



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0101977  
(43) 공개일자 2023년07월07일

- |   |   |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)<br/>G01R 31/317 (2006.01) G01R 31/3177 (2006.01)<br/>G01R 31/3183 (2006.01) G06N 20/00 (2019.01)<br/>G06N 3/08 (2023.01) G06T 7/00 (2017.01)<br/>G06T 7/187 (2017.01)</p> <p>(52) CPC특허분류<br/>G01R 31/31707 (2013.01)<br/>G01R 31/3177 (2013.01)</p> <p>(21) 출원번호 10-2021-0190867<br/>(22) 출원일자 2021년12월29일<br/>심사청구일자 2021년12월29일</p> | <p>(71) 출원인<br/>연세대학교 산학협력단<br/>서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)</p> <p>(72) 발명자<br/>강성호<br/>서울특별시 마포구 양화로 45, 101동 2102호(서교동, 메세나폴리스)</p> <p>김태현<br/>서울특별시 서대문구 가재울로2길 31-4, 2층(남가좌동)</p> <p>(74) 대리인<br/>특허법인(유한)아이시스</p> |
|---|---|

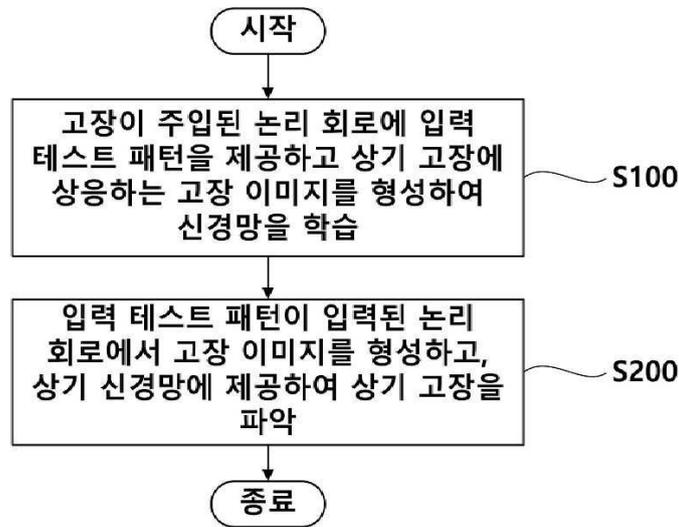
전체 청구항 수 : 총 18 항

(54) 발명의 명칭 회로의 고장 진단 방법 및 장치

(57) 요약

본 실시예의 고장 검출 방법은: 고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하고 상기 고장에 상응하는 고장 이미지를 형성하여 신경망을 학습시키는 단계와, 상기 입력 테스트 패턴이 입력된 논리 회로에서 고장 이미지를 형성하고, 상기 신경망에 제공하여 상기 고장을 파악하는 단계를 포함한다.

대표도 - 도1



(52) CPC특허분류

- G01R 31/318307 (2013.01)
- G01R 31/318342 (2013.01)
- G06N 20/00 (2021.08)
- G06N 3/08 (2023.01)
- G06T 7/0004 (2013.01)
- G06T 7/187 (2017.01)
- G06T 2207/20081 (2013.01)
- G06T 2207/30148 (2013.01)
- G06T 2219/004 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711131125
과제번호	2019R1A2C3011079
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	중견연구자지원사업
연구과제명	인-메모리 컴퓨팅의 로버스트니스 향상을 위한 반도체 설계 기술
기 여 율	1/1
과제수행기관명	연세대학교
연구기간	2021.03.01 ~ 2022.02.28

---

## 명세서

### 청구범위

#### 청구항 1

고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하고 상기 고장에 상응하는 고장 이미지를 형성하여 신경망을 학습시키는 단계와,

상기 입력 테스트 패턴이 입력된 논리 회로에서 고장 이미지를 형성하고, 상기 신경망에 제공하여 상기 고장을 파악하는 단계를 포함하는 고장 검출 방법.

#### 청구항 2

제1항에 있어서,

상기 고장이 주입된 논리 회로로부터 상기 고장 이미지를 형성하는 단계는,

주입된 상기 고장과 상기 입력 테스트 패턴에 의하여 형성된 출력 패턴을 수신하는 단계 및

상기 출력 패턴에 포함된 출력 비트 값(output bit value)을 픽셀 값으로 매핑하는 단계를 수행하여 상기 고장 이미지를 형성하는 고장 검출 방법.

#### 청구항 3

제2항에 있어서,

상기 테스트 패턴을 입력하는 단계는,

상기 논리 회로에 서로 다른 고장을 주입하여 복수회 수행하되,

상기 복수회 수행되어 입력되는 테스트 패턴은 서로 동일한 것인 고장 검출 방법.

