '모듈'로 구성될 수 있다. 상기 '모듈'은 소프트웨어 또는 Field Programmable Gate Array(FPGA) 또는 주문형 반도체(ASIC, Application Specific Integrated Circuit)와 같은 하드웨어 구성요소를 의미하며, 모듈은 어떤 역할들을 수행한다. 그렇지만 모듈은 소프트웨어 또는 하드웨어에 한정되는 의미는 아니다. 모듈은 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 실행시키도록 구성될 수도 있다. 구성요소들과 모듈들에서 제공되는 기능은 더 작은 수의 구성요소들 및 모듈들로 결합되거나 추가적인 구성요소들과 모듈들로 더 분리될 수 있다.
- [0084] 상술된 스크립트 저장부(160) 및 데이터 저장부(180)는 캐쉬, ROM(Read Only Memory), PROM(Programmable ROM), EPROM(Erasable Programmable ROM), EEPROM(Electrically Erasable Programmable ROM) 및 플래쉬 메모리(Flash memory)와 같은 비휘발성 메모리 소자 또는 RAM(Random Access Memory)과 같은 휘발성 메모리 소자 또는 하드디스크 드라이브(HDD, Hard Disk Drive), CD-ROM과 같은 저장 매체 중 적어도 하나로 구현될 수 있으나 이에 한정되지는 않는다.
- [0085] 도 10은 본 발명의 다른 실시예에 따른 소프트웨어 테스트 장치의 블록도이다.
- [0086] 전체적인 구성은 도 1의 테스트 장치와 거의 동일하며, 하이브리드 테스트부(190)가 추가되었으며, 테스트 제어부(130)의 구성이 다소 상이하다.
- [0087] 먼저 테스트 제어부(130)는 사용자의 선택, 소프트웨어에 대한 테스트 설정값 등에 따라 제 1 테스트부(110), 제 2 테스트부(120) 및 하이브리드 테스트부(190) 중 어느 하나를 실행시키고 소프트웨어에 대한 테스트를 제어한다.
- [0088] 이벤트 기반으로 테스트를 수행하는 제 2 테스트부(120)를 기초로 테스트를 수행할 경우 이벤트로 표현하지 못하는 기능(버튼이 인식되지 않는 경우, 사용자 지정 패턴 등)이 테스트 중간에 필요한 경우에는 제 2 테스트부

(120)만을 이용하기 어려운 문제가 있다. 이를 해결하기 위해서는 캡처 및 재생을 기반으로 테스트를 수행하는 제 1 테스트부(110)의 테스트 방법을 기초로 테스트를 수행할 필요가 있다.

[0089] 이에, 테스트 제어부(130)는 테스트의 내용을 기초로 하이브리드 테스트가 필요한 경우에는 하이브리드 테스트부(190)를 통해 테스트가 수행되도록 한다.

[0090] 하이브리드 테스트부(190)는 캡처 및 재생을 기반으로 하는 제 1 테스트와 이벤트 기반으로 테스트를 수행하는 제 2 테스트를 혼합한 하이브리드 테스트 스크립트를 기초로 테스트를 수행하도록 한다.

[0091] 도 11은 본 발명의 일 실시예에 따른 하이브리드 테스트부의 블록도이다.

[0092] 하이브리드 테스트부(190)는 제 1 테스트 스크립트 생성부(192), 제 2 테스트 스크립트 생성부(194), 하이브리드 테스트 스크립트 실행부(196) 및 검증부(198)를 포함한다.

[0093] 제 1 테스트 스크립트 생성부(192)는 소프트웨어에 대해 수행된 입력수단의 입력을 캡처하여 제 1 테스트 스크립트로 생성한다. 여기서, 마우스, 키보드 및 터치 패드 등을 통한 입력이 캡처되어 제 1 테스트 스크립트로 생성될 수 있다. 예컨대 사용자가 소프트웨어의 GUI 컴포넌트(component)(또는 컨트롤)에 대해 마우스 또는 키보드를 통한 입력(클릭, 드래그, 등)을 수행하였을 경우, 이에 대한 캡처가 수행되어 해당 캡처된 값을 기초로 한 제 1 테스트 스크립트가 생성될 수 있다.

[0094] 이때, 사용자가 선택한 GUI 컴포넌트의 속성값에 따라 테스트 데이터가 함께 생성될 수 있다. 예컨대, GUI 컴포넌트가 반도체 공정제어 과정에서 기기의 회전속도 및 각도를 조절하는 모듈인 경우, 회전속도 및 각도 제어와 관련된 테스트 데이터가 해당 모듈에 대응되도록 자동 생성될 수 있다.

