



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2013년05월29일
(11) 등록번호 10-1268996
(24) 등록일자 2013년05월23일

(51) 국제특허분류(Int. Cl.)

G06F 11/28 (2006.01) G06F 11/07 (2006.01)

G06F 11/10 (2006.01)

(21) 출원번호 10-2012-0021150

(22) 출원일자 2012년02월29일

심사청구일자 2012년02월29일

(56) 선행기술조사문헌

KR1020100133194 A

KR1020090017382 A

(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 21 항

심사관 : 이정은

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

(57) 요약

본 발명은 프로그램머블 덧셈/뺄셈 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 그 오류 검출 장치를 포함하는 자가검사 프로그램머블 덧셈/뺄셈 연산 유닛에 관한 것으로, 보다 구체적으로는 입력 데이터에 대해 덧셈/뺄셈 연산을 수행하여 출력 데이터를 출력하는 프로그램머블 덧셈/뺄셈 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 덧셈/뺄셈 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 그 오류 검출 장치를 포함하는 자가검사 프로그램머블 덧셈/뺄셈 연산 유닛에 관한 것이다.

대표도 - 도6

2-bit SEDC Encoded Add Operation							
Operations	opAS	A	B	SEDC ₂ Cout		SEDC ₂ Sum	
				Cin = 0	Cin = 1	Cin = 0	Cin = 1
A + B + Cin	0	00	00	1	1	11	10
		00	01	1	1	10	10
		00	10	1	1	10	01
		00	11	1	0	01	11
		01	00	1	1	10	10
		01	01	1	1	10	01
		01	10	1	0	01	11
		01	11	0	0	11	10
		10	00	1	1	10	01
		10	01	1	0	01	11
		10	10	0	0	11	10
		10	11	0	0	10	10
		11	00	1	0	01	11
		11	01	0	0	11	10
		11	10	0	0	10	10
		11	11	0	0	10	01

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

이진 입력 데이터들에 대해 덧셈/뺄셈(Add/Subtract) 연산을 수행하여 연산결과인 캐리 아웃(Carry_out) 데이터와 썸 아웃(Sum_out) 데이터를 포함하는 이진 출력 데이터를 출력하는 프로그래머블 덧셈/뺄셈 연산 유닛(Programable Add/Subtract Operations Unit)의 오류를 검출하는 오류 검출 장치로써,

상기 이진 입력 데이터들 및 캐리 인(Carry_in) 비트를 입력받고, 덧셈/뺄셈 연산의 종류에 따라 상기 캐리 아웃 데이터에 대한 오류 검출코드인 캐리 아웃 입력 오류검출코드 및 상기 썸 아웃 데이터에 대한 오류 검출코드인 썸 아웃 입력 오류검출코드를 포함하는 입력 오류검출코드를 생성하는 입력 오류검출코드 생성기; 및

상기 캐리 아웃 데이터와 상기 썸 아웃 데이터를 입력받고, 상기 캐리 아웃 데이터에 대한 오류 검출코드인 캐리 아웃 출력 오류검출코드 및 상기 썸 아웃 데이터에 대한 오류 검출코드인 썸 아웃 출력 오류검출코드를 포함하는 출력 오류검출코드를 생성하고, 상기 출력 오류검출코드와 상기 입력 오류검출코드의 동일성을 판단하여 오류검출결과를 출력하는 오류 검출기;를 포함하며,

상기 입력 오류검출코드 및 상기 출력 오류검출코드는 상기 이진 입력 데이터들의 길이에 따라 길이가 가변하는 확장형 오류검출코드(SEDScalable Error Detection Coding)인 것을 특징으로 하는 오류 검출 장치.

청구항 2

제 1 항에 있어서,

상기 입력 오류검출코드 생성기:는

2비트 데이터에 대응하는 2비트 오류검출코드들(SED₂)을 저장한 2비트 오류검출코드 진리테이블(truth table); 및

상기 2비트 오류검출코드들(SED₂)을 이용하여 상기 이진 입력 데이터들의 길이에 따른 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(SED₂)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 데이터들(A,B)이 1비트 입력 데이터(A(A₀),B(B₀))이고, 연산의 종류가 덧셈 연산이며, 상기 캐리 인 비트가 '0'일 경우, 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 1비트 입력 데이터로 하는 2비트 가정

입력 데이터들의 논리곱 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 상기 2비트 가정 입력 데이터들의 배타적 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 썸 아웃 입력 오류검출코드로 생성하며,

상기 캐리 인 비트가 '1'인 경우에는 상기 2비트 가정 입력 데이터들의 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 상기 2비트 가정 입력 데이터들의 배타적 부정 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 썸 아웃 입력 오류검출코드로 생성하는 1비트 입력 오류검출코드 생성수단을 포함하는 것을 특징으로 하는 오류검출장치.