#### 청구항 4

제3항에 있어서,

상기 논리 회로에 주입되는 고장의 개수는 각 회별로 하나인 고장 검출 방법.

#### 청구항 5

제3항에 있어서,

상기 매핑 하는 단계는,

상기 출력 비트 값이 예상된 값과 일치하면 R, G 및 B 중 어느 하나의 픽셀로 매핑하여 수행하고,

상기 예상된 값이 논리 하이(high)이나 상기 비트값이 논리 로우(low)이면 상기 R, G 및 B 중 나머지 어느 둘 중 어느 하나의 픽셀로 매핑하여 수행하며,

상기 예상된 값이 논리 로우이나 상기 비트값이 논리 로우이면 상기 R, G 및 B 중 나머지 어느 하나의 픽셀로 매핑하여 수행하는 고장 검출 방법.

#### 청구항 6

제2항에 있어서,

상기 픽셀의 값은,

전체 테스트 패턴의 수에 대한 상기 픽셀에 상응하는 개수의 비를 연산하여 얻는 고장 검출 방법.

#### 청구항 7

제1항에 있어서,  
 상기 신경망을 학습시키는 단계는,  
 상기 고장 이미지를 주입된 상기 고장으로 라벨링하여 수행하는 고장 검출 방법.

**청구항 8**

제1항에 있어서,  
 상기 입력 테스트 패턴이 제공된 논리 회로에서 고장 이미지를 형성하는 단계는,  
 상기 입력 테스트 패턴을 상기 논리 회로에 제공하는 단계;  
 상기 입력 테스트 패턴에 의하여 형성된 출력 패턴을 수신하는 단계 및  
 상기 출력 패턴에 포함된 출력 비트 값(output bit value)을 픽셀 값으로 매핑하는 단계를 수행하여 상기 고장 이미지를 형성하는 고장 검출 방법.

**청구항 9**

제1항에 있어서,  
 상기 학습시키는 단계의 상기 고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하는 단계는  
 상기 논리 회로에 대한 고장 주입 모의 실험으로 수행하고,  
 상기 고장을 파악하는 단계의 상기 입력 테스트 패턴을 논리 회로에 입력하는 단계는,  
 칩에 상기 입력 테스트 패턴을 제공하여 수행하는 고장 검출 방법.

**청구항 10**

회로의 고장 검출 장치로, 상기 장치는:  
 적어도 하나 이상의 프로세서; 및  
 상기 프로세서에 의해 실행되는 하나 이상의 프로그램을 저장하는 메모리를 포함하며, 상기 프로그램들은 하나 이상의 프로세서에 의해 실행될 때, 상기 하나 이상의 프로세서들은 고장 검출 방법을 수행하며, 상기 고장 검출 방법은:  
 고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하고 상기 고장에 상응하는 고장 이미지를 형성하여 신경망을 학습시키는 단계와,  
 상기 입력 테스트 패턴이 입력된 논리 회로에서 고장 이미지를 형성하고, 상기 신경망에 제공하여 상기 고장을 파악하는 단계를 포함하는 고장 검출 장치.

**청구항 11**

제10항에 있어서,  
 상기 고장이 주입된 논리 회로로부터 상기 고장 이미지를 형성하는 단계는,  
 주입된 상기 고장과 상기 입력 테스트 패턴에 의하여 형성된 출력 패턴을 수신하는 단계 및  
 상기 출력 패턴에 포함된 출력 비트 값(output bit value)을 픽셀 값으로 매핑하는 단계를 수행하여 상기 고장 이미지를 형성하는 고장 검출 장치.

**청구항 12**

제11항에 있어서,  
 상기 테스트 패턴을 입력하는 단계는,  
 상기 논리 회로에 서로 다른 고장을 주입하여 복수회 수행하되,

상기 복수회 수행되어 입력되는 테스트 패턴은 서로 동일한 것인 고장 검출 장치.

**청구항 13**

제12항에 있어서,

상기 논리 회로에 주입되는 고장의 개수는 각 회별로 하나인 고장 검출 장치.

**청구항 14**

제11항에 있어서,

상기 매핑 하는 단계는,

상기 출력 비트 값이 예상된 값과 일치하면 R, G 및 B 중 어느 하나의 픽셀로 매핑하여 수행하고,

상기 예상된 값이 논리 하이(high)이나 상기 비트값이 논리 로우(low)이면 상기 R, G 및 B 중 나머지 어느 둘 중 어느 하나의 픽셀로 매핑하여 수행하며,

상기 예상된 값이 논리 로우이나 상기 비트값이 논리 로우이면 상기 R, G 및 B 중 나머지 어느 하나의 픽셀로 매핑하여 수행하는 고장 검출 장치.

**청구항 15**

제11항에 있어서,

상기 픽셀의 값은,

전체 테스트 패턴의 수에 대한 상기 픽셀에 상응하는 개수의 비를 연산하여 얻는 고장 검출 장치.

