

	(19) 대한민국특허청(KR) (12) 공개특허공보(A)	(11) 공개번호 10-2013-0034713 (43) 공개일자 2013년04월08일
(51) 국제특허분류(Int. Cl.) G06F 11/10 (2006.01)		(71) 출원인 조선대학교산학협력단 광주광역시 동구 서석동 375 조선대학교 내
(21) 출원번호 10-2011-0098730		(72) 발명자 이정아 서울특별시 마포구 월드컵북로42가길 11, 203동 102호 (상암동, 휴먼시아) 소마순다람, 나타르잔 사이트 엔오. 10비 나가파 콜로니 제2크로스거리 카분담팔라얌, 코임바토레 641030 이정근 강원도 춘천시 서부대성로 34, 한신희플러스아파트 103동 1301호 (요선동)
(22) 출원일자 2011년09월29일 심사청구일자 2011년09월29일		(74) 대리인 특허법인아이엠

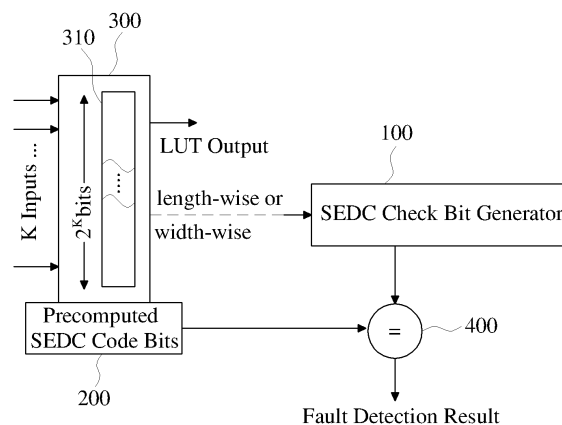
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 확장형 오류검출코드 생성기, 그 생성기를 구비한 자가검사 록업테이블 및 확장형 오류검출코드 생성방법

(57) 요약

본 발명은 확장형 오류검출코드 생성기, 그 생성기를 구비한 자가검사 록업테이블 및 확장형 오류검출코드 생성 방법에 관한 것으로, 보다 구체적으로는 서로 다른 길이의 이진 입력데이터에 대해 오류검출코드를 생성하는 복수 개의 오류검출코드 생성수단을 구비하여, 상기 오류검출코드 생성수단들 단독 또는 조합에 의해 오류검출코드의 길이를 유동적으로 확장할 수 있어, 최소한의 하드웨어 추가로 이진 입력데이터 전체의 단방향 오류를 검출할 수 있는 확장형 오류검출코드 생성기와 오류검출코드 생성방법 그리고 상기 확장형 오류검출코드 생성기를 구비하여 LUT데이터 전체를 자가검사하거나 매 동작시 LUT데이터를 자가검사할 수 있는 자가검사 록업테이블에 관한 것이다.

대표도 - 도10



이 발명을 지원한 국가연구개발사업

과제고유번호 2011-0017914

부처명 교과부

연구사업명 중견연구자지원사업(핵심연구)

연구과제명 오류 모듈에 안정적으로 대처하는 재구성 가능한 가상실험장치 구조 동시설계 연구

주관기관 조선대학교

연구기간 2010.05.01 ~ 2013.04.30

특허청구의 범위

청구항 1

오류검출의 대상이 되는 이진 입력데이터에 대해 생성될 확장형 오류검출코드(SED:scalable error detection coding)의 길이를 계산하는 코드 길이 계산모듈; 및

2비트 입력데이터에 대해 오류검출코드를 생성하는 2비트 입력 오류검출코드 생성수단, 3비트 입력데이터에 대해 오류검출코드를 생성하는 3비트 입력 오류검출코드 생성수단 또는 4비트 입력데이터에 대해 오류검출코드를 생성하는 4비트 입력 오류검출코드 생성수단을 포함하고, 상기 오류검출코드 생성수단들 단독 또는 조합에 의해 상기 코드 길이 계산모듈에서 계산된 길이의 확장형 오류검출코드를 생성하는 오류검출코드 생성모듈;을 포함하는 것을 특징으로 하는 확장형 오류검출코드 생성기.

청구항 2

제 1 항에 있어서,

상기 코드 길이 계산모듈:은

아래의 수학적 식 1을 이용하여 코드 길이의 계산을 위한 파라미터인 제1 파라미터(a) 및 제2 파라미터(b)를 계산하는 파라미터 계산수단; 및

아래의 수학적 식 2를 이용하여 상기 확장형 오류검출코드의 길이를 계산하는 코드 길이 계산수단;을 포함하는 것을 특징으로 하는 확장형 오류검출코드 생성기.

[수학적 식 1]

$$a = \frac{n - \max(b)}{3}$$

여기서, n은 이진입력데이터의 길이이고, a는 제1 파라미터로써 '0'에서부터 '∞'까지의 정수이고, b는 제2 파라미터로써 정수 '2', '3' 또는 '4'이고, max(b)는 정수 '2', '3' 및 '4' 중, 제1 파라미터가 정수가 되게 하는 최대값이다.

[수학적 식 2]

$$m = \lceil \log_2[n+1-(3 \times a)] \rceil + (2 \times a)$$

여기서, m은 확장형 오류검출코드의 길이이다.

청구항 3

제 1 항 또는 제 2 항에 있어서,

상기 2비트 입력 오류검출코드 생성수단(SED₂)은 아래의 수학적 식 3을 이용하여, 2비트 입력데이터(I₂)에 대해 2비트 오류검출코드(C₂)를 생성하되, 최상위 비트(C₂(1))는 상기 2비트 입력데이터(I₂)에서 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 생성하고, 그 외의 경우에는 '0'으로 생성하며, 최하위 비트(C₂(0))는 상기 2비트 입력데이터(I₂)에서 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성하며, 그 외의 경우에는 '1'로 생성하는 것을 특징으로 하는 확장형 오류검출코드 생성기.

[수학식 3]

$$C_2(1) = \begin{cases} 1, & \text{if number of 0's in } I_2 \geq \text{number of 1's in } I_2 \\ 0, & \text{otherwise} \end{cases}$$

$$C_2(0) = \begin{cases} 0, & \text{if number of 0's in } I_2 = \text{number of 1's in } I_2 \\ 1, & \text{otherwise} \end{cases}$$

청구항 4

제 3 항에 있어서,

상기 3비트 입력 오류검출코드 생성수단(SED_{C3})은 아래의 수학식 4를 이용하여, 3비트 입력데이터(I₃)에 대해 2비트 오류검출코드(C₃)를 생성하되, 상기 3비트 입력데이터(I₃)의 최상위 비트(I₃(2))가 '0'일 경우에는, 중간비트(I₃(1))와 최하위 비트(I₃(0))를 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단(SED_{C2})의 2비트 오류검출코드(C₂)와 동일하게 생성하고, 최상위 비트(I₃(2))가 '1'일 경우에는, 중간비트(I₃(1))와 최하위 비트(I₃(0))의 논리 부정 값을 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단(SED_{C2})의 2비트 오류검출코드(C₂)의 논리 부정 값으로 생성하는 것을 특징으로 하는 확장형 오류검출코드 생성기.

[수학식 4]

$$(C_3(1), C_3(0)) = \begin{cases} SEDC_2(I_3(1), I_3(0)), & \text{if } I_3(2) = 0 \\ 1's \text{ complement}(SEDC_2(I_3(1), I_3(0))), & \text{if } I_3(2) = 1 \end{cases}$$

청구항 5

제 4 항에 있어서,

상기 4비트 입력 오류검출코드 생성수단(SED_{C4})은 아래의 수학식 5를 이용하여, 4비트 입력데이터(I₄)에 대해 3비트 오류검출코드(C₄)를 생성하되, 상기 3비트 오류검출코드(C₄)의 중간비트(C₄(1))와 최하위 비트(C₄(0))는 상기 4비트 입력데이터(I₄)의 최상위 비트(I₄(3))를 제외한 세 비트(I₄(2), I₄(1), I₄(0))를 입력데이터로 하는 상기 3비트 입력 오류검출코드 생성수단(SED_{C3})의 2비트 오류검출코드(C₃)와 동일하게 생성하고, 최상위 비트(C₄(2))는 상기 4비트 입력데이터(I₄)의 최상위 비트(I₄(3))의 논리 부정 값으로 생성하는 것을 특징으로 하는 확장형 오류검출코드 생성기.

[수학식 5]

$$(C_4(1), C_4(0)) = SEDC_3(I_4(2), I_4(1), I_4(0))$$

$$C_4(2) = \overline{I_4(3)}$$

청구항 6

제 2 항에 있어서,

상기 2비트 입력 오류검출코드 생성수단(SED_{C2})은 2비트 입력데이터(I₂)에 대해 2비트 오류검출코드(C₂)를 생성하되, 최상위 비트(C₂(1))는 상기 2비트 입력데이터(I₂)에서 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '

1'로 생성하고, 그 외의 경우에는 '0'으로 생성하며, 최하위 비트($C_2(0)$)는 상기 2비트 입력데이터(I_2)에서 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성하며, 그 외의 경우에는 '1'로 생성하고,

상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)은 3비트 입력데이터(I_3)에 대해 2비트 오류검출코드(C_3)를 생성하되, 상기 3비트 입력데이터(I_3)의 최상위 비트($I_3(2)$)가 '0'일 경우에는, 중간 비트($I_3(1)$)와 최하위 비트($I_3(0)$)를 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단($SEDC_2$)의 2비트 오류검출코드(C_2)와 동일하게 생성하고, 최상위 비트($I_3(2)$)가 '1'일 경우에는, 중간 비트($I_3(1)$)와 최하위 비트($I_3(0)$)의 논리 부정 값을 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단($SEDC_2$)의 2비트 오류검출코드(C_2)의 논리 부정 값으로 생성하며,

상기 4비트 입력 오류검출코드 생성수단($SEDC_4$)은 4비트 입력데이터(I_4)에 대해 3비트 오류검출코드(C_4)를 생성하되, 상기 3비트 오류검출코드(C_4)의 중간 비트($C_4(1)$)와 최하위 비트($C_4(0)$)는 상기 4비트 입력데이터(I_4)의 최상위 비트($I_4(3)$)를 제외한 세 비트($I_4(2), I_4(1), I_4(0)$)를 입력데이터로 하는 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)의 2비트 오류검출코드(C_3)와 동일하게 생성하고, 최상위 비트($C_4(2)$)는 상기 4비트 입력데이터(I_4)의 최상위 비트($I_4(3)$)의 논리 부정 값으로 생성하며,

상기 오류 검출코드 생성모듈은 n비트 입력데이터에 대해 오류검출코드를 생성하는 n비트 입력 오류검출코드 생성수단($SEDC_n$)을 더 포함하고,

상기 n비트 입력 오류검출코드 생성수단($SEDC_n$)은 상기 오류검출코드 생성수단들($SEDC_2, SEDC_3, SEDC_4$) 중, 상기 제2 파라미터(b)의 숫자와 동일한 개수의 입력데이터를 처리하는 하나의 오류검출코드 생성수단의 오류검출코드 단독이나, 상기 제2 파라미터(b)의 숫자와 동일한 개수의 입력데이터를 처리하는 하나의 오류검출코드 생성수단의 오류검출코드와 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)의 하나 또는 복수 개의 2비트 오류검출코드(C_3)를 연결하여 생성하는 것을 특징으로 하는 확장형 오류검출코드 생성기.

청구항 7

컴퓨터를 제 6 항의 각 모듈 및 각 수단으로 기능하게 하여 확장형 오류검출코드를 생성하게 하는 확장형 오류검출코드 생성프로그램이 저장된 컴퓨터로 읽을 수 있는 기록매체.

청구항 8

제 7 항의 확장형 오류검출코드 생성프로그램이 저장되고 통신망을 통해 상기 확장형 오류검출코드 생성프로그램을 전송할 수 있는 서버 시스템.

청구항 9

제 6 항의 확장형 오류검출코드 생성기;

LUT(Look-up-table) 입력 데이터에 대해 일정한 함수를 생성하기 위한 LUT 데이터가 저장되는 룩업테이블;

상기 LUT 데이터를 상기 이진 입력데이터로 하여 미리 계산된 확장형 오류검출코드를 저장하는 코드 저장 메모리; 및

데이터의 동일성을 검사하여 오류검출신호를 출력하는 동치 검사기;를 포함하고,

상기 룩업테이블이 동작하면, 상기 코드 저장 메모리는 출력될 LUT 데이터에 대응하는 확장형 오류검출코드를 출력하고, 상기 확장형 오류검출코드 생성기는 출력될 LUT 데이터를 상기 이진 입력데이터로 입력받아 확장형 오류검출코드를 생성하여 출력하며, 상기 동치 검사기는 상기 코드 저장 메모리와 상기 확장형 오류검출코드 생성기에서 출력되는 오류검출코드들을 서로 비교하여 동일하지 않을 경우 상기 오류검출신호를 출력하는 것을 특

정으로 하는 자가검사 록업테이블.

