



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년02월12일

(11) 등록번호 10-1492743

(24) 등록일자 2015년02월05일

(51) 국제특허분류(Int. Cl.)

G06F 11/25 (2006.01)

(21) 출원번호 10-2013-0162712

(22) 출원일자 2013년12월24일

심사청구일자 2013년12월24일

(56) 선행기술조사문헌

KR100749753 B1*

KR1019920008506 A*

KR1020070063291 A*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

서울과학기술대학교 산학협력단

서울특별시 노원구 공릉로 232 (공릉동, 서울과학기술대학교)

(72) 발명자

이승은

서울 강서구 허준로 23, 106동 601호 (가양동, 한강타운아파트)

정영섭

강원도 춘천시 우석로 15

이성모

서울 도봉구 덕릉로60나길 4-8, 301 (창동)

(74) 대리인

김정현

전체 청구항 수 : 총 1 항

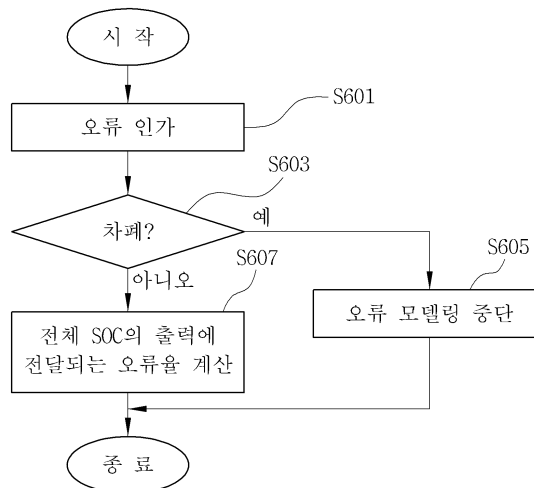
심사관 : 이정은

(54) 발명의 명칭 SoC에서의 게이트 레벨 오류 모델링 방법

(57) 요약

본 발명은 SoC 오류 모델링 방법에 관한 것으로서, 본 발명은 둘 이상의 게이트가 연결되어 구성되는 SoC(System on Chip)에서의 오류 모델링 방법에 있어서, 게이트 레벨에서의 오류 정보를 도출하는 단계 및 도출된 오류 정보를 기반으로 SoC의 게이트간 연결정보를 이용하여 전체 SoC 출력의 고장 유무를 판단하는 단계를 포함한다. 본 발명에 의하면 내고장형 SoC 설계 및 검증을 위해 필수적인 오류 모델을 개발하는데 있어서, 게이트 레벨에서 오류를 모델링하고 그에 대한 정확성을 갖는 오류 분석 방법을 제안함으로써, 보다 효율적인 오류 모델링을 수행할 수 있는 효과가 있다.

대표도 - 도6



특허청구의 범위

청구항 1

둘 이상의 게이트가 연결되어 구성되는 SoC(System on Chip)에서의 오류 모델링 방법에 있어서,

게이트 레벨에서의 오류 정보를 도출하는 단계;

도출된 오류 정보를 기반으로 SoC의 게이트간 연결정보를 이용하여 전체 SoC 출력의 고장 유무를 판단하는 단계;

상기 SoC 전체 출력의 고장 유무를 판단하기 위해, SoC의 연결망 정보와, 연결 게이트의 게이트 레벨에서의 오류 정보를 이용하여, 오류가 차폐되는지를 분석하는 단계; 및

상기 차폐 동작을 통해 모든 오류가 차폐되면 오류 모델링을 중단하는 단계를 포함하고,

상기 게이트 레벨에서의 오류 정보를 도출하는 단계에서, 각 게이트의 입력에 오류 신호를 인가하여 출력에 오류가 발생하는지 여부를 분석하는 방식으로 오류 정보를 도출하며,

상기 SOC의 내부 블록의 입력이나 블록 내부의 게이트 입력을 통해 오류를 인가하고, 상기 게이트의 출력에 오류가 발생하는 경우에 각 입력과, 각 입력의 오류 발생 상태를 나타낸 표를 참조하여 해당 게이트에서의 입력에 따른 출력의 오류 유무를 판단하고, 오류 유무가 판단된 게이트의 출력 정보를 바탕으로 블록의 최종 출력값과 오류 유무를 도출하는 것을 특징으로 하는 SoC에서의 오류 모델링 방법.