**청구항 16**

제10항에 있어서,

상기 신경망을 학습시키는 단계는,

상기 고장 이미지를 주입된 상기 고장으로 라벨링하여 수행하는 고장 검출 장치.

**청구항 17**

제10항에 있어서,

상기 입력 테스트 패턴이 제공된 논리 회로에서 고장 이미지를 형성하는 단계는,

상기 입력 테스트 패턴을 상기 논리 회로에 제공하는 단계;

상기 입력 테스트 패턴에 의하여 형성된 출력 패턴을 수신하는 단계 및

상기 출력 패턴에 포함된 출력 비트 값(output bit value)을 픽셀 값으로 매핑하는 단계를 수행하여 상기 고장 이미지를 형성하는 고장 검출 장치.

**청구항 18**

제10항에 있어서,

상기 학습시키는 단계의 상기 고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하는 단계는

상기 논리 회로에 대한 고장 주입 모의 실험으로 수행하고,

상기 고장을 파악하는 단계의 상기 입력 테스트 패턴을 논리 회로에 입력하는 단계는,

칩에 상기 입력 테스트 패턴을 제공하여 수행하는 고장 검출 장치.

**발명의 설명**

**기술 분야**

[0001] 본 기술은 회로의 고장 진단 방법 및 장치와 관련된다.

**배경 기술**

[0002] 기존의 논리 회로 고장 진단 방법은 고장이 발생 후 논리 회로에 대한 분석과 테스트 결과에 대한 분석을 통해 고장의 종류와 위치를 판단하는데, 이 과정은 회로의 크기가 커질수록 기하급수적으로 많은 시간이 필요하다.

**발명의 내용**

**해결하려는 과제**

[0003] 종래 기술에 의하면 고장 진단을 위해 많은 양의 데이터와 분석 시간을 요구하여 정확한 고장의 종류와 위치를 확인하는 데 어려움을 겪으며, 인공지능을 활용하는 방법 역시 고장 정보로부터 복잡한 과정을 거쳐 특징을 추출하고 인공지능을 학습시키는 데 큰 노력과 긴 시간이 필요하다.

[0004] 본 실시예로 해결하고자 하는 과제 중 하나는 종래 기술에 비하여 간단하게 신경망을 학습시키고, 학습된 신경망을 이용하여 간단하게 로직 회로의 고장을 검출하되, 학습과 추론에 필요한 시간을 단축하여 빠르게 고장을 진단하기 위한 것이다.

**과제의 해결 수단**

[0005] 본 실시예의 고장 검출 방법은: 고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하고 상기 고장에 상응하는 고장 이미지를 형성하여 신경망을 학습시키는 단계와, 상기 입력 테스트 패턴이 입력된 논리 회로에서 고장 이미지를 형성하고, 상기 신경망에 제공하여 상기 고장을 파악하는 단계를 포함한다.

[0006] 본 실시예의 회로의 고장 검출 장치는: 적어도 하나 이상의 프로세서; 및

[0007] 상기 프로세서에 의해 실행되는 하나 이상의 프로그램을 저장하는 메모리를 포함하며, 상기 프로그램들은 하나 이상의 프로세서에 의해 실행될 때, 상기 하나 이상의 프로세서들은 고장 검출 방법을 수행하며, 상기 고장 검출 방법은: 고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하고 상기 고장에 상응하는 고장 이미지를 형성하여 신경망을 학습시키는 단계와, 상기 입력 테스트 패턴이 입력된 논리 회로에서 고장 이미지를 형성하고, 상기 신경망에 제공하여 상기 고장을 파악하는 단계를 포함한다.

[0008] 본 실시예의 어느 한 측면에 의하면, 상기 고장이 주입된 논리 회로로부터 상기 고장 이미지를 형성하는 단계는, 주입된 상기 고장과 상기 입력 테스트 패턴에 의하여 형성된 출력 패턴을 수신하는 단계 및 상기 출력 패턴에 포함된 출력 비트 값(output bit value)을 픽셀 값으로 매핑하는 단계를 수행하여 상기 고장 이미지를 형성한다.

[0009] 본 실시예의 어느 한 측면에 의하면, 상기 테스트 패턴을 입력하는 단계는, 상기 논리 회로에 서로 다른 고장을 주입하여 복수회 수행하되, 상기 복수회 수행되어 입력되는 테스트 패턴은 서로 동일한 것이다.

[0010] 본 실시예의 어느 한 측면에 의하면, 상기 논리 회로에 주입되는 고장의 개수는 각 회별로 하나이다.