청구항 10

제 9 항에 있어서,

상기 록업테이블에 저장되는 LUT 데이터는 K-bit 입력 데이터에 대해 일정한 함수를 출력하는 하나 또는 복수 개의 2^k bits-LUT 데이터들로 이루어지고,

상기 확장형 오류검출코드 생성기는 상기 2^k bits-LUT 데이터 전체인 길이방향 LUT 데이터를 상기 이진 입력데이터로 동시에 입력받아 상기 2^k bits-LUT 데이터 전체에 대해 확장형 오류검출코드를 생성함으로써 상기 2^k bits-LUT 데이터 전체의 오류발생여부를 검출할 수 있는 것을 특징으로 하는 자가검사 록업테이블.

청구항 11

제 9 항에 있어서,

상기 록업테이블에 저장되는 LUT 데이터는 K-bit 입력 데이터에 대해 일정한 함수를 출력하는 하나의 2^k bits-LUT 데이터들로 이루어지고,

상기 확장형 오류검출코드 생성기는 상기 록업테이블이 동작할 때, 상기 2^k bits-LUT 데이터 중에서 출력되는 하나의 비트인 폭 방향 LUT 데이터를 상기 이진 입력데이터의 어느 하나의 비트로 입력받아 확장형 오류검출코드를 생성함으로써, 록업테이블이 동작할 때마다 오류발생여부를 검출할 수 있는 것을 특징으로 하는 자가검사 록업테이블.

청구항 12

제 11 항에 있어서,

상기 확장형 오류검출코드 생성기는 상기 2^k bits-LUT 데이터의 하나의 비트와 가상의 비트의 묶음을 상기 이진 입력데이터로 입력받아 확장형 오류검출코드를 생성하는 것을 특징으로 하는 자가검사 록업테이블.

청구항 13

제 9 항에 있어서,

상기 록업테이블에 저장되는 LUT 데이터는 K-bit 입력 데이터에 대해 복수 개의 함수를 출력하는 복수 개의 2^k bits-LUT 데이터들로 이루어지고,

상기 확장형 오류검출코드 생성기는 상기 록업테이블이 동작할 때, 상기 각 2^k bits-LUT 데이터들에서 출력되는 복수 개의 데이터인 폭 방향 LUT 데이터를 상기 이진 입력데이터로 입력받아 확장형 오류검출코드를 생성함으로써, 록업테이블이 동작할 때마다 오류발생여부를 검출할 수 있는 것을 특징으로 하는 자가검사 록업테이블.

청구항 14

제 6 항의 확장형 오류검출코드 생성기;

LUT(Look-up-table) 입력 데이터에 대해 일정한 함수를 생성하기 위한 LUT 데이터와 상기 LUT 데이터에 대해 미리 계산된 확장형 오류검출코드를 저장하며, 상기 LUT 입력 데이터가 입력되면, 대응하는 LUT 데이터와 미리 계

산된 확장형 오류검출코드를 함께 출력하는 룩업테이블; 및

데이터의 동일성을 검사하여 오류검출신호를 출력하는 동치 검사기;를 포함하고,

상기 룩업테이블이 동작하면, 상기 LUT 데이터는 상기 확장형 오류검출코드 생성기로 출력되고 상기 확장형 오류검출코드 생성기는 상기 LUT 데이터를 상기 이진 입력데이터로 하여 확장형 오류검출코드를 생성하여 출력하며, 상기 동치 검사기는 상기 룩업테이블에서 출력되는 미리 계산된 확장형 오류검출코드와 상기 확장형 오류검출코드 생성기에서 출력되는 확장형 오류검출코드를 서로 비교하여 동일하지 않을 경우 상기 오류검출신호를 출력하는 것을 특징으로 하는 자가검사 룩업테이블.

청구항 15

코드 길이 계산모듈이 오류검출의 대상이 되는 이진 입력데이터에 대해 생성될 확장형 오류검출코드 (SEDC:scalable error detection coding)의 길이를 계산하는 단계; 및

오류검출코드 생성 모듈이 2비트 입력데이터에 대해 오류검출코드를 생성하는 2비트 입력 오류검출코드 생성수단, 3비트 입력데이터에 대해 오류검출코드를 생성하는 3비트 입력 오류검출코드 생성수단 또는 4비트 입력데이터에 대해 오류검출코드를 생성하는 4비트 입력 오류검출코드 생성수단을 단독 또는 조합하여 상기 코드 길이 계산모듈에서 계산된 길이의 확장형 오류검출코드를 생성하는 단계;를 포함하는 것을 특징으로 하는 확장형 오류검출코드 생성방법.

청구항 16

제 15 항에 있어서,

상기 확장형 오류검출코드의 길이를 계산하는 단계:는

아래의 수학적식 6을 이용하여 제1 파라미터(a)와 제2 파라미터(b)를 계산하는 단계; 및

상기 제1 파라미터(a)를 이용하여 아래의 수학적식 7에 따라 상기 확장형 오류검출코드의 길이를 계산하는 단계;를 포함하는 것을 특징으로 하는 확장형 오류검출코드 생성방법.

[수학적식 6]

$$a = \frac{n - \max(b)}{3}$$

여기서, n은 이진입력데이터의 길이이고, a는 제1 파라미터로써 '0'에서부터 '∞'까지의 정수이고, b는 제2 파라미터로써 정수 '2', '3' 또는 '4'이고, max(b)는 정수 '2', '3' 및 '4' 중, 제1 파라미터가 정수가 되게 하는 최대값이다.

[수학적식 7]

$$m = \lceil \log_2[n+1-(3 \times a)] \rceil + (2 \times a)$$

여기서, m은 확장형 오류검출코드의 길이이다.

청구항 17

제 15 항 또는 제 16 항에 있어서,

상기 확장형 오류검출코드를 생성하는 단계에서,

상기 입력 데이터가 2비트 입력데이터(I₂)일 경우 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)을 이용하여 2

비트 오류검출코드(C_2)를 생성하되, 최상위 비트($C_2(1)$)는 상기 2비트 입력데이터(I_2)에서 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 생성하고, 그 외의 경우에는 '0'으로 생성하며, 최하위 비트($C_2(0)$)는 상기 2비트 입력데이터(I_2)에서 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성하며, 그 외의 경우에는 '1'로 생성하는 것을 특징으로 하는 확장형 오류검출코드 생성방법.

청구항 18

제 17 항에 있어서,

상기 확장형 오류검출코드를 생성하는 단계에서,

상기 입력 데이터가 3비트 입력데이터(I_3)일 경우 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)을 이용하여 2비트 오류검출코드(C_3)를 생성하되, 상기 3비트 입력데이터(I_3)의 최상위 비트($I_3(2)$)가 '0'일 경우에는, 중간비트($I_3(1)$)와 최하위 비트($I_3(0)$)를 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단($SEDC_2$)의 2비트 오류검출코드(C_2)와 동일하게 생성하고, 최상위 비트($I_3(2)$)가 '1'일 경우에는, 중간비트($I_3(1)$)와 최하위 비트($I_3(0)$)의 논리 부정 값을 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단($SEDC_2$)의 2비트 오류검출코드(C_2)의 논리 부정 값으로 생성하는 것을 특징으로 하는 확장형 오류검출코드 생성방법.

청구항 19

제 18 항에 있어서,

상기 확장형 오류검출코드를 생성하는 단계에서,

상기 입력 데이터가 4비트 입력데이터(I_4)일 경우 상기 4비트 입력 오류검출코드 생성수단($SEDC_4$)을 이용하여 3비트 오류검출코드(C_4)를 생성하되, 상기 3비트 오류검출코드(C_4)의 중간비트($C_4(1)$)와 최하위 비트($C_4(0)$)는 상기 4비트 입력데이터(I_4)의 최상위 비트($I_4(3)$)를 제외한 세 비트($I_4(2), I_4(1), I_4(0)$)를 입력데이터로 하는 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)의 2비트 오류검출코드(C_3)와 동일하게 생성하고, 최상위 비트($C_4(2)$)는 상기 4비트 입력데이터(I_4)의 최상위 비트($I_4(3)$)의 논리 부정 값으로 생성하는 것을 특징으로 하는 확장형 오류검출코드 생성방법.

청구항 20

제 16 항에 있어서,

상기 확장형 오류검출코드를 생성하는 단계에서,

상기 입력 데이터가 n 비트 입력데이터일 경우, n 비트 입력 오류검출코드 생성수단($SEDC_n$)을 이용하여 확장형 오류검출코드를 생성하고,

상기 n 비트 입력 오류검출코드 생성수단($SEDC_n$)은 상기 오류검출코드 생성수단들($SEDC_2, SEDC_3, SEDC_4$) 중, 상기 제2 파라미터(b)의 숫자와 동일한 개수의 입력데이터를 처리하는 하나의 오류검출코드 생성수단의 오류검출코드 단독이나, 상기 제2 파라미터(b)의 숫자와 동일한 개수의 입력데이터를 처리하는 하나의 오류검출코드 생성수단의 오류검출코드와 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)의 하나 또는 복수 개의 2비트 오류검출코드(C_3)를 연결하여 생성하는 것을 특징으로 하는 확장형 오류검출코드 생성방법.

명세서

기술분야

[0001] 본 발명은 확장형 오류검출코드 생성기, 그 생성기를 구비한 자가검사 록업테이블 및 확장형 오류검출코드 생성 방법에 관한 것으로, 보다 구체적으로는 서로 다른 길이의 이진 입력데이터에 대해 오류검출코드를 생성하는 복수 개의 오류검출코드 생성수단을 구비하여, 상기 오류검출코드 생성수단들 단독 또는 조합에 의해 오류검출코드의 길이를 유동적으로 확장할 수 있어, 최소한의 하드웨어 추가로 이진 입력데이터 전체의 단방향 오류를 검출할 수 있는 확장형 오류검출코드 생성기와 오류검출코드 생성방법 그리고 상기 확장형 오류검출코드 생성기를 구비하여 LUT데이터 전체를 자가검사하거나 매 동작시 LUT데이터를 자가검사할 수 있는 자가검사 록업테이블에 관한 것이다.

배경기술

- [0002] 내장형 시스템은 한정된 자원을 가지고 실시간 데이터를 처리하며, 다양한 속도의 수행속도를 갖는 주변장치와 상호반응하는 특징을 있다.
- [0003] 이러한 내장형 시스템의 목적은 제약시간 내에 원래의 목표한 기능을 수행하기 위한 것으로 수행되는 응용프로그램의 복잡성, 계산요구량, 개발비용과 시간 등으로 인하여 SRAM기반의 FPGA디바이스를 내장형 시스템의 설계에 이용하는 경향이 생겨났다.
- [0004] 실리콘 반도체 제조기술의 발달로 최근 몇 년 사이에 28nm 공정에서 2500만 게이트를 집적할 수 있는 FPGA디바이스가 출현했다.
- [0005] 이러한 초미세 공정으로 제작된 디바이스를 사용하는 재구성가능 시스템에서 가장 일반적인 결함은 외부적 요인과 반도체 물질의 불순도, 제작과정상의 문제, 사용시간의 증가로 인한 오류, 설계파라미터의 오류, 타이밍 오류 등의 내부적 요인으로 분류된다.
- [0006] 또한, 반도체 디바이스와 중성자, 알파입자의 반응으로 인해 설계와 실제 구현상의 차이에 의해 결함이 발생할 수도 있으며 이는 시스템의 오류로 귀결될 수 있다.
- [0007] 또한, 방사(radiation)에 의해 제작되는 CMOS 회로에서의 오류는 회로의 특정부분이 '0' 또는 '1'로 고정되는 stuck-at-0, stuck-at-1 오류를 초래할 수 있으며 이진 데이터의 논리 값을 단일 값으로 고정시키는 결함을 초래할 수 있다.
- [0008] 이러한 오류들을 검출하기 위한 오류검출기능은 반도체의 크기가 미세해짐에 따라 시스템의 신뢰성을 확보하기 위해 최근 더욱 중요한 관심을 받고 있으며, 오류에도 시스템의 동작이 중단되지 않고 원래 의도한 기능을 정상적으로 수행할 수 있는 결함허용가능 시스템의 연구가 활발하다.
- [0009] 또한, 결함허용기능을 갖추기 위해서는 기본 수행블럭들의 오류를 인지하고, 시스템의 동작 중에 오류를 극복하고 동작할 수 있는 회복조치들을 수행할 수 있어야 한다.
- [0010] 오류검출방법은 크게 Built-In-Self-Test(BIST)기법, 로빙테스트 기법, 중복기법, 논리값 기반의 기법, 오류검출코딩기법 등으로 분류된다.
- [0011] 먼저, BIST 기법은 별도의 테스트장비를 요구하지 않는 정적오류검출기법으로 동적결함검출 기능을 필요로 하는 재구성가능 내장형 시스템에는 적합하지 않다.
- [0012] 또한, 로빙테스트 기법은 계산용 데이터의 배열이 동일 크기의 영역으로 분할되며, 이 영역들 중 하나가 BIST를 수행할 수 있도록 구성되며, 나머지 영역들은 본래의 의도된 기능을 수행하도록 구성된다.
- [0013] 또한, 중복기법은 구조적 또는 시간적 중복성에 기반하여 오류를 검출하며 구조적 중복성을 이용한 오류검출기법은 동일한 기능블럭들을 복수 개로 구성하여 오류를 검출하며, 시간적 중복성을 이용한 오류검출기법은 동일한 기능블럭에서 복수 회 기능을 수행하여 오류를 검출한다. 그러나 이러한 중복기법은 시간적, 공간적 추가비용을 초래하는 단점이 있다.
- [0014] 또한, 논리값 기반의 기법은 오류를 검출하고자 하는 회로의 넷리스트를 입력으로 하여 논리값 패턴에 대한 회로의 모든 내부 노드들을 탐색하고, 오류의 발생가능성이 있는 회로의 노드패턴이 검색될 경우, 추가적인 검출 모듈을 추가하는 방식이다. 그러나, 이러한 논리값 기반의 기법은 모든 오류를 검출할 수 없는 단점이 있다.

