



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2008-0050858  
(43) 공개일자 2008년06월10일

(51) Int. Cl.

G11C 29/00 (2006.01)

(21) 출원번호 10-2006-0121702

(22) 출원일자 2006년12월04일

심사청구일자 없음

(71) 출원인

삼성전자주식회사

경기도 수원시 영통구 매탄동 416

(72) 발명자

심민규

경기 용인시 기흥구 농서동 151-8 아진빌라 A동 103호

(74) 대리인

권혁수, 송윤호, 오세준

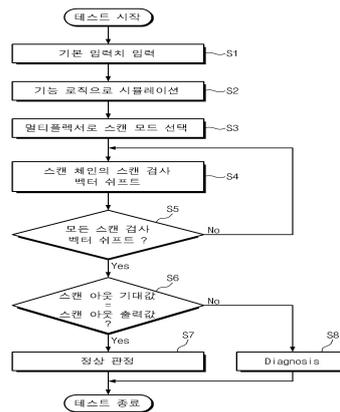
전체 청구항 수 : 총 4 항

(54) 기능 로직을 이용한 스캔 체인 검사 방법

(57) 요약

여기에 개시되는 스캔 체인 검사 방법은 기본 입력치에 응답하여 기능 로직에서 발생한 데이터를 스캔 체인에 저장하고 스캔 체인에 저장된 데이터를 출력하여 이를 기능 로직에서 발생한 데이터와 비교함으로써 스캔 체인 내의 어느 위치에 오류가 있는지를 판단한다.

대표도



## 특허청구의 범위

### 청구항 1

스캔 체인을 검사하는 방법에 있어서:  
 기능 로직에 데이터 비트들을 입력하여 테스트 데이터 비트들을 출력하는 단계와;  
 상기 테스트 데이터 비트들을 스캔 체인에 저장하는 단계와;  
 상기 스캔 체인에 저장된 데이터를 출력하는 단계와; 그리고  
 상기 테스트 데이터 비트들과 상기 스캔 체인으로부터 출력된 데이터를 비교하는 단계를 포함하는 방법.

### 청구항 2

제 1 항에 있어서,  
 상기 테스트 데이터 비트들은 상기 스캔 체인의 스캔 체인 셀들에 병렬로 저장되는 것을 특징으로 하는 방법.

### 청구항 3

제 2 항에 있어서,  
 상기 스캔 체인의 스캔 체인 셀들에 저장된 데이터는 직렬로 출력되는 것을 특징으로 하는 방법.

### 청구항 4

제 3 항에 있어서,  
 상기 테스트 데이터 비트들과 상기 스캔 체인으로부터 출력된 데이터를 최초의 스캔 체인 값부터 순차로 비교하는 것을 특징으로 하는 방법.