청구항 5

제 4 항에 있어서,

상기 2비트 가정 입력 데이터들은 피감수인 제1 2비트 가정 입력 데이터와 감수인 제2 2비트 가정 입력 데이터로 구분되고,

상기 1비트 입력 오류검출코드 생성수단은 연산의 종류가 뺄셈 연산이고, 상기 캐리 인 비트의 논리 부정값이 '0'일 경우, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 논리곱 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 배타적 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 썸 아웃 입력 오류검출코드로 생성하며,

상기 캐리 인 비트의 논리 부정값이 '1'인 경우에는, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 배타적 부정 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 썸 아웃 입력 오류검출코드로 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 6

제 3 항에 있어서,

상기 입력 오류검출코드 생성수단은,

상기 이진 입력 데이터들(A,B)이 2비트 입력 데이터($A(A_1, A_0), B(B_1, B_0)$)일 경우, 상기 2비트 입력데이터들의 덧셈/뺄셈 연산결과인 캐리 아웃 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 썸 아웃 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드를 상기 썸 아웃 입력 오류검출코드로 생성하는 2비트 입력 오류검출 코드 생성수단을 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 7

제 6 항에 있어서,

상기 캐리 아웃 데이터는 1비트 데이터이고,

상기 2비트 입력 오류검출코드 생성수단은 상위 비트는 '0'으로 가정하고 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 데이터에 대해 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드를 계산하고, 계산된 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 8

제 7 항에 있어서,

상기 입력 오류검출코드 생성수단은,

상기 이진 입력 데이터들(A,B)이 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)일 경우, 하위 2비트 입력 데이터($(A_1, A_0), (B_1, B_0)$)를 입력으로 하는 상기 2비트 입력 오류검출 코드 생성수단의 캐리 아웃 입력 오류검출코드를 제1 과도 캐리 아웃 입력 오류검출코드로 생성하고, 썸 아웃 입력 오류검출코드를 제1 과도 썸 아웃 입력 오류검출코드로 생성하며,

상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 모두 '0'이거나, 상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 서로 다르고 상기 제1 과도 캐리 아웃 입력 오류검출코드가 '1'인 경우, 상기 캐리 아웃 입력 오류 검출코드를 '1'로 생성하고, 그 외의 경우에는 상기 캐리 아웃 입력 오류검출코드를 '0'으로 생성하며,

상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 서로 동일하고 상기 제1 과도 캐리 아웃 입력 오류검출코드가 '0'인 경우나 상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 서로 다르고 상기 제1 과도 캐리 아웃 입력 오류검출코드가 '1'인 경우, 상기 제1 과도 썸 아웃 입력 오류검출코드에 '1'을 산술차(-)한 코드를 상기 썸 아웃 입력 오류검출코드로 생성하고, 그 외의 경우에는 상기 제1 과도 썸 아웃 입력 오류검출코드를 그대로 상기 썸 아웃 입력 오류검출코드로 생성하는 3비트 입력 오류검출 코드 생성수단;을 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 9

제 8 항에 있어서,

상기 입력 오류검출코드 생성수단은,

상기 이진 입력 데이터들(A,B)이 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)일 경우, 하위 3비트 입력 데이터($(A_2, A_1, A_0), (B_2, B_1, B_0)$)를 입력으로 하는 상기 3비트 입력 오류검출 코드 생성수단의 캐리 아웃 입력 오류검출코드를 제2 과도 캐리 아웃 입력 오류검출코드로 생성하고, 썸 아웃 입력 오류검출코드를 제2 과도 썸 아웃 입력 오류검출코드로 생성하며,

상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)의 최상위 비트들(A_3, B_3)의 값이 모두 '0'이거나, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)의 최상위 비트들(A_3, B_3)의 값이 서로 다르고 상기 제2 과도 캐리 아웃 입력 오류검출코드가 '1'인 경우, 상기 캐리 아웃 입력 오류 검출코드를 '1'로 생성하고, 그 외의 경우에는 상기 캐리 아웃 입력 오류검출코드를 '0'으로 생성하며,