- [0015] 또한, 오류검출코딩기법은 회로의 면적, 동작속도, 검출 오류의 범위 등에서 다른 기법들과 비교하여 매우 효율적이며, 이러한 방법은 에러검출코딩 알고리즘을 이용하여 이진데이터를 코딩함으로써 해당 이진데이터를 표현하는 코드를 검증하여 단방향 오류를 검출해 낼 수 있다.
- [0016] 또한, 오류검출코딩기법은 패리티코드(parity code), 해밍코드(hamming code), 리드솔로몬코드(Reed-Solomon code), 버저코드(berger code), 보스코드(Bose-lin code) 등과 같이 단방향 오류를 검출해 낼 수 있는 코딩기법들로 연구되어 왔으며, 그중 가장 간단하고 적은 비용으로 오류를 검출할 수 있는 코드는 원래의 이진데이터에 한 비트의 오류확인비트를 추가한 패리티코드이다.
- [0017] 이러한 패리티코드의 오류확인비트는 전체 데이터 비트에 '1'의 개수가 홀수개인지 짝수개인지에 따라 결정되며 매우 적은 하드웨어의 면적추가만으로도 신속한 오류검출기능을 수행할 수 있다. 그러나 패리티코드는 원래의 이진데이터에서 단 한 비트의 오류가 발생했을 경우와 홀수 비트에서 오류가 발생한 경우 이외에 다른 경우에는 오류검출기능을 수행할 수 없는 단점이 있다.
- [0018] 또한, 해밍코드는 에러정정기능을 제공하는 최초의 오류검출코드으로써, 해밍체크비트를 생성하기 위해 각각 다른 비트들에 패리티코딩을 수행한다. 즉, 해밍코드는 오류정정기능을 수행하기 위한 비트가 추가되므로 패리티코드에 비해 약간의 추가적인 하드웨어가 필요하다. 또한, 패리티코드와 유사하게 단일비트오류와 두 비트의 오류만을 검출할 수 있으므로 모든 이진 데이터들의 단방향 오류들을 검출할 수는 없다.
- [0019] 또한, 리드솔로몬코드는 오류검출과 오류정정기능을 함께 제공하는 다항식기반의 오류정정기법이다. 이 방법은 해밍코드와 비교하여 더 많은 하드웨어의 추가가 요구되며 이 역시 모든 이진데이터들의 단방향 오류를 검출할 수는 없다.
- [0020] 또한, 버저코드는 모든 단방향 오류들을 검출할 수는 있으나 오류정정기능을 제공하지 않는다. 이 코딩기법은 원래 이진데이터비트들의 논리 값 '0' 또는 '1'의 개수를 계산하여 그 값을 이진값으로 코딩하여 수행된다.
- [0021] 또한, 보스코드는 논리값 '0' 또는 '1'의 개수를 계수하여 코드를 생성한다는 점에서 버저코드와 유사하나, 단방향 오류를 검출하기 위한 모듈(modulo 4 or modulo 8)로 연산을 수행한다는 점이 서로 다르다. 또한, 보스코드는 t-단방향오류를 검출할 수 있으며, 여기서 't'는 원래 이진데이터의 두 비트 혹은 세 비트 오류를 의미한다. 또한, 보스코드는 버저코드와 마찬가지로, 오류정정기능을 제공하지 않는다.
- [0022] 이상에서 살펴본 바와 같이 오류검출코드 기법들 중에서 버저코드만이 입력 이진데이터들의 모든 비트에 대해 단방향 오류를 검출할 수 있다. 그러나 이 기법은 카운터회로를 위한 순차논리회로를 구성하는데 추가적인 하드웨어 면적부담이 초래되며 복수의 클록사이클이 필요하며, 긴 임계경로로 인한 계산 지연시간이 초래되는 문제점이 있다.

발명의 내용

해결하려는 과제

- [0023] 본 발명자들은 최소한의 하드웨어 추가로 입력되는 이진 입력데이터 전체에 대해 단방향 오류를 검출할 수 있고, 이진 입력데이터의 길이에 따라 선형적으로 오류검출코드를 확장하여 생성할 수 있는 확장형 오류검출코드 생성기와 상기 확장형 오류검출코드 생성기를 이용하여 자가검사를 통한 오류정정기능을 수행할 수 있는 록업테이블을 개발하기 위해 연구한 결과, 서로 다른 길이의 이진 입력데이터들에 대해 오류검출코드를 생성하는 복수 개의 오류검출코드 생성수단을 구비하여 유동적으로 오류검출코드의 길이를 확장할 수 있는 확장형 오류검출코드 생성기와 LUT 데이터 전체 또는 일부에 대해 자가검사를 통한 오류검출을 수행할 수 있는 자가검사 록업테이블의 기술적 구성을 개발하게 되어 본 발명을 완성하게 되었다.
- [0024] 따라서, 본 발명의 목적은 입력되는 이진 입력데이터의 길이에 따라 선형적으로 오류검출코드를 생성하여 입력되는 이진 입력데이터 전체의 단방향 오류를 검출할 수 있는 확장형 오류검출코드 생성기 및 확장형 오류검출코드 생성방법을 제공하는 것이다.
- [0025] 또한, 본 발명의 다른 목적은 상기 확장형 오류검출코드 생성기를 구비하여 자신이 갖고 있는 LUT데이터 전체의 오류를 한번에 검사하거나 매 동작시마다 출력되는 LUT 데이터의 오류를 검사할 수 있는 자가검사 록업테이블을 제공하는 것이다.

[0026] 본 발명의 목적들은 이상에서 언급한 목적들로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0027] 상기의 목적을 달성하기 위하여 본 발명은 오류검출의 대상이 되는 이진 입력데이터에 대해 생성될 확장형 오류검출코드(SED:scalable error detection coding)의 길이를 계산하는 코드 길이 계산모듈; 및 2비트 입력데이터에 대해 오류검출코드를 생성하는 2비트 입력 오류검출코드 생성수단, 3비트 입력데이터에 대해 오류검출코드를 생성하는 3비트 입력 오류검출코드 생성수단 또는 4비트 입력데이터에 대해 오류검출코드를 생성하는 4비트 입력 오류검출코드 생성수단을 포함하고, 상기 오류검출코드 생성수단들 단독 또는 조합에 의해 상기 코드 길이 계산모듈에서 계산된 길이의 확장형 오류검출코드를 생성하는 오류검출코드 생성모듈;을 포함하는 것을 특징으로 하는 확장형 오류검출코드 생성기를 제공한다.

[0028] 바람직한 실시예에 있어서, 상기 코드 길이 계산모듈:은 아래의 수학식 1을 이용하여 코드 길이의 계산을 위한 파라미터인 제1 파라미터(a) 및 제2 파라미터(b)를 계산하는 파라미터 계산수단; 및 아래의 수학식 2를 이용하여 상기 확장형 오류검출코드의 길이를 계산하는 코드 길이 계산수단;을 포함한다.

[0029] [수학식 1]

$$a = \frac{n - \max(b)}{3}$$

[0030]

[0031] 여기서, n은 이진입력데이터의 길이이고, a는 제1 파라미터로써 '0'에서부터 '∞'까지의 정수이고, b는 제2 파라미터로써 정수 '2', '3' 또는 '4'이고, max(b)는 정수 '2', '3' 및 '4' 중, 제1 파라미터가 정수가 되게 하는 최대값이다.

[0032] [수학식 2]

$$m = \lceil \log_2[n+1-(3 \times a)] \rceil + (2 \times a)$$

[0033]

[0034] 여기서, m은 확장형 오류검출코드의 길이이다.

[0035] 바람직한 실시예에 있어서, 상기 2비트 입력 오류검출코드 생성수단(SED₂)은 아래의 수학식 3을 이용하여, 2비트 입력데이터(I₂)에 대해 2비트 오류검출코드(C₂)를 생성하되, 최상위 비트(C₂(1))는 상기 2비트 입력데이터(I₂)에서 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 생성하고, 그 외의 경우에는 '0'으로 생성하며, 최하위 비트(C₂(0))는 상기 2비트 입력데이터(I₂)에서 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성하며, 그 외의 경우에는 '1'로 생성한다.

[0036] [수학식 3]

$$C_2(1) = \begin{cases} 1, & \text{if number of 0's in } I_2 \geq \text{number of 1's in } I_2 \\ 0, & \text{otherwise} \end{cases}$$

$$C_2(0) = \begin{cases} 0, & \text{if number of 0's in } I_2 = \text{number of 1's in } I_2 \\ 1, & \text{otherwise} \end{cases}$$

[0037]

[0038] 바람직한 실시예에 있어서, 상기 3비트 입력 오류검출코드 생성수단(SED₃)은 아래의 수학식 4를 이용하여, 3비트 입력데이터(I₃)에 대해 2비트 오류검출코드(C₃)를 생성하되, 상기 3비트 입력데이터(I₃)의 최상위 비트(I₃(2))가 '0'일 경우에는, 중간 비트(I₃(1))와 최하위 비트(I₃(0))를 입력데이터로 하는 상기 2비트 입력 오류

검출코드 생성수단(SED_{C2})의 2비트 오류검출코드(C₂)와 동일하게 생성하고, 최상위 비트(I₃(2))가 '1'일 경우에는, 중간 비트(I₃(1))와 최하위 비트(I₃(0))의 논리 부정 값을 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단(SED_{C2})의 2비트 오류검출코드(C₂)의 논리 부정 값으로 생성한다.

[0039] [수학식 4]

$$(C_3(1), C_3(0)) = \begin{cases} SEDC_2(I_3(1), I_3(0)), & \text{if } I_3(2)=0 \\ 1's \text{ complement}(SEDC_2(\overline{I_3(1)}, \overline{I_3(0)})), & \text{if } I_3(2)=1 \end{cases}$$

[0040]

[0041] 바람직한 실시예에 있어서, 상기 4비트 입력 오류검출코드 생성수단(SED_{C4})은 아래의 수학식 5를 이용하여, 4비트 입력데이터(I₄)에 대해 3비트 오류검출코드(C₄)를 생성하되, 상기 3비트 오류검출코드(C₄)의 중간 비트(C₄(1))와 최하위 비트(C₄(0))는 상기 4비트 입력데이터(I₄)의 최상위 비트(I₄(3))를 제외한 세 비트(I₄(2), I₄(1), I₄(0))를 입력데이터로 하는 상기 3비트 입력 오류검출코드 생성수단(SED_{C3})의 2비트 오류검출코드(C₃)와 동일하게 생성하고, 최상위 비트(C₄(2))는 상기 4비트 입력데이터(I₄)의 최상위 비트(I₄(3))의 논리 부정 값으로 생성한다.

[0042] [수학식 5]

$$(C_4(1), C_4(0)) = SEDC_3(I_4(2), I_4(1), I_4(0))$$

$$C_4(2) = \overline{I_4(3)}$$

[0043]

[0044] 바람직한 실시예에 있어서, 상기 오류 검출코드 생성모듈은 n비트 입력데이터에 대해 오류검출코드를 생성하는 n비트 입력 오류검출코드 생성수단(SED_{Cn})을 더 포함하고, 상기 n비트 입력 오류검출코드 생성수단(SED_{Cn})은 상기 오류검출코드 생성수단들(SED_{C2}, SED_{C3}, SED_{C4}) 중, 상기 제2 파라미터(b)의 숫자와 동일한 개수의 입력데이터를 처리하는 하나의 오류검출코드 생성수단의 오류검출코드 단독이나, 상기 제2 파라미터(b)의 숫자와 동일한 개수의 입력데이터를 처리하는 하나의 오류검출코드 생성수단의 오류검출코드와 상기 3비트 입력 오류검출코드 생성수단(SED_{C3})의 하나 또는 복수 개의 2비트 오류검출코드(C₃)를 연결하여 생성한다.