[0011] 본 실시예의 어느 한 측면에 의하면, 상기 매핑 하는 단계는, 상기 출력 비트 값이 예상된 값과 일치하면 R, G 및 B 중 어느 하나의 픽셀로 매핑하여 수행하고, 상기 예상된 값이 논리 하이(high)이나 상기 비트값이 논리 로우(low)이면 상기 R, G 및 B 중 나머지 어느 둘 중 어느 하나의 픽셀로 매핑하여 수행하며, 상기 예상된 값이 논리 로우이나 상기 비트값이 논리 로우이면 상기 R, G 및 B 중 나머지 어느 하나의 픽셀로 매핑하여 수행한다.

[0012] 본 실시예의 어느 한 측면에 의하면, 상기 픽셀의 값은, 전체 테스트 패턴의 수에 대한 상기 픽셀에 상응하는 개수의 비를 연산하여 얻는다.

[0013] 본 실시예의 어느 한 측면에 의하면, 상기 신경망을 학습시키는 단계는, 상기 고장 이미지를 주입된 상기 고장으로 라벨링하여 수행한다.

[0014] 본 실시예의 어느 한 측면에 의하면, 상기 입력 테스트 패턴이 제공된 논리 회로에서 고장 이미지를 형성하는 단계는, 상기 입력 테스트 패턴을 상기 논리 회로에 제공하는 단계; 상기 입력 테스트 패턴에 의하여 형성된 출력 패턴을 수신하는 단계 및 상기 출력 패턴에 포함된 출력 비트 값(output bit value)을 픽셀 값으로 매핑하는 단계를 수행하여 상기 고장 이미지를 형성한다.

[0015] 본 실시예의 어느 한 측면에 의하면, 상기 학습시키는 단계의 상기 고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하는 단계는 상기 논리 회로에 대한 고장 주입 모의 실험으로 수행하고, 상기 고장을 파악하는 단계의 상기 입력 테스트 패턴을 논리 회로에 입력하는 단계는, 칩에 상기 입력 테스트 패턴을 제공하여 수행한다.

**발명의 효과**

[0016] 본 실시예에 의하면, 인공지능을 활용한 칩의 고장 진단이 가능해지며, 고장 진단에 필요한 연산이 최소화되어 진단에 필요한 시간을 단축할 수 있다는 장점이 제공된다. 이로부터 반도체 생산 공정 초기 단계에 빠르게 고장의 원인을 찾아 수율을 향상시키고 생산 비용을 감소시킬 수 있다.

**도면의 간단한 설명**

[0017] 도 1은 본 실시예에 의한 회로의 고장 진단 방법의 개요를 도시한 순서도이다.  
 도 2는 본 실시예에 의한 고장 진단 장치의 개요를 도시한 도면이다.  
 도 3A로 예시된 것과 같이 복수의 스캔 셀들을 포함하는 스캔 체인에 입력 테스트 패턴을 시프트 인(shift in) 하고, 스캔 셀들에 제공된 입력 테스트 패턴을 테스트 대상 회로(DUT)에 제공하는 과정을 도시한 도면이고, 도 3B는 테스트 대상 회로(DUT)에서 형성된 출력 패턴(output pattern)을 캡처(capture)하여 스캔 셀로 제공되고, 스캔 셀이 형성된 출력 패턴(output pattern)을 출력하는 과정을 예시한 도면이다.  
 도 4는 고장 로그(fail log)를 개요적으로 도시한 도면이다.  
 도 5A는 고장 로그로부터 연산된 고장 계수(fail count)를 예시한 도면이고, 도 5B는 고장 계수로부터 픽셀값을 매핑하는 과정을 예시한 도면이다.

**발명을 실시하기 위한 구체적인 내용**

[0018] 이하에서는 첨부된 도면들을 참조하여 본 실시예를 설명한다. 도 1은 본 실시예에 의한 회로의 고장 진단 방법의 개요를 도시한 순서도이다. 도 1을 참조하면, 본 실시예에 의한 고장 진단 방법은 고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하고 상기 고장에 상응하는 고장 이미지를 형성하여 신경망을 학습시키는 단계(S100)와, 상기 입력 테스트 패턴이 입력된 논리 회로에서 고장 이미지를 형성하고, 상기 신경망에 제공하여 상기 고장을 파악하는 단계(S200)를 포함한다.

[0019] 도 2는 본 실시예에 의한 고장 진단 장치(10)의 개요를 도시한 도면이다. 도 2는 본 실시예에 의한 테스트 장치(1)의 개요를 나타낸 블록도이다. 도 2를 참조하면, 도 2는 본 실시예에 의한 테스트 장치(1)의 개요를 도시한 블록도이다. 도 2를 참조하면, 본 실시예에 따른 테스트 장치(1)는 입력부(21), 출력부(22), 프로세서(25), 메모리(24) 및 데이터베이스(23)를 포함한다.

