



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년08월17일

(11) 등록번호 10-1544649

(24) 등록일자 2015년08월07일

(51) 국제특허분류(Int. Cl.)
 G06F 19/00 (2011.01) G06F 11/00 (2006.01)
 (21) 출원번호 10-2013-0167254
 (22) 출원일자 2013년12월30일
 심사청구일자 2013년12월30일
 (65) 공개번호 10-2015-0078134
 (43) 공개일자 2015년07월08일
 (56) 선행기술조사문헌
 KR101297484 B1*
 JP평성05052912 A
 JP1993052912 A
 KR1020110071254 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 6 항

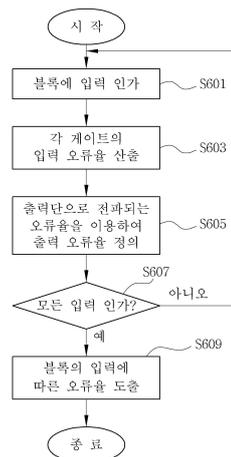
심사관 : 박재용

(54) 발명의 명칭 SoC에서의 오류율 분석 방법

(57) 요약

본 발명은 시스템반도체(SoC)의 고장 허용 기능 검증을 통한 신뢰성 향상을 위해, 본 발명은 둘 이상의 게이트가 연결되어 구성되는 하나 이상의 내부 블록을 포함하는 SoC(System on Chip)에서의 오류율 분석 방법에 있어서, 어느 한 내부 블록의 입력단에 입력신호를 인가하는 단계, 상기 내부 블록을 구성하는 각 게이트의 입력 오류율을 정의하는 단계 및 상기 각 게이트의 입력 오류율과 출력단으로 전파되는 오류율을 이용하여 상기 내부 블록의 출력 오류율을 정의하는 단계를 포함한다. 본 발명에 의하면 내고장형 SoC 설계 및 검증을 위해 필수적인 오류 모델을 개발하는데 있어서, 게이트 레벨에서 입력의 오류율에 따른 출력의 오류율의 변화를 분석하는 방법을 제시함으로써, 각 게이트에 대한 오류 분석 및 디지털 회로에서 게이트 간 입출력의 연관성을 포함하는 오류율 정보 모델을 라이브러리 형태로 정형화할 수 있는 효과가 있다.

대표도 - 도6



명세서

청구범위

청구항 1

오류율 분석 장치에 의한 둘 이상의 게이트가 연결되어 구성되는 하나 이상의 내부 블록을 포함하는 SoC(System on Chip)에서의 오류율 분석 방법에 있어서,

어느 한 내부 블록의 입력단에 입력신호를 인가하는 단계;

상기 내부 블록을 구성하는 각 게이트의 입력단의 입력 오류율을 정의하는 단계; 및

상기 각 게이트의 입력단의 입력 오류율과 상기 각 게이트의 출력단으로 전파되는 오류율을 이용하여 상기 입력 신호에 대응하는 상기 내부 블록의 최종 출력단의 출력 오류율을 정의하는 단계를 포함하고,

방사선 동위 원소에서 방출되는 α 입자, 열 잡음, 중성자, 외부환경 변화를 포함하는 요인으로 인해 게이트 입력단이나 게이트 내부에서 발생하는 오류율을 소프트 오류율이라고 할 때, 상기 각 게이트의 입력단의 입력 오류율은, 상기 각 게이트의 입력단(x)의 이전 출력에서 전파된 오류율($P_p(x)$)과 상기 각 게이트의 입력단의 소프트 오류율(γ)의 합이며,

상기 각 게이트의 출력단으로 전파되는 오류율은, 상기 각 게이트의 입력단에 상기 입력신호가 입력된 경우 상기 각 게이트의 기능에 따른 상기 각 게이트의 출력단의 오류 발생 확률로서, 상기 각 게이트의 입력단의 입력 오류율을 상기 각 게이트의 기능에 따라 기설정된 오류 발생 확률 산출용 수식에 대입하여 산출되는 것인 SoC에서의 오류율 분석 방법.

청구항 2

청구항 1에 있어서,

상기 내부 블록의 출력 오류율을 정의하는 단계 후에,

가능한 모든 경우의 입력신호를 순차적으로 인가하여 상기 내부 블록의 출력 오류율을 정의하는 과정을 반복 수행하는 단계; 및

정의된 출력 오류율을 분석하여 상기 내부 블록의 입력에 따른 오류율을 도출하는 단계를 더 포함하는 것을 특징으로 하는 SoC에서의 오류율 분석 방법.