상기 제2 과도 썸 아웃 입력 오류검출코드를 하위 두 비트로 하고 최상위 비트는 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)의 최상위 비트들(A_3, B_3)의 서로 다를 경우, 상기 제2 과도 캐리 아웃 입력 오류검출코드의 논리 부정 값으로 계산되고, 그 외의 경우에는 상기 제2 과도 캐리 아웃 입력 오류검출코드를 최상위 비트로 계산하여 상기 썸 아웃 입력 오류검출코드를 3비트의 썸 아웃 입력 오류검출코드로 생성하는 4비트 입력 오류검출코드 생성수단;을 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 10

제 9 항에 있어서,

상기 입력 오류검출코드 생성수단은,

상기 이진 입력 데이터들이 n 비트 입력 데이터일 경우, 상기 2비트 입력 오류검출코드 생성수단, 상기 3비트 입력 오류검출코드 생성수단 및 상기 4비트 입력 오류검출코드 생성수단 중, 어느 하나의 입력 오류검출코드 생성수단에 의해 생성된 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드를 상기 입력 오류검출코드로 생성하거나,

상기 2비트 입력 오류검출코드 생성수단, 상기 3비트 입력 오류검출코드 생성수단 및 상기 4비트 입력 오류검출코드 생성수단 중, 어느 하나의 입력 오류검출코드 생성수단에 의해 생성된 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드와 상기 3비트 입력 오류검출코드 생성수단에 의해 생성된 하나 또는 복수 개의 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드의 조합을 상기 입력 오류검출코드로 생성하는 n 비트 입력 오류검출코드 생성수단;을 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 11

제 10 항에 있어서,

상기 n 비트 입력 오류검출코드 생성수단은,

상기 n 비트 입력 데이터가 2비트, 3비트 또는 4비트이고, 연산의 종류가 덧셈 연산일 경우, 상기 n 비트 입력 데이터들과 상기 캐리 인 비트를 덧셈 연산하여 상기 캐리 아웃 데이터와 상기 썸 아웃 데이터를 계산하고, 계산된 캐리 아웃 데이터와 썸 아웃 데이터에 대응하는 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드를 생성하며,

연산의 종류가 뺄셈 연산일 경우, 상기 n 비트 입력 데이터들 중, 감수가 되는 n 비트 입력 데이터는 감수가 되는 n 비트 입력 데이터의 논리 부정 값으로 대체하고, 상기 캐리 인 비트는 입력되는 캐리 인 비트의 논리 부정 값으로 대체하여 덧셈 연산을 수행하여 상기 캐리 아웃 데이터와 상기 썸 아웃 데이터를 계산하고, 계산된 캐리 아웃 데이터와 썸 아웃 데이터에 대응하는 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드를 생성하는 것을 특징으로 하는 오류 검출 장치.

청구항 12

제 10 항에 있어서,

상기 n 비트 입력 오류검출코드 생성수단은,

상기 n 비트 입력 데이터들이 5비트 이상의 입력 데이터일 경우, 상기 각 n 비트 입력 데이터를 하나의 b 비트 입력 데이터와 하나 또는 복수 개의 3비트 입력 데이터 세트로 구분하고, 상기 b 비트 입력 데이터에 대한 오류검출코드인 b 비트 입력 오류검출코드와 상기 각 3비트 입력 데이터 세트에 대한 오류검출코드인 3비트 입력 오류검출코드를 생성하여, 상기 b 비트 입력 오류검출코드와 상기 3비트 입력 오류검출코드를 연결함으로써 상기 입력 오류검출코드를 생성하며,

연산의 종류가 덧셈 연산일 경우, 상기 b 비트 입력 오류검출코드는 상기 캐리 인 비트를 상기 3비트 입력 데이터 세트들 중, 최상위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드로 대체하여 계산하고, 상기 각 3비트 입력 오류검출코드는 상기 캐리 인 비트를 해당 3비트 입력 데이터 세트의 바로 하위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드로 대체되어 계산되되, 해당 3비트 입력 데이터 세트가 최하위 3비트 입력 데이터 세트일 경우, 입력되는 캐리 인 비트를 그대로 이용하여 계산하고,