[0095] 제 2 테스트 스크립트 생성부(194)는 사용자에게 의하여 작성된 복수의 이벤트 정보를 포함하는 제 2 테스트 스크립트를 생성한다. 예컨대, 이벤트 정보는 소프트웨어의 특정 모듈을 테스트하기 위한 정보를 포함할 수 있다. 여기서, 사용자는 인터페이스를 통해 이벤트 정보를 직접 입력하거나, 테스트하고자 하는 이벤트 정보를 선택하여 복수의 이벤트 정보를 작성할 수 있다.

[0096] 또한, 기본적인 이벤트 정보가 미리 설정되어 있는 경우, 사용자는 해당 인터페이스를 통해 이벤트 정보를 추가하거나 삭제할 수 있다. 이후, 제 2 테스트 스크립트 생성부(194)는 사용자에게 의해 입력 또는 선택된 이벤트 정보를 파싱(parsing)하여 제 2 테스트 스크립트로 생성할 수 있다.

[0097] 하이브리드 테스트 스크립트 실행부(196)는 제 1 테스트 스크립트 생성부(192)에서 생성된 제 1 테스트 스크립트 및 제 2 테스트 스크립트 생성부(194)에서 생성된 제 2 테스트 스크립트를 결합하여 하이브리드 모드로 실행한다. 이때, 하이브리드 테스트 스크립트 실행부(196)로 전달되는 테스트 스크립트는 제 1 테스트 스크립트 및 제 2 테스트 스크립트가 혼합된 것으로서, 이와 같은 구성에 따라, 한번의 테스트 과정에 제 1 테스트와 제 2 테스트를 혼합하여 실행할 수 있다.

[0098] 하이브리드 테스트 스크립트 실행부(196)는 제 1 테스트 스크립트가 있는 경우 제 1 테스트 스크립트를 재생(playback or replay)시켜 소프트웨어에 대한 테스트를 수행한다. 이때, 설정값에 따라 반복해서 테스트를 수행할 수 있다. 또한, 제 2 테스트 스크립트가 있는 경우 제 2 테스트 스크립트를 기초로 소프트웨어에 대한 테스트를 수행한다. 이때, 제 2 테스트 스크립트의 이벤트 정보에 따라 소프트웨어에 대한 테스트가 수행될 때, 해당 이벤트 정보에 대응되는 테스트 데이터가 함께 제공되어 특정 테스트 데이터에 대한 테스트가 이루어질 수 있다.

[0099] 검증부(198)는 테스트 스크립트 실행 결과에 기초하여 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다. 제 1 테스트 스크립트에 대해서는 앞서 도 2를 통해 설명한 이미지 검증부(118)와 마찬가지로, 제 1 이미지와 제 2 이미지의 차이점을 비교하여 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다. 예컨대, 상술한 반도체 공정제어를 수행하는 소프트웨어의 경우, 테스트 결과 데이터로서 그래프 이미지가 생성되므로, 검증부(198)는 입력수단의 입력을 캡처할 때의 제 1 이미지와 제 1 테스트 스크립트의 재생에 따라 생성된 제 2 이미지의 차이점을 비교하여 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다.

[0100] 만약, 제 1 이미지와 제 2 이미지의 차이점이 설정된 기준치를 초과하지 않을 경우, 검증부(198)는 해당 소프트웨어를 정상으로 판단하고, 그렇지 않을 경우에는 비정상으로 판단할 수 있다.

[0101] 또한, 제 2 테스트 스크립트에 대해서는 이벤트 검증부(128)와 마찬가지로, 제 2 테스트 스크립트에 포함된 이벤트 정보에 따라 테스트가 수행된 소프트웨어로부터 테스트 결과 데이터를 수신하고, 테스트 결과 데이터를 해

당 이벤트 정보를 기초로 기 생성된 예상 결과 데이터와 비교하여 소프트웨어의 정상 여부를 판단하기 위한 검증을 수행한다.

