



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년05월29일
(11) 등록번호 10-1268998
(24) 등록일자 2013년05월23일

(51) 국제특허분류(Int. Cl.)

G06F 11/10 (2006.01)

(21) 출원번호 10-2012-0006065

(22) 출원일자 2012년01월19일

심사청구일자 2012년01월19일

(56) 선행기술조사문헌

KR1020090017382 A*

KR1020100133194 A*

*는 심사관에 의하여 인용된 문헌

(73) 특허권자

조선대학교산학협력단

광주광역시 동구 서석동 375 조선대학교 내

(72) 발명자

이정아

서울특별시 마포구 상암동 휴먼시아아파트 203동 102호

소마순다람, 나타르잔

사이트 엔오. 10비 나가파 콜로니 제2크로스거리 카분담팔라얌, 코임바토레 641030

(74) 대리인

특허법인아이엠

전체 청구항 수 : 총 23 항

심사관 : 이정은

(54) 발명의 명칭 프로그램머블 시프트/로테이트 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 그 오류 검출 장치를 포함하는 자가검사 프로그램머블 시프트/로테이트 연산 유닛

(57) 요약

본 발명은 프로그램머블 시프트/로테이트 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 그 오류 검출 장치를 포함하는 자가검사 프로그램머블 시프트/로테이트 연산 유닛에 관한 것으로, 보다 구체적으로는 입력 데이터에 대해 시프트/로테이트 연산을 수행하여 출력 데이터를 출력하는 프로그램머블 시프트/로테이트 연산 유닛의 단방향 연산오류를 최소한의 하드웨어 추가로 연속적으로 검출할 수 있고, 입력 데이터들 및 출력 데이터의 길이에 따라 오류검출코드의 길이를 유동적으로 확장할 수 있으며, 오류검출시간을 최소화할 수 있는 프로그램머블 시프트/로테이트 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 그 오류 검출 장치를 포함하는 자가검사 프로그램머블 시프트/로테이트 연산 유닛에 관한 것이다.

대표도 - 도5

112

Operations	opSR	2-bit SEDC Encoded Shift/Rotate Operations
		$(C_1(2bit), C_0(2bit)) = SEDC_{2-in}(A(A_1, A_0), si, opSR)$
Shift Left A	0	$SEDC_2([A_0 si])$
Shift Right A	1	$SEDC_2([si A_1])$
Rotate Left A	2	$SEDC_2(A)$
Rotate Right A	3	$SEDC_2(A)$

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

삭제

청구항 2

이진 입력 데이터에 대해 시프트/로테이트(Shift/Rotate) 연산을 수행하여 연산결과인 이진 출력데이터를 출력하는 프로그래머블 시프트/로테이트 연산 유닛(Programable Shift/Rotate Operations Unit)의 오류를 검출하는 오류 검출 장치로써,

상기 이진 입력 데이터 및 시프트 비트를 입력받고, 시프트/로테이트 연산의 종류에 따라 상기 이진 입력 데이터의 오류 검출코드인 입력 오류검출코드를 생성하는 입력 오류검출코드 생성기; 및

상기 이진 출력데이터 및 상기 입력 오류 검출코드를 입력받고, 상기 이진 출력 데이터에 대한 오류검출코드인 출력 오류검출코드를 생성하며, 상기 출력 오류검출코드와 상기 입력 오류검출코드의 동일성을 판단하여 오류검출결과를 출력하는 오류 검출기;를 포함하고,

상기 입력 오류검출코드 및 상기 출력 오류검출코드는 상기 이진 입력 데이터 또는 상기 이진 출력 데이터의 길이에 따라 가변하는 확장형 오류검출코드(SED Scalable Error Detection Coding)이며,

상기 입력 오류검출코드 생성기:는

2비트 데이터에 대응하는 2비트 오류검출코드들(SED_2)을 저장한 2비트 오류검출코드 진리테이블(truth table); 및

상기 이진 입력 데이터의 길이에 따라 가변하는 상기 입력 오류검출코드를 생성하는 입력 오류검출코드 생성수단;을 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 3

제 2 항에 있어서,

상기 2비트 오류검출코드들(SED_2)의 상위 비트는 2비트 데이터 '00', '01', '10' 또는 '11'에 대해 비트들의 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 계산되어 저장되고, 그 외의 경우에는 '0'으로 계산되어 저장되며, 하위 비트는 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 계산되어 저장되고, 그 외의 경우에는 '1'로 계산되어 저장되거나,

상위 비트는 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 계산되어 저장되고, 그 외의 경우에는 '1'로 계산되어 저장되고, 하위 비트는 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 계산되어 저장되고, 그 외의 경우에는 '0'으로 계산되어 저장되는 것을 특징으로 하는 오류 검출 장치.

청구항 4

제 3 항에 있어서,

상기 입력 오류검출코드 생성수단은,

상기 이진 입력 데이터가 2비트 입력 데이터($A(A_1, A_0)$)일 경우, 상기 2비트 입력 데이터에 대한 시프트/로테이트 연산결과에 대응하는 2비트 오류검출코드(SED_2)인 2비트 입력 오류검출코드(SED_{2-in})를 계산하고, 상기 2비트 입력 오류검출코드(SED_{2-in})를 상기 입력 오류검출코드로 생성하는 2비트 입력 오류검출코드 생성수단;을 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 5

제 4 항에 있어서,

상기 2비트 입력 오류검출코드 생성수단은,

상기 시프트/로테이트 연산이 왼쪽 시프트(Shift Left) 연산일 경우, 상기 2비트 입력 데이터($A(A_1, A_0)$)의 하위 비트(A_0)를 상위 비트로 하고, 상기 시프트 비트(s_i)를 하위 비트로 하는 2비트 데이터(A_0, s_i)의 2비트 오류검출코드($SEDC_2$)를 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)로 생성하고,

상기 시프트/로테이트 연산이 오른쪽 시프트(Shift Right) 연산일 경우, 상기 2비트 입력 데이터($A(A_1, A_0)$)의 상위 비트(A_1)를 하위 비트로 하고, 상기 시프트 비트(s_i)를 상위 비트로 하는 2비트 데이터(s_i, A_1)의 2비트 오류검출코드($SEDC_2$)를 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)로 생성하고,

상기 시프트/로테이트 연산이 왼쪽 로테이트(Rotate Left) 연산 또는 오른쪽 로테이트(Rotate Right) 연산일 경우 상기 시프트 비트에 관계없이 상기 2비트 입력 데이터(A)에 대응하는 2비트 오류검출코드($SEDC_2$)를 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)로 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 6

제 5 항에 있어서,

상기 이진 입력 데이터가 1비트 입력 데이터일 경우, 상기 2비트 입력 오류검출코드 생성수단은 상기 시프트/로테이트 연산이 시프트 연산일 경우, 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 시프트 비트로 하는 2비트 데이터에 대해 2비트 오류검출코드($SEDC_2$)의 계산을 수행한 후, 계산된 2비트 오류검출코드($SEDC_2$) 중, 하위 비트만을 상기 입력 오류검출코드로 생성하고,

상기 시프트/로테이트 연산이 로테이트 연산일 경우, 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 1비트 입력 데이터로 하는 2비트 데이터에 대해 2비트 오류검출코드($SEDC_2$)의 계산을 수행한 후, 계산된 2비트 오류검출코드($SEDC_2$) 중, 하위 비트만을 상기 입력 오류검출코드로 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 7

제 5 항에 있어서,

상기 입력 오류검출코드 생성수단은,

상기 이진 입력 데이터가 3비트 입력 데이터($A(A_2, A_1, A_0)$)일 경우, 상기 3비트 입력 데이터에 대한 2비트의 오류검출코드인 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 계산하여, 계산된 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 상기 입력 오류검출코드로 생성하는 3비트 입력 오류검출코드 생성수단을 더 포함하고,

상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 입력 데이터($A(A_2, A_1, A_0)$)의 하위 두 비트(A_1, A_0)를 입력으로 하는 상기 2비트 입력 오류검출코드 생성수단의 2비트 입력 오류검출코드($SEDC_{2-in}$)를 과도 2비트 입력 오류검출코드로 계산하고, 상기 시프트/로테이트 연산의 종류 및 상기 3비트 입력 데이터의 상위 두 비트(A_2, A_1)의 값에 따라 상기 과도 2비트 입력 오류검출코드 자체 또는 상기 과도 2비트 입력 오류검출코드에 '1'을 산술차(-)하여 상기 3비트 입력 오류검출코드($SEDC_{3-in}$)를 계산하는 것을 특징으로 하는 오류 검출 장치.

청구항 8

제 7 항에 있어서,

상기 3비트 입력 오류검출코드 생성수단이 상기 과도 2비트 입력 오류검출코드에 '1'을 산술차(-)하는 경우는, 상기 시프트/로테이트 연산이 왼쪽 시프트 연산이고, 상기 3비트 입력 데이터($A(A_2, A_1, A_0)$)의 가운데 비트(A_1)의 값이 '1'인 경우와

상기 시프트/로테이트 연산이 오른쪽 시프트 연산이고, 상기 3비트 입력 데이터($A(A_2, A_1, A_0)$)의 최상위 비트(A_2)의 값이 '1'인 경우인 것을 특징으로 하는 오류 검출 장치.

청구항 9

제 8 항에 있어서,

상기 입력 오류검출코드 생성수단은,

상기 이진 입력 데이터가 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)일 경우, 상기 4비트 입력 데이터에 대한 3비트의 오류 검출코드인 4비트 입력 오류 검출코드($SEDC_{4-in}$)를 계산하여, 계산된 4비트 입력 오류 검출코드($SEDC_{4-in}$)를 상기 입력 오류검출코드로 생성하는 4비트 입력 오류검출코드 생성수단을 더 포함하고,

상기 3비트 입력 오류검출코드 생성수단은 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 하위 세 비트(A_2, A_1, A_0)를 입력으로 하는 상기 3비트 입력 오류검출코드 생성수단의 3비트 입력 오류검출코드($SEDC_{3-in}$)를 상기 4비트 입력 오류검출코드($SEDC_{4-in}$)의 하위 두 비트로 생성하되, 상기 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 상기 3비트 입력 오류검출코드($SEDC_{3-in}$)의 계산에 사용되는 시프트 비트는 상기 4비트 입력 데이터의 최상위 비트(A_3)로 대체하여 계산되고,

상기 4비트 입력 오류 검출코드($SEDC_{4-in}$)의 최상위 비트는, 상기 시프트/로테이트 연산이 왼쪽 시프트 연산일 경우, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 상위 두 번째 비트(A_2)의 논리 부정 값으로 생성되고, 상기 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 입력되는 시프트 비트의 논리 부정 값으로 생성되고, 상기 시프트/로테이트 연산이 왼쪽 로테이트 연산일 경우, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 상위 두 번째 비트(A_2)의 논리 부정 값으로 생성되고, 상기 시프트/로테이트 연산이 오른쪽 로테이트 연산일 경우, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 최하위 비트(A_0)의 논리 부정 값으로 생성되는 것을 특징으로 하는 오류 검출 장치.

청구항 10

제 9 항에 있어서,

상기 이진 입력 데이터가 n비트 입력 데이터일 경우 상기 n비트 입력 데이터에 대한 오류검출코드인 n비트 입력 오류검출코드($SEDC_{n-in}$)를 상기 입력 오류검출코드로 생성하는 n비트 입력 오류검출코드 생성수단을;을 더 포함하고,

상기 n비트 입력 오류검출코드 생성수단은 상기 입력 오류검출코드 생성수단들 각각의 오류검출코드($SEDC_{2-in}, SEDC_{3-in}, SEDC_{4-in}$)나, 상기 입력 오류검출코드 생성수단들 중 어느 하나의 입력 오류검출코드 생성수단의 오류 검출코드와 하나 또는 복수 개의 3비트 입력 오류검출코드($SEDC_{3-in}$)의 조합을 상기 입력 오류검출코드로 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 11

제 10 항에 있어서,

상기 n 비트 입력 데이터는 5비트 이상의 데이터이고,

상기 n 비트 입력 오류검출코드 생성수단은 상기 입력 오류검출코드 생성수단들 중 어느 하나의 입력 오류검출코드 생성수단의 오류검출코드인 b 비트 입력 오류검출코드($SEDC_{b-in}$)와 하나 또는 복수 개의 3비트 입력 오류검출코드($SEDC_{3-in}$)의 조합을 상기 입력 오류검출코드로 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 12

제 11 항에 있어서,

상기 n 비트 입력 오류검출코드 생성수단은 상기 n 비트 입력 데이터 중, 최상위 비트에서 하위 b 비트만큼의 b 비트 데이터에 대해 상기 b 비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하고, 상기 b 비트와 동일한 개수의 비트의 입력에 대해 오류검출코드를 생성하는 입력 오류검출코드 생성수단인 b 비트 입력 오류검출코드 생성수단에 의해 상기 b 비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하되,

상기 시프트/로테이트 연산이 왼쪽 시프트 연산일 경우, 상기 b 비트 입력 오류검출코드 생성수단은 계산에 사용되는 시프트 비트를 상기 n 비트 입력 데이터 중, 상기 b 비트 데이터를 제외한 최상위 비트로 대체하여 상기 b 비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하고,

상기 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 상기 b 비트 입력 오류검출코드 생성수단은 입력되는 시프트 비트(s_i)를 그대로 이용하여 상기 b 비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하며,

상기 시프트/로테이트 연산이 왼쪽 로테이트 연산일 경우, 상기 b 비트 입력 오류검출코드 생성수단은 왼쪽 시프트 연산의 경우로 상기 b 비트 입력 오류 검출코드($SEDC_{b-in}$)를 계산하여 생성하되, 계산에 사용되는 시프트 비트는 상기 n 비트 입력 데이터 중, 상기 b 비트 데이터를 제외한 최상위 비트로 대체하여 상기 b 비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하고,

상기 시프트/로테이트 연산이 오른쪽 로테이트 연산일 경우, 상기 b 비트 입력 오류검출코드 생성수단은 오른쪽 시프트 연산의 경우로 상기 b 비트 입력 오류 검출코드($SEDC_{b-in}$)를 계산하여 생성하되, 계산에 사용되는 시프트 비트는 상기 n 비트 입력 데이터의 최하위 비트로 대체하여 상기 b 비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 13

제 12 항에 있어서,

상기 n 비트 입력 오류검출코드 생성수단은 상기 3비트 입력 오류검출코드 생성수단을 이용하여 상기 n 비트 입력 데이터 중, 최상위 비트에서 하위 b 비트만큼을 제외한 ' k '개의 연속적인 3비트 데이터 세트에 대해 각각 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고, (여기서, $k=0, \dots, (a-1)$, a 는 3비트 데이터 세트의 개수이다.)