## 명세서

### 발명의 상세한 설명

#### 발명의 목적

#### 발명이 속하는 기술 및 그 분야의 종래기술

- <5> 본 발명은 반도체 집적 회로 장치의 검사 방법에 관한 것으로, 좀 더 구체적으로는 기능 로직을 이용하여 스캔 체인을 검사하는 방법에 관한 것이다.
- <6> 고밀도 집적 회로 설계 및 공정 기술이 최근 들어 급격히 발전함에 따라 회로의 집적도 및 복잡도가 크게 증가하고 있다. 이에 따라 완성된 고밀도 집적 회로 시스템의 정상 동작 여부를 판단하는 검사에 소요되는 비용도 크게 늘고 있다. 설계 사양을 만족시켜주는 완벽한 설계 못지 않게 완성된 시스템의 완벽한 동작을 보장해주는 테스트 또한 매우 중요한 문제이다. 따라서 설계 단계에서부터 테스트를 쉽게 할 수 있는 기법으로서 테스트를 위한 설계 방법(DFT : Design For Testability)이 도입되었다.
- <7> 최근에 개발되는 대부분의 마이크로 프로세서나 마이크로 컨트롤러들은 테스트 자체가 거의 불가능하며, 따라서 다양한 테스트 설계 방법들이 초기 설계단계에서부터 폭넓게 적용되고 있다. 회로를 효율적으로 테스트하기 위해서는 회로의 특성에 맞도록 여러 가지 테스트를 고려한 설계 방법이 사용되어야 하며, 이에 따른 성능저하 및 소요 면적 증가 등 부과되는 오버헤드를 최소화할 수 있는 기법이 적용되어야만 한다.
- <8> 테스트를 고려한 설계 방법은 테스트를 쉽게 하기 위해 주어진 회로를 변형시키거나 여분의 하드웨어를 추가하는 방식으로 이루어진다. 그러나 테스트 설계 방법을 적용할 경우 부수적으로 회로의 크기, 입출력단자수의 증가 및 성능 저하가 생기게 된다. 그러나 이러한 부작용에 비해 테스트 비용 및 테스트 시간 절감 등, 테스트 설계 방법에 따른 이득이 매우 크고, 또한 최근에 개발되는 시스템들에 있어서 테스트 설계 방법을 적용하지 않을 시에는 검사 자체가 거의 불가능하기 때문에 이를 적극적으로 활용하고 있다.

- <9> 테스트를 고려한 여러 가지 설계 기법 중 하나가 스캔 경로 설계이다. 스캔 경로 설계에 따라, 반도체 집적 회로 장치의 구성 요소 회로에서 발견되는 순차 회로내의 모든 메모리 소자들(플립플롭과 같은 소자들)은 그들의 고장 진단 테스트가 수행되는 한 일련의 링크된 쉬프트 레지스터를 형성하도록 설계된다. 스캔 경로 기법에 기초한 고장 진단 테스트를 사용하면, 반도체 집적 회로 장치의 구성 요소 회로에서 발견되는 모든 메모리 소자마다 원하는 고장 진단 테스트 데이터를 연속으로 공급하기 위해 상기와 같은 쉬프트 레지스터의 비트 쉬프트 동작이 사용된다. 또한, 구성 요소 회로들이 그와 같은 테스트 데이터에 따라 실제로 동작할 때는 각각의 메모리 소자에 저장되어 유지된 내용들이 상기와 같은 쉬프트 레지스터들의 비트 쉬프트 동작을 사용하여 연속으로 출력된다. 그와 같은 쉬프트 레지스터들의 비트 쉬프트 경로를 스캔 경로라 부른다.
- <10> 자동 테스트 패턴 생성을 위한 여러 가지 기법들의 등장으로 인해 조합 회로에 대한 테스트 패턴 생성은 비교적 쉽게 할 수 있지만, 순차 회로에 대해서는 아직까지 자동 테스트 패턴 생성 기법이 많이 적용되고 있지 않다. 스캔 기법은 순차 회로의 기본을 이루는 기억 소자들을 테스트 모드에서 외부로부터 쉽게 제어할 수 있도록 한 기법이다.