연산의 종류가 뺄셈 연산일 경우, 상기 b 비트 입력 오류검출코드는 상기 캐리 인 비트를 상기 최상위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드의 논리 부정 값으로 대체하고, 상기 b 비트 입력 데이터들 중, 감수가 되는 b 비트 입력 데이터는 감수가 되는 b 비트 입력 데이터의 논리 부정 값으로 대체하여 덧셈 연산을 수행함으로써 계산하고, 상기 3비트 입력 오류검출코드는 상기 캐리 인 비트를 해당 3비트 입력 데이터 세트가 상기 최하위 3비트 입력 데이터 세트가 아닐 경우, 해당 3비트 입력 데이터 세트의 바로 하위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드의 논리 부정 값으로 대체하고, 해당 3비트 입력 데이터 세트가 최하위 3비트 입력 데이터 세트일 경우, 입력되는 캐리 인 비트의 논리 부정 값으로 대체하며, 상기 3비트 입력 데이터 세트들 중, 감수가 되는 3비트 입력 데이터 세트는 감수가 되는 3비트 입력 데이터 세트의 논리 부정 값으로 대

체하여 덧셈 연산을 수행함으로써 계산하는 것을 특징으로 하는 오류 검출 장치.

청구항 13

제 2 항 내지 제 12 항 중 어느 한 항에 있어서,

상기 오류 검출기:는

상기 썸 아웃 데이터가 1비트 출력 데이터일 경우, 상기 썸 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 1비트 출력 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하고, 상기 캐리 아웃 출력 오류 검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하여, 상기 출력 오류검출코드를 생성하는 1비트 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 14

제 13 항에 있어서,

상기 오류 검출기:는

상기 썸 아웃 데이터가 2비트 출력 데이터일 경우,

상기 썸 아웃 출력 오류검출코드는 상기 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드로 생성하고,

상기 캐리 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하여,

상기 출력 오류검출코드를 생성하는 2비트 출력 오류검출코드 생성수단;을 더 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 15

제 14 항에 있어서,

상기 오류 검출기:는

상기 썸 아웃 데이터가 3비트 출력 데이터일 경우,

상기 썸 아웃 출력 오류검출코드는 상기 3비트 출력 데이터의 최상위 비트가 '0'일 경우에는, 중간비트 및 최하위 비트를 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 썸 아웃 출력 오류검출코드와 동일하게 생성하고, 상기 3비트 출력 데이터의 최상위 비트가 '1'일 경우에는 중간비트 및 최하위 비트의 논리 부정값을 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 썸 아웃 출력 오류검출코드로 생성하며,

상기 캐리 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하여,

상기 출력 오류검출코드를 생성하는 3비트 출력 오류검출코드 생성수단;을 더 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 16

제 15 항에 있어서,

상기 오류 검출기:는

상기 썸 아웃 데이터가 4비트 출력 데이터일 경우,

상기 썸 아웃 출력 오류검출코드는 하위 2비트는 상기 4비트 출력 데이터 중, 최상위 비트를 제외한 3비트를 입력으로 하는 상기 3비트 출력 오류검출코드 생성수단의 썸 아웃 출력 오류검출코드로 생성하고 최상위 비트는 상기 4비트 출력 데이터의 최상위 비트의 논리 부정 값으로 생성하여 3비트의 썸 아웃 출력 오류검출코드로 생성되며,

상기 캐리 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하여,

상기 출력 오류검출코드를 생성하는 4비트 출력 오류검출코드 생성수단;을 더 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 17

제 16 항에 있어서,

상기 오류 검출기:는

상기 썸 아웃 데이터가 n비트 출력 데이터일 경우,

상기 출력 오류검출코드 생성수단들 각각의 오류검출코드나, 상기 출력 오류검출코드 생성수단들 중 어느 하나의 출력 오류검출코드 생성수단의 오류검출코드와 하나 또는 복수 개의 3비트 출력 오류검출코드의 조합을 상기 출력 오류검출코드로 생성하는 n비트 출력 오류검출코드 생성수단;을 더 포함하는 것을 특징으로 하는 오류 검출 장치.

청구항 18

제 17 항의 오류 검출 장치:와

이진 입력 데이터들에 대해 덧셈/뺄셈(Add/Subtract) 연산을 수행하여 연산결과인 캐리 아웃(Carry_out) 데이터와 썸 아웃(Sum_out) 데이터를 포함하는 이진 출력 데이터를 출력하는 프로그래머블 덧셈/뺄셈 연산 유닛(Programmable Add/Subtract Operations Unit);을 포함하는 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛(Self-checking Programmable Add/Subtract Operations Unit).