상기 시프트/로테이트 연산이 왼쪽 시프트 연산일 경우, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 데이터 세트들 중, 상기 n 비트 입력 데이터의 최하위 비트를 포함하는 최하위 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(s_i)를 그대로 이용하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고, 그 외의 각 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(s_i)를 바로 하위 3비트 데이터 세트의 최상위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하며,

상기 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 데이터 세트들 중, 상기 b 비트 데이터의 바로 하위 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(s_i)를 상기 b 비트 데이터의 최하위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고, 그 외의 각 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(s_i) 바로 상위 3비트 데이터 세트의 최하위 비트로

대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하며,

상기 시프트/로테이트 연산이 왼쪽 로테이트 연산일 경우, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 데이터 세트들 중, 상기 n 비트 입력 데이터의 최하위 비트를 포함하는 최하위 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si)를 상기 n 비트 입력 데이터의 최상위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고, 그 외의 각 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si)를 바로 하위 3비트 데이터 세트의 최상위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하며,

상기 시프트/로테이트 연산이 오른쪽 로테이트 연산일 경우, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 데이터 세트들 중, 상기 b 비트 데이터의 바로 하위 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si)를 상기 b 비트 데이터의 최하위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고, 그 외의 각 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si) 바로 상위 3비트 데이터 세트의 최하위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 14

제 2 항 내지 제 13 항 중 어느 한 항에 있어서,

상기 오류 검출기는:

상기 이진 출력 데이터가 2비트 출력 데이터일 경우, 상기 2비트 출력 데이터에 대한 2비트의 오류검출코드인 2비트 출력 오류검출코드($SEDC_{2-out}$)를 상기 출력 오류검출코드로 생성하는 2비트 출력 오류검출코드 생성수단; 및

상기 입력 오류검출코드와 상기 출력 오류검출코드의 동일성을 비교하여 동일하지 않을 경우 오류 발생신호를 출력하는 코드비교수단;을 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 15

제 14 항에 있어서,

상기 2비트 출력 오류검출코드 생성수단은,

상기 2비트 출력 오류검출코드들($SEDC_{2-out}$)의 최상위 비트는 2비트 데이터 '00', '01', '10' 또는 '11'에 대해 비트들의 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 생성하고, 그 외의 경우에는 '0'으로 생성하며, 최하위 비트는 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성하고, 그 외의 경우에는 '1'로 생성하거나, 최상위 비트는 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성되고, 그 외의 경우에는 '1'로 생성하며, 최하위 비트는 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 생성하고, 그 외의 경우에는 '0'으로 생성하되,

상기 2비트 출력 오류검출코드($SEDC_{2-out}$)는 동일한 입력에 대해 상기 2비트 오류검출코드($SEDC_2$)와 동일한 오류 검출코드로 생성되는 것을 특징으로 하는 오류 검출 장치.

청구항 16

제 15 항에 있어서,

상기 오류 검출기는,

상기 이진 출력 데이터가 3비트 출력 데이터일 경우, 상기 3비트 출력 데이터에 대한 2비트의 오류검출코드인 3비트 출력 오류검출코드($SEDC_{3-out}$)를 상기 출력 오류검출코드로 생성하는 3비트 출력 오류검출코드 생성수단;을 더 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 17

제 16 항에 있어서,

상기 3비트 출력 오류검출코드 생성수단은 상기 3비트 출력 데이터의 최상위 비트가 '0'일 경우에는, 중간비트 및 최하위 비트를 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 2비트 출력 오류검출코드($SEDC_{2-out}$)와 동일하게 상기 3비트 출력 오류검출코드($SEDC_{3-out}$)를 생성하고, 상기 3비트 출력 데이터의 최상위 비트가 '1'일 경우에는 중간비트 및 최하위 비트의 논리 부정값을 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 2비트 출력 오류검출코드($SEDC_{2-out}$)의 논리 부정값을 상기 3비트 출력 오류검출코드($SEDC_{3-out}$)로 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 18

제 17 항에 있어서,

상기 오류 검출기는,

상기 이진 출력 데이터가 4비트 출력 데이터일 경우, 상기 4비트 출력 데이터에 대한 3비트의 오류검출코드인 4비트 출력 오류검출코드($SEDC_{4-out}$)를 상기 출력 오류검출코드로 생성하는 4비트 출력 오류검출코드 생성수단;을 더 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 19

제 18 항에 있어서,

상기 4비트 출력 오류검출코드 생성수단은 상기 4비트 출력 오류검출코드($SEDC_{4-out}$)의 하위 2비트는 상기 4비트 출력 데이터 중, 최상위 비트를 제외한 3비트를 입력으로 하는 상기 3비트 출력 오류검출코드 생성수단의 3비트 출력 오류검출코드($SEDC_{3-out}$)로 생성하고, 상기 4비트 출력 오류검출코드($SEDC_{4-out}$)의 최상위 비트는 상기 4비트 출력 데이터의 최상위 비트의 논리 부정 값으로 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 20

제 19 항에 있어서,

상기 오류 검출기는,

상기 이진 출력 데이터가 n비트 출력 데이터일 경우, 상기 n비트 출력 데이터에 대한 오류검출코드인 n비트 출력 오류검출코드($SEDC_{n-out}$)를 상기 출력 오류검출코드로 생성하는 n비트 출력 오류검출코드 생성수단;을 더 포함하고,

상기 n비트 출력 오류검출코드 생성수단은 상기 출력 오류검출코드 생성수단들 각각의 오류검출코드($SEDC_{2-out}, SEDC_{3-out}, SEDC_{4-out}$)나, 상기 출력 오류검출코드 생성수단들 중 어느 하나의 출력 오류검출코드 생성수단의 오류검출코드와 하나 또는 복수 개의 3비트 출력 오류검출코드($SEDC_{3-out}$)의 조합을 상기 출력 오류검출코드로 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 21

제 19 항의 오류 검출 장치; 및

이진 입력 데이터에 대해 시프트/로테이트(Shift/Rotate) 연산을 수행하여 연산결과인 이진 출력데이터를 출력하는 프로그래머블 시프트/로테이트 연산 유닛;을 포함하는 자가검사 프로그래머블 시프트/로테이트 연산 유닛(Self-checking Programmable Programmable Shift/Rotate Operations Unit).

청구항 22

제 19 항의 오류 검출 장치; 및

이진 입력 데이터에 대해 시프트/로테이트(Shift/Rotate) 연산을 수행하여 연산결과인 이진 출력데이터를 출력하는 프로그래머블 산술/논리 연산유닛을 포함하는 자가검사 프로그래머블 산술/논리 연산유닛(Self-checking Programmable Arithmetic and Logic Unit).

청구항 23

컴퓨터를 제 19 항의 오류 검출 장치로 기능하게 하는 오류 검출 프로그램이 저장된 컴퓨터로 읽을 수 있는 기록매체.

청구항 24

제 23 항의 오류 검출 프로그램이 저장되고 통신망을 통해 상기 오류 검출 프로그램을 전송할 수 있는 서버 시스템.

명세서

기술분야

[0001] 본 발명은 프로그래머블 시프트/로테이트 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 그 오류 검출 장치를 포함하는 자가검사 프로그래머블 시프트/로테이트 연산 유닛에 관한 것으로, 보다 구체적으로는 입력 데이터에 대해 시프트/로테이트 연산을 수행하여 출력 데이터를 출력하는 프로그래머블 시프트/로테이트 연산 유닛의 단방향 연산오류를 최소한의 하드웨어 추가로 연속적으로 검출할 수 있고, 입력 데이터들 및 출력 데이터의 길이에 따라 오류검출코드의 길이를 유동적으로 확장할 수 있으며, 오류검출시간을 최소화할 수 있는 프로그래머블 시프트/로테이트 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 그 오류 검출 장치를 포함하는 자가검사 프로그래머블 시프트/로테이트 연산 유닛에 관한 것이다.

배경기술

[0002] 결함허용(fault-tolerance)이란 시스템에서 이나 연산환경에서 결함이 발생하였을 때, 어떠한 제약도 없이 의도된 기능을 수행할 수 있게 하는 능력을 말한다.

[0003] 또한, 결함허용을 수행하기 위해서는 기본 연산 블록들(basic computational blocks)이 결함을 인식할 수 있어야하고, 시스템이 기능하는 도중에도 결함을 복구하기 위한 신호를 출력할 수 있어야한다.

[0004] 오류검출방법은 크게 Built-In-Self-Test(BIST)기법, 로빙테스트 기법, 중복기법, 논리값 기반의 기법, 오류검출코딩기법 등으로 분류된다.

[0005] 먼저, BIST 기법은 별도의 테스트장비를 요구하지 않는 정적오류검출기법으로 동적결함검출 기능을 필요로 하는 재구성가능 내장형 시스템에는 적합하지 않다.

[0006] 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.

[0007] 또한, 중복기법은 구조적 또는 시간적 중복성에 기반하여 오류를 검출하며 구조적 중복성을 이용한 오류검출기법은 동일한 기능블럭들을 복수 개로 구성하여 오류를 검출하며, 시간적 중복성을 이용한 오류검출기법은 동일

한 기능블럭에서 복수 회 기능을 수행하여 오류를 검출한다. 그러나 이러한 중복기법은 시간적, 공간적 추가비용을 초래하는 단점이 있다.

- [0008] 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.
- [0009] 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.
- [0010] 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.
- [0011] 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.
- [0012] 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.
- [0013] 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.
- [0014] 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.
- [0015] 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.
- [0016] 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

- [0017] 본 발명자들은 결함 허용 재구성가능(fault-tolerant reconfigurable)하고, 연산 속도나 레이턴시(latency)와 같은 특성을 유지하면서도 입력 데이터의 길이에 따라 선형적으로 오류검출코드를 확장하여 자가검사(self-checking)할 수 있는 프로그래머블 시프트/로테이트 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 자가 자가검사 프로그래머블 시프트/로테이트 연산 유닛을 개발하기 위해 노력한 결과, 시프트/로테이트 연산의 기능은 유지시키되 입력 데이터의 연산오류를 검출하여 결함 허용 가능하고, 입력 데이터와 출력 데이터의 길이대 대해 선형적으로 오류검출코드를 확장하여 오류 검출을 수행할 수 있는 프로그래머블 시프트/로테이트 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치와 상기 오류 검출 장치를 구비하여 자가 검사를 수행할 수 있는 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 기술적 구성을 개발하게 되어 본

발명을 완성하게 되었다.