[0020] 도 2의 고장 진단 장치(1)는 일 실시예에 따른 것으로서, 도 2에 도시된 모든 블록이 필수 구성요소는 아니며, 다른 실시예에서 고장 진단 장치(1)에 포함된 일부 블록이 추가, 변경 또는 삭제될 수 있다. 한편, 고장 진단 장치(1)는 고장 진단 방법을 수행하는 컴퓨팅 장치로 구현될 수 있으며, 고장 진단 장치(1)에 포함된 각 구성요소들은 각각 별도의 소프트웨어 장치로 구현되거나, 소프트웨어가 결합된 별도의 하드웨어 장치로 구현될 수 있다.

[0021] 고장 진단 장치(1)는 초고장이 주입된 논리 회로에 입력 테스트 패턴을 제공하고 상기 고장에 상응하는 고장 이미지를 형성하여 신경망을 학습시키는 단계(S100)와, 상기 입력 테스트 패턴이 입력된 논리 회로에서 고장 이미지를 형성하고, 상기 신경망에 제공하여 상기 고장을 파악하는 단계(S200)를 수행한다.

[0022] 입력부(21)는 고장 정보에 상응하는 신호 또는 데이터를 입력받는 수단을 의미한다. 입력부(21)는 외부 장치로부터 테스트 대상 회로의 고장 정보를 제공받을 수 있다. 또한, 입력부(21)는 프로세서(25)과 연동하여 다양한 형태의 신호 또는 데이터를 입력하거나, 외부 장치와 연동하여 직접 데이터를 획득하여 프로세서(25)로 전달할 수도 있다. 입력부(21)는 고장 로그(fail log) 등의 로그 정보(log), 다양한 조건 정보 또는 제어신호 등을 입력하거나 입력 받기 위한 장치 또는 서버일 수 있으나 반드시 이에 한정되는 것은 아니다.

[0023] 출력부(22)는 프로세서(25)와 연동하여 고장 진단 결과, 성공 또는 실패 로그 정보 등을 표시할 수 있다. 출력부(22)는 소정의 정보를 출력하기 위하여 고장 진단 장치(1)에 구비된 디스플레이(미도시), 스피커 등을 통해 다양한 정보를 표시하는 것이 바람직하나 반드시 이에 한정되는 것은 아니다.