[0045] 또한, 본 발명은 컴퓨터를 상기 각 모듈 및 각 수단으로 기능하게 하여 확장형 오류검출코드를 생성하게 하는 확장형 오류검출코드 생성프로그램이 저장된 컴퓨터로 읽을 수 있는 매체를 더 제공한다.

[0046] 또한, 본 발명은 상기 확장형 오류검출코드 생성프로그램이 저장되고 통신망을 통해 상기 확장형 오류검출코드 생성프로그램을 전송할 수 있는 서버 시스템을 더 제공한다.

[0047] 또한, 본 발명은 상기 확장형 오류검출코드 생성기; LUT(Look-up-table) 입력 데이터에 대해 일정한 함수를 생성하기 위한 LUT 데이터가 저장되는 룩업테이블; 상기 LUT 데이터를 상기 이진 입력데이터로 하여 미리 계산된 확장형 오류검출코드를 저장하는 코드 저장 메모리; 및 데이터의 동일성을 검사하여 오류검출신호를 출력하는 동치 검사기;를 포함하고, 상기 룩업테이블이 동작하면, 상기 코드 저장 메모리는 출력될 LUT 데이터에 대응하는 확장형 오류검출코드를 출력하고, 상기 확장형 오류검출코드 생성기는 출력될 LUT 데이터를 상기 이진 입력데이터로 입력받아 확장형 오류검출코드를 생성하여 출력하며, 상기 동치 검사기는 상기 코드 저장 메모리와 상기 확장형 오류검출코드 생성기에서 출력되는 오류검출코드들을 서로 비교하여 동일하지 않을 경우 상기 오류검출신호를 출력하는 것을 특징으로 하는 자가검사 룩업테이블을 더 제공한다.

[0048] 바람직한 실시예에 있어서, 상기 룩업테이블에 저장되는 LUT 데이터는 K-bit 입력 데이터에 대해 일정한 함수를 출력하는 하나 또는 복수 개의 2^kbits-LUT 데이터들로 이루어지고, 상기 확장형 오류검출코드 생성기는 상기 2^kbits-LUT 데이터 전체인 길이방향 LUT 데이터를 상기 이진 입력데이터로 동시에 입력받아 상기 2^kbits-LUT 데이터 전체에 대해 확장형 오류검출코드를 생성함으로써 상기 2^kbits-LUT 데이터 전체의 오류발생여부를 검출할 수 있다.

[0049] 바람직한 실시예에 있어서, 상기 룩업테이블에 저장되는 LUT 데이터는 K-bit 입력 데이터에 대해 일정한 함수를

출력하는 하나의 2^k bits-LUT 데이터들로 이루어지고, 상기 확장형 오류검출코드 생성기는 상기 룩업테이블이 동작할 때, 상기 2^k bits-LUT 데이터 중에서 출력되는 하나의 비트인 폭 방향 LUT 데이터를 상기 이진 입력데이터의 어느 하나의 비트로 입력받아 확장형 오류검출코드를 생성함으로써, 룩업테이블이 동작할 때마다 오류발생여부를 검출할 수 있다.

[0050] 바람직한 실시예에 있어서, 상기 확장형 오류검출코드 생성기는 상기 2^k bits-LUT 데이터의 하나의 비트와 가상의 비트의 묶음을 상기 이진 입력데이터로 입력받아 확장형 오류검출코드를 생성한다.

[0051] 바람직한 실시예에 있어서, 상기 룩업테이블에 저장되는 LUT 데이터는 K-bit 입력 데이터에 대해 복수 개의 함수를 출력하는 복수 개의 2^k bits-LUT 데이터들로 이루어지고, 상기 확장형 오류검출코드 생성기는 상기 룩업테이블이 동작할 때, 상기 각 2^k bits-LUT 데이터들에서 출력되는 복수 개의 데이터인 폭 방향 LUT 데이터를 상기 이진 입력데이터로 입력받아 확장형 오류검출코드를 생성함으로써, 룩업테이블이 동작할 때마다 오류발생여부를 검출할 수 있다.

[0052] 또한, 본 발명은 상기 확장형 오류검출코드 생성기; LUT(Look-up-table) 입력 데이터에 대해 일정한 함수를 생성하기 위한 LUT 데이터와 상기 LUT 데이터에 대해 미리 계산된 확장형 오류검출코드를 저장하며, 상기 LUT 입력 데이터가 입력되면, 대응하는 LUT 데이터와 미리 계산된 확장형 오류검출코드를 함께 출력하는 룩업테이블; 및 데이터의 동일성을 검사하여 오류검출신호를 출력하는 동치 검사기;를 포함하고, 상기 룩업테이블이 동작하면, 상기 LUT 데이터는 상기 확장형 오류검출코드 생성기로 출력되고 상기 확장형 오류검출코드 생성기는 상기 LUT 데이터를 상기 이진 입력데이터로 하여 확장형 오류검출코드를 생성하여 출력하며, 상기 동치 검사기는 상기 룩업테이블에서 출력되는 미리 계산된 확장형 오류검출코드와 상기 확장형 오류검출코드 생성기에서 출력되는 확장형 오류검출코드를 서로 비교하여 동일하지 않을 경우 상기 오류검출신호를 출력하는 것을 특징으로 하는 자가검사 룩업테이블을 더 제공한다.

[0053] 또한, 본 발명은 코드 길이 계산모듈이 오류검출의 대상이 되는 이진 입력데이터에 대해 생성될 확장형 오류검출코드(SED:scalable error detection coding)의 길이를 계산하는 단계; 및 오류검출코드 생성 모듈이 2비트 입력데이터에 대해 오류검출코드를 생성하는 2비트 입력 오류검출코드 생성수단, 3비트 입력데이터에 대해 오류검출코드를 생성하는 3비트 입력 오류검출코드 생성수단 또는 4비트 입력데이터에 대해 오류검출코드를 생성하는 4비트 입력 오류검출코드 생성수단을 단독 또는 조합하여 상기 코드 길이 계산모듈에서 계산된 길이의 확장형 오류검출코드를 생성하는 단계;를 포함하는 것을 특징으로 하는 확장형 오류검출코드 생성방법을 더 제공한다.

[0054] 바람직한 실시예에 있어서, 상기 확장형 오류검출코드의 길이를 계산하는 단계:는 아래의 수학식 6을 이용하여 제1 파라미터(a)와 제2 파라미터(b)를 계산하는 단계; 및 상기 제1 파라미터(a)를 이용하여 아래의 수학식 7에 따라 상기 확장형 오류검출코드의 길이를 계산하는 단계;를 포함한다.

[0055] [수학식 6]

$$a = \frac{n - \max(b)}{3}$$

[0056]

[0057] 여기서, n은 이진입력데이터의 길이이고, a는 제1 파라미터로써 '0'에서부터 ' ∞ '까지의 정수이고, b는 제2 파라미터로써 정수 '2', '3' 또는 '4'이고, max(b)는 정수 '2', '3' 및 '4' 중, 제1 파라미터가 정수가 되게 하는 최대값이다.

[0058] [수학식 7]

$$m = \lceil \log_2[n+1-(3 \times a)] \rceil + (2 \times a)$$

[0059]

[0060] 여기서, m은 확장형 오류검출코드의 길이이다.

[0061] 바람직한 실시예에 있어서, 상기 확장형 오류검출코드를 생성하는 단계에서, 상기 입력 데이터가 2비트 입력데이터(I_2)일 경우 상기 2비트 입력 오류검출코드 생성수단(SED₂)을 이용하여 2비트 오류검출코드(C_2)를

생성하되, 최상위 비트($C_2(1)$)는 상기 2비트 입력데이터(I_2)에서 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 생성하고, 그 외의 경우에는 '0'으로 생성하며, 최하위 비트($C_2(0)$)는 상기 2비트 입력데이터(I_2)에서 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성하며, 그 외의 경우에는 '1'로 생성한다.

[0062] 바람직한 실시예에 있어서, 상기 확장형 오류검출코드를 생성하는 단계에서, 상기 입력 데이터가 3비트 입력데이터(I_3)일 경우 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)을 이용하여 2비트 오류검출코드(C_3)를 생성하되, 상기 3비트 입력데이터(I_3)의 최상위 비트($I_3(2)$)가 '0'일 경우에는, 중간 비트($I_3(1)$)와 최하위 비트($I_3(0)$)를 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단($SEDC_2$)의 2비트 오류검출코드(C_2)와 동일하게 생성하고, 최상위 비트($I_3(2)$)가 '1'일 경우에는, 중간 비트($I_3(1)$)와 최하위 비트($I_3(0)$)의 논리 부정 값을 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단($SEDC_2$)의 2비트 오류검출코드(C_2)의 논리 부정 값으로 생성한다.

[0063] 바람직한 실시예에 있어서, 상기 확장형 오류검출코드를 생성하는 단계에서, 상기 입력 데이터가 4비트 입력데이터(I_4)일 경우 상기 4비트 입력 오류검출코드 생성수단($SEDC_4$)을 이용하여 3비트 오류검출코드(C_4)를 생성하되, 상기 3비트 오류검출코드(C_4)의 중간 비트($C_4(1)$)와 최하위 비트($C_4(0)$)는 상기 4비트 입력데이터(I_4)의 최상위 비트($I_4(3)$)를 제외한 세 비트($I_4(2), I_4(1), I_4(0)$)를 입력데이터로 하는 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)의 2비트 오류검출코드(C_3)와 동일하게 생성하고, 최상위 비트($C_4(2)$)는 상기 4비트 입력데이터(I_4)의 최상위 비트($I_4(3)$)의 논리 부정 값으로 생성한다.

[0064] 바람직한 실시예에 있어서, 상기 확장형 오류검출코드를 생성하는 단계에서, 상기 입력 데이터가 n비트 입력데이터일 경우, n비트 입력 오류검출코드 생성수단($SEDC_n$)을 이용하여 확장형 오류검출코드를 생성하고, 상기 n비트 입력 오류검출코드 생성수단($SEDC_n$)은 상기 오류검출코드 생성수단들($SEDC_2, SEDC_3, SEDC_4$) 중, 상기 제2 파라미터(b)의 숫자와 동일한 개수의 입력데이터를 처리하는 하나의 오류검출코드 생성수단의 오류검출코드 단독이나, 상기 제2 파라미터(b)의 숫자와 동일한 개수의 입력데이터를 처리하는 하나의 오류검출코드 생성수단의 오류검출코드와 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)의 하나 또는 복수 개의 2비트 오류검출코드(C_3)를 연결하여 생성한다.

발명의 효과

[0065] 본 발명은 다음과 같은 우수한 효과를 가진다.

[0066] 먼저, 본 발명의 확장형 오류검출코드 생성기에 의하면, 2비트 입력 오류검출코드 생성수단, 3비트 입력 오류검출코드 생성수단, 4비트 입력 오류검출코드 생성수단 및 n비트 입력 오류검출코드 생성수단을 구비하여 상기 오류검출코드 생성수단들 각 단독 또는 조합에 의해 오류검출의 대상이 되는 이진 입력데이터에 대해 선택적이고 유연적인 확장형 오류검출코드를 생성할 수 있는 장점이 있다.

[0067] 또한, 본 발명의 자가검사 록업테이블에 의하면, 길이방향 LUT 데이터에 대해 확장형 오류검출코드를 생성할 수 있으므로 전체 LUT 데이터에 대해 단방향 오류검사를 자가수행할 수 있는 장점이 있다.

[0068] 또한, 본 발명의 자가검사 록업테이블에 의하면, 폭 방향 LUT 데이터에 대해 확장형 오류검출코드를 생성할 수 있으므로, 록업테이블의 매 동작마다 출력되는 LUT 데이터의 오류검사를 자가수행할 수 있는 장점이 있다.