청구항 19

제 17 항의 오류 검출 장치; 및

이진 입력 데이터들에 대해 덧셈/뺄셈(Add/Subtract) 연산을 수행하여 연산결과인 캐리 아웃(Carry_out) 데이터와 썸 아웃(Sum_out) 데이터를 포함하는 이진 출력 데이터를 출력하는 프로그래머블 덧셈/뺄셈 연산 유닛(Programmable Add/Subtract Operations Unit)으로 기능하는 프로그래머블 산술/논리 연산유닛;를 포함하며, 상기 오류 검출 장치는 상기 프로그래머블 산술/논리 연산유닛의 오류를 검출하는 것을 특징으로 하는 자가검사 프로그래머블 산술/논리 연산유닛(Self-checking Programmable Arithmetic and Logic Unit).

청구항 20

컴퓨터를 제 17 항의 오류 검출 장치로 기능하게 하는 오류 검출 프로그램이 저장된 컴퓨터로 읽을 수 있는 기

록매체.

청구항 21

제 20 항의 오류 검출 프로그램이 저장되고 통신망을 통해 상기 오류 검출 프로그램을 전송할 수 있는 서버 시스템.

명세서

기술분야

- [0001] 본 발명은 프로그래머블 덧셈/뺄셈 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 그 오류 검출 장치를 포함하는 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛에 관한 것으로, 보다 구체적으로는 입력 데이터에 대해 덧셈/뺄셈 연산을 수행하여 출력 데이터를 출력하는 프로그래머블 덧셈/뺄셈 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 덧셈/뺄셈 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 그 오류 검출 장치를 포함하는 자가검사 프로그래머블 덧셈/뺄셈 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 있어야 하고, 시스템이 기능하는 도중에도 결함을 복구하기 위한 신호를 출력할 수 있어야 한다.
- [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 동일한 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 오류검출코드로서, 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 덧셈/뺄셈 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치 및 자가 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛을 개발하기 위해 노력한 결과, 덧셈/뺄셈 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 덧셈/뺄셈 연산 유닛을 위한 확장형 오류검출코드 기반의 오류 검출 장치와 상기 오류 검출 장치를 구비하여 자가검사를 수행할 수 있는 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 기술적 구성을 개발하게 되어 본 발명을 완성하게 되었다.

[0018] 따라서, 본 발명의 목적은 프로그래머블 덧셈/뺄셈 연산 유닛의 기능은 유지시키되 연산의 오류를 검출할 수 있는 결함허용 가능한 오류 검출 장치를 제공하는 것이다.

[0019] 또한, 본 발명의 다른 목적은 프로그래머블 덧셈/뺄셈 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 덧셈/뺄셈 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 데이터들에 대해 덧셈/Add/Substract 연산을 수행하여 연산결과인 캐리 아웃(Carry_out) 데이터와 썸 아웃(Sum_out) 데이터를 포함하는 이진 출력 데이터를 출력하

는 프로그래머블 덧셈/뺄셈 연산 유닛(Programable Add/Substract Operations Unit)의 오류를 검출하는 오류 검출 장치로서, 상기 이진 입력 데이터들 및 캐리 인(Carry_in) 비트를 입력받고, 덧셈/뺄셈 연산의 종류에 따라 상기 캐리 아웃 데이터에 대응하는 오류 검출코드인 캐리 아웃 입력 오류검출코드 및 상기 뺄 아웃 데이터에 대응하는 오류 검출코드인 뺄 아웃 입력 오류검출코드를 포함하는 입력 오류검출코드를 생성하는 입력 오류검출코드 생성기; 및 상기 캐리 아웃 데이터와 상기 뺄 아웃 데이터를 입력받고, 상기 캐리 아웃 데이터에 대한 오류 검출코드인 캐리 아웃 출력 오류검출코드 및 상기 뺄 아웃 데이터에 대한 오류 검출코드인 뺄 아웃 출력 오류검출코드를 포함하는 출력 오류검출코드를 생성하고, 상기 출력 오류검출코드와 상기 입력 오류검출코드의 동일성을 판단하여 오류검출결과를 출력하는 오류 검출기;를 포함하며, 상기 입력 오류검출코드 및 상기 출력 오류검출코드는 상기 이진 입력 데이터들의 길이에 따라 길이가 가변하는 확장형 오류검출코드(SEDScalable 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(SED_2)을 저장한 2비트 오류검출코드 진리테이블(truth table); 및 상기 2비트 오류검출코드들(SED_2)을 이용하여 상기 이진 입력 데이터들의 길이에 따른 가변하는 상기 입력 오류검출코드를 생성하는 입력 오류검출코드 생성수단;을 포함한다.