청구항 3

청구항 1에 있어서,

상기 각 게이트가 AND 게이트이면, 그 입력신호 (i, j)가 각기 (0, 0), (0, 1), (1, 0), (1, 1)일 때의 상기 각 게이트의 출력단으로 전파되는 오류율 $P_p(o)$ 은, 각기

$$P(i)*P(j), P(i)*(1-P(j)), (1-P(i))*P(j), 1-\{(1-P(i))*(1-P(j))\},$$

여기서, $P(i)=P_p(i)+\gamma_{11}, P(j)=P_p(j)+\gamma_{12}$ 이며,

상기 각 게이트가 OR 게이트이면, 그 입력신호 (i, j)가 각기 (0, 0), (0, 1), (1, 0), (1, 1)일 때의 상기 각 게이트의 출력단으로 전파되는 오류율 $P_p(o)$ 은, 각기

$$1-\{(1-P(i))*(1-P(j))\}, (1-P(i))*P(j), P(i)*(1-P(j)), P(i)*P(j)$$

여기서, $P(i)=P_p(i)+\gamma_{21}, P(j)=P_p(j)+\gamma_{22}$ 이며,

상기 각 게이트가 NOT 게이트이면, 그 입력신호 i 의 값에 상관없이 상기 각 게이트의 출력단으로 전파되는 오류

율 $P_p(o)$ 은, $P_p(o) = P(i) = P_p(i)+\gamma_3$ 이며,

여기서, $\gamma_{11}, \gamma_{12}, \gamma_{21}, \gamma_{22}, \gamma_3$ 은, 상기 각 게이트가 각기 AND, OR 및 NOT 게이트일 때에 상기 각 게이트의 입력단이나 상기 각 게이트의 내부에서 발생하는 소프트 오류율인 SoC에서의 오류율 분석 방법.

청구항 4

청구항 1에 있어서,

상기 내부 블록의 출력 오류율을 정의하는 단계에서,

상기 내부 블록을 구성하는 게이트의 연결망 정보를 기반으로 게이트 간에 오류율이 전파되는 정보를 이용하여 상기 내부 블록의 출력 오류율을 정의하는 것을 특징으로 하는 SoC에서의 오류율 분석 방법.

청구항 5

청구항 1에 있어서,

시뮬레이션을 수행하여 각 게이트의 입력 통계 정보를 분석하고, 상기 입력 통계 정보를 이용하여 각 게이트의 입력 오류율 및 출력 오류율을 모델링하는 것을 특징으로 하는 SoC에서의 오류율 분석 방법.

청구항 6

청구항 1에 있어서,

각 게이트의 입력단에 인가되는 입력신호가 동일한 빈도로 나타나는 것을 가정하여 각 게이트의 입력 오류율 및 출력 오류율을 모델링하는 것을 특징으로 하는 SoC에서의 오류율 분석 방법.

발명의 설명

기술 분야

[0001] 본 발명은 시스템반도체(SoC)의 고장 허용 기능 검증을 통한 신뢰성 향상을 위해, 특정 게이트 입력의 오류율이 게이트 출력의 오류율에 미치는 영향, 시스템 블록의 입력이나 블록 내부에 오류가 인가되었을 때 게이트 간에 오류율이 전파되는 과정, 그리고 전체 시스템에 적합한 오류 정정 수단을 유도하는 방법에 관한 것이다.

배경 기술

[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 결합 여부를 검출하기 위한 테스트의 중요성이 대두되고 있다.
- [0005] 공정기술의 발달에 따라서 디지털 회로들의 기능에 문제가 생기는 신뢰성 문제가 대두되었다. 이를 해결하고자 구현된 에러 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] SoC에서 발생하는 회로의 오류는 회로 내부에서 Logic Masking, Temporal Masking, 또는 Electrical Masking 등의 요인으로 차폐되어 정상 동작될 수도 있다.
- [0012] 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.
- [0013] SoC에 오류가 발생하는 빈도, 위치, 및 시간에 대한 모델링 뿐만 아니라, 단일 또는 다수의 오류로 인하여 SoC에서 발생하는 고장(failure) 여부에 대한 모델링이 요구된다.
- [0014] 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.
- [0015] 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.
- [0016] 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.
- [0017] 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

[0018] (특허문헌 0001) 한국공개특허 10-2011-0071254

발명의 내용

해결하려는 과제

[0019] 본 발명은 상기와 같은 문제점을 해결하기 위하여 안출된 것으로서, 본 발명은 시스템반도체(SoC)의 고장 허용 기능 검증을 통한 신뢰성 향상을 위해, 특정 게이트 입력의 오류율 및 게이트 내부의 오류율이 게이트 출력의 오류율에 미치는 영향을 분석하고, 시스템 반도체의 게이트 연결망 정보를 이용하여 전체 시스템 반도체의 오류율을 분석하는 방법을 제안하는데 그 목적이 있다.