- [0024] 프로세서(25)은 메모리(24)에 포함된 적어도 하나의 명령어 또는 프로그램을 수행한다. 본 실시예에 따른 프로세서(25)는 입력부(21) 또는 데이터베이스(23)로부터 획득한 데이터를 기반으로 각 단계를 수행하기 위한 데이터를 연산한다.
- [0025] 메모리(24)는 프로세서(25)에 의해 실행 가능한 적어도 하나의 명령어 또는 프로그램을 포함한다. 메모리(24)는 처리를 수행하기 위한 명령어 또는 프로그램을 포함할 수 있다. 메모리(24)는 각 단계에서 수행된 결과, 중간값 등의 연관된 값들을 저장할 수 있다.
- [0026] 데이터베이스(23)는 데이터베이스 관리 프로그램(DBMS)을 이용하여 컴퓨터 시스템의 저장공간(하드디스크 또는 메모리)에 구현된 일반적인 데이터구조를 의미하는 것으로, 데이터의 검색(추출), 삭제, 편집, 추가 등을 자유롭게 행할 수 있는 데이터 저장형태를 뜻하는 것으로, 오라클(Oracle), 인포믹스(Infomix), 사이베이스(Sybase), DB2와 같은 관계형 데이터베이스 관리 시스템(RDBMS)이나, 겜스톤(Gemston), 오리온(Orion), O2 등과 같은 객체 지향 데이터베이스 관리 시스템(OODBMS) 및 엑셀론(Excelon), 타미노(Tamino), 세카이주(Sekaiju) 등의 XML 전용 데이터베이스(XML Native Database)를 이용하여 본 발명의 일 실시예의 목적에 맞게 구현될 수 있고, 자신의 기능을 달성하기 위하여 적당한 필드(Field) 또는 엘리먼트들을 가지고 있다.
- [0027] 본 실시예에 따른 데이터베이스(23)은 로그 정보, 진단된 고장 정보, 조건 정보, 고장 진단 결과정보 등을 저장하고, 저장된 데이터를 제공할 수 있다. 한편, 데이터베이스(24)는 고장 진단 장치(1) 내에 구현되는 것으로 기재하고 있으나 반드시 이에 한정되는 것은 아니며, 별도의 데이터 저장장치로 구현될 수도 있다.
- [0028] 계속해서 도 1 및 도 3을 참조하면, 칩의 시뮬레이션 단계에서 칩에 고장을 주입한다. 일 실시예로, 칩에 고장을 주입하는 단계는 시뮬레이션 단계에서 수행될 수 있다. 칩에 주입되는 고장(F)은 하나의 고장으로, 칩에 포함된 복수의 논리 소자를 대상으로 수행할 수 있으며, 각 논리 소자의 출력이 0으로 고착(Stuck at 0)되거나, 혹은 1로 고착(Stuck at 1)되는 고장으로 상정되어 주입될 수 있다. 고장 주입시 이전에 주입된 고장과 다른 하나의 고장을 설정하여 주입될 수 있다.
- [0029] 일 실시예로, 테스트 대상 회로(DUT)에 주입되는 고장은 어느 한 소자의 입력 혹은 출력이 논리 하이로 고착되는(Stuck at 1) 고장이거나, 논리 로우로 고착되는(Stuck at 0) 고장 중 어느 하나일 수 있다.
- [0030] 고장 주입 후, 도 3A로 예시된 것과 같이 복수의 스캔 셀들을 포함하는 스캔 체인에 입력 테스트 패턴을 시프트 인(shift in)하고, 스캔 셀들에 제공된 입력 테스트 패턴을 테스트 대상 회로(DUT)에 제공한다. 일 실시예로, 하나의 고장 주입 후, 복수의 서로 다른 테스트 패턴들을 테스트 대상 회로(DUT)에 제공할 수 있으며, 또 다른 하나의 고장을 테스트 대상 회로(DUT)에 주입하는 경우에 제공되는 테스트 패턴들은 서로 동일할 수 있다. 따라서, 동일한 하나 이상의 테스트 패턴을 매 서로 다른 고장 주입시 마다 제공할 수 있다.
- [0031] 입력 테스트 패턴이 제공된 테스트 대상 회로(DUT)는 출력 패턴(output pattern)을 형성하여 제공한다. 출력 패턴(output pattern)은 입력 테스트 패턴, 테스트 대상 회로(DUT)의 구성 및 주입된 고장(F)에 의하여 영향을 받을 수 있다. 도 3B로 예시된 것과 같이 테스트 대상 회로(DUT)에서 형성된 출력 패턴(output pattern)은 캡처(capture)되어 스캔 셀로 제공되고, 스캔 셀은 형성된 출력 패턴(output pattern)을 출력한다.
- [0032] 각 입력 패턴 별로 출력 패턴은 서로 다를 수 있으며, 출력 패턴을 포함하는 고장 로그(fail log)를 개요적으로 도시하면 도 4와 같다. 도 4를 참조하면, 0번 패턴에서 첫 비트에서 예상되었던 0이 아닌 1 값이 나오는 것을 알 수 있으며, 0번 패턴에서 두번째 비트에서 예상되었던 값인 1이 아닌 0 이 출력되는 것을 알 수 있다.
- [0033] 즉, 테스트 대상 회로(DUT)에서 출력된 패턴 0, 패턴 1, ..., 패턴 n-1 가지의 총 n 개의 패턴에 대하여 각 패턴별 순서별 비트에서 발생하는 예측값과 출력되는 값을 얻을 수 있다.
- [0034] 도 5A는 고장 로그로부터 연산된 고장 계수(fail count)를 예시한 도면이다. 도 5A를 참조하면, 고장 로그(fail log)에서 패턴의 순서와 고장의 종류 별로 계수한다. 첫 번째 비트에서 예상된 값인 0이 아니라 1이 출력된 경우는 45회, 예상된 값인 1이 아니라 0이 출력된 경우는 18회이다. 총 테스트 패턴의 개수가 300개 라면 예상된 값과 동일한 값이 출력된 경우는  $300 - (45 + 18) = 237$ 로 연산될 수 있다.
- [0035] 마찬가지로, 두 번째 비트에서 예상된 값인 0이 아니라 1이 출력된 경우는 78회, 예상된 값인 1이 아니라 0이 출력된 경우는 188회이다. 총 테스트 패턴의 개수가 1000개 라면 예상된 값과 동일한 값이 출력된 경우는  $300 - (78 + 188) = 34$ 로 연산될 수 있다.
- [0036] 도 5B는 고장 계수로부터 픽셀값을 매핑하는 과정을 예시한 도면이다. 도 5B를 참조하면, 일 실시예로, 예측된 값이 0 이나 1이 출력된 경우에는 R, G, B 픽셀 중 어느 하나로 매핑될 수 있다. 또한, 예측된 값이 1 이나, 0

이 출력된 경우에는 나머지 두 개 중 어느 하나의 픽셀로 매핑될 수 있다. 나머지 예측된 값과 출력된 값이 같은 경우에는 나머지 하나의 픽셀로 매핑될 수 있다.

[0037] 도시된 실시예에서 예측된 값이 0 이나 1이 출력된 경우에는 R 픽셀로 매핑을 수행한다. 또한, 예측된 값이 1 이나, 0이 출력된 경우에는 G 픽셀로 매핑된다. 나머지 예측된 값과 출력된 값이 같은 경우에는 B 픽셀로 매핑 된다.