[0024] 바람직한 실시예에 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 저장된다.

[0025] 바람직한 실시예에 있어서, 상기 입력 오류검출코드 생성수단은, 상기 이진 입력 데이터들(A,B)이 1비트 입력 데이터($A(A_0), B(B_0)$)이고, 연산의 종류가 덧셈 연산이며, 상기 캐리 인 비트가 '0'일 경우, 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 1비트 입력 데이터로 하는 2비트 가정 입력 데이터들의 논리곱 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 상기 2비트 가정 입력 데이터들의 배타적 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 뺄 아웃 입력 오류검출코드로 생성하며, 상기 캐리 인 비트가 '1'인 경우에는 상기 2비트 가정 입력 데이터들의 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 상기 2비트 가정 입력 데이터들의 배타적 부정 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 뺄 아웃 입력 오류검출코드로 생성하는 1비트 입력 오류검출코드 생성수단을 포함한다.

[0026] 바람직한 실시예에 있어서, 상기 2비트 가정 입력 데이터들은 피감수인 제1 2비트 가정 입력 데이터와 감수인 제2 2비트 가정 입력 데이터로 구분되고, 상기 1비트 입력 오류검출코드 생성수단은 연산의 종류가 뺄셈 연산이고, 상기 캐리 인 비트의 논리 부정값이 '0'일 경우, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 논리곱 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 배타적 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 뺄 아웃 입력 오류검출코드로 생성하며, 상기 캐리 인 비트의 논리 부정값이 '1'인 경우에는, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 배타적 부정 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드의 하위 비트를 상기 뺄 아웃 입력 오류검출코드로 생성한다.

[0027] 바람직한 실시예에 있어서, 상기 입력 오류검출코드 생성수단은, 상기 이진 입력 데이터들(A,B)이 2비트 입력 데이터($A(A_1, A_0), B(B_1, B_0)$)일 경우, 상기 2비트 입력데이터들의 덧셈/뺄셈 연산결과인 캐리 아웃 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드를 상기 캐리 아웃 입력 오류검출코드로 생성하고, 뺄 아웃 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드를 상기 뺄 아웃 입력 오류검출코드로 생성하는 2비트 입력 오류검출 코드 생성수단;을 포함한다.