[0020] 본 발명의 목적은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0021] 이와 같은 목적을 달성하기 위한 본 발명은 둘 이상의 게이트가 연결되어 구성되는 하나 이상의 내부 블록을 포함하는 SoC(System on Chip)에서의 오류율 분석 방법에 있어서, 어느 한 내부 블록의 입력단에 입력신호를 인가하는 단계, 상기 내부 블록을 구성하는 각 게이트의 입력 오류율을 정의하는 단계 및 상기 각 게이트의 입력 오류율과 출력단으로 전파되는 오류율을 이용하여 상기 내부 블록의 출력 오류율을 정의하는 단계를 포함한다.

[0022] 상기 내부 블록의 출력 오류율을 정의하는 단계 후에, 가능한 모든 경우의 입력신호를 순차적으로 인가하여 상기 내부 블록의 출력 오류율을 정의하는 과정을 반복 수행하는 단계 및 정의된 출력 오류율을 분석하여 상기 내부 블록의 입력에 따른 오류율을 도출하는 단계를 더 포함할 수 있다.

[0023] 방사선 동위 원소에서 방출되는 α 입자, 열 잡음, 중성자, 외부환경 변화를 포함하는 요인으로 인해 게이트 입력단이나 게이트 내부에서 발생하는 오류율을 소프트 오류율이라고 할 때, 각 게이트에서 입력 오류율은 이전 출력에서 전파된 오류율과 소프트 오류율의 합으로 나타낼 수 있다.

[0024] 상기 내부 블록의 출력 오류율을 정의하는 단계에서, 상기 내부 블록을 구성하는 게이트의 연결망 정보를 기반으로 게이트 간에 오류율이 전파되는 정보를 이용하여 상기 내부 블록의 출력 오류율을 정의할 수 있다.

[0025] 본 발명의 일 실시예에서 시뮬레이션을 수행하여 각 게이트의 입력 통계 정보를 분석하고, 상기 입력 통계 정보를 이용하여 각 게이트의 입력 오류율 및 출력 오류율을 모델링할 수 있다.

[0026] 또는 본 발명의 다른 실시예에서 각 게이트의 입력단에 인가되는 입력신호가 동일한 빈도로 나타나는 것을 가정하여 각 게이트의 입력 오류율 및 출력 오류율을 모델링할 수 있다.

발명의 효과

[0027] 본 발명에 의하면 내고장형 SoC 설계 및 검증을 위해 필수적인 오류 모델을 개발하는데 있어서, 게이트 레벨에서 입력의 오류율에 따른 출력의 오류율의 변화를 분석하는 방법을 제시함으로써, 각 게이트에 대한 오류 분석

및 디지털 회로에서 게이트 간 입출력의 연관성을 포함하는 오류율 정보 모델링을 라이브러리 형태로 정형화할 수 있는 효과가 있다.

[0028] 또한, 본 발명의 오류 모델링 방법은 정형화된 라이브러리를 이용하여 시스템 출력의 오류율을 분석하는데 정확성과 신뢰성을 제공하는 효과가 있다.

[0029] 또한, 본 발명에 의하면 오류가 자주 발생하는 블록의 오류 정정 기법의 규모를 판단하여 신뢰도 향상을 위한 비용을 절감할 수 있는 효과가 있다.

도면의 간단한 설명

[0030] 도 1은 본 발명의 일 실시예에 따른 AND 게이트에서 입력의 오류율이 출력의 오류율에 전파되는 것을 설명하기 위한 예시도이다.

도 2는 본 발명의 일 실시예에 따른 OR 게이트에서 입력의 오류율이 출력의 오류율에 전파되는 것을 설명하기 위한 예시도이다.

도 3은 본 발명의 일 실시예에 따른 NOT 게이트에서 입력의 오류율이 출력의 오류율에 전파되는 것을 설명하기 위한 예시도이다.

도 4는 본 발명의 일 실시예에 따른 게이트 레벨에서의 SoC 내부 블록의 오류율 분석 방법을 예시하기 위한 블록도이다.

도 5는 본 발명의 일 실시예에 따른 전체 SoC의 게이트 레벨 오류 분석 방법을 예시하기 위한 블록도이다.

도 6은 본 발명의 일 실시예에 따른 SoC에서의 오류율 분석 방법을 보여주는 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0031] 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.

[0032] 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.

[0033] 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.

[0034] 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.

[0035] 본 발명은 둘 이상의 게이트가 연결되어 구성되는 하나 이상의 내부 블록을 포함하는 SoC(System on Chip)에서의 오류율 분석 방법에 관한 것이다.