- <11> 도 1은 일반적인 기능 로직 (10)과 스캔 체인 (20)의 연결관계를 보여주는 도면이다.
- <12> 기능 로직 (10)은 조합회로 또는 순차회로일 수 있다. 기능 로직 (10)의 케환 경로에는 멀티플렉서 및 플립플롭이 연결되며 멀티플렉서 및 플립플롭이 스캔 체인 (20)을 구성한다. 플립플롭은 데이터를 저장하고 멀티플렉서는 일반 모드 경로 또는 스캔 모드 경로를 선택한다.
- <13> 실시예에서는 멀티플렉서의 셀렉트 단자에 0이 입력되는 경우 기능 로직 (10)의 출력이 기능 로직 (10)의 입력으로 케환하는 정상 경로를 이루며, 멀티플렉서의 셀렉트 단자에 1이 입력되는 경우는 각 플립플롭들이 직렬연결되어 스캔 경로를 형성한다. 이를 통해 다수의 외부 연결 단자 없이 스캔 인과 스캔 아웃 단자만으로 기능 로직 (10)의 오류 여부를 검사할 수 있는 장점이 있다.
- <14> 그러나 이러한 스캔 기법에 있어서는 기능 로직 (10) 상의 오류를 검출하기 위해 엮는 스캔 체인 (20) 상에 오류가 있을 경우 스캔 체인 (20)의 직렬 연결 특성상 정확한 오류의 위치를 찾기 어려운 문제가 생긴다. 이와 같은 문제가 도 2에 도시되어있다.
- <15> 도 2는 일반적인 스캔 체인 검사 방법을 보여주는 도면이다.
- <16> 도 2를 참조하면, 스캔 체인 검사 시 먼저 스캔 인 단자를 통해 검사 데이터를 입력한다. 도면에 나타난 실시예에서는 스캔 체인 (20)에 순차로 11001100의 값을 입력한다. 그러나 스캔 체인 (20)의 네 번째 셀 (24)이 0으로 고정된 경우(Stuck - At 0), 각 셀들의 직렬 연결 특성상 첫 번째 셀 (21)부터 네 번째 셀 (24)까지의 저장된 값은 모두 0이 된다. 따라서 실제로 스캔 체인 (20)에 저장되는 값은 11000000이 된다. 실제로 스캔 체인 (20)에 저장된 값으로 시물레이션을 수행하는 경우 본래 입력하고자 했던 데이터에 의한 시물레이션 결과값과 상이한 출력값을 얻게 된다.
- <17> 실시예에서 정상적인 스캔 체인으로 판정하기 위해서는 스캔 아웃을 통해 11001100에 대응하는 시물레이션 값인 01001101이 출력되어야 한다. 그러나 스캔 체인으로의 데이터 입력과정에서 이미 데이터 값이 변하기 때문에 (11001100 -> 11000000), 그에 대응하는 시물레이션 결과값 역시 기대값과는 다른 값(00000001)을 가진다. 따라서 기대값(01001101)과 시물레이션 결과값(00000001)을 비교할 경우, 세 번째 셀 (23)이 처음으로 상이한 값을 가지므로 실제로는 네 번째 셀 (24)에 오류가 있음에도 불구하고 세 번째 셀 (23)에 오류가 있는 것으로 잘못 판단하게 된다.
- <18> 또한 스캔 체인 검사를 하기에 앞서 스캔 인을 통해 스캔 체인에 검사 데이터를 저장하는 단계가 필요하므로 검사 시간이 오래 걸리는 단점이 있다. 실시예에서 총 8개의 스캔 체인 셀 (21~28)에 데이터를 저장하기 위해서는 총 8개의 클럭이 인가되어야 하기 때문이다.