- [0018] 따라서, 본 발명의 목적은 프로그래머블 시프트/로테이트 연산 유닛의 기능은 유지시키되 연산의 오류를 검출할 수 있는 결합허용 가능한 오류 검출 장치를 제공하는 것이다.
- [0019] 또한, 본 발명의 다른 목적은 프로그래머블 시프트/로테이트 연산 유닛의 입력 데이터 및 출력 데이터의 길이에 따라 선형적으로 오류검출코드를 확장하여 오류 검출을 수행할 수 있는 오류 검출 장치를 제공하는 것이다.
- [0020] 또한, 본 발명의 또 다른 목적은 상기 오류 검출 장치를 구비하여 자가 검사를 할 수 있는 자가검사 프로그래머블 시프트/로테이트 연산 유닛을 제공하는 것이다.
- [0021] 본 발명의 목적들은 이상에서 언급한 목적들로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

- [0022] 상기의 목적을 달성하기 위하여 본 발명은 이진 입력 데이터에 대해 시프트/로테이트(Shift/Rotate) 연산을 수행하여 연산결과인 이진 출력데이터를 출력하는 프로그래머블 시프트/로테이트 연산 유닛(Programable Shift/Rotate Operations Unit)의 오류를 검출하는 오류 검출 장치로써, 상기 이진 입력 데이터 및 시프트 비트를 입력받고, 시프트/로테이트 연산의 종류에 따라 상기 이진 입력 데이터의 오류 검출코드인 입력 오류검출코드를 생성하는 입력 오류검출코드 생성기; 및 상기 이진 출력데이터 및 상기 입력 오류 검출코드를 입력받고, 상기 이진 출력 데이터에 대한 오류검출코드인 출력 오류검출코드를 생성하며, 상기 출력 오류검출코드와 상기 입력 오류검출코드의 동일성을 판단하여 오류검출결과를 출력하는 오류 검출기;를 포함하고, 상기 입력 오류검출코드 및 상기 출력 오류검출코드는 상기 이진 입력 데이터 또는 상기 이진 출력 데이터의 길이에 따라 가변하는 확장형 오류검출코드(SEDCScalable Error Detection Coding)인 것을 특징으로 하는 오류 검출 장치를 제공한다.
- [0023] 바람직한 실시예에 있어서, 상기 입력 오류검출코드 생성기:는 2비트 데이터에 대응하는 2비트 오류검출코드들($SEDC_2$)을 저장한 2비트 오류검출코드 진리테이블(truth table); 및 상기 이진 입력 데이터의 길이에 따라 가변하는 상기 입력 오류검출코드를 생성하는 입력 오류검출코드 생성수단;을 포함한다.
- [0024] 바람직한 실시예에 있어서, 상기 2비트 오류검출코드들($SEDC_2$)의 상위 비트는 2비트 데이터 '00', '01', '10' 또는 '11'에 대해 비트들의 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 계산되어 저장되고, 그 외의 경우에는 '0'으로 계산되어 저장되며, 하위 비트는 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 계산되어 저장되고, 그 외의 경우에는 '1'로 계산되어 저장되거나, 상위 비트는 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 계산되어 저장되고, 그 외의 경우에는 '1'로 계산되어 저장되고, 하위 비트는 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 계산되어 저장되고, 그 외의 경우에는 '0'으로 계산되어 저장된다.
- [0025] 바람직한 실시예에 있어서, 상기 입력 오류검출코드 생성수단은, 상기 이진 입력 데이터가 2비트 입력 데이터($A(A_1, A_0)$)일 경우, 상기 2비트 입력 데이터에 대한 시프트/로테이트 연산결과에 대응하는 2비트 오류검출코드($SEDC_2$)인 2비트 입력 오류검출코드($SEDC_{2-in}$)를 계산하고, 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)를 상기 입력 오류검출코드로 생성하는 2비트 입력 오류검출코드 생성수단;을 포함한다.
- [0026] 바람직한 실시예에 있어서, 상기 2비트 입력 오류검출코드 생성수단은, 상기 시프트/로테이트 연산이 왼쪽 시프트(Shift Left) 연산일 경우, 상기 2비트 입력 데이터($A(A_1, A_0)$)의 하위 비트(A_0)를 상위 비트로 하고, 상기 시프트 비트(s_i)를 하위 비트로 하는 2비트 데이터(A_0, s_i)의 2비트 오류검출코드($SEDC_2$)를 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)로 생성하고, 상기 시프트/로테이트 연산이 오른쪽 시프트(Shift Right) 연산일 경우, 상기 2비트 입력 데이터($A(A_1, A_0)$)의 상위 비트(A_1)를 하위 비트로 하고, 상기 시프트 비트(s_i)를 상위 비트로 하는 2비트 데이터(s_i, A_1)의 2비트 오류검출코드($SEDC_2$)를 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)로 생성하고, 상기 시프트/로테이트 연산이 왼쪽 로테이트(Rotate Left) 연산 또는 오른쪽 로테이트(Rotate Right) 연산일 경우 상기 시프트 비트에 관계없이 상기 2비트 입력 데이터(A)에 대응하는 2비트 오류검출코드($SEDC_2$)를 상기 2비트 입력

오류검출코드($SEDC_{2-in}$)로 생성한다.

[0027] 바람직한 실시예에 있어서, 상기 이진 입력 데이터가 1비트 입력 데이터일 경우, 상기 2비트 입력 오류검출코드 생성수단은 상기 시프트/로테이트 연산이 시프트 연산일 경우, 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 시프트 비트로 하는 2비트 데이터에 대해 2비트 오류검출코드($SEDC_2$)의 계산을 수행한 후, 계산된 2비트 오류검출코드($SEDC_2$) 중, 하위 비트만을 상기 입력 오류검출코드로 생성하고, 상기 시프트/로테이트 연산이 로테이트 연산일 경우, 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 1비트 입력 데이터로 하는 2비트 데이터에 대해 2비트 오류검출코드($SEDC_2$)의 계산을 수행한 후, 계산된 2비트 오류검출코드($SEDC_2$) 중, 하위 비트만을 상기 입력 오류검출코드로 생성한다.

[0028] 바람직한 실시예에 있어서, 상기 입력 오류검출코드 생성수단은, 상기 이진 입력 데이터가 3비트 입력 데이터($A(A_2, A_1, A_0)$)일 경우, 상기 3비트 입력 데이터에 대한 2비트의 오류 검출코드인 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 계산하여, 계산된 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 상기 입력 오류검출코드로 생성하는 3비트 입력 오류검출코드 생성수단을 더 포함하고, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 입력 데이터($A(A_2, A_1, A_0)$)의 하위 두 비트(A_1, A_0)를 입력으로 하는 상기 2비트 입력 오류검출코드 생성수단의 2비트 입력 오류검출코드($SEDC_{2-in}$)를 과도 2비트 입력 오류검출코드로 계산하고, 상기 시프트/로테이트 연산의 종류 및 상기 3비트 입력 데이터의 상위 두 비트(A_2, A_1)의 값에 따라 상기 과도 2비트 입력 오류검출코드 자체 또는 상기 과도 2비트 입력 오류검출코드에 '1'을 산술차(-)하여 상기 3비트 입력 오류검출코드($SEDC_{3-in}$)를 계산한다.

[0029] 바람직한 실시예에 있어서, 상기 3비트 입력 오류검출코드 생성수단이 상기 과도 2비트 입력 오류검출코드에 '1'을 산술차(-)하는 경우는, 상기 시프트/로테이트 연산이 왼쪽 시프트 연산이고, 상기 3비트 입력 데이터($A(A_2, A_1, A_0)$)의 가운데 비트(A_1)의 값이 '1'인 경우와 상기 시프트/로테이트 연산이 오른쪽 시프트 연산이고, 상기 3비트 입력 데이터($A(A_2, A_1, A_0)$)의 최상위 비트(A_2)의 값이 '1'인 경우이다.

[0030] 바람직한 실시예에 있어서, 상기 입력 오류검출코드 생성수단은, 상기 이진 입력 데이터가 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)일 경우, 상기 4비트 입력 데이터에 대한 3비트의 오류 검출코드인 4비트 입력 오류 검출코드($SEDC_{4-in}$)를 계산하여, 계산된 4비트 입력 오류 검출코드($SEDC_{4-in}$)를 상기 입력 오류검출코드로 생성하는 4비트 입력 오류검출코드 생성수단을 더 포함하고, 상기 3비트 입력 오류검출코드 생성수단은 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 하위 세 비트(A_2, A_1, A_0)를 입력으로 하는 상기 3비트 입력 오류검출코드 생성수단의 3비트 입력 오류검출코드($SEDC_{3-in}$)를 상기 4비트 입력 오류검출코드($SEDC_{4-in}$)의 하위 두 비트로 생성하되, 상기 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 상기 3비트 입력 오류검출코드($SEDC_{3-in}$)의 계산에 사용되는 시프트 비트는 상기 4비트 입력 데이터의 최상위 비트(A_3)로 대체하여 계산되고, 상기 4비트 입력 오류 검출코드($SEDC_{4-in}$)의 최상위 비트는, 상기 시프트/로테이트 연산이 왼쪽 시프트 연산일 경우, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 상위 두 번째 비트(A_2)의 논리 부정 값으로 생성되고, 상기 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 입력되는 시프트 비트의 논리 부정 값으로 생성되고, 상기 시프트/로테이트 연산이 왼쪽 로테이트 연산일 경우, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 상위 두 번째 비트(A_2)의 논리 부정 값으로 생성되고, 상기 시프트/로테이트 연산이 오른쪽 로테이트 연산일 경우, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 최하위 비트(A_0)의 논리 부정 값으로 생성된다.

[0031] 바람직한 실시예에 있어서, 상기 이진 입력 데이터가 n비트 입력 데이터일 경우 상기 n비트 입력 데이터에 대한 오류검출코드인 n비트 입력 오류검출코드($SEDC_{n-in}$)를 상기 입력 오류검출코드로 생성하는 n비트 입력 오류검출코드 생성수단을 더 포함하고, 상기 n비트 입력 오류검출코드 생성수단은 상기 입력 오류검출코드 생성수단들 각각의 오류검출코드($SEDC_{2-in}, SEDC_{3-in}, SEDC_{4-in}$)나, 상기 입력 오류검출코드 생성수단들 중 어느 하나의 입력 오류검출코드 생성수단의 오류검출코드와 하나 또는 복수 개의 3비트 입력 오류검출코드($SEDC_{3-in}$)의 조합을 상기 입력 오류검출코드로 생성한다.

[0032] 바람직한 실시예에 있어서, 상기 n비트 입력 데이터는 5비트 이상의 데이터이고, 상기 n비트 입력 오류검출코드

생성수단은 상기 입력 오류검출코드 생성수단들 중 어느 하나의 입력 오류검출코드 생성수단의 오류검출코드인 b비트 입력 오류검출코드($SEDC_{b-in}$)와 하나 또는 복수 개의 3비트 입력 오류검출코드($SEDC_{3-in}$)의 조합을 상기 입력 오류검출코드로 생성한다.

[0033] 바람직한 실시예에 있어서, 상기 n비트 입력 오류검출코드 생성수단은 상기 n비트 입력 데이터 중, 최상위 비트에서 하위 b비트만큼의 b비트 데이터에 대해 상기 b비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하고, 상기 b비트와 동일한 개수의 비트의 입력에 대해 오류검출코드를 생성하는 입력 오류검출코드 생성수단인 b비트 입력 오류검출코드 생성수단에 의해 상기 b비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하되, 상기 시프트/로테이트 연산이 왼쪽 시프트 연산일 경우, 상기 b비트 입력 오류검출코드 생성수단은 계산에 사용되는 시프트 비트를 상기 n비트 입력 데이터 중, 상기 b비트 데이터를 제외한 최상위 비트로 대체하여 상기 b비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하고, 상기 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 상기 b비트 입력 오류검출코드 생성수단은 입력되는 시프트 비트(si)를 그대로 이용하여 상기 b비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하며, 상기 시프트/로테이트 연산이 왼쪽 로테이트 연산일 경우, 상기 b비트 입력 오류검출코드 생성수단은 왼쪽 시프트 연산의 경우로 상기 b비트 입력 오류 검출코드($SEDC_{b-in}$)를 계산하여 생성하되, 계산에 사용되는 시프트 비트는 상기 n비트 입력 데이터 중, 상기 b비트 데이터를 제외한 최상위 비트로 대체하여 상기 b비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성하고, 상기 시프트/로테이트 연산이 오른쪽 로테이트 연산일 경우, 상기 b비트 입력 오류검출코드 생성수단은 오른쪽 시프트 연산의 경우로 상기 b비트 입력 오류 검출코드($SEDC_{b-in}$)를 계산하여 생성하되, 계산에 사용되는 시프트 비트는 상기 n비트 입력 데이터의 최하위 비트로 대체하여 상기 b비트 입력 오류 검출코드($SEDC_{b-in}$)를 생성한다.