[0036] 본 발명은 SoC의 게이트 레벨 오류율 분석을 수행하기 위해, 내부 게이트 연결 소자들의 연결 정보를 바탕으로 전체 회로의 게이트 오류 모델을 분석적인 방법으로 구현한다.

[0037] 도 6은 본 발명의 일 실시예에 따른 SoC에서의 오류율 분석 방법을 보여주는 흐름도이다.

[0038] 도 6을 참조하면, SoC의 어느 한 내부 블록의 입력단에 입력신호를 인가한다(S601).

- [0039] 그리고, 내부 블록을 구성하는 각 게이트의 입력 오류율을 정의한다(S603).
- [0040] 다음, 각 게이트의 입력 오류율과 출력단으로 전파되는 오류율을 이용하여 내부 블록의 출력 오류율을 정의한다(S605).
- [0041] 다음, 가능한 모든 경우의 입력신호가 인가되었는지 여부를 확인하고(S607), 모든 경우의 입력신호가 인가될 때까지 순차적으로 입력신호를 인가하여 내부 블록의 출력 오류율을 정의하는 과정을 반복 수행한다(S607).
- [0042] 모든 경우의 입력신호가 인가되면, 정의된 출력 오류율을 분석하여 내부 블록의 입력에 따른 오류율을 도출한다(S609).
- [0043] 방사선 동위 원소에서 방출되는 α 입자, 열 잡음, 중성자, 외부환경 변화를 포함하는 요인으로 인해 게이트 입력단이나 게이트 내부에서 발생하는 오류율을 소프트 오류율이라고 할 때, 각 게이트에서 입력 오류율은 이전 출력에서 전파된 오류율과 소프트 오류율의 합으로 나타낼 수 있다.
- [0044] 본 발명의 일 실시예에서 S605 단계에서 내부 블록을 구성하는 게이트의 연결망 정보를 기반으로 게이트 간에 오류율이 전파되는 정보를 이용하여 내부 블록의 출력 오류율을 정의할 수 있다.
- [0045] 본 발명의 일 실시예에서 시뮬레이션을 수행하여 각 게이트의 입력 통계 정보를 분석하고, 입력 통계 정보를 이용하여 각 게이트의 입력 오류율 및 출력 오류율을 모델링할 수 있다.
- [0046] 본 발명의 다른 실시예에서 각 게이트의 입력단에 인가되는 입력신호가 동일한 빈도로 나타나는 것을 가정하여 각 게이트의 입력 오류율 및 출력 오류율을 모델링할 수 있다.
- [0047] 도 1은 본 발명의 일 실시예에 따른 AND 게이트에서 입력의 오류율이 출력의 오류율에 전파되는 것을 설명하기 위한 예시도이다.
- [0048] 도 1을 참조하면, AND 게이트의 입력은 i 와 j 이고 출력은 o 이며, γ 는 방사선 동위 원소에서 방출되는 α 입자, 열 잡음, 중성자, 외부환경 변화 등으로 인해 게이트 입력단 혹은 게이트 내부에서 발생하는 소프트 오류율이다.
- [0049] 본 발명에서 γ 는 게이트 내부에서 발생하는 소프트 오류율로 정의되며, 관련 수학식에서 편의상 사용된다. 예를 들어, AND 게이트의 입력 i 와, j 의 내부 소프트 오류율은 동일한 γ 로 표시되지만, 다를 수 있으며, 소프트 오류율이 같은 경우는 특수한 경우라고 할 수 있다. γ 에 관한 구체적인 설명은 관련 수학식이 나오는 부분에서 후술하기로 한다.
- [0050] $P(i)$ 는 입력 i 의 오류 발생 확률이며, 이전 출력에서 전파된 오류율 $P_p(i)$ 와 해당하는 게이트 내부에 발생하는 소프트 오류율 γ 의 합으로 나타낸다.
- [0051] 또한 $P(j)$ 는 입력 j 의 오류 발생 확률이며, 이전 출력에서 전파된 오류율 $P_p(j)$ 와 해당하는 게이트의 소프트 오류율 γ 의 합으로 나타낸다.
- [0052] $P(o)$ 는 입력 i 와 j 에 따른 출력 o 의 오류 발생 확률을 나타낸다.
- [0053] 도 1에 도시된 표에 있는 AND 게이트의 입력 i 와 j 에 따른 출력 o 의 오류 발생 확률을 도출하는 방법을 상세히 설명하면 다음과 같다.
- [0054] AND 게이트에서 입력 i, j 가 각각 $i=0, j=0$ 일 때 정상적인 출력은 $o=0$ 이고, 비정상 출력에 해당하는 $o=1$ 이 나오는 경우는 입력 i, j 에서 모두 오류가 발생하는 경우이다. 따라서 출력 o 의 오류 발생 확률 $P(o)$ 는 $P(i) \times P(j)$ 로 나타낸다.
- [0055] AND 게이트에서 입력 i, j 가 각각 $i=1, j=1$ 일 때, 정상적인 출력은 $o=1$ 이며 비정상 출력에 해당하는 $o=0$ 이 나오는 경우는 입력 i, j 이 모두 오류가 발생하지 않는 경우를 제외한 모든 경우이다.
- [0056] 따라서 출력 o 의 오류 발생 확률 $P(o)$ 는 $1 - \{(1 - P(i)) \times (1 - P(j))\}$ 로 나타낸다. 같은 방법으로 입력 i, j 가 $i=0, j=1$ 일 경우와 $i=1, j=0$ 일 경우 $P(o)$ 는 각각 $P(i) \times (1 - P(j))$, $(1 - P(i)) \times P(j)$ 으로 나타낸다.
- [0057] 도 2는 본 발명의 일 실시예에 따른 OR 게이트에서 입력의 오류율이 출력의 오류율에 전파되는 것을 설명하기