청구항 2

삭제

청구항 3

삭제

명세서

기술분야

[0001] 본 발명은 시스템반도체(SoC)의 신뢰성 향상을 위해 고장 허용 기능 검증을 위한 오류 모델링 방법에 있어서, 게이트 레벨에서 특정 입력의 변화에 따른 내부 도선들의 상태변화를 분석하는 방법과 입력 포트나 내부 게이트 입력(node)에 원하는 여러 종류의 오류를 주입하여 SoC의 고장을 모델링 하고, 모든 오류가 차폐되었을 때 오류 분석을 중단하여 분석 시간을 줄이는 방법에 관한 것이다.

배경기술

[0002] 반도체 제조 공정 기술의 발전으로 고집적도 구현이 가능해짐에 따라 여러 가지 반도체 부품, 예를 들어, 프로세서, 메모리, 주변 장치 등을 하나의 칩에 구현하는 시스템 온 칩(System on Chip, SoC)이 제안되고 있다.

[0003] SoC는 전체 시스템을 칩 하나에 담은 반도체를 말하는 것으로서, 연산 기억 데이터 전환 소자 등 주요 반도체 소자가 하나의 칩에 구현되는 기술을 의미한다. 즉, 컴퓨터 중앙처리장치(CPU), 디지털 신호처리 칩(DSP), 마이크로 컨트롤러(MCU) 등을 하나의 반도체 다이에 통합하여, 칩 자체가 하나의 시스템이 되도록 하는 것이다. 이처럼 여러가지 기능을 가진 반도체가 하나의 칩으로 통합되면 보드 공간이 크게 줄어들어 시스템 크기가 대폭 줄어들게 되어서, 각종 전자시스템들의 크기를 축소시킬 수 있다. 또한 여러 개의 반도체를 별도로 만드는 것에 비해 반도체 제조비용이 훨씬 저렴해지고 전체 시스템 가격도 낮아진다.

[0004] 따라서 모든 부품 기능을 하나의 칩에 집적하는 SoC 기술은 고성능, 저비용, 소형화로 집약되는 첨단 디지털시대의 핵심 부품기술로 떠오르고 있다. 이러한 SoC에 대한 지속적인 성능 향상으로 하나의 칩에 포함되는 반도체 부품의 수가 점차적으로 증가하고 있어서, SoC의 결함 여부를 검출하기 위한 테스트의 중요성이 대두되고 있다.