도면의 간단한 설명

[0069] 도 1은 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기의 2비트 오류검출코드 생성수단의 입출력명세를 보여주는 도면,

도 2는 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기의 2비트 오류검출코드 생성수단이 생성하는 2비트 오류검출코드의 진리표,

- 도 3은 본 발명의 일 실시예에 따른 오류검출코드 생성기의 2비트 오류검출코드 생성수단의 논리회로,
- 도 4는 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기의 3비트 오류검출코드 생성수단의 입출력명세를 보여주는 도면,
- 도 5는 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기의 3비트 오류검출코드 생성수단이 생성하는 2비트 오류검출코드의 진리표,
- 도 6은 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기의 3비트 오류검출코드 생성수단의 논리회로,
- 도 7은 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기의 4비트 오류검출코드 생성수단의 입출력명세를 보여주는 도면,
- 도 8은 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기의 4비트 오류검출코드 생성수단이 생성하는 3비트 오류검출코드의 진리표,
- 도 9는 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기의 4비트 오류검출코드 생성수단의 논리회로,
- 도 10은 본 발명의 다른 실시예에 따른 자가검사 룩업테이블을 보여주는 도면,
- 도 11은 본 발명의 다른 실시예에 따른 자가검사 룩업테이블이 길이방향 LUT 데이터에 대해 확장형 오류검출코드를 생성하는 것을 설명하기 위한 도면,
- 도 12는 본 발명의 다른 실시예에 따른 자가검사 룩업테이블이 하나의 LUT 데이터를 가질 경우, 길이방향 LUT 데이터에 대해 확장형 오류검출코드를 생성하는 일례를 보여주는 도면,
- 도 13은 본 발명의 다른 실시예에 따른 자가검사 룩업테이블이 복수의 LUT 데이터를 가질 경우, 길이방향 LUT 데이터에 대해 확장형 오류검출코드를 생성하는 것을 설명하기 위한 도면,
- 도 14는 본 발명의 다른 실시예에 따른 자가검사 룩업테이블이 복수의 LUT 데이터를 가질 경우, 길이방향 LUT 데이터에 대해 확장형 오류검출코드를 생성하는 일례를 보여주는 도면,
- 도 15는 본 발명의 다른 실시예에 따른 자가검사 룩업테이블이 폭 방향 LUT 데이터에 대해 확장형 오류검출코드를 생성하는 것을 설명하기 위한 도면,
- 도 16은 본 발명의 다른 실시예에 따른 자가검사 룩업테이블이 복수의 LUT 데이터를 가질 경우, 폭 방향 LUT 데이터에 대해 확장형 오류검출코드를 생성하는 것을 설명하기 위한 도면,
- 도 17은 본 발명의 다른 실시예에 따른 자가검사 룩업테이블이 두 개의 LUT 데이터를 갖는 경우, 폭 방향 LUT 데이터에 대해 확장형 오류검출코드를 생성하는 일례를 보여주는 도면,
- 도 18은 본 발명의 다른 실시예에 따른 자가검사 룩업테이블이 다섯 개의 LUT 데이터를 갖는 경우, 폭 방향 LUT 데이터에 대해 확장형 오류검출코드를 생성하는 일례를 보여주는 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0070] 본 발명에서 사용되는 용어는 가능한 현재 널리 사용되는 일반적인 용어를 선택하였으나, 특정한 경우는 출원인이 임의로 선정한 용어도 있는데 이 경우에는 단순한 용어의 명칭이 아닌 발명의 상세한 설명 부분에 기재되거나 사용된 의미를 고려하여 그 의미가 파악되어야 할 것이다.
- [0071] 이하, 첨부한 도면에 도시된 바람직한 실시예들을 참조하여 본 발명의 기술적 구성을 상세하게 설명한다.
- [0072] 그러나, 본 발명은 여기서 설명되는 실시예에 한정되지 않고 다른 형태로 구체화 될 수도 있다. 명세서 전체에 걸쳐 동일한 참조번호는 동일한 구성요소를 나타낸다.
- [0073] 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기는 입력되는 이진 입력데이터의 길이에 선형적으로 모든 이진 입력데이터의 오류를 검사할 수 있는 오류검출코드 생성기이다.
- [0074] 또한, 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성방법은 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기에 의해 수행되며 상기 확장형 오류검출코드 생성기의 동작순서와 실질적으로 동일하므로, 별도로 설명하지 않고 상기 확장형 오류검출코드 생성기의 설명과 함께 설명하기로 한다.

[0075] 또한, 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기는 확장형 오류검출코드 생성프로그램에 의해 기능하는 컴퓨터일 수 있고, 상기 확장형 오류검출코드 생성프로그램은 컴퓨터로 읽을 수 있는 기록매체나 통신망을 통해 통신가능한 서버 시스템을 통해 컴퓨터에 전송되어 상기 컴퓨터가 각 수단으로 기능하여 오류검출코드를 생성할 수 있다.

[0076] 이하에서는 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기의 구성을 상세히 설명하기로 한다.

[0077] 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기는 코드 길이 계산모듈과 오류검출코드 생성모듈을 포함하여 이루어진다.

[0078] 상기 코드 길이 계산모듈은 오류검출의 대상이 되는 이진 입력데이터에 대해 생성될 확장형 오류검출코드(SED:scalable error detection coding)의 길이를 계산한다.

[0079] 또한, 상기 코드 길이 계산모듈은 코드 길이의 계산을 위한 파라미터인 제1 파라미터(a)와 제2 파라미터(b)를 계산하는 파라미터 계산수단과 상기 파라미터들을 이용하여 상기 확장형 오류검출코드의 전체 길이를 계산하는 코드 길이 계산수단을 포함한다.

[0080] 또한, 상기 제1 파라미터(a)와 상기 제2 파라미터(b)는 코드 길이의 계산을 위한 파라미터들로서 아래의 수학식 1을 이용하여 계산된다.

수학식 1

$$a = \frac{n - \max(b)}{3}$$

[0081]

[0082] 여기서, n은 이진 입력데이터의 길이이고, a는 제1 파라미터로써 '0'에서부터 '∞'까지의 정수이고, b는 제2 파라미터로써 '2', '3' 또는 '4'의 정수이며, max(b)는 정수 '2', '3' 및 '4' 중, 제1 파라미터가 정수가 되게 하는 최대값이다.

[0083] 또한, 상기 코드 길이 계산수단은 아래의 수학식 2를 이용하여 생성될 오류검출코드의 전체 길이를 계산한다.

수학식 2

$$m = \lceil \log_2 [n+1-(3 \times a)] \rceil + (2 \times a)$$

[0084]

[0085] 여기서, m은 확장형 오류검출코드의 길이이다.

[0086] 예를 들어, 이진 입력데이터의 길이가 '8비트'일 경우, 상기 수학식 1의 max(b)는 정수 '2'로 선택되고, 제1 파라미터는 정수 '2'로 계산된다.

[0087] 여기서, 상기 제1 파라미터가 정수가 되게 하는 제2 파라미터가 복수 개 존재할 경우 가장 큰 정수를 max(b)로 선택한다.

[0088] 또한, 상기 제1 파라미터가 정수 '2'로 계산되었으므로, 수학식 2에 대입하면, 가우스 ' $\log_2 3$ '은 정수 '2'가 되고, 여기에 정수 '4'를 더하면 정수 '6'이 되어, 상기 이진 입력데이터의 길이가 '8비트'일 경우, 확장형 오류검출코드의 길이는 '6비트'로 계산되는 것이다.

[0089] 즉, 상기 코드 길이 계산모듈은 입력되는 이진 입력데이터의 길이에 따라 생성될 확장형 오류검출코드의 길이를 유동적으로 계산할 수 있는 기능을 한다.

[0090] 또한, 상기 제2 파라미터 'b'는 n비트의 이진 입력데이터에 대해 확장형 오류검출코드를 계산하는 파라미터로 활용된다. 자세한 설명은 아래의 n비트 오류검출코드 생성수단(SED_n)의 설명에서 상세하게 하기로 한다.

- [0091] 상기 오류검출코드 생성모듈은 상기 코드 길이 계산모듈에서 계산된 코드 길이를 갖는 확장형 오류검출코드를 생성한다.
- [0092] 또한, 상기 오류검출코드 생성모듈은 2비트 입력데이터에 대해 2비트의 오류검출코드를 생성하는 2비트 입력 오류검출코드 생성수단(SEDC₂), 3비트 입력데이터에 대해 2비트의 오류검출코드를 생성하는 3비트 입력 오류검출코드 생성수단(SEDC₃), 4비트 입력데이터에 대해 3비트의 오류검출코드를 생성하는 4비트 입력 오류검출코드 생성수단(SEDC₄) 및 n비트 입력데이터에 대해 오류검출코드를 생성하는 n비트 입력 오류검출코드 생성수단(SEDC_n)을 포함한다.
- [0093] 그러나, 상기 이진 입력데이터가 2비트로 정해져 있는 경우, 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)만을 구비할 수 있고, 3비트로 정해져 있는 경우 상기 3비트 입력 오류검출코드 생성수단(SEDC₃)만을 구비할 수 있고, 4비트로 정해져 있는 경우에는 상기 4비트 입력 오류검출코드 생성수단(SEDC₄)만을 구비하여 오류검출코드를 생성할 수 있다.
- [0094] 한편, 상기 n비트 입력 오류검출코드 생성수단(SEDC_n)은 상기 2비트 입력 오류검출코드 생성수단(SEDC₂), 상기 3비트 입력 오류검출코드 생성수단(SEDC₃) 및 상기 4비트 입력 오류검출코드 생성수단(SEDC₄)을 조합하여 하나의 확장형 오류검출코드를 생성하는 기능을 하며, 상기 이진 입력데이터가 2비트로 정해져 있다 하더라도, 상기 n비트 오류검출코드 생성수단(SEDC_n)에 의해 2비트 이진 입력데이터에 대해 오류검출코드가 생성되게 할 수 있다.
- [0095] 이하에서는 상기 각 오류검출코드 생성수단들(SEDC₂, SEDC₃, SEDC₄, SEDC_n)에 대해서 상세하게 설명하기로 한다.
- [0096] 도 1은 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)의 입출력 명세를 보여주는 도면이고, 도 2는 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)의 진리표를 보여주는 도면이며, 도 3은 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)의 논리회로를 보여주는 도면이다.
- [0097] 도면들을 참조하면, 상기 2비트 입력 오류검출코드 생성수단(SEDC₂, 100)은 2비트 입력데이터(I₂)에 대해 2비트의 오류검출코드(C₂)를 생성한다.
- [0098] 또한, 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)은 아래의 수학식 3을 이용하여 생성되는 2비트의 오류검출코드(C₂) 중, 최상위 비트(C₂(1))는 상기 2비트 입력데이터(I₂)에서 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 생성하고, 그 외의 경우에는 '0'으로 생성하며, 최하위 비트(C₂(0))는 상기 2비트 입력데이터(I₂)에서 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성하며, 그 외의 경우에는 '1'로 생성한다.

수학식 3

$$C_2(1) = \begin{cases} 1, & \text{if number of 0's in } I_2 \geq \text{number of 1's in } I_2 \\ 0, & \text{otherwise} \end{cases}$$

$$C_2(0) = \begin{cases} 0, & \text{if number of 0's in } I_2 = \text{number of 1's in } I_2 \\ 1, & \text{otherwise} \end{cases}$$

- [0099]
- [0100] 또한, 본 발명의 실시예에서는 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)을 2비트 입력에 대해 exclusive-NOT-OR 연산을 수행하는 하나의 XNOR게이트(111)와 2비트 입력에 대해 NOT-AND 연산을 수행하는 하나의 NAND게이트(112)를 이용하여 구현하였다.
- [0101] 그러나, 상기 2비트 입력 오류검출코드 생성수단은 2비트 입력데이터에 대해 도 2의 진리표의 오류검출코드(C₂)를 출력할 수 있는 논리회로라면 어떠한 게이트들의 조합으로도 설계가 가능하다.

[0102] 도 4는 상기 3비트 입력 오류검출코드 생성수단(SEDC₃)의 입출력 명세를 보여주는 도면이고, 도 5는 상기 3비트 입력 오류검출코드 생성수단(SEDC₃)의 진리표를 보여주는 도면이며, 도 6은 상기 3비트 입력 오류검출코드 생성수단(SEDC₃)의 논리회로를 보여주는 도면이다.

[0103] 도면들을 참조하면, 상기 3비트 입력 오류검출코드 생성수단(SEDC₃, 120)은 3비트 입력데이터(I₃)에 대해 2비트의 오류검출코드(C₃)를 생성한다.

[0104] 또한, 상기 3비트 입력 오류검출코드 생성수단(SEDC₃)은 아래의 수학식 4를 이용하여 2비트의 오류검출코드(C₃)를 생성하되, 상기 3비트 입력데이터(I₃)의 최상위 비트(I₃(2))가 '0'일 경우에는, 상기 3비트 입력데이터(I₃)의 중간 비트(I₃(1))와 최하위 비트(I₃(0))를 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)의 2비트 오류검출코드(C₂)와 동일하게 생성하고, 상기 3비트 입력데이터(I₃)의 최상위 비트(I₃(2))가 '1'일 경우에는, 중간 비트(I₃(1))와 최하위 비트(I₃(0))의 논리 부정 값을 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)의 2비트 오류검출코드(C₂)의 논리 부정 값으로 생성한다.

[0105] 예를 들어, 상기 3비트 입력데이터(I₃)가 '000'의 값을 갖는다면, 상기 3비트 입력데이터(I₃)의 최상위 비트(I₃(2))가 '0'이므로, 상기 3비트 입력데이터(I₃)의 중간비트(I₃(1)) '0'과 최하위비트(I₃(0)) '0'을 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)의 2비트 오류검출코드(C₂)인 '11'로 생성된다.