위한 예시도이다.

- [0058] 도 2를 참조하면, AND 게이트의 경우와 동일하게 입력은 i 와 j 이고 출력은 o 이다. 마찬가지로 $P(i)$ 는 입력 i 의 오류 발생 확률이며, 이전 출력에서 전파된 오류율 $P_p(i)$ 와 해당하는 게이트의 소프트 오류율 γ 의 합으로 나타낸다.
- [0059] $P(j)$ 는 입력 j 의 오류 발생 확률이며, 이전 출력에서 전파된 오류율 $P_p(j)$ 와 해당하는 게이트의 소프트 오류율 γ 의 합으로 나타낸다. $P(o)$ 는 입력 i 와 j 에 따른 출력 o 의 오류 발생 확률을 나타낸다.
- [0060] 도 2에 도시된 표에 있는 OR 게이트의 입력 i 와 j 에 따른 출력 o 의 오류 발생 확률을 도출하는 방법을 자세히 설명하면 다음과 같다.
- [0061] OR 게이트에서 입력 i, j 가 각각 $i=0, j=0$ 일 때, 정상적인 출력은 $o=0$ 이고, 비정상 출력에 해당하는 $o=1$ 이 나오는 경우는 입력 i, j 에서 모두 오류가 발생하지 않는 경우를 제외한 모든 경우이다. 따라서 출력 o 의 오류 발생 확률 $P(o)$ 는 $1 - \{(1 - P(i)) \times (1 - P(j))\}$ 로 나타낸다.
- [0062] OR 게이트에서 입력 i, j 가 각각 $i=1, j=1$ 일 때, 정상적인 출력은 $o=1$ 이며 비정상 출력에 해당하는 $o=0$ 이 나오는 경우는 입력 i, j 이 모두 오류가 발생하는 경우이다. 따라서 출력 o 의 오류 발생 확률 $P(o)$ 는 $P(i) \times P(j)$ 로 나타낸다. 같은 방법으로 입력 i, j 가 $i=0, j=1$ 일 경우와 $i=1, j=0$ 일 경우 $P(o)$ 는 각각 $(1 - P(i)) \times P(j)$, $P(i) \times (1 - P(j))$ 으로 나타낸다.
- [0063] 도 3은 본 발명의 일 실시예에 따른 NOT 게이트에서 입력의 오류율이 출력의 오류율에 전파되는 것을 설명하기 위한 예시도이다.
- [0064] $P(i)$ 는 입력 i 의 오류 발생 확률이며 이전 출력에서 전파된 오류율 $P_p(i)$ 와 해당하는 게이트의 소프트 오류율 γ 의 합으로 나타낸다. 입력 i 가 $P(i)$ 로 오류가 발생할 경우, 출력 o 의 오류율은 $P(o) = P(i)$ 의 식을 따른다.
- [0065] 도 3에 도시된 표에 있는 NOT 게이트의 입력 i 와 j 에 따른 출력 o 의 오류 발생 확률을 도출하는 방법을 자세히 설명하면 다음과 같다.
- [0066] NOT 게이트의 경우 입력과 출력이 각각 하나이므로 입력 단에서 오류가 발생하면 출력 단에서도 오류가 발생하는 특징을 갖는다. 따라서 NOT 게이트에서 출력 o 의 오류 발생 확률 $P(o)$ 는 입력 i 에서 오류 발생 확률 $P(i)$ 로 나타낸다.
- [0067] 이외에 SoC를 구성하는 나머지 NAND 게이트, NOR 게이트, XOR 게이트도 도 1, 도 2 및 도 3과 같은 표를 작성하여 설명할 수 있다.
- [0068] 도 4는 본 발명의 일 실시예에 따른 게이트 레벨에서의 SoC 내부 블록의 오류율 분석 방법을 예시하기 위한 블록도이다.
- [0069] 도 4는 전체 SoC 중 하나의 내부 블록(100)만을 나타낸 실시예이다. 이 블록(100)의 입력은 i 와 j 이다.
- [0070] 도 4를 참조하면, 블록(100) 내부는 여러 게이트로 이루어져 있으며, 각 게이트의 연결노드는 각각의 오류율을 가지고 있다. 이 블록(100)의 출력은 o 이다.
- [0071] $P(o)$ 는 출력의 오류 발생 확률을 나타낸다.
- [0072] 도 4에서 편의상 5개의 게이트와 2개의 블록 입력 포트 그리고 1개의 출력 포트를 사용하여 설명하나, 본 발명에서 게이트의 개수와 종류 그리고 블록 입출력 개수는 제한되지 않는다.
- [0073] 게이트 레벨 오류율 분석을 위해 오류 모델을 완성하고자 블록에 입력을 인가하고 출력의 오류율을 확인한다.
- [0074] 오류율 분석을 위해 입력으로 $i=0, j=0$ 을 인가한다.
- [0075] 외부 블록의 출력에서 전파되어 입력 i, j 로 유입되는 오류율과 OR 게이트, NOT 게이트의 입력단자 또는 게이트 소자에서 발생하는 소프트 에러율을 고려하여 $P(a), P(b), P(c)$ 를 정의한다.
- [0076] $P(a), P(b), P(c)$ 는 다음과 같다.