- [0005] 공정기술의 발달에 따라서 더 작은 전력만으로 더 빠르게 동작하는 디지털 회로들의 기능에 문제가 생기는 신뢰성 문제가 대두되었다. 이를 해결하고자 구현된 여러 정정 방법에 점점 많은 비용이 소모됨에 따라서, 시스템 반도체 설계자들은 더 이상 성능과 저전력 요소뿐만 아니라 내고장형 기능을 디자인 요소로서 고려하기 시작했다.
- [0006] 고장의 원인에는 여러 가지가 있으며 주로 제품의 제조과정이나 이후의 환경적인 요인에 의해 시스템 기능의 결함(failure)이 발생한다. 제품이 사용됨에 따라 반도체의 노후화로 인한 스위칭 타이밍 문제가 발생하거나, 우주에서 날아오는 광선(Cosmic ray)이 대기 중에서 충돌하여 회로에 데이터 에러를 유발하는 알파입자(alpha particle)가 생성된다. 그 밖에도 여러 도선 사이의 거리가 좁아짐에 따라 전기적인 간섭이 일어나는 도선간섭(crosstalk)과, 방사선 붕괴(radioactive decay)에서 방출되는 다양한 방사선, 반도체의 문턱전압에 영향을 주는 온도 잡음(Thermal Noise) 등은 전체 시스템의 오동작을 심화시킨다.
- [0007] SoC는 Cosmic Ray Particle, 전원 잡음, crosstalk, 등의 환경 변화에 의해 고장을 초래할 수 있으며, 디지털 회로를 파괴하지 않으면서 일시적으로 정상 동작을 방해하는 오류를 소프트 오류(Soft Error)라 한다.
- [0008] 1962년 cosmic ray particle이 회로에 오류를 일으킬 수 있다고 예견되었으며, 1975년 처음으로 cosmic ray에 의한 회로의 오동작(failure)이 보고되었다. 1978년 지표면에서 SRAM의 오류가 관측되어 이를 해결하고자 하는 연구와, 오류 모델링이 시작되었다.
- [0009] SoC에 있어서, 전원 잡음(Voltage Drop) 및 radiation은 오류율을 증가시키는데 상당히 기여하고 있으며, 온도 잡음(Thermal Noise) 또한 회로의 정상 동작을 저해하는 오류의 원인이다. 이외 다양한 외부 환경의 변화는 SoC 반도체의 정상 동작을 저해하며 오류를 발생시킨다.
- [0010] 이와 같이 다양한 원인에 기인한 회로의 오류는 SoC의 정상 동작을 방해하여 실제 고장(failure)으로 나타날 수도 있으며, 또는 회로 내부에서 차폐되어 정상동작될 수도 있다.
- [0011] 도 5는 SoC에서 회로 내부의 차폐(Masking) 관련 회로를 보여주는 도면이다. 도 5에서 (a)는 Logic Masking이고, (b)는 Temporal Masking이고, (c)는 Electrical Masking이다.
- [0012] 도 5에서 보는 바와 같이, SoC에서 발생하는 회로의 오류는 회로 내부에서 Logic Masking, Temporal Masking, 또는 Electrical Masking 으로 차폐되어 정상 동작될 수도 있다.
- [0013] 이와 같은 여러 원인들로 인하여 회로에는 많은 종류의 고장이 발생된다. 하드웨어의 고장에는 얼마나 회로에 존재하는지에 따라서 영구적인(permanent), 간헐적인(intermittent), 일시적인(transient) fault로 나눌 수 있다. 이러한 고장들은 회로에 악영향을 끼쳐 실제 데이터가 바뀌는 데이터 에러(error)를 일으킨다. 그 중에 소프트에러(soft error)는 주변 환경적인 요인에 의해 하드웨어에 손상을 주지 않고 데이터만 변화하는 에러이다. 이러한 회로의 오류를 파악하고 대처하고자 오류가 언제, 어디서, 얼마나 발생하는지 그리고 오류에 따라서 시스템에 어떻게 반응하는지에 대한 모델링이 필요하다.
- [0014] SoC에 오류가 발생하는 빈도, 위치, 및 시간에 대한 모델링 뿐만 아니라, 단일 또는 다수의 오류로 인하여 SoC에서 발생하는 고장(failure) 여부에 대한 모델링이 요구된다.
- [0015] 회로(Circuit) 레벨에서의 모델링은 상용 시뮬레이터를 사용하여 오류 모델링을 수행하기 용이하며, 시뮬레이션 환경에서 회로 내부의 노드 값들을 변경 및 모니터링하는데 용이한 반면, 회로 레벨의 오류 모델을 기반으로 상대적으로 복잡한 SoC의 내고장형 설계 방법의 효율성을 검증하는데 시간이 많이 요구된다.
- [0016] 게이트 레벨에서의 오류 모델링은 SoC에 사용되는 게이트의 종류별로 오류 모델링을 수행하여, 각각의 게이트의 오류 모델링을 시뮬레이션할 때 사용하거나, 분석적인 방법으로 도출하는데 용이하며, SoC의 넷리스트(netlist)에 적용하여 fault injection 플랫폼 개발이 상대적으로 쉬운 편이나, SoC 레벨에서 내고장형 설계 방법의 효율성을 검증하는데 시간이 오래 걸린다.
- [0017] 칩 레벨에서의 오류 모델링은 상위 레벨에서의 모델링을 수행함으로써 분석적인 방법을 기반으로 하여 상위 레벨에서의 시뮬레이션으로 모델링이 가능하고, 또한 도출된 모델이 상대적으로 간단하여 오류검증 플랫폼의 개발 및 내고장형 설계 방법의 검증 시간이 적게 걸리는 반면, 칩 레벨에서의 오류 유무를 판단하여 SoC를 구성하는 내부 블록들의 오류에 대한 동작을 분석하기에는 어려움이 있다.
- [0018] 모델링 방법은 회로에 접근하는 수준에 따라서 system, RTL, gate, switch, physical 등 여러 가지 방법으로 분류할 수가 있다. 각 방법은 정확성, 복잡도 등의 차이가 있으며 시스템 수준에서부터 반도체수준까지 자세히

접근할수록 시스템의 신뢰성 검증 시간이 오래 걸린다. 따라서 설계자는 디자인 사양과 상황에 맞는 분석 방법을 선택하여 회로의 내고장 기능을 검증해야한다. 그 중에서 게이트 수준(gate level)의 모델링 방법은 반도체의 동작까지 고려하는 스위치 수준(switch level) 방법보다 빠르고, 시스템과 RTL 수준의 모델링 방법보다 정확한 장점을 지닌다.