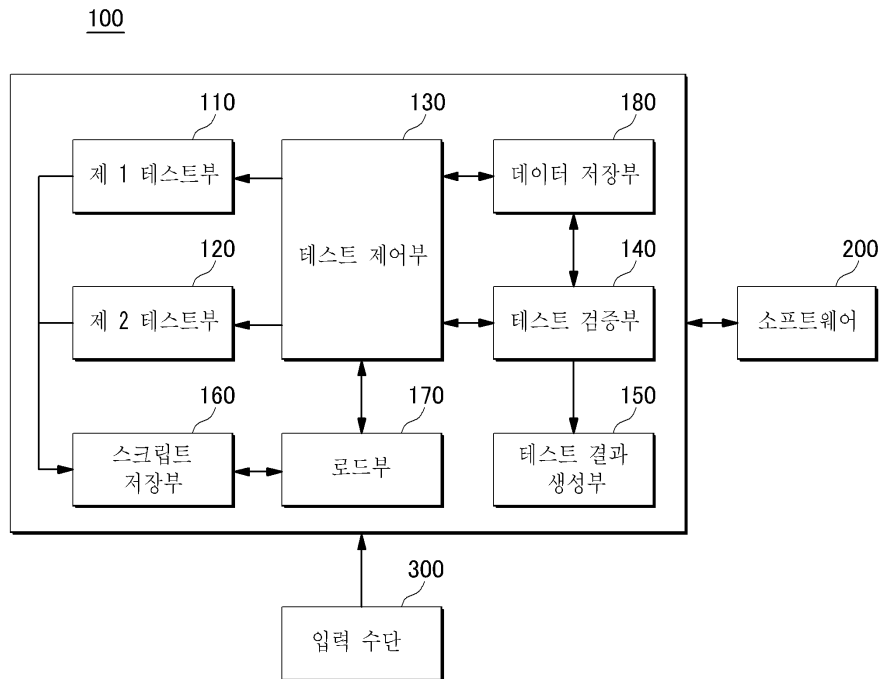
- [0102] 도 12는 본원 발명의 다른 실시예에 따른 하이브리드 테스트부를 통한 소프트웨어 테스트 방법의 순서도이다.
- [0103] 먼저, 사용자에게 의하여 설정된 테스트 조건을 분석하고(S1210), 하이브리드 테스트가 필요한지 여부에 대한 판단을 수행한다(S1220). 예를 들어, 테스트 제어부(120)를 통해 테스트 조건을 분석하여, 이벤트 기반의 테스트로 수행하지 못하는 기능(버튼이 인식되지 않는 경우, 사용자 지정 패턴 등)이 테스트 중간에 필요한 경우에는 하이브리드 테스트를 수행하도록 한다.
- [0104] 판단 결과, 하이브리드 테스트가 불필요한 경우에는 앞서 설명한 내용에 따라 제 1 테스트 또는 제 2 테스트를 선택적으로 실행한다(S1240).
- [0105] 그러나, 하이브리드 테스트가 필요한 것으로 판단된 경우에는 하이브리드 테스트부(190)를 통해, 제 1 테스트 및 제 2 테스트를 혼합시킨 하이브리드 테스트를 실행하도록 한다(S1230).
- [0106] 본 발명의 일 실시예는 컴퓨터에 의해 실행되는 프로그램 모듈과 같은 컴퓨터에 의해 실행 가능한 명령어를 포함하는 기록 매체의 형태로도 구현될 수 있다. 컴퓨터 판독 가능 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 가용 매체일 수 있고, 휘발성 및 비휘발성 매체, 분리형 및 비분리형 매체를 모두 포함한다. 또한, 컴퓨터 판독가능 매체는 컴퓨터 저장 매체 및 통신 매체를 모두 포함할 수 있다. 컴퓨터 저장 매체는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함한다. 통신 매체는 전형적으로 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈, 또는 반송파와 같은 변조된 데이터 신호의 기타 데이터, 또는 기타 전송 메커니즘을 포함하며, 임의의 정보 전달 매체를 포함한다.
- [0107] 본 발명의 방법 및 시스템은 특정 실시예와 관련하여 설명되었지만, 그것들의 구성 요소 또는 동작의 일부 또는 전부는 범용 하드웨어 아키텍처를 갖는 컴퓨터 시스템을 사용하여 구현될 수 있다.
- [0108] 진술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.
- [0109] 본 발명의 범위는 상기 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