- [0077] $P(a)=P_p(a)+\gamma$ (1)
- [0078] $P(b)=P_p(b)+\gamma$ (2)
- [0079] $P(c)=P_p(c)+\gamma$ (3)
- [0080] 여기서 AND 게이트의 입력 i ,와 j 의 내부 소프트 오류율을 다룰 수 있으며 γ 로 표시되었지만 실제로는 γ_1 , γ_2 이다. 입력 i 와 j 의 소프트 오류율이 같은 경우가 특별한 경우라고 할 수 있다.
- [0081] 또한, 도 4에 도시된 회로에서 각각의 게이트마다 γ 값이 다를 수 있다.
- [0082] 도 2 및 도 3에서 도시된 OR 게이트와 NOT 게이트의 입력 오류율이 출력으로 전파되는 오류율에 대한 수식을 참조하여 오류율 $P_p(d)$, $P_p(e)$, $P_p(f)$ 를 다음과 같이 정의한다.
- [0083] $P_p(d)=1-[(1-(P_p(a)+\gamma))\times(1-(P_p(b)+\gamma))]$ (4)
- [0084] $P_p(e)=1-[(1-(P_p(a)+\gamma))\times(1-(P_p(b)+\gamma))]$ (5)
- [0085] $P_p(f)=P_p(c)+\gamma$ (6)
- [0086] 여기서 각 게이트에서 소프트 오류가 일어났을 경우를 가정하면, 입력 $i=0$, $j=0$ 에서 OR 게이트의 출력은 1이며 NOT 게이트의 출력은 0이다.
- [0087] 다음 게이트에 해당하는 NOT 게이트와 AND 게이트의 입력은 각각 $d=1$, $e=1$, $f=0$ 이다. 다시 NOT 게이트와 AND 게이트의 입력단자 또는 게이트 소자에서 발생하는 소프트 오류율을 고려하여 $P(d)$, $P(e)$, $P(f)$ 를 정의할 수 있으며 그 식은 다음과 같다.
- [0088] $P(d)=P_p(d)+\gamma$ (7)
- [0089] $P(e)=P_p(e)+\gamma$ (8)
- [0090] $P(f)=P_p(f)+\gamma$ (9)
- [0091] d, e, f 에 해당하는 입력과 NOT 게이트, AND 게이트의 입력오류율 및 출력으로 전파되는 오류율에 대한 수식을 참조하여 출력으로 전파되는 오류율 $P_p(g)$, $P_p(h)$ 을 판단하면 다음과 같이 정의할 수 있다.
- [0092] $P_p(g)=P_p(d)+\gamma$ (10)
- [0093] $P_p(h)=[1-(P_p(e)+\gamma)]\times[P_p(f)+\gamma]$ (11)
- [0094] 이상에서 설명한 바와 같이, 블록의 입력부터 순차적으로 게이트의 식을 대입하여 정리하면, 최종 출력단의 오류율을 정의할 수 있으며, 최종 출력단의 오류율 $P(o)$ 은 처음 입력으로부터 전파되는 $P_p(a)$, $P_p(b)$, $P_p(c)$ 와 내부의 각 게이트에서 정의한 소프트 오류율 γ 만으로 나타낼 수 있다.
- [0095] 본 발명에서는 도 4에서 도출된 오류율들을 분석하여 상대적으로 높은 오류율을 야기하는 입력에 대해서는 특별한 오류정정기법을 적용시키도록 유도할 수 있다.
- [0096] 이상에서 설명한 오류율 도출 방법은 각 게이트의 입력 조합에 따른 출력 오류율을 계산하는 방법으로서, 복잡한 SoC의 모델링의 경우 SoC의 입력 정보를 기반으로 모든 게이트의 입력 정보를 분석하여 모델링을 하는 것이 너무 복잡하고 물리적으로 불가능할 수도 있다. 이에 본 발명에서는 각 게이트의 오류 모델을 도출함에 있어, 시뮬레이션을 실행하고 각 게이트별로 입력 조합의 통계 정보를 이용하여 게이트의 모델을 완성하는 방법을 추가로 제안한다.
- [0097] 도 1에서 $P_p(o)$ 는 4가지의 가능한 입력에 따라 $P(i)*P(j)$, $P(i)*(1-P(j))$, $(1-P(i))*P(j)$, $1-((1-P(i))*(1-P(j)))$ 로 결정된다. 예를 들면, SoC의 게이트레벨 모델을 이용하여 시뮬레이션을 수행하여 특정 NAND게이트의