[0034] 바람직한 실시예에 있어서, 상기 n비트 입력 오류검출코드 생성수단은 상기 3비트 입력 오류검출코드 생성수단을 이용하여 상기 n비트 입력 데이터 중, 최상위 비트에서 하위 b비트만큼을 제외한 'k'개의 연속적인 3비트 데이터 세트에 대해 각각 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고(여기서, $k=0, \dots, (a-1)$, a는 3비트 데이터 세트의 개수이다.), 상기 시프트/로테이트 연산이 왼쪽 시프트 연산일 경우, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 데이터 세트들 중, 상기 n비트 입력 데이터의 최하위 비트를 포함하는 최하위 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si)를 그대로 이용하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고, 그 외의 각 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si)를 바로 하위 3비트 데이터 세트의 최상위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하며, 상기 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 데이터 세트들 중, 상기 b비트 데이터의 바로 하위 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si)를 상기 b비트 데이터의 최하위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고, 그 외의 각 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si) 바로 상위 3비트 데이터 세트의 최하위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하며, 상기 시프트/로테이트 연산이 왼쪽 로테이트 연산일 경우, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 데이터 세트들 중, 상기 n비트 입력 데이터의 최하위 비트를 포함하는 최하위 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si)를 상기 n비트 입력 데이터의 최상위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고, 그 외의 각 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si)를 바로 하위 3비트 데이터 세트의 최상위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하며, 상기 시프트/로테이트 연산이 오른쪽 로테이트 연산일 경우, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 데이터 세트들 중, 상기 b비트 데이터의 바로 하위 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si)를 상기 b비트 데이터의 최하위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성하고, 그 외의 각 3비트 데이터 세트에 대해서는 입력되는 시프트 비트(si) 바로 상위 3비트 데이터 세트의 최하위 비트로 대체하여 상기 3비트 입력 오류 검출코드($SEDC_{3-in}$)를 생성한다.

[0035] 바람직한 실시예에 있어서, 상기 오류 검출기:는 상기 이진 출력 데이터가 2비트 출력 데이터일 경우, 상기 2비트 출력 데이터에 대한 2비트의 오류검출코드인 2비트 출력 오류검출코드($SEDC_{2-out}$)를 상기 출력 오류검출코드로 생성하는 2비트 출력 오류검출코드 생성수단; 및 상기 입력 오류검출코드와 상기 출력 오류검출코드의 동일성을

비교하여 동일하지 않을 경우 오류 발생신호를 출력하는 코드비교수단;을 포함한다.

- [0036] 바람직한 실시예에 있어서, 상기 2비트 출력 오류검출코드 생성수단은, 상기 2비트 출력 오류검출코드들($SEDC_{2-out}$)의 최상위 비트는 2비트 데이터 '00', '01', '10' 또는 '11'에 대해 비트들의 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 생성하고, 그 외의 경우에는 '0'으로 생성하며, 최하위 비트는 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성하고, 그 외의 경우에는 '1'로 생성하거나, 최상위 비트는 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 생성되고, 그 외의 경우에는 '1'로 생성하며, 최하위 비트는 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 생성하고, 그 외의 경우에는 '0'으로 생성하되, 상기 2비트 출력 오류검출코드($SEDC_{2-out}$)는 동일한 입력에 대해 상기 2비트 오류검출코드($SEDC_2$)와 동일한 오류검출코드로 생성된다.
- [0037] 바람직한 실시예에 있어서, 상기 오류 검출기는, 상기 이진 출력 데이터가 3비트 출력 데이터일 경우, 상기 3비트 출력 데이터에 대한 2비트의 오류검출코드인 3비트 출력 오류검출코드($SEDC_{3-out}$)를 상기 출력 오류검출코드로 생성하는 3비트 출력 오류검출코드 생성수단;을 더 포함한다.
- [0038] 바람직한 실시예에 있어서, 상기 3비트 출력 오류검출코드 생성수단은 상기 3비트 출력 데이터의 최상위 비트가 '0'일 경우에는, 중간비트 및 최하위 비트를 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 2비트 출력 오류검출코드($SEDC_{2-out}$)와 동일하게 상기 3비트 출력 오류검출코드($SEDC_{3-out}$)를 생성하고, 상기 3비트 출력 데이터의 최상위 비트가 '1'일 경우에는 중간비트 및 최하위 비트의 논리 부정값을 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 2비트 출력 오류검출코드($SEDC_{2-out}$)의 논리 부정값을 상기 3비트 출력 오류검출코드($SEDC_{3-out}$)로 생성한다.
- [0039] 바람직한 실시예에 있어서, 상기 오류 검출기는, 상기 이진 출력 데이터가 4비트 출력 데이터일 경우, 상기 4비트 출력 데이터에 대한 3비트의 오류검출코드인 4비트 출력 오류검출코드($SEDC_{4-out}$)를 상기 출력 오류검출코드로 생성하는 4비트 출력 오류검출코드 생성수단;을 더 포함한다.
- [0040] 바람직한 실시예에 있어서, 상기 4비트 출력 오류검출코드 생성수단은 상기 4비트 출력 오류검출코드($SEDC_{4-out}$)의 하위 2비트는 상기 4비트 출력 데이터 중, 최상위 비트를 제외한 3비트를 입력으로 하는 상기 3비트 출력 오류검출코드 생성수단의 3비트 출력 오류검출코드($SEDC_{3-out}$)로 생성하고, 상기 4비트 출력 오류검출코드($SEDC_{4-out}$)의 최상위 비트는 상기 4비트 출력 데이터의 최상위 비트의 논리 부정 값으로 생성한다.
- [0041] 바람직한 실시예에 있어서, 상기 오류 검출기는, 상기 이진 출력 데이터가 n비트 출력 데이터일 경우, 상기 n비트 출력 데이터에 대한 오류검출코드인 n비트 출력 오류검출코드($SEDC_{n-out}$)를 상기 출력 오류검출코드로 생성하는 n비트 출력 오류검출코드 생성수단;을 더 포함하고, 상기 n비트 출력 오류검출코드 생성수단은 상기 출력 오류검출코드 생성수단들 각각의 오류검출코드($SEDC_{2-out}$, $SEDC_{3-out}$, $SEDC_{4-out}$)나, 상기 출력 오류검출코드 생성수단들 중 어느 하나의 출력 오류검출코드 생성수단의 오류검출코드와 하나 또는 복수 개의 3비트 출력 오류검출코드($SEDC_{3-out}$)의 조합을 상기 출력 오류검출코드로 생성한다.
- [0042] 또한, 본 발명은 상기 오류 검출 장치와 프로그래머블 시프트/로테이트 연산 유닛을 포함하는 자가검사 프로그래머블 시프트/로테이트 연산 유닛(Self-checking Programmable Shift/Rotate Operations Unit)을 더 제공한다.
- [0043] 또한, 본 발명은 상기 오류 검출 장치 및 상기 시프트/로테이트 연산 유닛으로 기능하는 프로그래머블 산술/논리 연산유닛을 포함하는 자가검사 프로그래머블 산술/논리 연산유닛(Self-checking Programmable Arithmetic and Logic Unit)을 더 제공한다.
- [0044] 또한, 본 발명은 컴퓨터를 상기 오류 검출 장치로 기능하게 하는 오류 검출 프로그램이 저장된 컴퓨터로 읽을 수 있는 기록매체를 더 제공한다.
- [0045] 또한, 본 발명은 상기 오류 검출 프로그램이 저장되고 통신망을 통해 상기 오류 검출 프로그램을 전송할 수 있는 서버 시스템을 더 제공한다.

발명의 효과

- [0046] 본 발명은 다음과 같은 우수한 효과를 가진다.
- [0047] 먼저, 본 발명의 오류 검출 장치 및 자가 검사 프로그래머블 시프트/로테이트 연산 유닛에 의하면 입력 오류검출코드 생성기 및 오류 검출기가 각각 2비트, 3비트, 4비트 및 n비트의 입력에 대해 선형적이고 유동적인 확장형 오류검출코드를 각각 생성하여 동일성을 비교함으로써 시프트/로테이트 연산의 결함을 검출할 수 있는 장점이 있다.
- [0048] 또한, 본 발명의 오류 검출 장치 및 자가 검사 프로그래머블 시프트/로테이트 연산 유닛에 의하면, 오류검출코드의 생성이 입력 데이터에 대한 시프트/로테이트 연산과는 별개로 이루어지므로 입력 데이터의 연산의 속도나 레이턴시와 같은 성능의 하락 없이 결함 허용 가능 시스템을 구현할 수 있는 장점이 있다.

도면의 간단한 설명

- [0049] 도 1은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛을 설명하기 위한 도면,
- 도 2는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 2비트 오류검출코드 진리테이블을 보여주는 도면,
- 도 3은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 2비트 오류검출코드를 생성하는 논리회로의 일례를 보여주는 도면,
- 도 4는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 2비트 오류검출코드를 생성하는 논리회로의 다른 예를 보여주는 도면,
- 도 5는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 연산종류에 따른 2비트 입력 오류검출코드의 생성 로직을 설명하기 위한 도면,
- 도 6은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 연산종류에 따른 1비트 입력 오류검출코드의 생성 로직을 설명하기 위한 도면,
- 도 7은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 연산종류에 따른 3비트 입력 오류검출코드의 생성 로직을 설명하기 위한 도면,
- 도 8은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 연산종류에 따른 4비트 입력 오류검출코드의 생성 로직을 설명하기 위한 도면,
- 도 9는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 연산종류에 따른 n비트 입력 오류검출코드의 생성 로직을 설명하기 위한 도면
- 도 10은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 2비트 출력 오류검출코드 진리 테이블을 보여주는 도면,
- 도 11은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 3비트 출력 오류검출코드 진리 테이블을 보여주는 도면,
- 도 12는 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 3비트 출력 오류검출코드를 생성하기 위한 논리회로를 보여주는 도면,
- 도 13은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 4비트 출력 오류검출코드 진리 테이블을 보여주는 도면,
- 도 14는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 시프트/로테이트 연산 유닛의 4비트 출력 오류검출코드를 생성하기 위한 논리회로를 보여주는 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0050] 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 기재되어

나 사용된 의미를 고려하여 그 의미가 파악되어야 할 것이다.

[0051] 이하, 첨부한 도면에 도시된 바람직한 실시예들을 참조하여 본 발명의 기술적 구성을 상세하게 설명한다.

[0052] 그러나, 본 발명은 여기서 설명되는 실시예에 한정되지 않고 다른 형태로 구체화 될 수도 있다. 명세서 전체에 걸쳐 동일한 참조번호는 동일한 구성요소를 나타낸다.

[0053] 본 발명의 일 실시예에 따른 오류 검출 장치(100)는 시프트/로테이트 연산(Shift/Rotate Operations)을 수행하는 프로그래머블 시프트/로테이트 연산 유닛(Programmable Shift/Rotate Operations Unit, 10)의 이진 입력 데이터(A)와 상기 이진 입력 데이터(A)에 대해 시프트/로테이트 연산된 출력 데이터(F)에 대한 오류검출코드를 각각 생성하여 서로 동일성을 비교함으로써 상기 시프트/로테이트 연산 유닛(10)에 결함이 있는지 여부를 판단한다.

[0054] 또한, 상기 오류 검출 장치(100)는 상기 프로그래머블 시프트/로테이트 연산 유닛(10)과 하나의 자가검사 프로그래머블 시프트/로테이트 연산 유닛(Self-checking Programmable Shift/Rotate Operations Unit)으로 구성되어 제공될 수 있다.

[0055] 또한, 상기 프로그래머블 시프트/로테이트 연산 유닛(10)은 시프트/로테이트 연산을 수행하는 프로그래머블 산술/논리 연산유닛(Programmable Arithmetic and Logic Unit)으로 대체될 수 있으며, 이 경우, 상기 오류 검출 장치(100)는 상기 프로그래머블 산술/논리 연산유닛과 함께 하나의 자가검사 프로그래머블 산술/논리 연산유닛(Self-checking Programmable Arithmetic and Logic Unit)으로 구성되어 제공될 수 있다.

[0056] 또한, 상기 오류 검출 장치는 실질적으로 오류 검출 프로그램에 의해 기능하는 컴퓨터일 수 있고, 상기 오류 검출 프로그램은 컴퓨터로 읽을 수 있는 기록매체나 통신망을 통해 통신가능한 서버 시스템에 의해 상기 컴퓨터에 전송되고, 상기 컴퓨터는 상기 오류 검출 프로그램에 의해 기능하여 시프트/로테이트 연산에 대한 오류 검출을 수행할 수 있다.

[0057] 이하에서는 본 발명의 일 실시예에 따른 오류 검출 장치(100)의 구성을 상세히 설명하기로 한다.

[0058] 도 1을 참조하면, 본 발명의 일 실시예에 따른 오류 검출 장치(100)는 입력 오류검출코드 생성기(110) 및 오류 검출기(120)를 포함하여 이루어진다.

[0059] 또한, 상기 입력 오류검출코드 생성기(110)는 상기 프로그래머블 시프트/로테이트 연산 유닛(10)의 이진 입력 데이터(A)를 입력받아 선택된 시프트/로테이트 연산의 결과에 대한 오류검출코드인 입력 오류검출코드를 생성하고, 상기 오류 검출기(120)는 상기 프로그래머블 시프트/로테이트 연산 유닛(10)의 이진 출력 데이터(F)를 입력받아 상기 이진 출력 데이터에 대한 오류검출코드인 출력 오류검출코드를 생성하며, 상기 입력 오류검출코드와 상기 출력 오류검출코드의 동일성을 판단하여 오류검출결과(Fault Detection Result)를 출력한다.