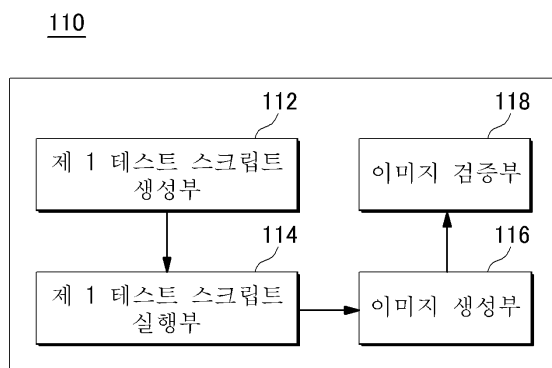
- | | | |
|--------|-----------------|---------------|
| [0110] | 110: 제 1 테스트부 | 120: 제 2 테스트부 |
| | 130: 테스트 제어부 | 140: 테스트 검증부 |
| | 150: 테스트 결과 생성부 | 160: 스크립트 생성부 |
| | 170: 로드부 | 180: 데이터 저장부 |
| | 190: 하이브리드 테스트부 | |

도면

도면1

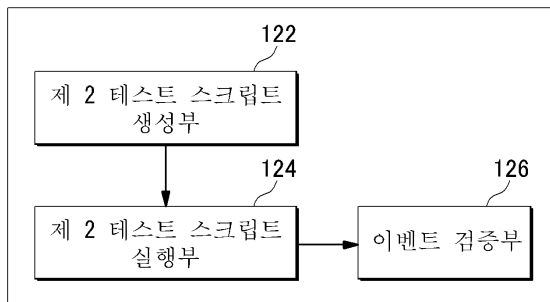


도면2

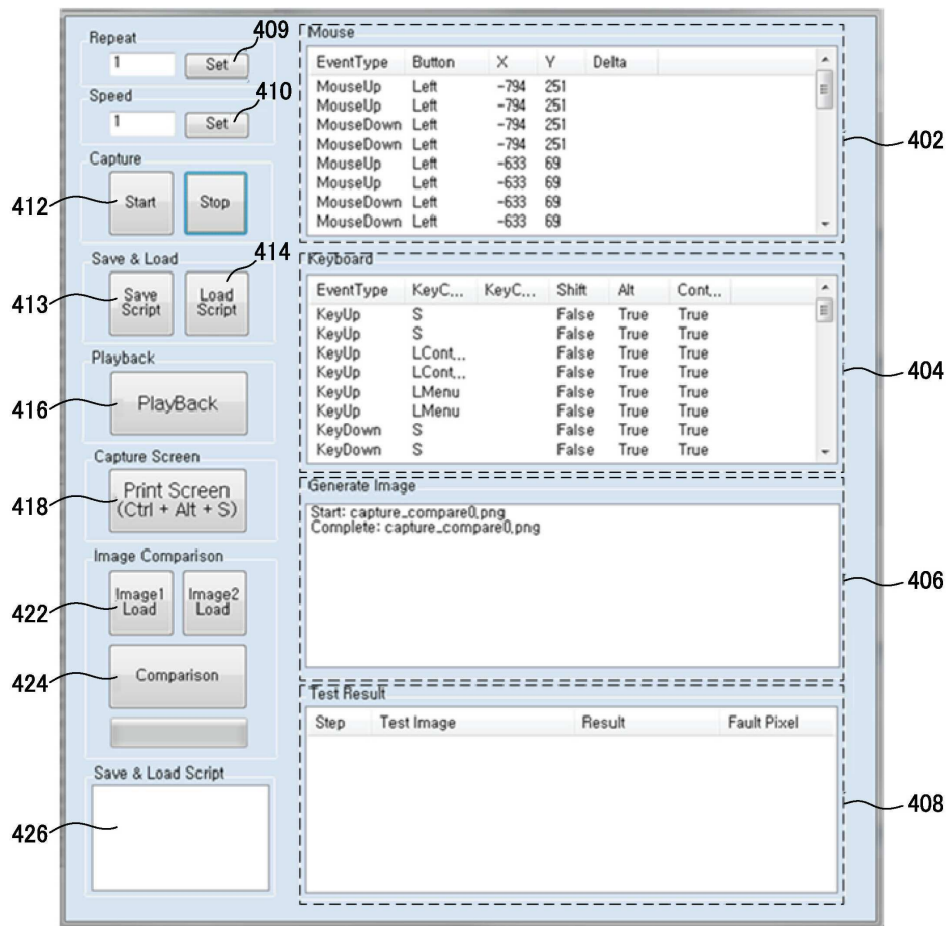


도면3

120



도면4



도면6a

502

```
<?xml version="1.0" encoding="utf-8"?>
<test>

  <step>
    <comment>Menu->View->Trace Data</comment>
    <type>click</type>
    <content>Trace Data</content>
    <eventName>listViewToolStripMenuItem_Click</eventName>
  </step>

  <step>
    <comment>TREE 선택</comment>
    <type>text</type>
    <content>CCCC</content>
    <eventName>textBox1</eventName>
  </step>

  <step>
    <comment>Data Type 선택</comment>
    <type>text</type>
    <content>User Define</content>
    <eventName>comboBox1</eventName>
  </step>

  <step>
    <comment>조회 버튼</comment>
    <type>click</type>
    <content>검색</content>
    <eventName>button2_Click_1</eventName>
  </step>

  <step>
    <comment>Lot File 경로 선택</comment>
    :
  </step>
```