선행기술문헌

특허문헌

[0019] (특허문헌 0001) 한국공개특허 10-2011-0071254

발명의 내용

해결하려는 과제

[0020] 본 발명은 상기와 같은 문제점을 해결하기 위하여 안출된 것으로서, 시스템반도체(SoC)의 신뢰성 향상을 위해 고장 허용 기능 검증에 사용되는 오류 모델을 개발하는데 있어서, 게이트 레벨에서 특정 입력의 변화에 따른 내부 도선들의 상태변화를 분석하는 방법과 모든 오류가 차폐되었을 때 오류 분석을 중단하여 시간을 줄이는 방법, 전체 시스템에 적합한 오류 정정 수단을 유도하는 방법을 제안하는데 그 목적이 있다.

[0021] 본 발명의 목적은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0022] 이와 같은 목적을 달성하기 위한 본 발명은 둘 이상의 게이트가 연결되어 구성되는 SoC(System on Chip)에서의 오류 모델링 방법에 있어서, 게이트 레벨에서의 오류 정보를 도출하는 단계 및 도출된 오류 정보를 기반으로 SoC의 게이트간 연결정보를 이용하여 전체 SoC 출력의 고장 유무를 판단하는 단계를 포함한다.

[0023] 상기 SoC 출력의 고장 유무를 판단하기 위하여, 회로 내부 또는 입력에 오류를 인가하고, 연결망 정보를 이용하여 인가된 오류가 차폐되어 전체 SoC의 출력에 영향을 주지 않으면, 상기 인가된 오류에 대한 오류 모델링을 중단하여 오류 모델링 시간을 단축시키고, 차폐되지 않은 경우에는 전체 SoC의 출력에 전달되는 오류율을 계산하여 SoC의 오류율을 모델링 하는 것을 포함한다.

[0024] 상기 게이트 레벨에서의 오류 정보를 도출하는 단계에서, 각 게이트의 입력에 오류 신호를 인가하여 출력에 오류가 발생하는지 여부를 분석하는 방식으로 오류 정보를 도출할 수 있다.

발명의 효과

[0025] 본 발명에 의하면 내고장형 SoC 설계 및 검증을 위해 필수적인 오류 모델을 개발하는데 있어서, 게이트 레벨에서 오류를 모델링하고 그에 대한 정확성을 갖는 오류 분석 방법을 제안함으로써, 보다 효율적인 오류 모델링을 수행할 수 있는 효과가 있다.

[0026] 시스템 반도체에 사용되는 논리게이트의 종류는 제한적이며, 게이트의 입력에 대한 출력은 진리표를 통하여 정확히 표현이 가능하므로, 본 발명에서 각 게이트에 대한 오류 분석 및 디지털 회로에서 게이트 간 입출력의 연관성을 포함하는 오류 모델링은 라이브러리 형태로 정형화할 수 있다.

[0027] 본 발명의 오류 모델링 방법은 정형화된 라이브러리를 이용하여 오류를 검출함에 있어서, 정확성과 신뢰성을 제공하며, 검증 모의실험 시간을 단축할 수 있는 효과가 있다.

[0028] 본 발명의 게이트 레벨 오류 모델링 방법은 SoC의 오류회피 및 복구 방법에 대한 신뢰성을 검증하기 위한 시스템의 오류 검증 플랫폼에 사용될 수 있다.