[0038] 이어서 픽셀의 값을 연산한다. 일반적인 이미지에서 R 픽셀, G 픽셀 및 B 픽셀의 값은 0 ~ 255의 값을 가질 수 있다. 따라서, 도 5A로 계수된 값을 이용하여 픽셀의 값을 연산한다. 일 예로, 첫 번째 비트의 R 픽셀의 경우에, 예측된 값이 0이나 1이 출력된 경우는 총 45회로, 전체 테스트 패턴의 수인 300에 대한 비율을 연산하면 0.15이다. 이를 255 단계로 환산하면 38.25이다. 일 실시예로, 소수점 이하의 수는 버림, 올림 및 반올림 중 어느 하나로 연산될 수 있다.

[0039] 다른 예로, 일 예로, 1번째 비트의 G 픽셀의 경우에, 예측된 값이 1이나 0이 출력된 경우는 총 18회로, 전체 테스트 패턴의 수인 300에 대한 비율을 연산하면 0.06이다. 이를 255 단계로 환산하면 15.3이다. 각 값에 대하여 반올림을 수행하면 도 5B로 예시된 픽셀 값을 연산할 수 있다.

[0040] 형성된 픽셀들을 이미지 형태로 배열하여 고장 이미지를 형성한다. 일 실시예로, 1에서 k 까지의 각 픽셀들을 총 j 개의 픽셀을 포함하는 행으로 배열하여 고장 이미지를 형성할 수 있다. 즉, 1 ~ j 까지의 픽셀을 포함하는 제1행, j+1 ~ 2j 까지의 제2행, ... , k 까지의 픽셀을 포함하는 n행으로 각 픽셀들을 배치하여 고장 이미지를 배치할 수 있다.

[0041] 다른 예로, 1에서 k 까지의 각 픽셀들을 총 j 개의 픽셀을 포함하는 열로 배열하여 고장 이미지를 형성할 수 있다. 또한, 1 ~ k까지의 픽셀들이 j 개로 나누어떨어지지 않는 경우에는 더미 픽셀을 부가하여 고장 이미지를 형성하는 것도 가능하다. 이와 같이 형성된 고장 이미지는 주입된 고장과 함께 라벨되고, 신경망을 학습시킨다 (S100).

[0042] 이어서, 테스트 대상 회로(DUT)에 상기한 입력 테스트 패턴을 제공하고, 테스트 대상 회로(DUT)에서 제공된 출력 패턴을 수신한다.

[0043] 상술한 과정을 거쳐 출력 패턴으로부터 고장 이미지를 형성한다. 일 실시예로, 출력된 패턴의 고장 로그로부터 고장 계수를 연산하고, 고장 계수 결과로부터 픽셀값을 매핑하는 과정을 거쳐 고장 이미지를 형성할 수 있다.

[0044] 이와 같이 형성된 고장 이미지를 학습된 신경망에 입력하여 테스트 대상 회로(DUT) 내의 어느 요소에 고장이 발생하였는지, 어떠한 유형의 고장이 발생하였는지 파악할 수 있다(S200).

[0046] 본 실시예에 의하면, 인공지능을 활용한 칩의 고장 진단이 가능해지며, 고장 진단에 필요한 연산이 최소화되어 진단에 필요한 시간을 단축할 수 있다는 장점이 제공된다. 이로부터 반도체 생산 공정 초기 단계에 빠르게 고장의 원인을 찾아 수율을 향상시키고 생산 비용을 감소시킬 수 있다.