도면6b

504

Test Scenario

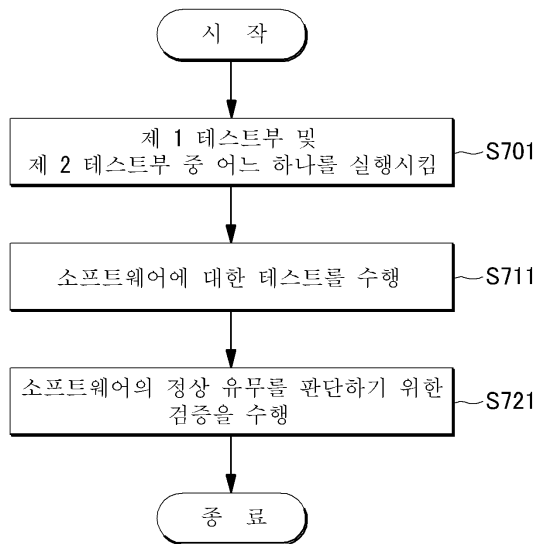
- 1) Menu->View-> Trace Data
- 2) TREE 선택
- 3) Data Type 선택
- 4) 조회 버튼
- 5) Lot File 경로 선택
- 6) Lot File Search
- 7) 삭제할 위한 파일선택
- 8) 선택된 Lot 파일 삭제

408

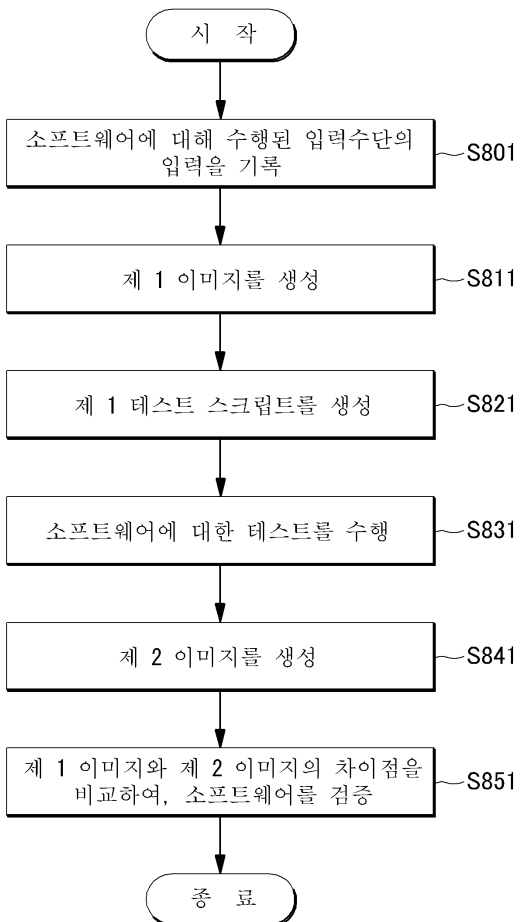
Test Result

Result	Step	Type	Content	eventName
Pass	Menu->View-> Trace Data	click	Trace Data	listViewToolStripMenuulte..
Pass	TREE 선택	text	CCCC	textBox1
Pass	Data Type 선택	text	User Define	comboBox1
Pass	조회 버튼	click	검색	button2_Click_1
Pass	Lot File 경로 선택	text	C:\WWWLotList	bt_Dir
Pass	Lot File Search	click	검색	btn_Search_Click
Pass	삭제할 위한 파일선택	click	체크 버튼	Ist_View_SelectedIndexC.
Pass	선택된 Lot 파일 삭제	click	Delete	button1_Click