[0060] 즉, 본 발명의 일 실시예에 따른 오류 검출 장치(100)는 상기 프로그래머블 시프트/로테이트 연산 유닛(10)의 연산에 대해서 관여하지 않으므로 결함 허용 가능한 구조(fault-tolerant reconfigurable architecture)이다.

[0061] 또한, 상기 입력 오류검출코드 생성기(110)와 상기 오류 검출기(120)는 상기 이진 입력 데이터(A)나 상기 이진 출력 데이터(F)의 길이에 선형적으로 가변하는 확장형 오류검출코드(SEDC:scalable error detection coding)를 생성한다.

[0062] 상기 입력 오류검출코드 생성기(100)는 2비트 오류검출코드 진리테이블(truth table, 111) 및 상기 이진 입력 데이터(A)에 대해 입력 오류검출코드를 생성하는 입력 오류검출코드 생성수단을 포함하여 이루어진다.

[0063] 또한, 도 2를 참조하면, 상기 2비트 오류검출코드 진리테이블(111)은 2비트 데이터(A, 111a)에 대응하는 2비트의 오류검출코드인 2비트 오류검출코드들(SEDC₂, 111b, 111c)이 계산되어 저장된다

[0064] 여기서, 상기 2비트 데이터(111a)는 상기 프로그래머블 시프트/로테이트 연산 유닛의 이진 입력 데이터(A)와 대응하는 데이터로써, 상기 2비트 오류검출코드들(SEDC₂)은 상기 프로그래머블 시프트/로테이트 연산 유닛의 이진 입력 데이터(A)의 시프트/로테이트 연산 결과에 대해 생성되는 2비트 입력 오류검출코드(SEDC_{2-in})와는 서로 구분

된다.

[0065] 또한, 상기 2비트 오류검출코드들(SED_{C2})은 미리 계산되어 저장될 수 있고, 도 3 및 도 4에 도시한 논리회로에 의해 오류검출시 실시간으로 계산될 수 있다.

[0066] 또한, 상기 2비트 오류검출코드들(SED_{C2})은 서로 다른 두 가지 케이스(Case1, Case2)로 계산될 수 있다.

[0067] 또한, 상기 2비트 오류검출코드(SED_{C2})의 첫 번째 케이스(Case1, 111b)는 아래의 수학식1에 의해 계산된다.

수학식 1

$$C_1 = \begin{cases} 1, & \text{if number of 0's in } A \geq \text{number of 1's in } A \\ 0, & \text{otherwise} \end{cases}$$

$$C_0 = \begin{cases} 0, & \text{if number of 0's in } A = \text{number of 1's in } A \\ 1, & \text{otherwise} \end{cases}$$

[0069] 즉, 첫 번째 케이스(111b)의 경우, 2비트 오류검출코드(SED_{C2})의 최상위 비트(C₁)는 상기 2비트 데이터(111a) '00', '01', '10' 또는 '11'에 대해 비트들의 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 계산되어 저장되고, 그 외의 경우에는 '0'으로 계산되어 저장되며, 최하위 비트(C₀)는 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 계산되어 저장되고, 그 외의 경우에는 '1'로 계산되어 저장된다.

[0070] 또한, 상기 첫 번째 케이스(111b)는 도 3에 도시한 제1 2비트 오류검출코드 생성 논리회로(111-1)에 의해 생성되며, 상기 제1 2비트 오류검출코드 생성 논리회로(111-1)은 2비트 입력에 대해 exclusive-NOT-OR 연산을 수행하는 하나의 XNOR게이트(111-1a)와 2비트 입력에 대해 NOT-AND 연산을 수행하는 하나의 NAND게이트(111-1b)를 이용하여 구현하였다.

[0071] 그러나, 상기 제1 2비트 오류검출코드 생성 논리회로(111-1)의 구성은 하나의 예시일 뿐, 다양한 논리 게이트의 조합으로 구현할 수 있음은 물론이다.

[0072] 또한, 상기 2비트 오류검출코드(SED_{C2})의 두 번째 케이스(Case2, 111c)는 첫 번째 케이스(111b)와 비교하여 상위 비트와 하위 비트가 서로 역전된 코드이다.

[0073] 즉, 상기 두 번째 케이스(111c)의 경우, 2비트 오류검출코드(SED_{C2})의 최상위 비트(C₁)는 상기 2비트 데이터(111a) '00', '01', '10' 또는 '11'에 대해 비트들의 '0'의 개수와 '1'의 개수가 동일할 경우 '0'으로 계산되어 저장되고, 그 외의 경우에는 '1'로 계산되어 저장되고, 최하위 비트(C₀)는 '0'의 개수가 '1'의 개수보다 크거나 같을 경우 '1'로 계산되어 저장되고, 그 외의 경우에는 '0'으로 계산되어 저장된다.

[0074] 또한, 상기 두 번째 케이스(111c)의 경우, 도 4에 도시한 제2 2비트 오류검출코드 생성 논리회로(111-1)에 의해 생성되며, 제1 2비트 오류검출코드 생성 논리회로(111-1)와 구성은 서로 동일하나, XNOR게이트(111-2a)의 출력이 상기 2비트 오류검출코드(SED_{C2})의 상위 비트로 생성되고, NAND게이트(111-2b)의 출력이 상기 2비트 오류검출코드(SED_{C2})의 하위 비트로 생성된다.

[0075] 또한, 상기 제2 2비트 오류검출코드 생성 논리회로(111-2) 역시 다양한 논리 게이트의 조합으로 구현될 수 있다.

[0076] 상기 입력 오류검출코드 생성수단은 상기 이진 입력 데이터(A)의 길이에 따라 가변하는 입력 오류검출코드를 생성하며, 2비트 입력 데이터에 대한 오류검출코드인 2비트 입력 오류검출코드(SED_{C2_in})를 생성하는 2비트 입력 오류검출코드 생성수단, 3비트 입력 데이터에 대한 오류검출코드인 3비트 입력 오류검출코드(SED_{C3_in})를 생성하는 3비트 입력 오류검출코드 생성수단, 4비트 입력 데이터에 대한 오류검출코드인 4비트 입력 오류검출코드(SED_{C4_in})를 생성하는 4비트 입력 오류검출코드 생성수단 및 n비트 입력 데이터에 대한 오류검출코드인 n비트 입

력 오류검출코드($SEDC_{n_in}$)를 생성하는 n 비트 입력 오류검출코드 생성수단을 포함한다.

[0077] 그러나, 상기 입력 오류검출코드 생성수단은 상기 이진 입력 데이터(A)의 길이가 정해져 있는 경우 상기 2비트 입력 오류검출코드 생성수단, 상기 3비트 입력 오류검출코드 생성수단 및 상기 4비트 입력 오류검출코드 생성수단 중 해당 길이의 입력 데이터에 대해 오류검출코드를 생성하는 하나의 입력 오류검출코드 생성수단이나 상기 입력 오류검출코드 생성수단들 중, 적어도 두 개의 입력 오류검출코드 생성수단의 조합으로 구성될 수도 있다.

[0078] 도 5는 상기 2비트 입력 오류검출코드 생성수단이 상기 2비트 입력 오류검출코드($SEDC_{2_in}$)($C_{1(2bit)}, C_{0(2bit)}$)를 생성하는 로직(112)을 설명하기 위한 것으로, 상기 2비트 입력 오류검출코드 생성수단은 상기 이진 입력 데이터(A), 시프트/로테이트 연산의 종류를 선택하기 위한 연산선택신호(opSR) 및 연산에 이용될 시프트 비트(si)를 입력받고 상기 2비트 입력 오류검출코드($SEDC_{2_in}$)를 생성한다.

[0079] 또한, 상기 시프트/로테이트 연산은 왼쪽 시프트 연산(Shift Left), 오른쪽 시프트 연산(Shift Right), 왼쪽 로테이트 연산(Rotate Left) 및 오른쪽 로테이트 연산(Rotate Right)으로 구분된다.

[0080] 한편, 상기 각 시프트/로테이트 연산을 예를 들어 설명하면, 상기 이진 입력 데이터(A)가 '01'이고 시프트 비트(si)가 '1'일 경우, 상기 왼쪽 시프트 연산은 하위비트 '1'이 상위비트로 시프트되고 시프트 비트 '1'이 하위비트가 되어 '11'으로 계산되고, 상기 오른쪽 시프트 연산의 결과는 상위비트 '0'이 하위비트로 시프트되고 시프트 비트 '1'이 상위비트가 되어 '10'으로 계산된다.

[0081] 또한, 상기 왼쪽 로테이트 연산의 경우에는 하위비트 '1'이 왼쪽 즉, 상위비트로 시프트되고 상위비트 '0'은 하위비트로 로테이트되어 '10'으로 계산되고, 상기 오른쪽 로테이트 연산은 상위비트 '1'이 오른쪽 즉, 하위비트로 시프트되고 하위비트 '1'은 상위 비트로 로테이트되어 '10'으로 계산된다.

[0082] 즉, 상기 2비트 입력 오류검출코드 생성수단은 2비트의 이진 입력 데이터(A)와 시프트 비트(si)를 입력받아 해당 시프트/로테이트 연산결과에 대응하는 데이터에 대해 오류검출코드를 계산하게 된다.

[0083] 더욱 자세하게는 상기 2비트 입력 오류검출코드 생성수단은 상기 2비트 입력 오류검출코드 생성수단은, 상기 시프트/로테이트 연산이 왼쪽 시프트 연산(opSR=0)일 경우, 상기 2비트 입력 데이터($A(A_1, A_0)$)의 하위 비트(A_0)를 상위 비트로 하고, 상기 시프트 비트(si)를 하위 비트로 하는 2비트 데이터(A_0, si)와 대응하는 2비트 오류검출코드($SEDC_2$)를 상기 2비트 입력 오류검출코드($SEDC_{2_in}$)로 생성한다.

[0084] 또한, 상기 시프트/로테이트 연산이 오른쪽 시프트 연산(opSR=1)일 경우, 상기 2비트 입력 데이터($A(A_1, A_0)$)의 상위 비트(A_1)를 하위 비트로 하고, 상기 시프트 비트(si)를 상위 비트로 하는 2비트 데이터(si, A_1)의 2비트 오류검출코드($SEDC_2$)를 상기 2비트 입력 오류검출코드($SEDC_{2_in}$)로 생성한다.

[0085] 또한, 상기 2비트 입력 오류검출코드 생성수단은 상기 시프트/로테이트 연산이 왼쪽 로테이트 연산(opSR=2) 또는 오른쪽 로테이트 연산(opSR=2)일 경우 상기 시프트 비트에 관계없이 상기 2비트 입력 데이터(A)에 대응하는 2비트 오류검출코드($SEDC_2$)를 상기 2비트 입력 오류검출코드($SEDC_{2_in}$)로 생성한다.

[0086] 그 이유는 상기 시프트/로테이트 연산의 경우 시프트 비트를 계산에 이용하지 않기 때문이다.

[0087] 도 6은 상기 이진 입력 데이터(A)가 1비트 데이터일 경우 1비트 입력 오류검출코드($C_{0(1bit)}$)를 생성하는 로직(113)을 설명하기 위한 것으로 상기 1비트 입력 오류검출 코드는 실질적으로 상기 2비트 입력 오류검출코드 생성수단에 의해 생성된다.

[0088] 그러나, 상기 1비트 입력 오류검출코드($C_{0(1bit)}$)는 별도의 1비트 입력 오류검출코드 생성수단에 의해 생성될 수 있음은 물론이다.

[0089] 또한, 상기 1비트 입력 오류검출코드($C_{0(1bit)}$)를 생성할 경우, 상기 2비트 입력 오류검출코드 생성수단은 상기 이진 입력 데이터(A)가 1비트 데이터이고, 상기 시프트/로테이트 연산이 시프트 연산일 경우, 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 시프트 비트(si)로 하는 2비트 데이터에 대해 2비트 오류검출코드($SEDC_2$)의 계산을 수행한 후, 계산된 2비트 오류검출코드($SEDC_2$) 중, 하위 비트만을 1비트의 입력 오류검출코드로 생성하고, 상기 시프트/로테이트 연산이 로테이트 연산일 경우, 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 1비트

입력 데이터로 하는 2비트 데이터에 대해 2비트 오류검출코드(SED_{C2})의 계산을 수행한 후, 계산된 2비트 오류검출코드(SED_{C2}) 중, 하위 비트만을 상기 입력 오류검출코드로 생성한다.

[0090] 그러나 상기 2비트 입력 오류검출코드 생성수단은 상위 비트를 '1'로 가정하여 1비트의 입력 오류검출코드를 생성할 수도 있다.