도면의 간단한 설명

- [0029] 도 1은 본 발명의 일 실시예에 따른 AND 게이트에서 입력에 따라 오류가 발생하는 것을 설명하기 위한 예시도이다.
- 도 2는 본 발명의 일 실시예에 따른 OR 게이트에서 입력에 따라 오류가 발생하는 것을 설명하기 위한 예시도이다.
- 도 3은 본 발명의 일 실시예에 따른 시스템 내부 블록의 오류 모델링 방법을 예시하기 위한 논리 게이트를 도시한 도면이다.
- 도 4는 본 발명의 일 실시예에 따른 SoC 시스템의 게이트 레벨 오류 분석 방법을 예시하기 위한 블록도이다.
- 도 5는 SoC에서 회로 내부의 차폐(Masking) 관련 회로를 보여주는 도면이다.
- 도 6은 본 발명의 일 실시예에 따른 SoC에서의 오류 모델링 방법을 보여주는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0030] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.
- [0031] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0032] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 갖고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 갖는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0033] 또한, 첨부 도면을 참조하여 설명함에 있어, 도면 부호에 관계없이 동일한 구성 요소는 동일한 참조부호를 부여하고 이에 대한 중복되는 설명은 생략하기로 한다. 본 발명을 설명함에 있어서 관련된 공지 기술에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우 그 상세한 설명을 생략한다.
- [0034] 본 발명은 시스템 온 칩(System on Chip, SoC)에서의 오류 모델링 방법에 관한 것이다.
- [0035] 본 발명은 SoC를 구성하는 게이트 레벨에서의 오류 모델링을 수행하고, 이들 게이트들의 연결망 정보를 이용하여 분석하여 전체 SoC의 오류 모델을 완성한다.
- [0036] 본 발명의 SoC에서의 오류 모델링 방법은 둘 이상의 게이트가 연결되어 구성되는 SoC(System on Chip)에서의 오류 모델링 방법에 있어서, 먼저 게이트 레벨에서의 오류 정보를 도출하는 단계와, 도출된 오류 정보를 기반으로 SoC의 게이트간 연결정보를 이용하여 전체 SoC 출력의 고장 유무를 판단하는 단계를 포함한다. 본 발명에서 각 게이트의 입력에 오류 신호를 인가하여 출력에 오류가 발생하는지 여부를 분석하는 방식으로 오류 정보를 도출할 수 있다.
- [0037] 도 6은 본 발명의 일 실시예에 따른 SoC에서의 오류 모델링 방법을 보여주는 흐름도이다.
- [0038] 도 6을 참조하면, SoC 출력의 고장 유무를 판단하기 위하여, SoC 회로 내부 또는 SoC 입력에 오류를 인가한다(S601).
- [0039] 다음, 연결망 정보를 이용하여 인가된 오류가 차폐되는지 여부를 확인한다(S603).
- [0040] 인가된 오류가 차폐되면, 전체 SoC의 출력에 영향을 주지 않기 때문에 인가된 오류에 대한 오류 모델링을 중단하여 오류 모델링 시간을 단축시킨다(S605).

- [0041] 인가된 오류가 차폐되지 않은 경우, 전체 SoC의 출력에 전달되는 오류율을 계산하여 SoC의 오류율을 모델링한다(S607).
- [0042] 본 발명은 SoC의 게이트 레벨의 오류 모델링을 수행하기 위해, 내부 게이트 연결 소자들의 위치 정보를 기반으로 전체 SoC 회로의 오류 모델을 분석적인 방법으로 구현한다.