- [0028] 바람직한 실시예에 있어서, 상기 캐리 아웃 데이터는 1비트 데이터이고, 상기 2비트 입력 오류검출코드 생성수단은 상위 비트는 '0'으로 가정하고 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 데이터에 대해 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드를 계산하고, 계산된 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드로 생성한다.
- [0029] 바람직한 실시예에 있어서, 상기 입력 오류검출코드 생성수단은, 상기 이진 입력 데이터들(A,B)이 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)일 경우, 하위 2비트 입력 데이터($(A_1, A_0), (B_1, B_0)$)를 입력으로 하는 상기 2비트 입력 오류검출 코드 생성수단의 캐리 아웃 입력 오류검출코드를 제1 과도 캐리 아웃 입력 오류검출코드로 생성하고, 썸 아웃 입력 오류검출코드를 제1 과도 썸 아웃 입력 오류검출코드로 생성하며, 상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 모두 '0'이거나, 상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 서로 다르고 상기 제1 과도 캐리 아웃 입력 오류검출코드가 '1'인 경우, 상기 캐리 아웃 입력 오류 검출코드를 '1'로 생성하고, 그 외의 경우에는 상기 캐리 아웃 입력 오류검출코드를 '0'으로 생성하며, 상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 서로 동일하고 상기 제1 과도 캐리 아웃 입력 오류검출코드가 '0'인 경우나 상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 서로 다르고 상기 제1 과도 캐리 아웃 입력 오류검출코드가 '1'인 경우, 상기 제1 과도 썸 아웃 입력 오류검출코드에 '1'을 산술차(-)한 코드를 상기 썸 아웃 입력 오류검출코드로 생성하고, 그 외의 경우에는 상기 제1 과도 썸 아웃 입력 오류검출코드를 그대로 상기 썸 아웃 입력 오류검출코드로 생성하는 3비트 입력 오류검출 코드 생성수단;을 포함한다.
- [0030] 바람직한 실시예에 있어서, 상기 입력 오류검출코드 생성수단은, 상기 이진 입력 데이터들(A,B)이 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)일 경우, 하위 3비트 입력 데이터($(A_2, A_1, A_0), (B_2, B_1, B_0)$)를 입력으로 하는 상기 3비트 입력 오류검출 코드 생성수단의 캐리 아웃 입력 오류검출코드를 제2 과도 캐리 아웃 입력 오류검출코드로 생성하고, 썸 아웃 입력 오류검출코드를 제2 과도 썸 아웃 입력 오류검출코드로 생성하며, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)의 최상위 비트들(A_3, B_3)의 값이 모두 '0'이거나, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)의 최상위 비트들(A_3, B_3)의 값이 서로 다르고 상기 제2 과도 캐리 아웃 입력 오류검출코드가 '1'인 경우, 상기 캐리 아웃 입력 오류 검출코드를 '1'로 생성하고, 그 외의 경우에는 상기 캐리 아웃 입력 오류검출코드를 '0'으로 생성하며, 상기 제2 과도 썸 아웃 입력 오류검출코드를 하위 두 비트로 하고 최상위 비트는 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)의 최상위 비트들(A_3, B_3)의 서로 다를 경우, 상기 제2 과도 캐리 아웃 입력 오류검출코드의 논리 부정 값으로 계산되고, 그 외의 경우에는 상기 제2 과도 캐리 아웃 입력 오류검출코드를 최상위 비트로 계산하여 상기 썸 아웃 입력 오류검출코드를 3비트의 썸 아웃 입력 오류검출코드로 생성하는 4비트 입력 오류검출코드 생성수단;을 포함한다.
- [0031] 바람직한 실시예에 있어서, 상기 입력 오류검출코드 생성수단은, 상기 이진 입력 데이터들이 n비트 입력 데이터일 경우, 상기 2비트 입력 오류검출코드 생성수단, 상기 3비트 입력 오류검출코드 생성수단 및 상기 4비트 입력 오류검출코드 생성수단 중, 어느 하나의 입력 오류검출코드 생성수단에 의해 생성된 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드를 상기 입력 오류검출코드로 생성하거나, 상기 2비트 입력 오류검출코드 생성수단, 상기 3비트 입력 오류검출코드 생성수단 및 상기 4비트 입력 오류검출코드 생성수단 중, 어느 하나의 입력 오류검출코드 생성수단에 의해 생성된 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드와 상기 3비트 입력 오류검출코드 생성수단에 의해 생성된 하나 또는 복수 개의 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드의 조합을 상기 입력 오류검출코드로 생성하는 n비트 입력 오류검출코드 생성수단;을 포함한다.
- [0032] 바람직한 실시예에 있어서, 상기 n비트 입력 오류검출코드 생성수단은, 상기 n비트 입력 데이터가 2비트, 3비트 또는 4비트이고, 연산의 종류가 덧셈 연산일 경우, 상기 n비트 입력 데이터들과 상기 캐리 인 비트를 덧셈 연산하여 상기 캐리 아웃 데이터와 상기 썸 아웃 데이터를 계산하고, 계산된 캐리 아웃 데이터와 썸 아웃 데이터에 대응하는 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드를 생성하며, 연산의 종류가 뺄셈 연산일 경우, 상기 n비트 입력 데이터들 중, 감수가 되는 n비트 입력 데이터는 감수가 되는 n비트 입력 데이터의 논리 부정 값으로 대체하고, 상기 캐리 인 비트는 입력되는 캐리 인 비트의 논리 부정 값으로 대체하여 덧셈 연산을 수행하여 상기 캐리 아웃 데이터와 상기 썸 아웃 데이터를 계산하고, 계산된 캐리 아웃 데이터와 썸 아웃 데이터에 대응하는 캐리 아웃 입력 오류검출코드 및 썸 아웃 입력 오류검출코드를 생성한다.