[0106] 또한, 예를 들어, 상기 3비트 입력데이터(I₃)가 '111'의 값을 갖는다면, 상기 3비트 입력데이터(I₃)의 최상위 비트(I₃(2))가 '1'이므로, 상기 3비트 입력데이터(I₃)의 중간비트(I₃(1)) '1'과 최하위비트(I₃(0)) '1'의 논리 부정 값인 '00'을 입력데이터로 하는 상기 2비트 입력 오류검출코드 생성수단(SEDC₂)의 2비트 오류검출코드(C₂) '11'의 논리 부정 값인 '00'이 2비트 오류검출코드(C₂)로 생성된다.

수학식 4

$$(C_3(1), C_3(0)) = \begin{cases} SEDC_2(I_3(1), I_3(0)), & \text{if } I_3(2)=0 \\ 1's \text{ complement}(SEDC_2(\overline{I_3(1)}, \overline{I_3(0)})), & \text{if } I_3(2)=1 \end{cases}$$

[0108] 또한, 본 발명의 실시예에서는 상기 3비트 입력 오류검출코드 생성수단(SEDC₃)을 세 개의 입력에 대해 exclusive-NOT-OR 연산을 수행하는 하나의 XNOR게이트(121), 네 개의 입력에 대해 or-and-invert 연산을 수행하는 하나의 OAI게이트(122) 및 두 개의 입력에 대해 AND 연산을 수행하는 AND게이트(123)로 설계하였다.

[0109] 그러나, 상기 3비트 입력 오류검출코드 생성수단(SEDC₃)은 입력 데이터와 출력 데이터가 도 5의 진리표를 만족할 수 있다면 어떠한 게이트들의 조합으로도 설계가 가능하다.

[0110] 도 7은 상기 4비트 입력 오류검출코드 생성수단(SEDC₄)의 입출력 명세를 보여주는 도면이고, 도 8은 상기 4비트 입력 오류검출코드 생성수단(SEDC₄)의 진리표를 보여주는 도면이며, 도 9는 상기 4비트 입력 오류검출코드 생성수단(SEDC₄)의 논리회로를 보여주는 도면이다.

[0111] 도면들을 참조하면, 상기 4비트 입력 오류검출코드 생성수단(SEDC₄, 130)은 4비트 입력데이터(I₄)에 대해 3비트의 오류검출코드(C₄)를 생성한다.

[0112] 또한, 상기 4비트 입력 오류검출코드 생성수단(SEDC₄)은 아래의 수학식 5를 이용하여 3비트의 오류검출코드(C₄)를 생성하되, 상기 3비트 오류검출코드(C₄)의 중간 비트(C₄(1))와 최하위 비트(C₄(0))는 상기 4비트 입력데이터(I₄)의 최상위 비트(I₄(3))를 제외한 세 비트(I₄(2), I₄(1), I₄(0))를 입력데이터로 하는 상기 3비트 입력 오류검출코드 생성수단(SEDC₃)의 2비트 오류검출코드(C₃)와 동일하게 생성하고, 최상위 비트(C₄(2))는 상기 4비트 입력

데이터(I_4)의 최상위 비트($I_4(3)$)의 논리 부정 값으로 생성한다.

수학식 5

$$(C_4(1), C_4(0)) = SEDC_3(I_4(2), I_4(1), I_4(0))$$

$$C_4(2) = \overline{I_4(3)}$$

[0113]

[0114]

예를 들어, 상기 4비트 입력 오류검출코드 생성수단($SEDC_4$)으로 입력되는 4비트 입력데이터(I_4)가 '0101'일 경우, 생성되는 3비트 오류검출코드(C_4)의 중간 비트($C_4(1)$)와 최하위 비트($C_4(0)$)는 상기 4비트 입력데이터(I_4)의 최상위 비트($I_4(3)$)를 제외한 세 비트($I_4(2), I_4(1), I_4(0)$), 즉, '101'을 입력데이터로 하는 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$)의 오류검출코드인 '01'로 생성되고, 최상위 비트($C_4(2)$)는 상기 4비트 입력데이터(I_4)의 최상위 비트($I_4(3)$)인 '0'의 논리 부정 값인 '1'로 생성되어 '101'의 3비트 오류검출코드(C_4)를 생성된다.

[0115]

또한, 본 발명의 실시예에서는 상기 4비트 입력 오류검출코드 생성수단($SEDC_4$)을 두 개의 입력에 대해 exclusive-NOT-OR 연산을 수행하는 하나의 XNOR게이트(131), 두 개의 입력에 대해 exclusive-OR 연산을 수행하는 하나의 XOR 게이트(132), 두 개의 입력과 하나의 선택 입력에 대해 선택적 입력을 출력하는 하나의 MUX(133), 하나의 입력에 대해 invert 연산을 수행하는 하나의 invert 게이트(134)로 설계하였다.

[0116]

그러나, 상기 4비트 입력 오류검출코드 생성수단($SEDC_4$)은 입력 데이터 및 출력 데이터가 도 8의 진리표를 만족할 수 있다면 어떠한 게이트들의 조합으로도 설계가 가능하다.

[0117]

상기 n비트 입력 오류검출코드 생성수단($SEDC_n$)은 n비트의 입력에 대해 오류검출코드를 생성하고, 상기 2비트 입력 오류검출코드 생성수단($SEDC_2$), 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$), 상기 4비트 입력 오류검출코드 생성수단($SEDC_4$) 중, 상기 파라미터 계산수단에서 계산된 제2 파라미터(b)의 숫자와 동일한 개수의 이진 입력데이터에 대해 오류검출코드를 생성하는 하나의 오류검출코드 생성수단과, 하나 또는 복수 개의 3비트 입력 오류검출코드 생성수단($SEDC_3$)을 조합하여 생성한다.

[0118]

예를 들면, 2비트 입력데이터가 '8비트'일 경우, 제2 파라미터(b)는 정수 '2'로 계산되고, 확장형 오류검출코드의 길이는 '6비트'로 계산됨을 앞서 보인바 있다.

[0119]

이 경우, 제2 파라미터(b)인 정수 '2'의 개수의 입력에 대해 오류검출코드를 생성하는 2비트 입력 오류검출코드 생성수단($SEDC_2$) 하나와 '3비트'의 입력에 대해 '2비트'의 오류검출코드를 생성하는 3비트 입력 오류검출코드 생성수단($SEDC_3$) 두 개를 조합하여 총 '6비트'의 오류검출코드를 생성하는 것이다.

[0120]

그러나, 이진 입력데이터가 '2비트', '3비트' 또는 '4비트'로 고정되어 있을 경우, 상기 2비트 입력 오류검출코드 생성수단($SEDC_2$), 상기 3비트 입력 오류검출코드 생성수단($SEDC_3$) 또는 상기 4비트 입력 오류검출코드 생성수단($SEDC_4$) 각각 만을 구비하여 오류검출코드를 생성할 수 있고, 이진 입력데이터가 '2비트'일 경우에도 상기 n비트 입력 오류검출코드 생성수단($SEDC_n$)을 이용하여 오류검출코드를 생성할 수도 있다.

[0121]

예를 들면, 이진 입력데이터의 길이가 '2비트'일 경우, 제2 파라미터 'b'는 '2'가 되고 제1 파라미터 'a'는 '0'이 되며 확장형 오류검출코드의 길이는 가우스 ' $\log_2 3$ '이 되어 '2비트'가 되고, 상기 n비트 입력 오류검출코드 생성수단($SEDC_n$)은 제2 파라미터 'b'는 '2'이므로 하나의 2비트 입력 오류검출코드 생성수단($SEDC_2$)만으로 오류검출코드를 생성할 수 있는 것이다.

[0122]

따라서, 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기 및 생성방법에 의하면 입력되는 이진 입력데이터의 길이에 선형적으로 오류검출코드를 확장하여 생성할 수 있는 장점이 있다.

- [0123] 도 10은 본 발명의 다른 실시예에 따른 자가검사 룩업테이블을 보여주는 도면으로써, 확장형 오류검출코드 생성기(100)와 코드 저장 메모리(200), 룩업테이블(300, Look-up-table, 'LUT') 및 동치 검사기(400)를 포함하여 이루어진다.
- [0124] 즉, 본 발명의 다른 실시예에 따른 자가검사 룩업테이블은 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기(100)를 이용하여 자신의 LUT 데이터 전체 또는 매 동작마다 출력되는 LUT 데이터의 오류를 검사할 수 있는 룩업테이블이다.
- [0125] 상기 확장형 오류검출코드 생성기(100)는 아래서 설명할 룩업테이블(300)의 LUT 데이터(310) 또는 LUT 출력 데이터(LUT Output)를 이진 입력데이터로 하여 확장형 오류검출코드를 생성한다. 생성되는 오류검출코드는 본 발명의 일 실시예에 따른 확장형 오류검출코드 생성기를 통해 생성되는 오류검출코드와 실질적으로 동일하므로 설명을 생략하기로 한다.
- [0126] 상기 코드 저장 메모리(200)는 아래에서 설명할 룩업테이블(200)에 저장된 LUT 데이터에 대해 본 발명의 일 실시예에 따른 확장형 오류검출코드 방법을 이용하여 미리 계산된 오류검출코드들이 저장된다.
- [0127] 그러나, 상기 코드 저장 메모리(200)는 별도로 구비되지 않을 수 있고, 이 경우, 미리 계산된 오류검출코드들은 상기 룩업테이블(200)의 LUT 데이터 중 일부에 오류검출코드들이 할당되어 저장될 수 있다. 자세한 설명은 도 17 내지 도 18의 설명에서 하기로 한다.
- [0128] 상기 룩업테이블(300)은 LUT 입력 데이터(K Input)에 대해 일정한 함수의 계산을 위한 LUT 데이터(310)가 저장된다
- [0129] 또한, 상기 룩업테이블(300)은 'K-bit'의 입력에 대해 ' 2^k bit'의 LUT 데이터를 갖는다.
- [0130] 또한, 상기 룩업테이블의 LUT 데이터는 길이방향(length-wise) 또는 폭 방향(width-wise)으로 상기 확장형 오류검출코드 생성기(100)로 출력된다.
- [0131] 도 11은 'K-bit'의 입력에 대해 하나의 출력을 위한 ' 2^k bit'의 LUT 데이터(310)를 갖는 룩업테이블(300)에서 LUT 데이터(310)가 길이방향으로 상기 확장형 오류검출코드 생성기(100)로 출력되는 것을 설명하기 위한 도면이다.
- [0132] 여기서 '길이방향으로 출력된다' 함은 상기 LUT 데이터(310) 전체를 병렬로 출력하는 것을 뜻하고, 예를 들어, 최상위 비트에서 최하위 비트 방향으로 또는 최하위 비트에서 최상위 비트 방향으로 연속된 데이터가 모두 한번에 출력되면, 상기 확장형 오류검출코드 생성기(100)는 상기 LUT 데이터(310) 모두에 대해 하나의 오류검출코드를 생성하게 된다.
- [0133] 도 12를 참조하여, 더욱 자세하게 설명하면 도 12는 '3비트'의 LUT 입력 데이터(K input)에 대해 'OR' 함수의 구현을 위한 하나의 '8비트' LUT 데이터(310)를 구비하는 룩업테이블(300)에서 길이방향으로 오류검출코드를 생성하는 것을 보여주는 일례이다.
- [0134] 여기서 상기 LUT 데이터(310)는 상기 확장형 오류검출코드 생성기(100)로 병렬적으로 동시에 출력되고, 상기 확장형 오류검출코드 생성기(100)는 상기 LUT 데이터(310)에 대해 '6비트'의 길이를 갖는 하나의 오류검출코드를 생성한다.
- [0135] 또한, '8비트' 길이의 이진 입력데이터에 대해서는 '6비트'의 오류검출코드가 생성되고, 하나의 2비트 입력 오류검출코드 생성수단(110)과 두 개의 3비트 입력 오류검출코드 생성수단(120)을 조합하여 '6비트' 오류검출코드를 생성하는 것을 앞서 설명한 바 있다.
- [0136] 또한, 도 12에서는 상기 LUT 데이터(310) 중, 최상위 비트와 그 다음 비트의 2비트(311, '11')를 상기 2비트 입력 오류검출코드 생성수단(110)의 이진 입력데이터로 사용하고, 세 번째 비트에서 다섯 번째 비트까지 3비트(312, '111')와 여섯 번째 비트부터 최하위 비트까지의 3비트(313, '110')를 각각 3비트 입력 오류검출코드 생성수단(120)의 이진 입력데이터로 사용하여, '010001'의 진리로 이루어지는 오류검출코드가 생성되는 것을 상정하였다.
- [0137] 그러나, 역으로, 최하위 비트와 그 상위 비트('01')를 상기 2비트 입력 오류검출코드 생성수단(110)의 이진 입력데이터로 사용하고, 나머지 여섯 비트를 반으로 나누어 각각 상기 3비트 입력 오류검출코드 생성수단(120)으

로 입력으로 사용할 수 있다.