- [0043] 도 1은 본 발명의 일 실시예에 따른 AND 게이트에서 입력에 따라 오류가 발생하는 것을 설명하기 위한 예시도이다.
- [0044] 도 1을 참조하면, AND 게이트의 입력은 i 와 j 이고 출력은 o 이다.
- [0045] o_f 는 출력 o 의 오류 발생 유무를 나타내며, 출력 o 이 오류를 가지면 1이 되고 정상 출력일 때는 0이 된다. 마찬가지로 i_f 와 j_f 은 입력 i 와 j 의 오류 발생 유무를 나타낸다. 즉, 입력 i 와 j 에 오류가 없으면 i_f 와 j_f 은 0이고, 오류가 있으면 i_f 와 j_f 은 1이 된다.
- [0046] 도 1에서 오른쪽 표를 보면, AND 게이트에서의 출력에 오류가 발생하는 경우(=1), 이에 해당하는 입력과 입력의 오류 발생 상태를 나타내고 있다.
- [0047] 도 2는 본 발명의 일 실시예에 따른 OR 게이트에서 입력에 따라 오류가 발생하는 것을 설명하기 위한 예시도이다.
- [0048] 도 2를 참조하면, OR 게이트에서 입력은 i 와 j 이고 출력은 o 이다.
- [0049] o_f 는 출력 o 의 오류 발생 유무를 나타내며, 출력 o 이 오류를 가지면 1이 되고 정상 출력일 때는 0이 된다. 마찬가지로 i_f 와 j_f 은 입력 i 와 j 의 오류 발생 유무를 나타낸다. 즉, 입력 i 와 j 에 오류가 없으면 i_f 와 j_f 은 0이고, 오류가 있으면 i_f 와 j_f 은 1이 된다.
- [0050] 마찬가지로 도 2에서 오른쪽 표를 보면 OR 게이트에서의 출력에 오류가 발생하는 경우(=1), 이에 해당하는 입력과 입력의 오류 발생 상태를 나타낸다.
- [0051] 이제, 본 발명에서 게이트 정보를 이용한 오류 정보를 도출하는 방법을 자세히 설명하면 다음과 같다.
- [0052] i_f 와 j_f 은 입력 i 와 j 의 오류 발생 유무를 나타내므로 $i_f=0$ 와 $j_f=0$ 이면 입력 i 와 j 는 정상상태이고, 반대로 $i_f=1$ 와 $j_f=1$ 이면 입력 i 와 j 는 오류가 발생한 상태가 된다.
- [0053] 이와 같은 방식으로 오류가 있는 게이트 입력 값을 반전시키면 정상적인 입력 값을 알 수가 있다. 예를 들어, 도 1에서 만약 게이트로 들어오는 입력이 $i=1$, $j=1$ 이고 그 정보가 $i_f=0$ 와 $j_f=1$ 이라는 것은 원래의 정상적인 입력 신호는 $i=1$, $j=0$ 이라는 의미가 된다.
- [0054] 도 1의 표를 참조해보면 $i=1$ 와 $j=0$ 이고 $i_f=0$ 와 $j_f=1$ 일 때, AND 게이트 출력은 오류가 발생하는 경우이다($o_f=1$). 발생한 오류는 다른 게이트 입력으로 계속 전파(propagation)된다.
- [0055] 도 2에서, $i=1$, $j=1$ 이고 $i_f=0$ 와 $j_f=1$ 이라면 원래의 정상적인 입력 신호는 $i=1$, $j=0$ 이다. 따라서 도 2에서 $i=1$, $j=0$ 일 때 $i_f=0$ 와 $j_f=1$ 이면 OR 게이트 출력에 오류가 발생하지 않는 경우이다($o_f=0$).
- [0056] 이러한 방법을 통하여 본 발명에서는 SoC 회로 내에서 데이터의 값과 그 오류 정보를 계속 추적할 수가 있다.
- [0057] 본 발명의 일 실시예에서 NOT 게이트의 경우 입력에 오류가 있을 경우 출력으로 오류가 모두 전달된다. 이처럼 본 발명에서 SoC를 구성하는 다른 모든 게이트들도 도 1 및 도 2와 같은 표를 작성하여 설명할 수 있다.
- [0058] 이러한 게이트의 오류 발생 정보를 이용하여 본 발명에서는 SoC 시스템 내부 블록의 오류 모델링 방법을 구현할 수 있다.
- [0059] 도 3은 본 발명의 일 실시예에 따른 시스템 내부 블록의 오류 모델링 방법을 예시하기 위한 논리 게이트를 도시