[0091] 또한, 상기 2비트 입력 오류검출코드 생성수단은 1비트의 입력 데이터에 대해 2비트의 입력 오류검출코드를 생성할 수도 있으며, 이 경우에는 상기 오류 검출기(120)가 입력 오류검출코드와 출력 오류검출코드의 최하위 비트만을 서로 비교하여 오류검출결과를 출력할 수 있다.

[0092] 도 7은 상기 3비트 입력 오류검출코드 생성수단이 상기 3비트 입력 오류 검출코드(SED_{C3-in})(C_{1(3bit)}, C_{0(3bit)})를 생성하는 생성 로직(114)을 보여주는 도면이다.

[0093] 도 7을 참조하면, 상기 3비트 입력 오류검출코드 생성수단은 3비트의 이진 입력 데이터(A(A₂, A₁, A₀)), 시프트 비트(si) 및 연산선택신호(opSR)를 입력받아 상기 3비트 입력 오류 검출코드(SED_{C3-in})를 생성한다.

[0094] 먼저, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 입력 데이터(A(A₂, A₁, A₀))의 하위 두 비트((A₁, A₀), (B₁, B₀))를 입력으로 하는 상기 2비트 입력 오류검출코드 생성수단의 2비트 입력 오류검출코드(SED_{C2-in}(A(A₁, A₀), si, opSR))를 과도 2비트 입력 오류검출코드(C_{1(2bit)}, C_{0(2bit)})로 생성한다.

[0095] 다음, 상기 연산선택신호(opSR)와 상기 3비트 입력 데이터(A(A₂, A₁, A₀))의 상위 두 비트(A₂, A₁) 값에 따라 상기 과도 2비트 입력 오류검출코드 자체 또는 상기 과도 2비트 입력 오류검출코드에 '1'을 산술차(-)하여 상기 3비트 입력 오류검출코드(SED_{C3-in})를 생성한다.

[0096] 여기서, '1'을 산술차(-)하는 경우는, 도 7에 도시한 바와 같이, 상기 연산선택신호(opSR)가 왼쪽 시프트 연산(opSR=0)이고, 상기 3비트 입력 데이터(A(A₂, A₁, A₀))의 가운데 비트(A₁)의 값이 '1'인 경우와 상기 연산선택신호(opSR)가 오른쪽 시프트 연산(opSR=1)이고, 상기 3비트 입력 데이터(A(A₂, A₁, A₀))의 최상위 비트(A₂)의 값이 '1'인 경우이며, 상기 연산선택신호(opSR)가 로테이트 연산(opSR=2,3)일 경우에는 상기 3비트 입력 데이터(A(A₂, A₁, A₀))의 상위 두 비트(A₂, A₁) 값에 관계없이(don't care) 상기 과도 2비트 입력 오류검출코드를 그대로 상기 3비트 입력 오류검출코드(SED_{C3-in})로 생성한다.

[0097] 도 8은 상기 4비트 입력 오류검출코드 생성수단이 상기 4비트 입력 오류 검출코드(SED_{C4-in})(C_{2(4bit)}, C_{1(4bit)}, C_{0(4bit)})를 생성하는 생성 로직(115)을 보여주는 도면이다.

[0098] 도 8을 참조하면, 상기 4비트 입력 오류검출코드 생성수단은 4비트의 이진 입력 데이터(A(A₃, A₂, A₁, A₀)), 시프트 비트(si) 및 연산선택신호(opSR)를 입력받아 상기 4비트 입력 오류 검출코드(SED_{C4-in})를 생성한다.

[0099] 먼저, 상기 4비트 입력 오류검출코드 생성수단은 상기 4비트 입력 데이터(A(A₃, A₂, A₁, A₀))의 하위 세 비트((A₂, A₁, A₀))를 입력으로 하는 상기 3비트 입력 오류검출코드 생성수단의 3비트 입력 오류검출코드(SED_{C3-in})(C_{1(3bit)}, C_{0(3bit)})를 상기 4비트 입력 오류검출코드(SED_{C4-in})의 하위 두 비트(C_{1(4bit)}, C_{0(4bit)})로 생성한다.

[0100] 다만, 여기서, 시프트/로테이트 연산의 종류가 오른쪽 시프트 연산(opSR=1)일 경우에는 상기 시프트 비트(si)를 상기 4비트 입력 데이터(A(A₃, A₂, A₁, A₀))의 최상위 비트(A₃)를 대체 시프트 비트(sibit)로 대체하여 3비트 입력 오류검출코드(SED_{C3-in})를 계산한다.

[0101] 다음, 상기 4비트 입력 오류검출코드 생성수단은 상기 4비트 입력 오류 검출코드(SED_{C4-in})의 최상위 비트(C_{2(4bit)})를 계산하여 생성한다.

[0102] 여기서 상기 4비트 입력 오류 검출코드(SED_{C4-in})의 최상위 비트(C_{2(4bit)})는 상기 시프트/로테이트 연산이 왼쪽 시프트 연산(opSR=0)일 경우, 상기 4비트 입력 데이터(A(A₃, A₂, A₁, A₀))의 상위 두 번째 비트(A₂)의 논리 부정 값(NOT(A₂))으로 생성되고, 상기 시프트/로테이트 연산이 오른쪽 시프트 연산(opSR=1)일 경우, 입력되는 시프트

비트(s_i)의 논리 부정 값($\text{NOT}(s_i)$)으로 생성되고, 상기 시프트/로테이트 연산이 왼쪽 로테이트 연산($\text{opSR}=2$)일 경우, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 상위 두 번째 비트(A_2)의 논리 부정 값($\text{NOR}(A_2)$)으로 생성되고, 상기 시프트/로테이트 연산이 오른쪽 로테이트 연산($\text{opSR}=3$)일 경우, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0)$)의 최하위 비트(A_0)의 논리 부정 값($\text{NOT}(A_0)$)으로 생성한다.

[0103] 도 9는 상기 n 비트 입력 오류검출코드 생성수단이 상기 n 비트 입력 오류검출코드($\text{SEDC}_{n\text{-in}}$)를 생성하는 로직 (116)을 보여주는 도면이다.

[0104] 또한, 상기 n 비트 입력 오류검출코드 생성수단은 상기 이진 입력 데이터(A)가 5비트 이상인 경우에 입력 오류검출코드를 생성하며, 상기 이진 입력 데이터(A)가 5비트 미만의 데이터일 경우에는 상기 2비트 입력 오류검출코드 생성수단, 상기 3비트 입력 오류검출코드 생성수단 또는 상기 4비트 입력 오류검출코드 생성수단 각각에 의해 입력 오류검출코드를 생성한다.

[0105] 또한, 상기 n 비트 입력 오류검출코드 생성수단은 상기 입력 오류검출코드 생성수단들 중 어느 하나의 입력 오류검출코드 생성수단의 오류검출코드인 b 비트 입력 오류검출코드($\text{SEDC}_{b\text{-in}}$)와 하나 또는 복수 개의 3비트 입력 오류검출코드($\text{SEDC}_{3\text{-in}}$)의 조합을 상기 입력 오류검출코드로 생성한다.

[0106] 또한, 상기 n 비트 입력 오류검출코드 생성수단은 아래의 수학적식 2와 같이 입력되는 입력 데이터의 길이에 따라 생성될 입력 오류검출코드의 길이를 계산한다.

수학적식 2

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

[0107]

[0108] 여기서, m 은 상기 n 비트 입력 오류검출코드($\text{SEDC}_{n\text{-in}}$)의 길이이고, 파라미터 ' a '는 '0'에서부터 ' ∞ '까지의 정수 중 어느 하나이며, 상기 파라미터 ' a '는 아래의 수학적식 3에 의해 계산된다.

수학적식 3

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

[0109]

[0110] 여기서, n 은 상기 입력데이터의 길이이고, 파라미터 ' b '는 '2', '3' 또는 '4'의 정수이며, $\max(b)$ 는 정수 '2', '3' 및 '4' 중, 상기 파라미터 ' a '가 정수가 되게 하는 최대값이다.

[0111] 예를 들어, 상기 입력 데이터(n)의 길이가 '8비트'일 경우, 상기 수학적식 3의 $\max(b)$ 는 정수 '2'로 선택되고, 상기 파라미터 ' a '는 정수 '2'로 계산된다.

[0112] 또한, 상기 파라미터 ' a '가 정수 '2'로 계산되었으므로, 수학적식 2에 대입하면, 가우스 ' $\log_2 3$ '은 정수 '2'가 되고, 여기에 정수 '4'를 더하면 정수 '6'이 되어, 상기 입력 입력데이터의 길이가 '8비트'일 경우, 오류검출코드의 길이는 '6비트'로 계산되는 것이다.

[0113] 또한, 상기 파라미터 ' b '는 상기 2비트 입력 오류검출코드 생성수단, 상기 3비트 입력 오류검출코드 생성수단 및 상기 4비트 입력 오류검출코드 생성수단 중, 상기 b 비트 입력 오류검출코드($\text{SEDC}_{b\text{-in}}$)를 생성하기 위한 입력 오류검출코드 생성수단(이하, " b 비트 입력 오류검출코드 생성수단"이라 함)을 지정하기 위한 파라미터이고, 상기 파라미터 ' a '는 상기 b 비트 입력 오류검출코드($\text{SEDC}_{b\text{-in}}$)와 조합될 3비트 입력 오류검출코드($\text{SEDC}_{3\text{-in}}$)의 개수를 의미한다.

[0114] 또한, 상기 b 비트 입력 오류검출코드 생성수단은 n 비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b}, \dots)$) 중, 최상위 비트부터

'b'비트 만큼의 b비트 데이터에 대해 상기 b비트 입력 오류검출코드($SEDC_{b-in}$)를 생성하되 시프트/로테이트 연산의 종류가 왼쪽 시프트 연산, 왼쪽 로테이트 연산 및 오른쪽 로테이트 연산일 경우, 계산에 사용될 시프트 비트(si)를 제1 대체 시프트 비트($sibit1$)으로 대체하여 입력 오류검출코드를 계산한다.

[0115] 또한, 상기 왼쪽 로테이트 연산의 경우, 비트의 이동이 상기 왼쪽 시프트 연산과 동일하므로 상기 b비트 입력 오류검출코드 생성수단은 선택된 시프트/로테이트 연산이 왼쪽 로테이트 연산($opSRbits='10'$)일 경우는 왼쪽 시프트 연산의 경우($opSR=0$)로 입력 오류검출코드를 계산하고, 상기 오른쪽 로테이트 연산의 경우, 비트의 이동이 상기 오른쪽 시프트 연산과 동일하므로 상기 b비트 입력 오류검출코드 생성수단은 선택된 시프트/로테이트 연산이 오른쪽 로테이트 연산($opSRbits='11'$)일 경우는 오른쪽 시프트 연산의 경우($opSR=1$)로 입력 오류검출코드를 계산한다.

[0116] 또한, 상기 제1 대체 시프트 비트($sibit1$)는 시프트/로테이트 연산이 왼쪽 시프트 연산일 경우 상기 n비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b},)$) 중 최상위 비트부터 b비트 만큼의 데이터를 제외한 입력 데이터의 최상위 비트로 설정된다.

[0117] 다시 말해서 상기 n비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b},)$) 중, 상기 b비트 데이터의 바로 하위 비트(A_{n-b-1})가 상기 제1 대체 시프트 비트($sibit1$)로 설정되는 것이다.

[0118] 그 이유는 왼쪽 시프트 연산일 경우 상기 b비트 데이터의 최하위 비트는 상기 b비트 데이터의 바로 하위 비트가 시프트되어 변화하기 때문이다.

[0119] 또한, 상기 제1 대체 시프트 비트($sibit1$)는 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 입력되는 시프트 비트(si)가 상기 b비트 데이터의 최상위 비트가 되므로 입력되는 시프트 비트(si)와 동일하다.

[0120] 또한, 상기 제1 대체 시프트 비트($sibit1$)는 시프트/로테이트 연산이 왼쪽 로테이트 연산일 경우, 상기 왼쪽 시프트 연산과 동일하게 상기 b비트 데이터의 바로 하위 비트(A_{n-b-1})로 설정된다.

[0121] 또한, 상기 제1 대체 시프트 비트($sibit1$)는 시프트/로테이트 연산이 오른쪽 로테이트 연산일 경우, 상기 n비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b},)$)의 최하위 비트(A_{n-b})(A_0)가 상기 b비트 데이터의 최상위 비트에 대체되므로 상기 n비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b},)$)의 최하위 비트(A_0)로 설정된다.

[0122] 또한, 상기 n비트 입력 오류검출코드 생성수단은 상기 n비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b},)$) 들 중, 입력 오류검출코드가 생성된 b비트 데이터를 제외한 3비트 데이터 세트에 대해 각각 3비트 입력 오류검출코드($SEDC_{3-in}$)를 생성한다.