입력 통계 정보를 분석하고, 입력 i 와 j 의 빈도가 $i=0, j=0$ 이 x , $i=1, j=0$ 이 y , $i=0, j=1$ 이 z , $i=1, j=1$ 이 w 라고 하면 $x+y+z+w=1$ 이 만족된다.

[0098] 이때, 이 게이트의 오류율 $P_p(o) = x * P(i) * P(j) + y * P(i) * (1 - P(j)) + z * (1 - P(i)) * P(j) + w * 1 - \{(1 - P(i)) * (1 - P(j))\}$ 로 결정할 수 있다.

[0099] 본 발명의 다른 실시예에서 SoC 내부 게이트의 입력 통계정보를 분석하지 않고, 각 입력이 동일한 빈도로 나타나는 것을 가정하여 ($x=y=z=w=0.25$), 게이트의 오류율 $P_p(o) = \{P(i) * P(j) + P(i) * (1 - P(j)) + (1 - P(i)) * P(j) + 1 - \{(1 - P(i)) * (1 - P(j))\}\} / 4$ 로 결정하여, 모델링 시간 및 복잡성을 감소시킬 수 있다. 이 경우 회로 내의 같은 게이트는 같은 비율의 오류 전파율을 가지게 된다.

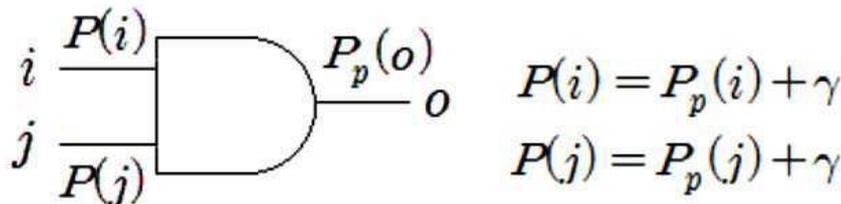
[0100] 도 5는 본 발명의 일 실시예에 따른 전체 SoC의 게이트 레벨 오류 분석 방법을 예시하기 위한 블록도이다.

[0101] 도 5는 전체 시스템의 게이트 레벨 오류율 분석 방법 예시를 나타낸다. 도 4에서 내부 블록들의 게이트 오류율 분석 방법을 이용하여, 시스템의 각 블록에서 발생하는 오류율들이 최종 출력으로 전파되는 과정을 파악하고 그 오류율들을 구할 수가 있다. 이를 통해 전체 시스템의 각 블록에서의 오류율을 비교하여, 오류가 자주 발생하는 블록의 오류 정정 기법의 규모를 판단하여 신뢰도 향상을 위한 비용을 절감할 수 있다.

[0102] 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 특허청구범위에 제시된 권리범위에서 벗어나지 않으면서 다양한 변화와 수정을 가할 수 있음을 이해할 것이다.