**발명이 이루고자 하는 기술적 과제**

- <19> 본 발명의 목적은 스캔 체인 내의 정확한 오류 위치를 찾아내어 스캔 체인 검사의 신뢰도를 향상시키는 스캔 체인 검사 방법을 제공하는 것이다.
- <20> 본 발명의 다른 목적은 스캔 체인 검사에 소요되는 시간을 줄일 수 있는 스캔 체인 검사 방법을 제공하는 것이다.

**발명의 구성 및 작용**

- <21>     상술한 제반 목적들을 달성하기 위한 본 발명의 특징에 따른 스캔 체인 검사 방법은, 기능 로직에 데이터 비트들을 입력하여 테스트 데이터 비트들을 출력하는 단계와, 테스트 데이터 비트들을 스캔 체인에 저장하는 단계와, 스캔 체인에 저장된 데이터를 출력하는 단계와, 그리고 테스트 데이터 비트들과 스캔 체인으로부터 출력된 데이터를 비교하는 단계를 포함한다.
- <22>     예시적인 실시예에 있어서, 상기 테스트 데이터 비트들은 상기 스캔 체인의 스캔 체인 셀들에 병렬로 저장된다.
- <23>     예시적인 실시예에 있어서, 상기 스캔 체인의 스캔 체인 셀들에 저장된 데이터는 직렬로 출력된다.
- <24>     예시적인 실시예에 있어서, 상기 테스트 데이터 비트들과 상기 스캔 체인으로부터 출력된 데이터는 최초의 스캔 체인 값부터 순차로 비교된다.
- <25>     스캔 검사 방법은 기능 로직에 기본 입력치를 입력하는 단계와; 기능 로직이 기본 입력치에 응답하여 스캔 체인 입력값을 발생하는 단계와; 스캔 체인 입력값을 스캔 아웃을 통해 출력하는 단계와; 스캔 아웃을 통해 출력한 값을 스캔 체인 입력값과 비교하는 단계를 포함한다.
- <26>     본 발명의 예시적인 실시예들이 참조 도면들에 의거하여 이하 상세히 설명될 것이다.
- <27>     도 3은 본 발명에 따른 스캔 체인 검사 방법을 보여주는 도면이다.
- <28>     도 3을 참조하면, 먼저 기능 로직 (10)에 기본 입력치를 입력하고, 기능 로직 (10)은 기본 입력치에 응답하여 시뮬레이션 결과값을 출력한다. 도 3에 나타난 실시예에서는 기본 입력치에 응답하여 시뮬레이션 결과값으로 11001100을 출력한다.
- <29>     정상 모드 하에서, 시뮬레이션 결과값은 각 스캔 체인 셀 (21~28)에 병렬로 입력된다. 실시예에서는 8개의 스캔 체인 셀에 각각 시뮬레이션 결과값이 입력되는데, 예를 들어 네 번째 스캔 체인 셀 (24)이 0으로 고정된 경우 (Stuck - At 0) 스캔 체인의 직렬 특성상 실제로 입력되는 값은 11000100이 된다. 이후 스캔 체인 셀 (20)에 저장된 값을 출력하기 위하여 스캔 모드를 선택하여 각 스캔 체인이 직렬 연결되도록 한다. 스캔 모드의 선택은 멀티플렉서의 셀렉터 단자에 의해 이루어진다. 이후 스캔 아웃을 통해 스캔 체인 셀 (20)에 저장된 값을 순차로 쉬프트시켜 스캔 패스를 통해 직렬로 출력한다.
- <30>     스캔 아웃에 의해 출력된 값을 스캔 아웃 기대값과 비교하는데 이 경우 스캔 아웃 기대값은 시뮬레이션 결과값과 동일하다. 따라서 실시예에서 스캔 아웃 기대값은 11001100이 된다. 실시예에서 스캔 아웃 출력값은 네 번째 스캔 체인 셀 (24)이 0으로 고정되었기 때문에 네 번째 값부터는 모두 0을 출력한다. 따라서 00000100이 출력된다. 이 경우 스캔 아웃 기대값과 스캔 아웃 출력값을 비교할 경우 네 번째 값이 다르게 되어 관찰 결과 네 번째 스캔 체인 셀 (24)에 0 고정(Stuck - At 0) 오류가 있음을 알 수 있으며 이는 올바른 검사 결과이다.
- <31>     도 4는 본 발명에 따른 스캔 체인 검사 방법을 검사순서에 따라 나타낸 순서도이다.
- <32>     도 4를 참조하면, S1 단계에서는 기능 로직에 기본 입력치를 입력한다.
- <33>     S2 단계에서는 기본 입력치에 응답하여 기능 로직은 시뮬레이션을 수행하고 기본 입력치에 대응하는 시뮬레이션 결과값을 출력한다. 이 경우 스캔 모드가 아닌 정상 모드 상태에 있으므로 출력된 시뮬레이션 결과값은 각 스캔 체인 셀에 병렬로 저장된다. 따라서 종래의 기술이 스캔 인을 통해 직렬로 검사 데이터를 저장하는 방법에 비해 검사 시간이 적게 걸린다.
- <34>     S3 단계에서는 각 스캔 체인 셀에 저장된 값을 출력하기 위하여 스캔 모드를 선택하는데 이 경우 스캔 체인 셀을 구성하는 멀티플렉서를 이용한다.
- <35>     S4 단계에서는 스캔 체인 셀을 구성하는 플립플롭에 인가되는 클럭에 동기되어 스캔 체인 셀에 저장된 데이터가 순차로 스캔 아웃을 통하여 출력된다. 이 동작은 스캔 체인 셀의 개수만큼의 클럭 인가로 이루어진다. 실시예에서는 8개의 스캔 체인 셀이 있으므로 8개의 클럭이 필요하다.
- <36>     S5 단계에서는 모든 스캔 체인 셀의 데이터가 출력되었는지를 판단한다.
- <37>     S6 단계에서는 스캔 아웃 기대값과 스캔 아웃 출력값을 비교하는데 두 값이 동일할 경우 S7 단계에서 스캔 체인에 오류가 없는 것으로 판정하지만 두 값이 동일하지 않을 경우 S8 단계에서 스캔 체인의 어느 부분에 오류가 있는지 진단(Diagnosis)한다.