- [0138] 또한, 이 경우, 상기 코드 저장메모리(200)에는 상기 LUT 데이터(310) 중, 최상위 비트와 그 다음 비트의 2비트(311, '11')에 대한 오류검출코드인 '01'(211), 세 번째 비트에서 다섯 번째 비트까지 3비트(312, '111')에 대한 오류검출코드인 '00'(212), 여섯 번째 비트부터 최하위 비트까지의 3비트(313, '110')에 대한 오류검출코드인 '01'(213)이 저장되어 있다.
- [0139] 상기 동치 검사기(400)는 저장된 오류검출코드(210)와 상기 확장형 오류검출코드 생성기(100)에서 생성된 오류검출코드의 동일성을 비교하고 서로 차이가 있을 경우 오류검출신호를 출력한다.
- [0140] 다시 말해서, 길이방향 LUT 데이터를 오류검출코드의 생성을 위한 이진 입력데이터로 입력받는 것은 LUT 데이터 전체에 대해 하나의 오류검출코드를 생성하는 것으로 설명할 수 있고, 모든 이진 입력데이터에 대해 선형적으로 오류검출을 수행할 수 있는 것을 의미한다.
- [0141] 도 13은 LUT 입력 데이터(K Input)에 대해 서로 다른 함수의 계산을 위한 복수 개의 LUT 데이터(310, ... 310n)가 구비될 경우 길이방향으로 오류검출코드를 생성하는 것을 설명하기 위한 것으로, 모든 LUT 데이터(310, ... 310n)가 병렬적으로 상기 확장형 오류검출코드 생성기(100)로 입력되고, 상기 확장형 오류검출코드 생성기(100)는 입력되는 데이터를 분리하여 상기 오류검출코드 생성수단들(SED₂, SED₃, SED₄)의 조합에 이용하여 하나의 오류검출코드를 생성한다.
- [0142] 여기서 상기 LUT 데이터(310, ... 310n)는 최상위 비트에서 최하위 비트방향으로 분리되어 상기 오류검출코드 생성수단들(SED₂, SED₃, SED₄)의 조합에 의해 오류검출코드가 생성될 수 있고, 최하위 비트에서 최상위 비트방향으로 분리되어 오류검출코드가 생성될 수도 있다.
- [0143] 그러나, 상기 LUT 데이터(310, ... 310n)가 갖는 모든 비트가 서로 중복되어 상기 확장형 오류검출코드 생성기로 입력되지 않는다면 어떠한 방법으로도 분리하여 상기 오류검출코드 생성수단들(SED₂, SED₃, SED₄)의 이진 입력데이터로 사용할 수 있다.
- [0144] 도 14는 LUT 입력 데이터(K Input)에 대해 'NOR', 'NAND' 및 'OR' 함수를 구현하기 위한 복수 개의 2^kbits-LUT 데이터(310)가 구비된 룩업테이블(300)에서 길이방향으로 LUT 데이터를 입력받아 확장형 오류검출코드를 생성하기 위한 자가검사 룩업테이블의 일례를 보여주는 것으로, 상기 LUT 데이터(310)는 'NOR' 함수의 구현을 위한 8비트 LUT데이터(310a), 'NAND' 함수의 구현을 위한 8비트 LUT데이터(310b) 및 'OR' 함수의 구현을 위한 8비트 LUT데이터(310c)로 구성되며, 총 24비트의 데이터 길이를 갖는다.
- [0145] 또한, 이진 입력데이터의 길이가 24비트일 경우, 상기 수학적 1을 통해 제1 파라미터(a)는 정수 '7'로 계산되고, 제2 파라미터(b)는 정수 '3'으로 계산되며, 확장형 오류검출코드의 길이는 '16'비트로 계산된다.
- [0146] 또한, 제2 파라미터(b)가 정수 '3'이므로 하나의 3비트 입력 오류검출코드 생성수단(SED₃)과 일곱 개의 3비트 입력 오류검출코드 생성수단(SED₃), 즉, 총 여덟 개의 3비트 입력 오류검출코드 생성수단을 이용하여 하나의 확장형 오류검출코드를 생성한다.
- [0147] 또한, 도 14에서는 'NOR' 함수의 구현을 위한 8비트 LUT데이터(310a)의 최하위 비트에서 'OR' 함수의 구현을 위한 8비트 LUT데이터(310c)의 최상위 비트방향으로 3비트씩(314, 315, 316, 317) 구분하여 오류검출코드를 생성하였으나 역으로 'OR' 함수의 구현을 위한 8비트 LUT데이터(310c)의 최상위 비트에서 'NOR' 함수의 구현을 위한 8비트 LUT데이터(310a)의 최하위 비트방향으로 3비트씩 분리하여 오류검출코드를 생성할 수 있으며, 'OR' 함수의 구현을 위한 8비트 LUT데이터(310c)의 최하위 비트에서 'NOR' 함수의 구현을 위한 8비트 LUT데이터(310a)의 최상위 비트 방향으로 3비트씩 구분하여 오류검출코드를 생성할 수도 있다.
- [0148] 또한, 'NOR' 함수의 구현을 위한 8비트 LUT데이터(310a)의 최하위 비트에서 'OR' 함수의 구현을 위한 8비트 LUT데이터(310c)의 최상위 비트방향으로 3비트씩(314, 315, 316, 317) 구분하여 오류검출코드를 생성할 경우, 분리된 각 3비트(314, 315, 316, 317)에 대응하여 미리 계산된 오류검출코드들(214, 215, 216, 217)이 상기 코드 저장 메모리에 저장된다.
- [0149] 도 15는 'K-bit'의 입력에 대해 하나의 출력을 위한 '2^kbit'의 LUT 데이터(310)를 갖는 룩업테이블(300)에서 LUT 데이터(310)가 폭 방향으로 상기 확장형 오류검출코드 생성기(100)로 출력되는 것을 설명하기 위한 도면이

다.

- [0150] 여기서 '폭 방향으로 출력된다' 함은 상기 LUT 데이터(310)의 하나의 비트(LSB)에 대해 오류검출코드를 생성하는 것을 뜻한다.
- [0151] 다시 말해서 최상위 비트에서 최하위 비트로 각각의 비트에 대해 오류검출코드가 생성되고 생성된 각 오류검출코드들은 자가검사를 위한 동치검사를 통해 오류검출이 수행되므로, 룩업테이블의 매 동작마다 출력되는 LUT 데이터에 대해 오류검출을 수행할 수 있는 것을 의미한다.
- [0152] 또한, 도 16은 'K-bit'의 입력에 대해 하나의 출력을 위한 복수의 ' 2^k bit'의 LUT 데이터들(310a, ..., 310n)을 갖는 룩업테이블(300)에서 LUT 데이터(310)가 폭 방향으로 상기 확장형 오류검출코드 생성기(100)로 출력되는 것을 설명하기 위한 도면으로써, 이 경우, 각 ' 2^k bit'의 LUT 데이터들(310a, ..., 310n)의 서로 동일한 비트(LSB1, LSB2), 예를 들면, 최하위 비트들을 동시에 입력받아 오류검출코드를 생성한다. 즉, 폭 방향 LUT 데이터라고 하더라도 복수 개의 LUT 데이터를 오류검출코드를 생성하기 위한 이진 입력데이터로 사용할 수 있다.
- [0153] 도 17은 LUT 입력 데이터(K Input)에 대해 'OR'함수 구현을 위한 하나의 출력을 갖는 룩업테이블(300)에서 폭 방향으로 오류검출코드를 생성하는 것을 설명하기 위한 것으로 LUT 데이터(310)는 '8비트' 길이의 데이터로 구성되며, LUT 입력 데이터(K Input)에 대해 어느 하나의 비트가 출력된다.
- [0154] 또한, 상기 LUT 데이터(310)는 'OR'함수 구현을 위한 LUT 데이터(310a) 이외에 미리 계산된 확장형 오류검출코드(310b)가 더 저장된다.
- [0155] 예를 들면, 'K-bit'의 입력에 대해 하나의 출력을 위한 복수의 ' 2^k bit'의 LUT 데이터들(310a, ..., 310n)을 갖는 룩업테이블(300)은 동일한 LUT 입력 데이터(K Input)를 갖는 두 개의 8×1 MUX로 구성되고 하나의 8×1 MUX의 LUT 데이터(310a)는 'OR'함수 구현을 위한 데이터들로 구성되고, 다른 하나의 8×1 MUX의 LUT 데이터(310b)는 미리 계산된 오류검출코드들이 저장된다.
- [0156] 또한, 이러한 경우 하나의 비트의 출력이 상기 확장형 오류검출코드 생성기(100)의 이진 입력데이터로 입력되므로, 가상의 '1비트'가 있는 것으로 가정하여 2비트 입력 오류검출코드 생성수단(111)을 통해 오류검출코드를 생성한 후, 그 중 하나의 비트를 오류검출코드로 출력한다.
- [0157] 예를 들어, 'OR'함수 구현을 위한 LUT 데이터(310a)의 최하위 비트(LSB1) 값인 '0'이 출력될 경우, 가상의 비트 '0'을 상위비트에 덧붙인 '00'을 상기 확장형 오류검출코드 생성기(100)의 입력으로 출력하고, 상기 확장형 오류검출코드 생성기(100)는 이진 입력데이터 '00'에 대응하는 오류검출코드인 '11'중, 하위비트 '1'을 출력한다.
- [0158] 이때, 미리 계산된 오류검출코드들이 저장된 다른 하나의 8×1 MUX의 LUT 데이터(310b)의 최하위 비트(310b)가 출력되며, 상기 최하위 비트(310b)에는 미리 데이터 '1'이 저장된다.
- [0159] 즉, 폭 방향 LUT 데이터를 이진 입력데이터로 입력받아 확장형 자가검사하는 방법은 룩업테이블의 매 동작마다 발생한 오류를 검출할 수 있는 것이고, 길이 방향 LUT 데이터를 이용하여 자가검사하는 방법은 LUT 데이터 전체에 대해 한번에 발생한 오류를 할 수 있는 방법으로써 서로 구분된다.
- [0160] 도 18은 NOR, NAND, XOR 함수를 구현하는 세 개의 출력을 갖는 LUT 데이터(310a)가 구비된 룩업테이블(300)에서 폭 방향 LUT 데이터에 대해 오류검출코드를 생성하는 방법을 설명하기 위한 도면으로, 복수 개의 함수를 구현하는 LUT 데이터들(310a-1, 310a-2, 310a-3)이 구비될 경우 입력 LUT 데이터(K Input)에 대해 각 LUT 데이터들(310a-1, 310a-2, 310a-3)의 동일 비트(예를 들면 각 LUT 데이터들의 최하위 비트들(LSB-1, LSB-2, LSB-3))이 상기 확장형 오류검출코드 생성기(100)로 출력된다.
- [0161] 또한, 도 18에서는 세 개의 데이터가 상기 확장형 오류검출코드 생성기(100)로 출력되는 것을 상정하였으므로 상기 확장형 오류검출코드 생성기(100)는 3비트 입력 오류검출코드 생성수단(120)을 이용하여 '2비트'의 오류검출코드를 생성한다.
- [0162] 또한, NOR, NAND, XOR 함수를 구현하기 위한 LUT 데이터(310a)에서 출력되는 데이터와 대응하여 미리 계산된 오류검출코드도 함께 출력된다.
- [0163] 예를 들어, 도 18에서 입력 LUT 데이터(K Input)에 대해 NOR, NAND, XOR 함수를 구현하는 LUT 데이터들(310a-1, 310a-2, 310a-3) 중, 각각의 최하위 비트들 '110'이 출력되는 동시에 오류검출코드가 저장된 LUT 데이터(310b)의 최하위 비트(LSB2-1, LSB2-2) '01'이 함께 출력되고, 상기 동치 검사기(400)는 상기 확장형 오류검출코

드 생성기(100)에서 이진 입력데이터 '110'에 대한 오류검출코드 '01'과 오류검출코드가 미리 저장된 LUT 데이터(310b)의 최하위 비트(LSB2-1, LSB2-2) '01'을 서로 비교하여 동일성을 판단한 후, 서로 동일하지 않을 경우 오류검출신호를 출력하여 자가검사를 수행한다.

[0164]

이상에서 살펴본 바와 같이 본 발명은 바람직한 실시예를 들어 도시하고 설명하였으나, 상기한 실시예에 한정되지 아니하며 본 발명의 정신을 벗어나지 않는 범위 내에서 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변경과 수정이 가능할 것이다.