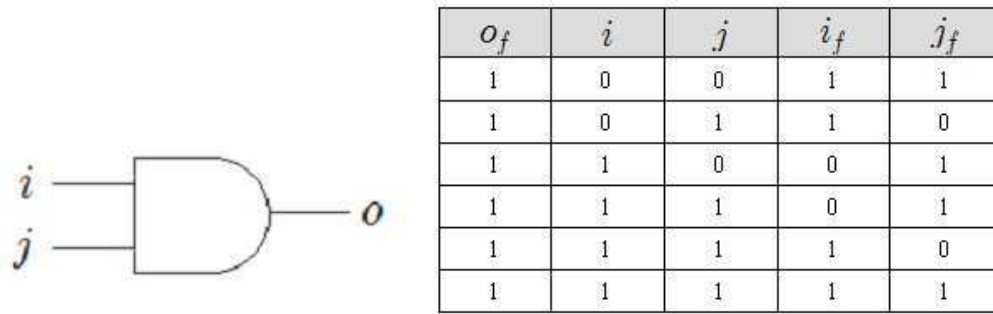
한 도면이다.

- [0060] 도 3은 전체 SoC 중 하나의 서브 블록만을 나타낸 실시예이다. 이 블록의 입력은 I_1 과 I_2 이다. I_{1f} 와 I_{2f} 는 각 입력들의 오류 발생 유무를 나타낸다.
- [0061] 블록 내부는 여러 게이트로 이루어져 있으며, 각 게이트의 연결은 a, b, c, d, e의 연결망을 가지고 있다. 이 블록의 출력은 O_1 과 O_2 이며, 일정한 데이터 폭을 갖는다. O_{1f} 와 O_{2f} 는 각 출력들의 오류 발생 유무를 나타낸다. 본 발명의 일 실시예에서 7개의 게이트와 2개의 블록 입출력 포트를 사용하여 설명하나, 본 발명에서 게이트의 개수와 종류, 그리고 블록 입출력 개수는 제한하지 않는다.
- [0062] 본 발명에서 게이트 레벨 오류 테스트를 위해 오류 모델을 완성하고자 블록에 오류를 인가하고 출력의 오류 발생 유무를 확인한다.
- [0063] 도면 3에서 게이트 레벨의 오류 테스트를 위한 과정은 다음과 같다.
- [0064] 순서 1. 블록의 입력이나 블록 내부의 게이트 입력을 통해 오류를 인가한다.
- [0065] 순서 2. 도1 및 도 2에 나타난 표를 참조하여 해당 게이트에서 입력에 따른 출력의 오류 유무를 판단한다.
- [0066] 순서 3. 확인된 게이트의 출력 정보를 바탕으로 블록의 최종 출력 값과 오류 유무를 도출한다.
- [0067] 순서 1에서 오류 주입은 블록의 입력 포트나 내부 게이트 입력(node)을 대상으로 주입되며, 원하는 여러 종류의 오류를 주입함으로써 SoC 시스템에 여러 가지 고장이 발생하는 경우를 모델링할 수 있다. 예를 들어 트랜지트 폴트(transient fault)를 모델링하기 위해서는 임의의 위치와 시간에 오류를 주입하고, 인터미턴트 폴트(intermittent fault)는 동일한 장소에 간헐적으로 에러를 주입하고, 그리고 퍼머넌트 폴트(permanent fault)는 동일한 장소에 영구적으로 에러를 주입한다. 그러면 시스템은 해당 폴트(fault)로 인한 데이터 오류를 가지게 되어, 특정 에러로 인한 반응 정보를 기대할 수 있게 된다.
- [0068] 오류가 주입되면, 순서 2와 같이 게이트의 오류 정보를 가지는 도 1 및 2에 도시된 표를 참조하여 해당 게이트 출력의 오류 유무를 판단하게 된다. 예를 들어 도면 3의 블록 입력 I_1 과 I_2 는 AND 게이트를 거쳐 출력 a를 얻게 되고, 입력 I_1 과 I_2 에 오류가 주입되면 도 1의 AND 게이트 정보를 근거로 출력 a의 오류 정보를 구할 수가 있게 된다. 동일한 방법으로 도 3의 게이트 출력 b, c, d, e는 입력 포트나 노드의 오류 신호가 전파되는지 판단하여 각각의 오류 정보를 구할 수 있다.
- [0069] 한편, SoC 시스템에서 오류는 로직 마스킹(Logic Masking), 템포럴 마스킹(Temporal Masking), 또는 일렉트릭얼 마스킹(Electrical Masking) 등으로 인해 차폐될 수 있다. 따라서 본 발명에서는 SoC 시스템의 게이트에 대한 모든 오류가 차폐되었을 경우, 전체 오류 모델링 과정을 중단하고, 다음 오류 입력을 준비하도록 한다. 이를 통해 본 발명에서는 전체 검증 모의 실험시간을 단축시킬 수 있다.
- [0070] 마지막으로 순서 3을 통해, 블록의 최종 출력 값과 그 오류의 유무를 알려주는 O_{1f} 와 O_{2f} 를 확인함으로써, 시스템의 출력에 오류가 발생했는지 여부를 알 수 있다.
- [0071] 도 4는 본 발명의 일 실시예에 따른 SoC 시스템의 게이트 레벨 오류 분석 방법을 예시하기 위한 블록도이다.
- [0072] 도 4는 서브 블록에서의 출력 오류 도출 방법을 예시한 것으로서, 내부 블록들의 오류 모델을 이용하여 게이트 레벨에서 전체 SoC의 오류 분석을 수행한다.
- [0073] 도 4를 참조하면 각 서브 블록에서 입력되는 오류들은, 전체 SoC 시스템에 입력된 오류 또는 이전 블록에서 발생한 출력의 오류가 전파(propagation)되는 오류 신호들에 기인한다. 본 발명에서는 이러한 오류 분석을 통해 각 블록에서의 오류율을 비교하여, 오류가 자주 발생하는 블록의 오류 정정 기법의 규모를 판단하여 신뢰도 향상을 위한 비용을 절감할 수 있다.
- [0074] 이상 본 발명을 몇 가지 바람직한 실시예를 사용하여 설명하였으나, 이들 실시예는 예시적인 것이며 한정적인 것이 아니다. 본 발명이 속하는 기술분야에서 통상의 지식을 지닌 자라면 본 발명의 사상과 첨부된 특허청구범