<38> 본 발명에 따른 검사 방법을 상기한 설명 및 도면에 따라 도시하였지만, 이는 예를 들어 설명한 것에 불과하며 본 발명의 기술적 사상 및 범위를 벗어나지 않는 범위 내에서 다양한 변화 및 변경이 가능함은 물론이다.

**발명의 효과**

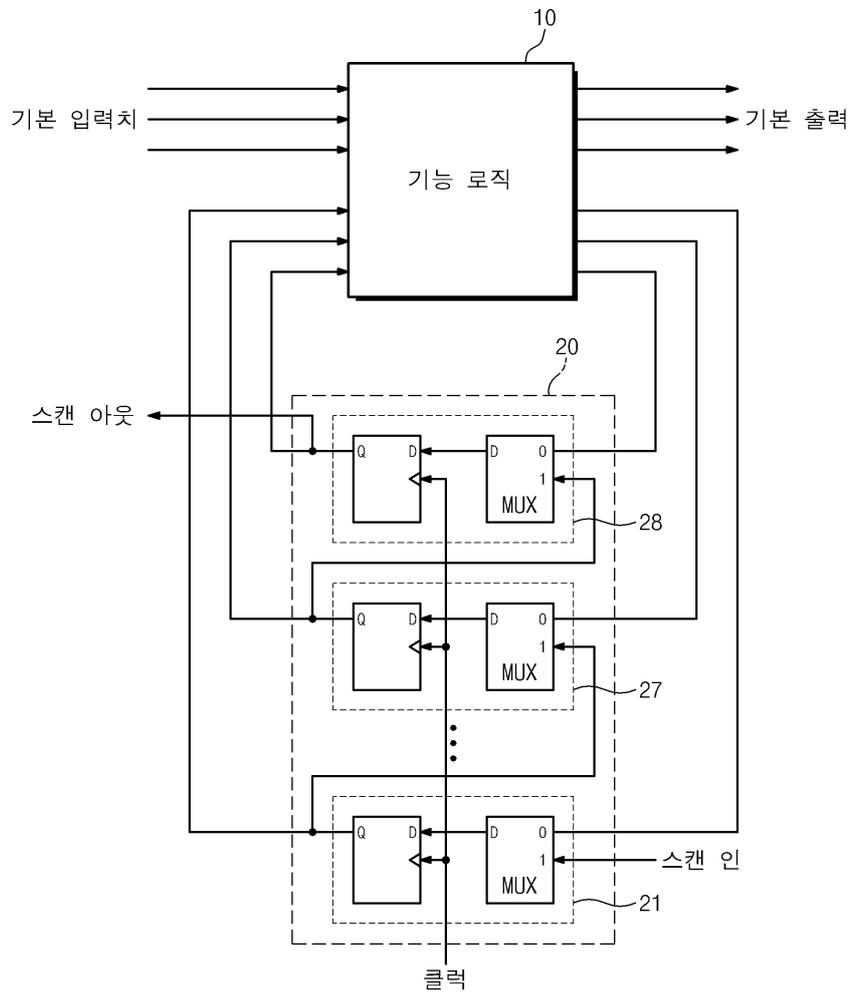
- <39> 상술한 바와 같이 스캔 체인에 오류가 있는지를 검사하는데 있어서,
- <40> 검사 데이터를 스캔 인을 통해 직렬로 입력하지 않고 기능 로직을 이용하여 병렬로 입력함으로써 스캔 체인 내의 오류로 인하여 스캔 체인 검사시 잘못된 판정을 하는 것을 막아 스캔 체인 검사의 신뢰도가 향상된다.
- <41> 또한 스캔 체인에 초기값을 입력하는 종래의 방식에 비하여 본원 발명은 초기값을 입력하는 과정이 필요 없으므로 검사시 소요되는 시간이 짧다.
- <42> 그리고 기존의 스캔 체인 검사 기법들 중에는 Dictionary를 이용하여 이를 출력값과 비교하여 순위를 따져 분석하는 기법이 있는데 본원발명은 별도의 Dictionary가 필요 없으므로 소요되는 메모리가 적다.