[0123] 예를 들어, 상기 n비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b},)$)가 8비트 데이터($A_7, A_6, A_5, A_4, A_3, A_2, A_1, A_0$)라면, 최상위 비트를 포함하는 2비트 데이터(A_7, A_6)가 상기 b비트 데이터가 되고 나머지 6비트 데이터($A_5, A_4, A_3, A_2, A_1, A_0$)는 두 개의 3비트 데이터 세트(A_5, A_4, A_3)(A_2, A_1, A_0)로 구분된다.

[0124] 또한, 상기 3비트 데이터 세트는 파라미터 'k'개로 이루어지고, 여기서, 상기 파라미터 'k'는 '0'에서 '(a-1)'까지의 정수이다. 즉, 상기 n비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b},)$)가 8비트 데이터일 경우, 상기 파라미터 'a'는 '2'로 계산되고, 상기 파라미터 'k'는 '0'과 '1'의 두 개의 값을 가지며, 두 개의 3비트 데이터 세트가 존재함을 의미한다.

[0125] 또한, 본 발명에서는 상기 파라미터 'k'가 '0'인 3비트 데이터 세트가 상기 b비트 데이터와 연결되는 데이터이고 나머지 3비트 데이터 세트는 상기 파라미터 'k'가 증가하며 순차적으로 연결되는 데이터 세트인 것으로 정의한다.

[0126] 예를 들어, 상기 n비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b},)$)가 8비트 데이터($A_7, A_6, A_5, A_4, A_3, A_2, A_1, A_0$) 라면, 최상위 비트(A_7)와 일곱 번째 비트(A_6)은 상기 b비트 데이터가 되고, 여섯 번째, 다섯 번째 및 네 번째 비트(A_5, A_4, A_3)는 상기 파라미터 'k'가 '0'인 3비트 데이터 세트가 되고, 세 번째, 두 번째 및 최하위 비트(A_2, A_1, A_0)는 'k'가 '1'인 3비트 데이터 세트가 된다.

[0127] 또한, 상기 n비트 입력 오류검출코드 생성수단은 각 3비트 데이터 세트에 대해 상기 3비트 오류 검출코드 생성

수단을 이용하여 3비트 입력 오류검출코드($SEDC_{3-in}$)를 생성하되, 오류검출코드 계산에 이용되는 시프트 비트를 제2 대체 시프트 비트($sibit2$)로 대체하여 생성한다.

[0128] 또한, 상기 3비트 입력 오류검출코드 생성수단이 상기 각 3비트 데이터 세트에 대해 오류검출코드를 생성할 때, 선택된 시프트/로테이트 연산이 왼쪽 로테이트 연산($opSRbits='10'$)일 경우에는 왼쪽 시프트 연산의 경우($opSR=0$)로 입력 오류검출코드를 계산하고, 선택된 시프트/로테이트 연산이 오른쪽 로테이트 연산($opSRbits='11'$)일 경우는 오른쪽 시프트 연산의 경우($opSR=1$)로 입력 오류검출코드를 계산한다.

[0129] 또한, 상기 3비트 입력 오류검출코드 생성수단은 선택된 시프트/로테이트 연산이 왼쪽 시프트 연산일 경우, 상기 n 비트 입력 데이터의 최하위 비트를 포함하는 데이터 세트, 즉, 상기 파라미터 ' k '가 ' $(a-1)$ '인 데이터 세트에 대해서는 시프트 비트(si)를 그대로 이용하여 3비트 입력 오류검출 코드($SEDC_{3-in}$)를 생성하고, 그 외의 데이터 세트(' $k \neq (a-1)$ ')에 대해서는 상기 시프트 비트(si)를 해당 3비트 데이터 세트의 바로 하위 3비트 데이터 세트의 최상위 비트($A_{n-b(3 \times (k+1))-1}$)로 대체하여 계산한다.

[0130] 다시 말해서, 상기 제2 대체 시프트 비트($sibit2$)는 선택된 시프트/로테이트 연산이 왼쪽 시프트 연산이고, 상기 파라미터 ' k '가 ' $(a-1)$ '일 경우, 입력되는 시프트 비트(si)와 동일한 값으로 설정되고, 상기 파라미터 ' k '가 ' $(a-1)$ '이 아닐 경우, 해당 3비트 데이터 세트의 바로 하위 3비트 데이터 세트의 최상위 비트($A_{n-b(3 \times (k+1))-1}$)로 설정되는 것이다.

[0131] 또한, 상기 3비트 입력 오류검출코드 생성수단은 선택된 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 상기 파라미터 ' k '가 ' 0 '인 3비트 데이터 세트에 대해서는 상기 시프트 비트를 상기 b 비트 데이터의 최하위 비트로 대체하여 상기 3비트 입력 오류검출 코드($SEDC_{3-in}$)를 계산하고, 상기 파라미터 ' k '가 ' 0 '이 아닌 3비트 데이터 세트에 대해서는 상기 시프트 비트를 바로 상위 3비트 데이터 세트의 최하위 비트($A_{n-b-(3 \times k)}$)로 대체하여 상기 3비트 입력 오류검출 코드($SEDC_{3-in}$)를 계산한다.

[0132] 또한, 선택된 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 상기 시프트 비트는 상기 파라미터 ' k '에 관계없이 비트 ' $A_{n-b-(3 \times k)}$ '인 제2 대체 시프트 비트($sibit2$)로 대체되는 것으로 설명할 수 있다.

[0133] 또한, 상기 3비트 입력 오류검출코드 생성수단은 선택된 시프트/로테이트 연산이 왼쪽 로테이트 연산일 경우, 상기 n 비트 입력 데이터의 최하위 비트를 포함하는 3비트 데이터 세트, 즉, 상기 파라미터 ' k '가 ' $(a-1)$ '에 해당하는 데이터 세트에 대해서는 상기 시프트 비트를 상기 n 비트 입력 데이터의 최상위 비트(A_{msb})로 대체하여 상기 3비트 입력 오류검출 코드($SEDC_{3-in}$)를 계산하고, 그외의 데이터 세트에 대해서는 상기 시프트 비트를 바로 하위 3비트 데이터 세트의 최상위 비트($A_{n-b(3 \times (k+1))-1}$)로 대체하여 계산한다.

[0134] 또한, 상기 3비트 입력 오류검출코드 생성수단은 선택된 시프트/로테이트 연산이 오른쪽 로테이트 연산일 경우, 상기 파라미터 ' k '가 ' 0 '인 3비트 데이터 세트에 대해서는 상기 시프트 비트를 상기 b 비트 데이터의 최하위 비트로 대체하여 상기 3비트 입력 오류검출 코드($SEDC_{3-in}$)를 계산하고, 상기 파라미터 ' k '가 ' 0 '이 아닌 3비트 데이터 세트에 대해서는 상기 시프트 비트를 바로 상위 3비트 데이터 세트의 최하위 비트($A_{n-b-(3 \times k)}$)로 대체하여 상기 3비트 입력 오류검출 코드($SEDC_{3-in}$)를 계산한다.

[0135] 또한, 선택된 시프트/로테이트 연산이 오른쪽 시프트 연산일 경우, 상기 시프트 비트는 상기 파라미터 ' k '에 관계없이 비트 ' $A_{n-b-(3 \times k)}$ '인 제2 대체 시프트 비트($sibit2$)로 대체되는 것으로 설명할 수 있다.

[0136] 상기 오류 검출기(120)는 상기 프로그래머블 시프트/로테이트 연산 유닛(10)에서 출력되는 연산결과인 출력 데이터(F)에 대해 출력 오류검출코드를 생성하며, 상기 입력 오류검출코드 생성기(110)에서 생성된 입력 오류검출코드와 상기 출력 오류검출코드를 서로 비교하여 동일하지 않을 경우 오류 발생 신호를 출력한다.

[0137] 또한, 상기 오류 검출기(120)는 상기 출력 데이터가 2비트의 출력데이터일 경우, 상기 출력 데이터에 대한 2비트의 오류검출코드인 2비트 출력 오류검출코드($SEDC_{2-out}$)를 상기 출력 오류검출코드로 출력하는 2비트 출력 오류검출코드 생성수단과 상기 입력 오류검출코드와 상기 출력 오류검출코드의 동일성을 비교하여 오류 발생 신호를 출력하는 코드비교수단을 포함한다.

[0138] 또한, 도 10은 상기 출력 데이터(F,121a)에 대한 2비트 출력 오류검출코드(SED_{C2-out})의 진리 테이블(121)을 보여주는 것으로 두 가지 케이스(121b,121c)(C_{1'}_(2bit),C_{0'}_(2bit))로 생성될 수 있으며, 상기 2비트 출력 오류검출코드(SED_{C2-out})의 진리 테이블(121)은 도 2에서 보인 2비트 오류검출코드(SED_{C2})의 진리 테이블(111)과 실질적으로 동일하므로 설명을 생략한다.

[0139] 다만, 상기 2비트 출력 오류검출코드 생성수단은 상기 입력 오류검출코드 생성기에서 사용되는 2비트 오류검출코드들(SED_{C2})의 특정 케이스와 동일한 케이스의 2비트 출력 오류검출코드(SED_{C2-out})를 상기 출력 오류검출코드로 생성하여야 한다.

[0140] 또한, 상기 오류 검출기(120)는 상기 출력 데이터가 3비트의 출력데이터일 경우, 상기 출력 데이터에 대한 2비트의 오류검출코드인 3비트 출력 오류검출코드(SED_{C3-out})를 상기 출력 오류검출코드로 출력하는 3비트 출력 오류검출코드 생성수단을 더 포함한다.

[0141] 도 11은 상기 3비트 출력 오류검출코드(SED_{C3-out})의 진리 테이블(122)을 보여주는 것으로 상기 3비트 출력 오류검출코드 생성수단은 아래의 수학적 식 4를 이용하여 상기 출력 데이터(F)를 3비트 출력 오류검출코드(C_{1'}_(3bit),C_{0'}_(3bit))로 생성하며, 더욱 자세하게는, 상기 출력 데이터(F)의 최상위 비트(F_{2(3bit)})가 '0'일 경우에는, 상기 출력 데이터(F) 중간비트(F_{1(3bit)}) 및 최하위 비트(F_{0(3bit)})를 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 2비트 출력 오류검출코드(SED_{C2-out})와 동일하게 상기 3비트 출력 오류검출코드(C_{1'}_(3bit),C_{0'}_(3bit))를 생성하고, 상기 출력 데이터(F)의 최상위 비트(F_{2(3bit)})가 '1'일 경우에는 상기 출력 데이터(F) 중간비트(F_{1(3bit)}) 및 최하위 비트(F_{0(3bit)})의 논리 부정값을 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 2비트 출력 오류검출코드(SED_{C2-out})의 논리 부정값을 상기 3비트 출력 오류검출코드(C_{1'}_(3bit),C_{0'}_(3bit))로 생성한다.

수학적 식 4

$$[0142] \quad (C_1'_{(3bit)}, C_0'_{(3bit)}) = \begin{cases} SEDC_{2-out}(F_{1(3bit)}, F_{0(3bit)}), & \text{if } F_{2(3bit)} = 0 \\ 1's \text{ complement}(SEDC_{2-out}(F_{1(3bit)}, F_{0(3bit)})), & \text{if } F_{2(3bit)} = 1 \end{cases}$$

[0143] 또한, 도 12는 상기 3비트 출력 오류검출코드 생성수단을 구현한 논리회로(122-1)를 보여주는 것으로 세 개의 입력에 대해 exclusive-NOT-OR 연산을 수행하는 하나의 XNOR게이트(122-1a), 네 개의 입력에 대해 or-and-invert 연산을 수행하는 하나의 OAI게이트(122-1b) 및 두 개의 입력에 대해 AND 연산을 수행하는 AND게이트(122-1c)로 설계하였다.

[0144] 그러나, 상기 3비트 출력 오류검출코드 생성수단은 3비트의 출력 데이터에 생성되는 오류검출코드가 도 13의 진리 테이블을 만족할 수 있다면, 어떠한 논리 게이트들의 조합으로도 설계가 가능하다.

[0145] 또한, 상기 오류 검출기(120)는 상기 출력 데이터가 4비트의 출력 데이터일 경우, 상기 출력 데이터에 대한 3비트의 오류검출코드인 4비트 출력 오류검출코드(SED_{C4-out})를 상기 출력 오류검출코드로 출력하는 4비트 출력 오류검출코드 생성수단을 더 포함한다.

[0146] 도 13은 상기 4비트 출력 오류검출코드(SED_{C4-out})의 진리 테이블(123)을 보여주는 것으로 상기 4비트 출력 오류검출코드 생성수단은 아래의 수학적 식 5를 이용하여 상기 출력 데이터(F)를 4비트 출력 오류검출코드(C_{2'}_(4bit),C_{1'}_(4bit),C_{0'}_(4bit))로 생성하며, 더욱 자세하게는, 상기 4비트 출력 오류검출코드(SED_{C4-out})의 하위 2비트(C_{1'}_(4bit),C_{0'}_(4bit))는 상기 4비트 출력 데이터들 중, 최상위 비트(F_{3(4bit)})를 제외한 3비트(F_{2(4bit)},F_{1(4bit)},F_{0(4bit)})를 입력으로 하는 상기 3비트 출력 오류검출코드 생성수단의 3비트 출력 오류검출코드(SED_{C3-out})로 생성하고, 상기 4비트 출력 오류검출코드(SED_{C4-out})의 최상위 비트(C_{2'}_(4bit))는 상기 4비트 출력 데이터의 최상위 비트(F_{3(4bit)})의 논리 부정 값으로 생성한다.