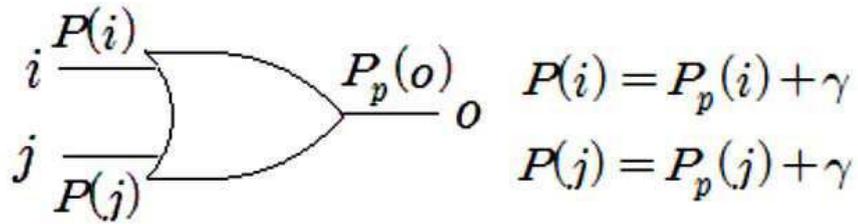
도면

도면1



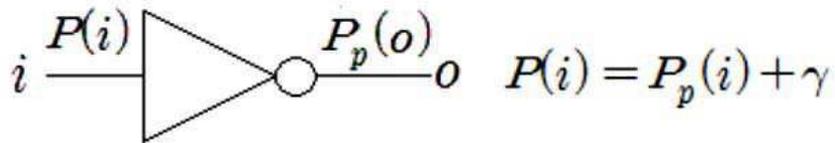
i	j	o	$P_p(o)$
0	0	0	$P(i) * P(j)$
0	1	0	$P(i) * (1 - P(j))$
1	0	0	$(1 - P(i)) * P(j)$
1	1	1	$1 - \{(1 - P(i)) * (1 - P(j))\}$

도면2



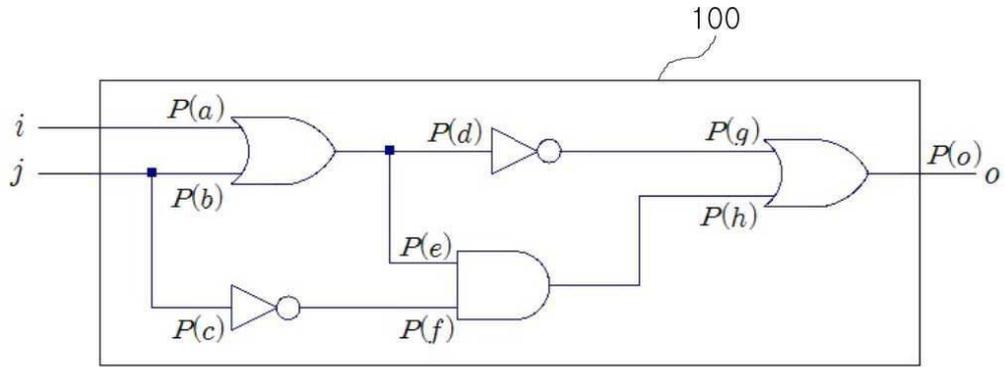
i	j	o	$P_p(o)$
0	0	0	$1 - \{(1 - P(i)) * (1 - P(j))\}$
0	1	1	$(1 - P(i)) * P(j)$
1	0	1	$P(i) * (1 - P(j))$
1	1	1	$P(i) * P(j)$

도면3

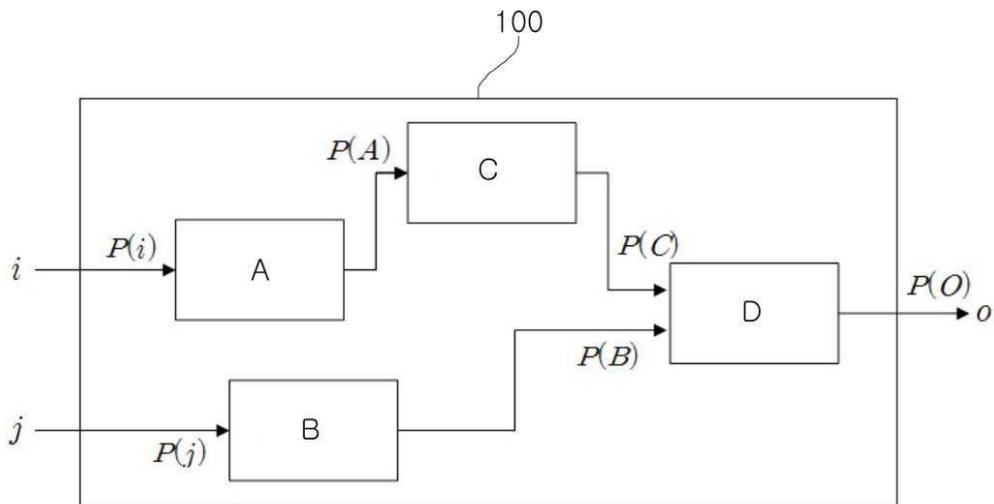


i	o	$P_p(o)$
0	1	$P(i)$
1	0	$P(i)$

도면4



도면5



도면6