**도면의 간단한 설명**

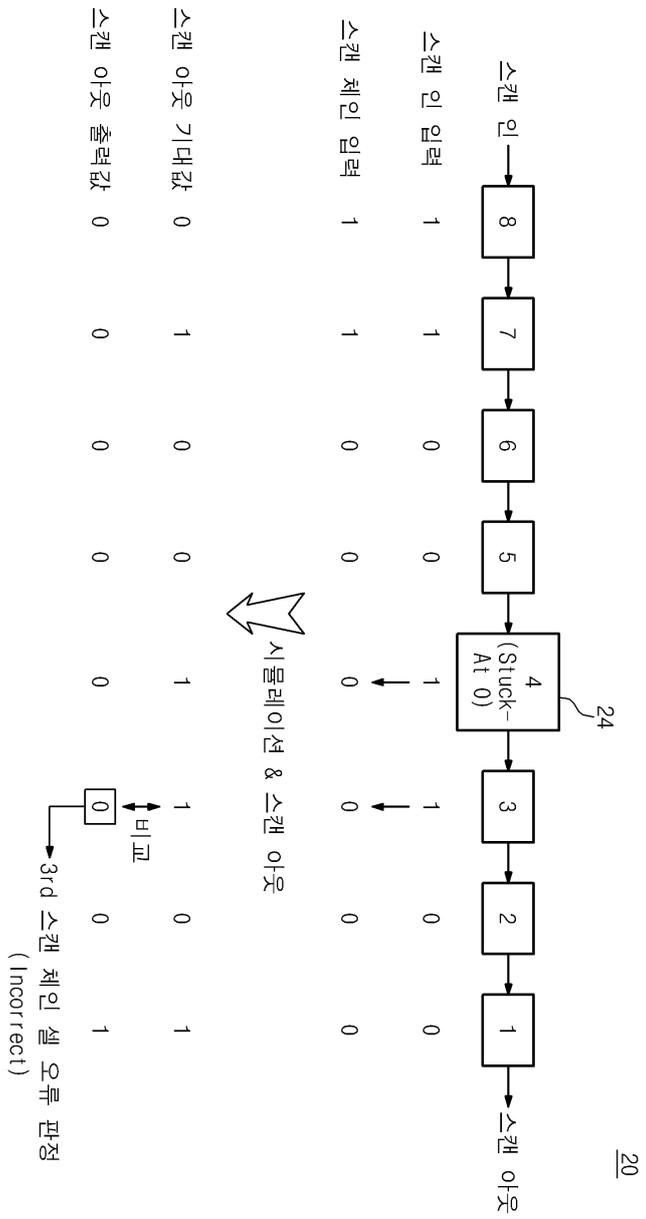
- <1> 도 1은 일반적인 기능 로직과 스캔 체인의 연결관계를 보여주는 도면;
- <2> 도 2는 일반적인 스캔 체인 검사 방법을 예시적으로 보여주는 도면;
- <3> 도 3은 본 발명에 따른 스캔 체인 검사 방법을 예시적으로 보여주는 도면;
- <4> 도 4는 본 발명에 따른 스캔 체인 검사 방법을 순서에 따라 보여주는 순서도이다.