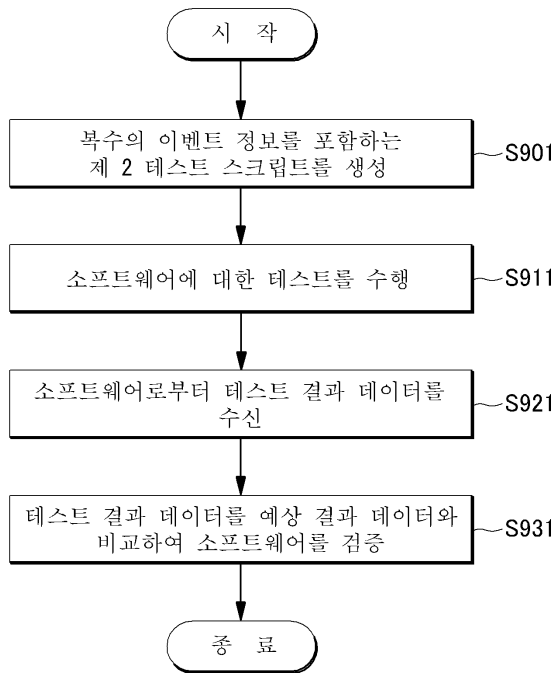
도면7



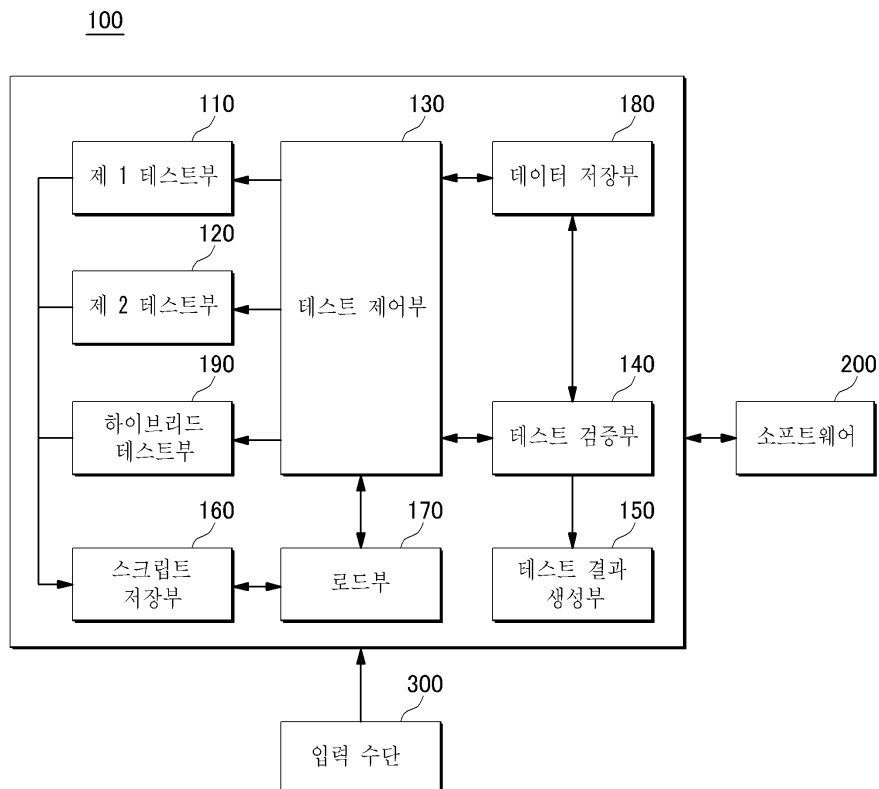
도면8



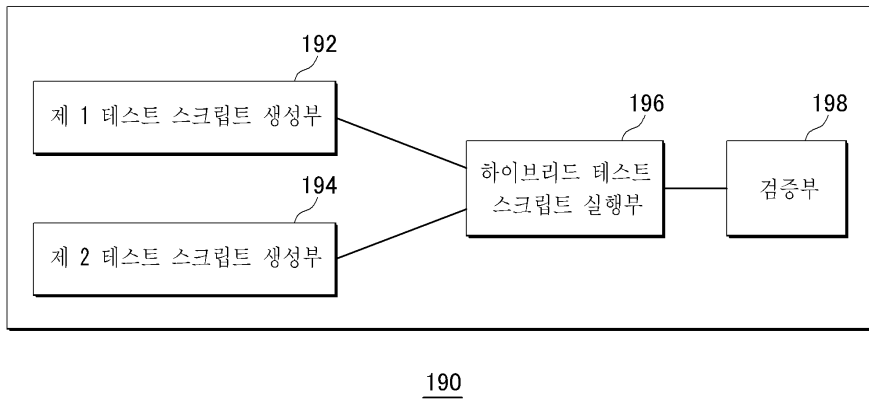
도면9



도면10



도면11



도면12