수학적식 5

$$(C_1'_{(4bit)}, C_0'_{(4bit)}) = \overline{SEDC_{3-out}(F_{2(4bit)}, F_{1(4bit)}, F_{0(4bit)})}$$

$$C_2'_{(4bit)} = F_{3(4bit)}$$

[0147]

[0148] 또한, 도 14는 상기 4비트 출력 오류검출코드 생성수단을 구현한 논리회로(123-1)를 보여주는 것으로 두 개의 입력에 대해 exclusive-NOT-OR 연산을 수행하는 하나의 XNOR게이트(123-1a), 두 개의 입력에 대해 exclusive-OR 연산을 수행하는 하나의 XOR 게이트(123-1b), 두 개의 입력과 하나의 선택 입력에 대해 선택적 입력을 출력하는 하나의 MUX(123-1c), 하나의 입력에 대해 invert 연산을 수행하는 하나의 invert 게이트(123-1d)로 설계하였다.

[0149] 또한, 상기 오류 검출기(120)는 상기 이진 출력 데이터(F)가 n비트 출력 데이터일 경우, 상기 n비트 출력 데이터에 대한 오류검출코드인 n비트 출력 오류검출코드($SEDC_{n-out}$)를 상기 출력 오류검출코드로 생성하는 n비트 출력 오류검출코드 생성수단을 더 포함한다.

[0150] 또한, 상기 n비트 출력 오류검출코드 생성수단은 상기 출력 오류검출코드 생성수단들 중 어느 하나의 오류검출코드($SEDC_{2-out}$, $SEDC_{3-out}$, $SEDC_{4-out}$)와 하나 또는 복수 개의 3비트 출력 오류검출코드($SEDC_{3-out}$)의 조합을 상기 출력 오류검출코드로 생성한다.

[0151] 또한, 상기 n비트 출력 오류검출코드 생성수단은 상기 출력 이진 데이터(F) b비트의 출력 데이터와 하나 또는 복수 개의 3비트 데이터 세트에 구분하여 각각 출력 오류검출코드를 계산한 후 하나의 출력 오류검출코드를 계산한다.

[0152] 또한, b비트의 데이터와 3비트의 데이터 세트에 대해서는 상기 n비트 입력 오류검출코드 생성수단의 설명에서 보인바 있으므로 설명을 생략하기로 한다.

[0153] 단, 여기서 주의할 점은 상기 n비트 입력 데이터와 상기 n비트 출력 데이터를 b비트 및 3비트 데이터 세트에 구분하는 방법은 동일하여야 하며, b비트 및 각 3비트 데이터 세트에 대해 생성된 오류검출코드를 동일한 방법으로 연결하여 서로 동일성 여부를 판단하여야 한다.

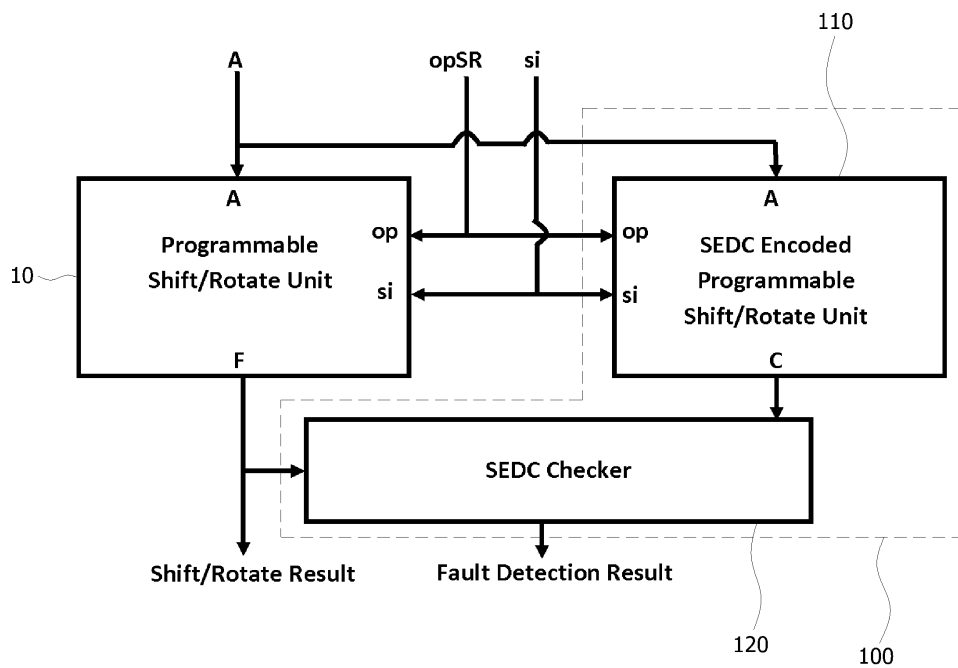
[0154] 이상에서 살펴본 바와 같이 본 발명은 바람직한 실시예를 들어 도시하고 설명하였으나, 상기한 실시예에 한정되지 아니하며 본 발명의 정신을 벗어나지 않는 범위 내에서 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변경과 수정이 가능할 것이다.

부호의 설명

[0155] 10:프로그래머블 시프트/로테이트 연산 유닛 100:오류 검출 장치
110:입력 오류검출코드 생성기 120:오류 검출기

도면

도면1



도면2

112

A		SEDC ₂ (A) Case1		SEDC ₂ (A) Case2	
A ₁	A ₀	C ₁	C ₀	C ₁	C ₀
0	0	1	1	1	1
0	1	1	0	0	1
1	0	1	0	0	1
1	1	0	1	1	0

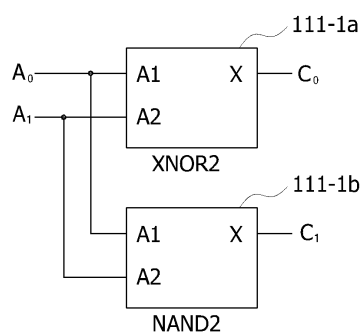
111a

111b

111c

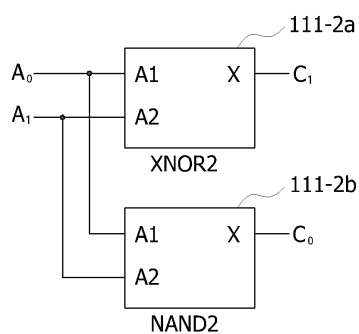
도면3

111-1



도면4

111-2



도면5

112

Operations	opSR	2-bit SEDC Encoded Shift/Rotate Operations
		$(C_1(2bit), C_0(2bit)) = SEDC_{2-in}(A(A_1, A_0), si, opSR)$
Shift Left A	0	$SEDC_2([A_0 si])$
Shift Right A	1	$SEDC_2([si A_1])$
Rotate Left A	2	$SEDC_2(A)$
Rotate Right A	3	$SEDC_2(A)$

도면6

113

Operations	opSR	1-bit SEDC Encoded Shift/Rotate Operations
		$C_0(1bit) = \text{lsb}\{SEDC_{2-in}(A(0, A_0), si, opSR)\}$
Shift Left A	0	$\text{lsb}\{SEDC_2(0[si])\}$
Shift Right A	1	$\text{lsb}\{SEDC_2(0[si])\}$
Rotate Left A	2	$\text{lsb}\{SEDC_2(0[A])\}$
Rotate Right A	3	$\text{lsb}\{SEDC_2(0[A])\}$

도면7

114

3-bit SEDC Encoded Shift/Rotate Operations				
$(C_{1(3bit)}, C_{0(3bit)}) = SEDC_{3-in}(A(A_2, A_1, A_0), si, opSR)$				
$(C_{1(2bit)}, C_{0(2bit)}) = SEDC_{2-in}(A(A_1, A_0), si, opSR)$ $(C_{1(3bit)}, C_{0(3bit)}) = \begin{cases} (C_{1(2bit)}, C_{0(2bit)})-1, & \text{if } S_1=1 \\ (C_{1(2bit)}, C_{0(2bit)}) & \text{otherwise} \end{cases}$				
Operations	opSR	A ₂	A ₁	S ₁
Shift Left A	0	×	1	1
Shift Right A	1	1	×	1
Rotate Left A	2	×	×	0
Rotate Right A	3	×	×	0

도면8

115

4-bit SEDC Encoded Shift/Rotate Operations				
$(C_{2(4bit)}, C_{1(4bit)}, C_{0(4bit)}) = SEDC_{4-in}(A(A_3, A_2, A_1, A_0), si, opSR)$				
$(C_{1(4bit)}, C_{0(4bit)}) = (C_{1(3bit)}, C_{0(3bit)}) = SEDC_{3-in}(A(A_2, A_1, A_0), sibit, opSR)$ $C_{2(4bit)} = \begin{cases} NOT(A_2), & \text{if } opSR=0 \\ NOT(si), & \text{if } opSR=1 \\ NOT(A_2), & \text{if } opSR=2 \\ NOT(A_0), & \text{if } opSR=3 \end{cases}$				
Operations	opSR	A ₃	si	sibit
Shift Left A	0	×	×	si
Shift Right A	1	×	×	A ₃
Rotate Left A	2	×	×	×
Rotate Right A	3	×	×	×

도면9

116

n-bit SEDC Encoded Shift/Rotate Operations ($n \geq 5$)			
b-bit SEDC Encoded Shift/Rotate Operations($(A_{n-1}, A_{n-2}, \dots, A_{n-b}), \text{sibit1}, \text{opSR}$)			
Operations	opSRbits	opSR	sibit1
Shift Left A	00	0	A_{n-b-1}
Shift Right A	01	1	Si
Rotate Left A	10	0	A_{n-b-1}
Rotate Right A	11	1	A_0
3-bit SEDC Encoded Shift/Rotate Operations($(A_{nn-1}, A_{nn-2}, A_{nn-3}), \text{sibit2}, \text{opSR}$)			
Operations	opSRbits	k $k = 0 \text{ to } (a-1)$	sibit2
Shift Left A	00	$= (a-1)$	si
		$\neq (a-1)$	$A_{n-b-(3 \times (k+1))-1}$
Shift Right A	01	$\times$	$A_{n-b-(3 \times k)}$
Rotate Left A	10	$= (a-1)$	A_{msb}
		$\neq (a-1)$	$A_{n-b-(3 \times (k+1))-1}$
Rotate Right A	11	$\times$	$A_{n-b-(3 \times k)}$
$a = \frac{n - \max(b)}{3}$ $k = 0 \text{ to } (a-1)$ $nn = n - (3 \times k) - b$			

도면10

121

F		SEDC _{2-out} (F) Case1		SEDC _{2-out} (F) Case2	
$F_1(2\text{bit})$	$F_0(2\text{bit})$	$C_1'(2\text{bit})$	$C_0'(2\text{bit})$	$C_1'(2\text{bit})$	$C_0'(2\text{bit})$
0	0	1	1	1	1
0	1	1	0	0	1
1	0	1	0	0	1
1	1	0	1	1	0
121a		121b		121c	

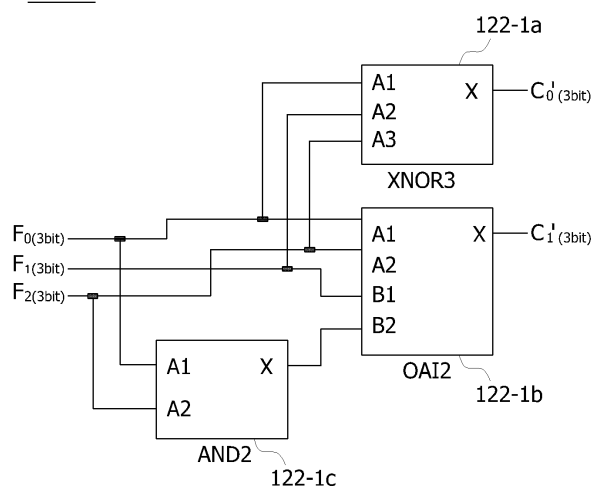
도면11

122

$F_2(3bit)$	$F_1(3bit)$	$F_0(3bit)$	$C'_1(3bit)$	$C'_0(3bit)$
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

도면12

122-1



도면13

123

$F_3(4\text{bit})$	$F_2(4\text{bit})$	$F_1(4\text{bit})$	$F_0(4\text{bit})$	$C_2'(4\text{bit})$	$C_1'(4\text{bit})$	$C_0'(4\text{bit})$
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

도면14

123-1