도면

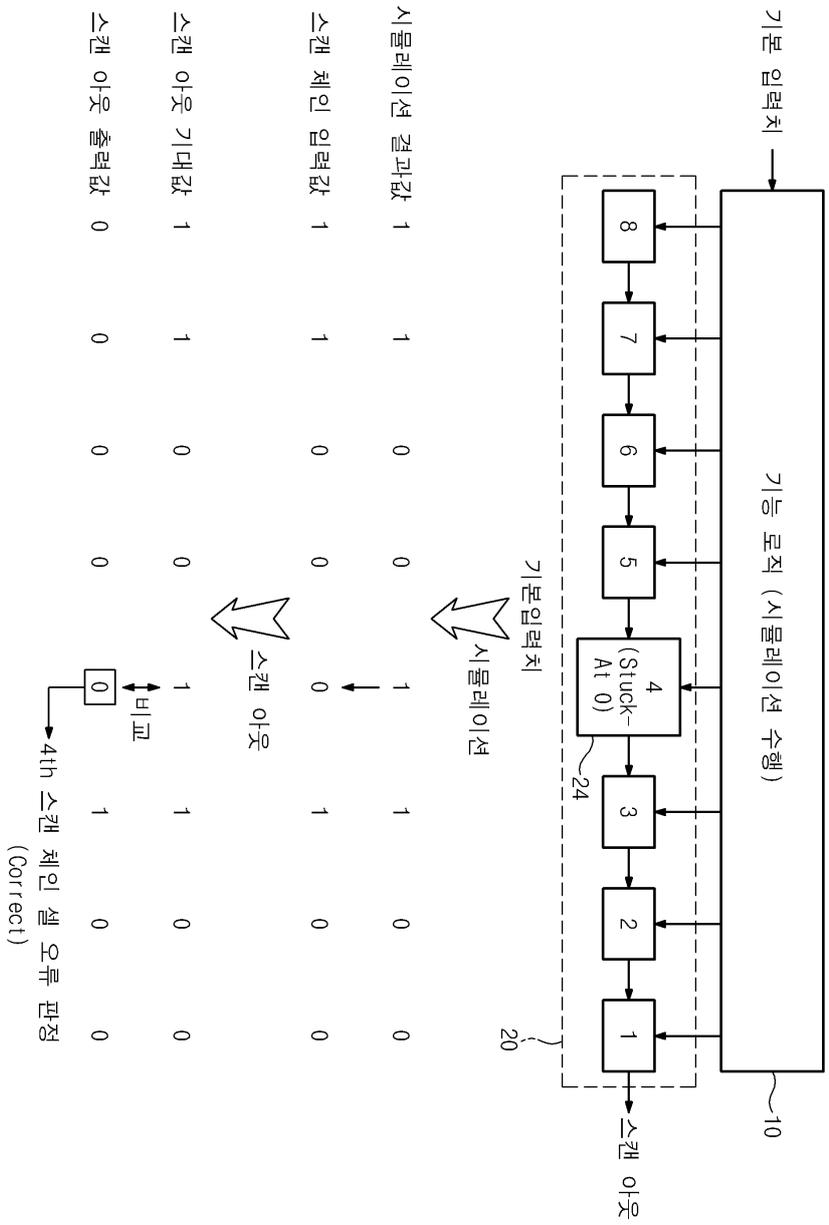
도면1



도면2



도면3



도면4