부호의 설명

[0165]

100: 확장형 오류검출코드 발생기

110: 2비트 입력 오류검출코드 생성수단

120: 3비트 입력 오류검출코드 생성수단

130: 4비트 입력 오류검출코드 생성수단

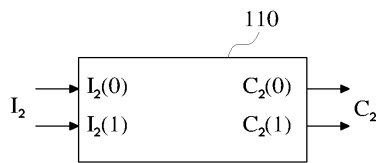
200: 코드 저장 메모리

300: 룩업테이블

310: LUT 데이터

도면

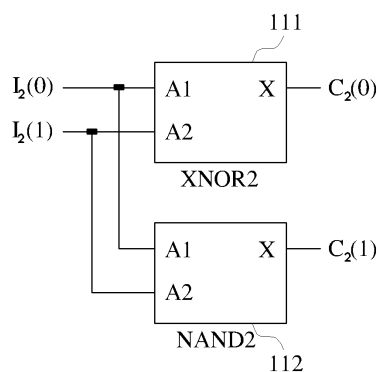
도면1



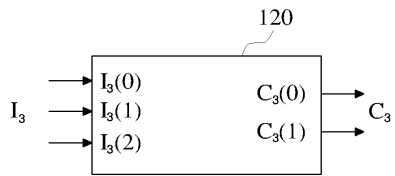
도면2

I ₂ (1)	I ₂ (0)	C ₂ (1)	C ₂ (0)
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	1

도면3



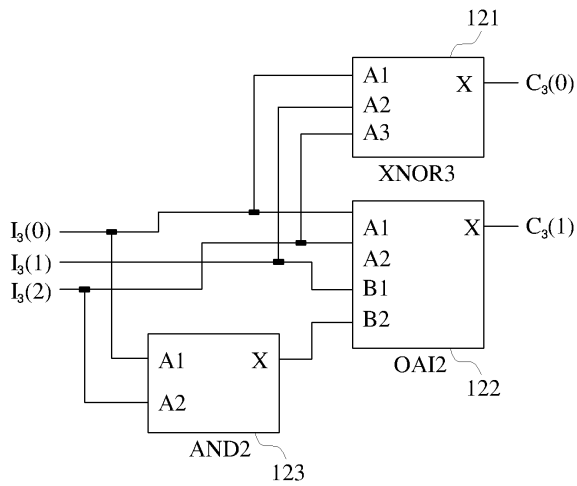
도면4



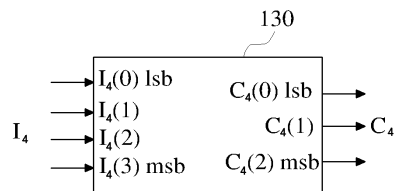
도면5

$I_3(2)$	$I_3(1)$	$I_3(0)$	$C_3(1)$	$C_3(0)$
0	0	0	1	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	0

도면6



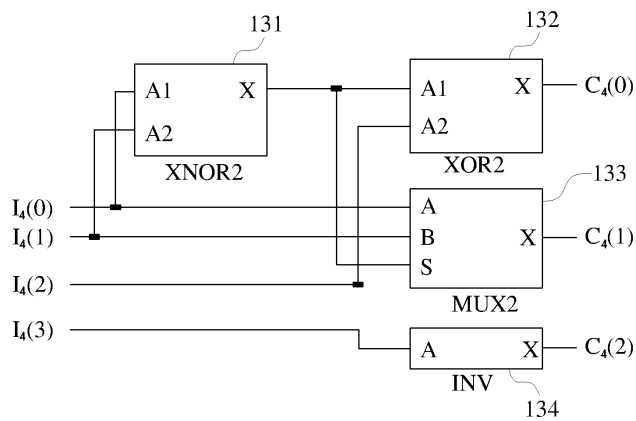
도면7



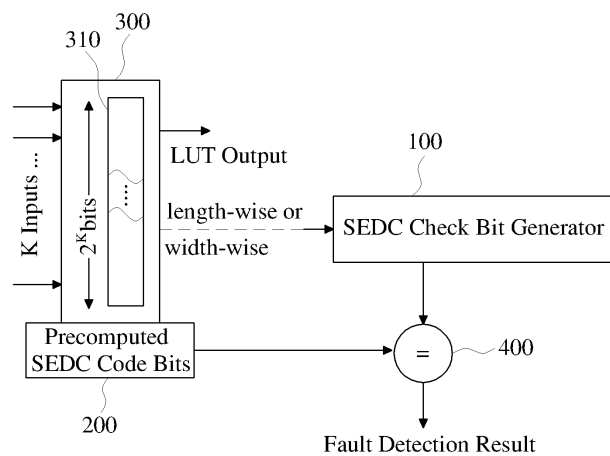
도면8

$I_4(3)$	$I_4(2)$	$I_4(1)$	$I_4(0)$	$C_4(2)$	$C_4(1)$	$C_4(0)$
0	0	0	0	1	1	1
0	0	0	1	1	1	0
0	0	1	0	1	1	0
0	0	1	1	1	0	1
0	1	0	0	1	1	0
0	1	0	1	1	0	1
0	1	1	0	1	0	1
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	0	1	0
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	0	0

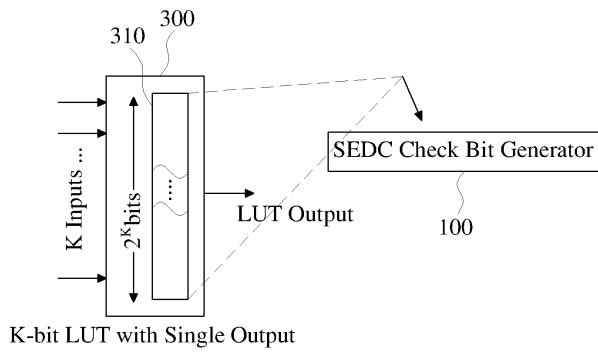
도면9



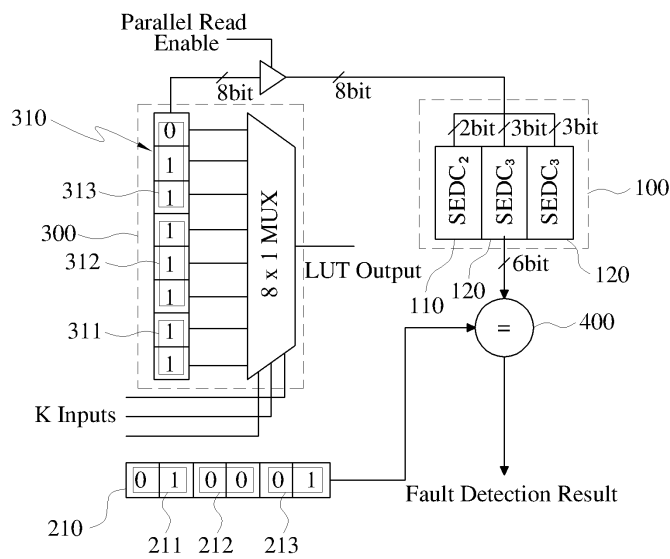
도면10



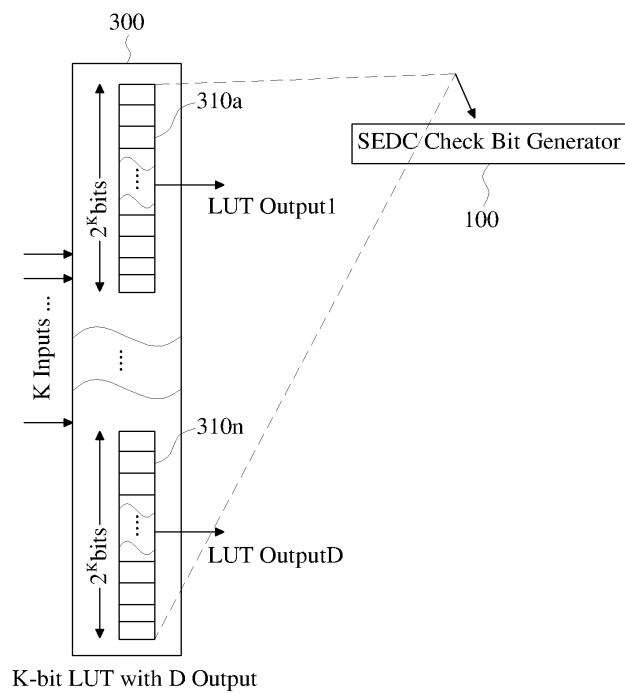
도면11



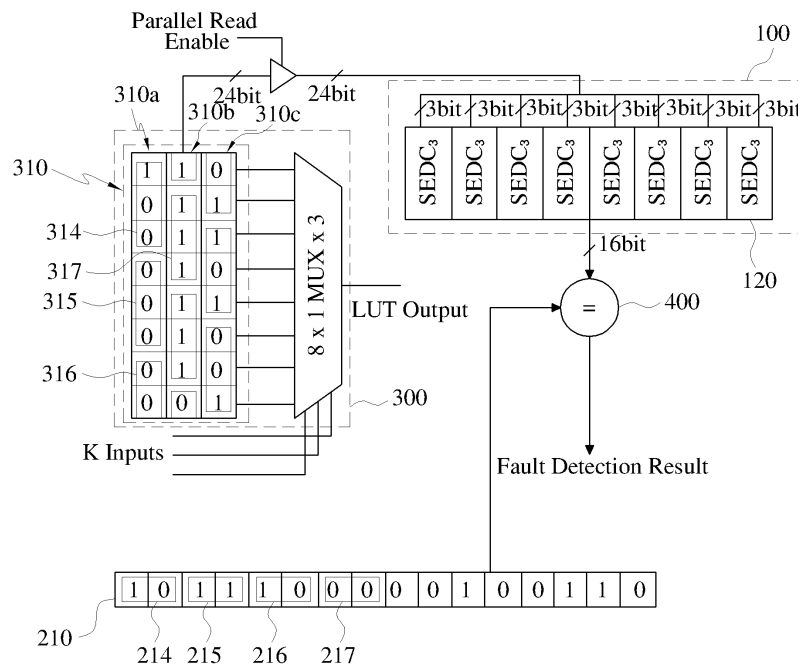
도면12



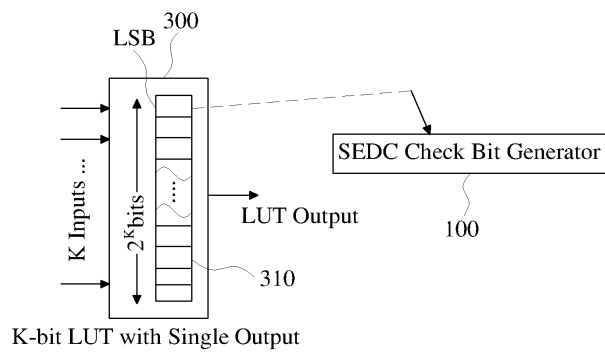
도면13



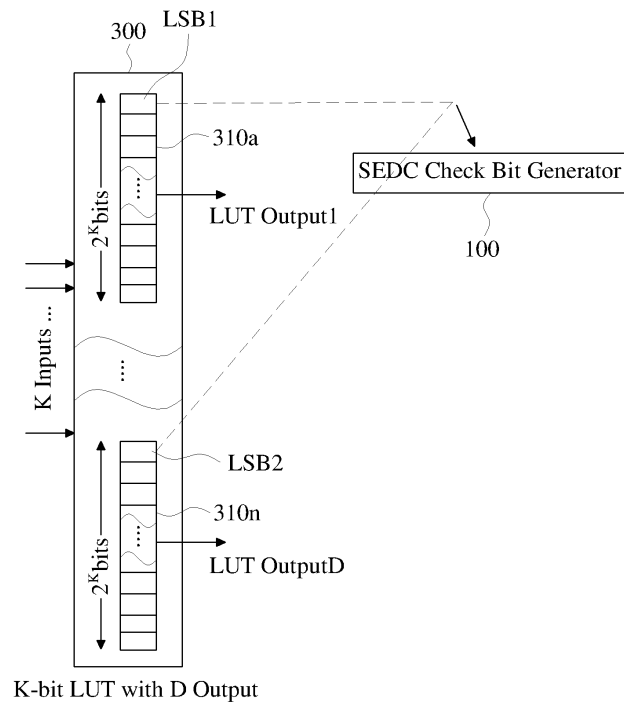
도면14



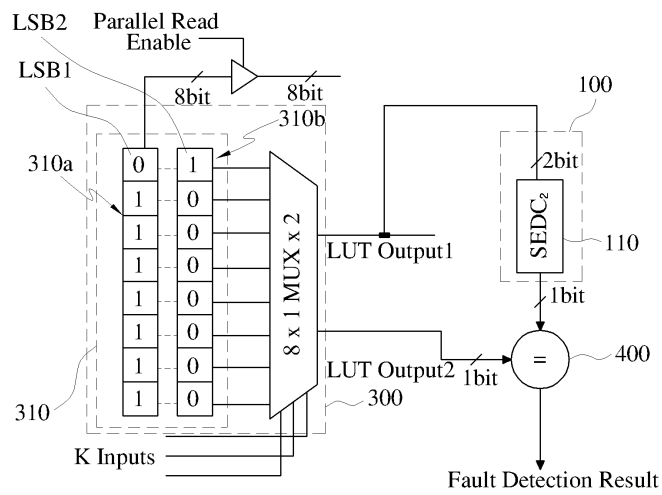
도면15



도면16



도면17



도면18