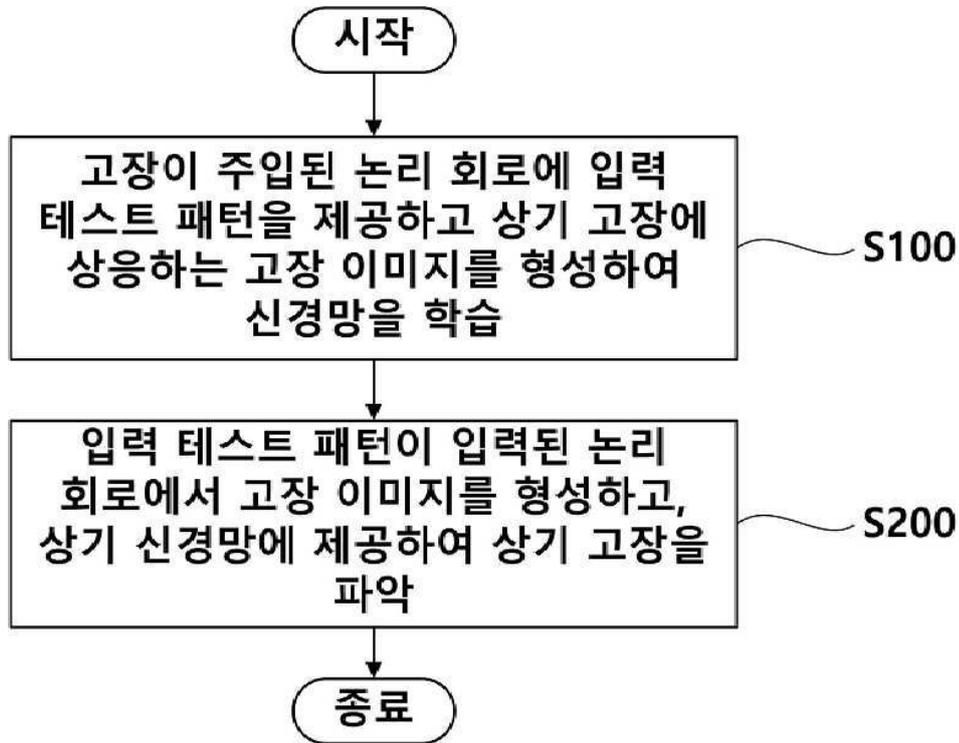
[0048] 본 발명에 대한 이해를 돕기 위하여 도면에 도시된 실시 예를 참고로 설명되었으나, 이는 실시를 위한 실시예로, 예시적인 것에 불과하며, 당해 분야에서 통상적 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호범위는 첨부된 특허청구범 위에 의해 정해져야 할 것이다.

**부호의 설명**

- [0049] 1: 고장 검출 장치
- 21: 입력부    22: 출력부
- 23: 데이터베이스    24: 메모리
- 25: 프로세서

도면

도면1

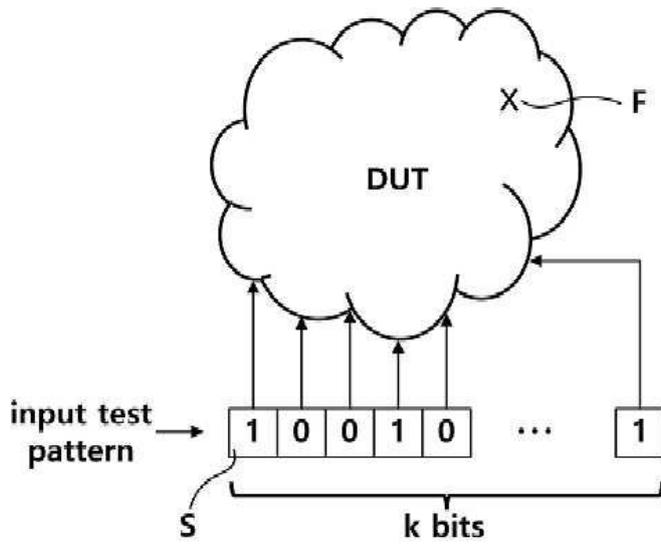


도면2

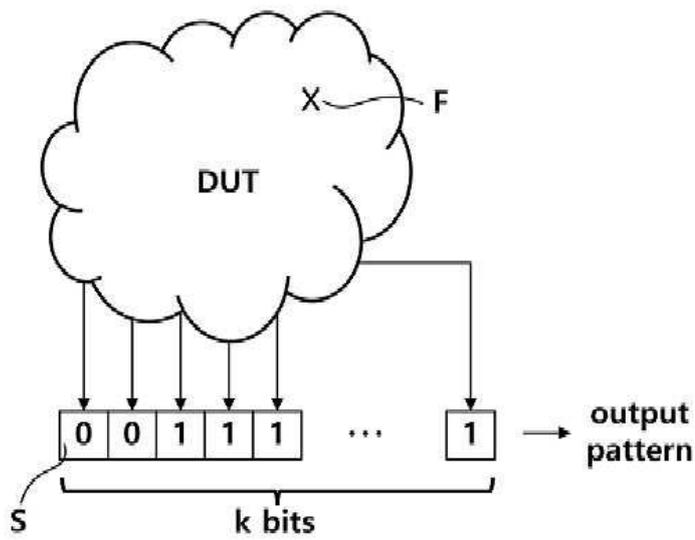
1



도면3



도 3A



도 3B

도면4

패턴 번호	순서	오류
pattern 0	1	0->1
pattern 0	2	1->0
pattern 3	2	1->0
pattern 7	3	1->0
...	...	...
pattern n-1	k	0->1

도면5

	1	2	3	...	k
0->1	45	78	33	...	84
1->0	18	188	12	...	102
0->0 또는 1->1	237	34	255		114

도 5A

	1	2	3	...	k
R	38	66	28	...	71
G	15	160	10	...	87
B	201	29	217	...	97

도 5B