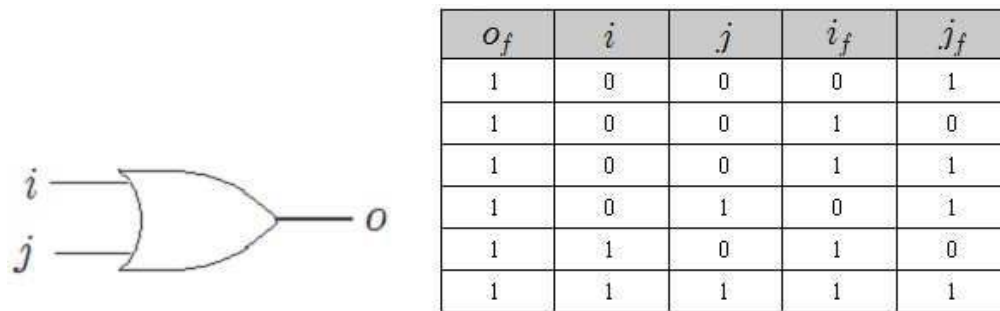
위에 제시된 권리범위에서 벗어나지 않으면서 다양한 변화와 수정을 가할 수 있음을 이해할 것이다.

도면

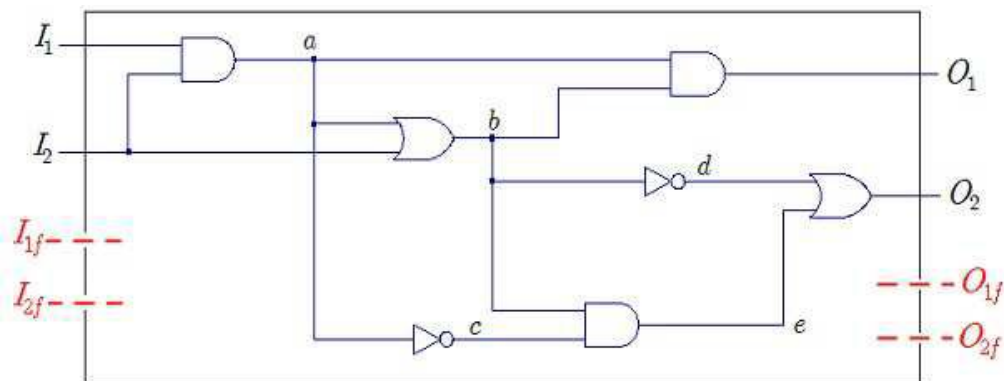
도면1



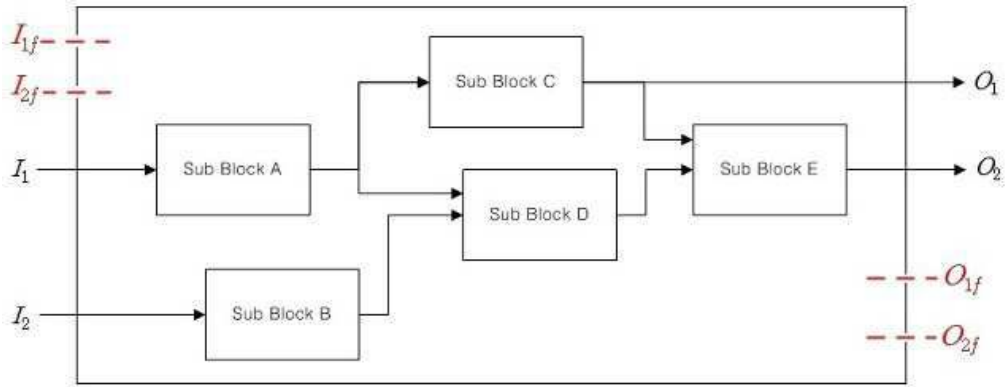
도면2



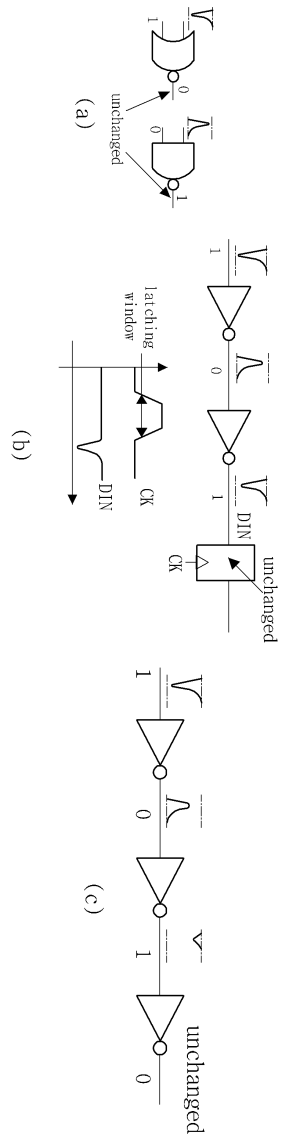
도면3



도면4



도면5



도면6