- [0033] 바람직한 실시예에 있어서, 상기 n 비트 입력 오류검출코드 생성수단은, 상기 n 비트 입력 데이터들이 5비트 이상의 입력 데이터일 경우, 상기 각 n 비트 입력 데이터를 하나의 b 비트 입력 데이터와 하나 또는 복수 개의 3비트 입력 데이터 세트로 구분하고, 상기 b 비트 입력 데이터에 대한 오류검출코드인 b 비트 입력 오류검출코드와 상기 각 3비트 입력 데이터 세트에 대한 오류검출코드인 3비트 입력 오류검출코드를 생성하여, 상기 b 비트 입력 오류검출코드와 상기 3비트 입력 오류검출코드를 연결함으로써 상기 입력 오류검출코드를 생성하며, 연산의 종류가 덧셈 연산일 경우, 상기 b 비트 입력 오류검출코드는 상기 캐리 인 비트를 상기 3비트 입력 데이터 세트들 중, 최상위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드로 대체하여 계산하고, 상기 각 3비트 입력 오류검출코드는 상기 캐리 인 비트를 해당 3비트 입력 데이터 세트의 바로 하위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드로 대체되어 계산하되, 해당 3비트 입력 데이터 세트가 최하위 3비트 입력 데이터 세트일 경우, 입력되는 캐리 인 비트를 그대로 이용하여 계산하고, 연산의 종류가 뺄셈 연산일 경우, 상기 b 비트 입력 오류검출코드는 상기 캐리 인 비트를 상기 최상위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드의 논리 부정 값으로 대체하고, 상기 b 비트 입력 데이터들 중, 감수가 되는 b 비트 입력 데이터는 감수가 되는 b 비트 입력 데이터의 논리 부정 값으로 대체하여 덧셈 연산을 수행함으로써 계산하고, 상기 3비트 입력 오류검출코드는 상기 캐리 인 비트를 해당 3비트 입력 데이터 세트가 상기 최하위 3비트 입력 데이터 세트가 아닐 경우, 해당 3비트 입력 데이터 세트의 바로 하위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드의 논리 부정 값으로 대체하고, 해당 3비트 입력 데이터 세트가 최하위 3비트 입력 데이터 세트일 경우, 입력되는 캐리 인 비트의 논리 부정 값으로 대체하며, 상기 3비트 입력 데이터 세트들 중, 감수가 되는 3비트 입력 데이터 세트는 감수가 되는 3비트 입력 데이터 세트의 논리 부정 값으로 대체하여 덧셈 연산을 수행함으로써 계산한다.
- [0034] 바람직한 실시예에 있어서, 상기 오류 검출기:는 상기 썸 아웃 데이터가 1비트 출력 데이터일 경우, 상기 썸 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 1비트 출력 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하고, 상기 캐리 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하여, 상기 출력 오류검출코드를 생성하는 1비트 출력 오류검출코드 생성수단; 및 상기 입력 오류검출코드와 상기 출력 오류검출코드의 동일성을 비교하여 동일하지 않을 경우 오류 발생신호를 출력하는 코드비교수단;을 포함한다.
- [0035] 바람직한 실시예에 있어서, 상기 오류 검출기:는 상기 썸 아웃 데이터가 2비트 출력 데이터일 경우, 상기 썸 아웃 출력 오류검출코드는 상기 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드로 생성하고, 상기 캐리 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하여, 상기 출력 오류검출코드를 생성하는 2비트 출력 오류검출코드 생성수단;을 더 포함한다.
- [0036] 바람직한 실시예에 있어서, 상기 오류 검출기:는 상기 썸 아웃 데이터가 3비트 출력 데이터일 경우, 상기 썸 아웃 출력 오류검출코드는 상기 3비트 출력 데이터의 최상위 비트가 '0'일 경우에는, 중간비트 및 최하위 비트를 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 썸 아웃 출력 오류검출코드와 동일하게 생성하고, 상기 3비트 출력 데이터의 최상위 비트가 '1'일 경우에는 중간비트 및 최하위 비트의 논리 부정값을 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 썸 아웃 출력 오류검출코드로 생성하며, 상기 캐리 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하여, 상기 출력 오류검출코드를 생성하는 3비트 출력 오류검출코드 생성수단;을 더 포함한다.
- [0037] 바람직한 실시예에 있어서, 상기 오류 검출기:는 상기 썸 아웃 데이터가 4비트 출력 데이터일 경우, 상기 썸 아웃 출력 오류검출코드는 하위 2비트는 상기 4비트 출력 데이터 중, 최상위 비트를 제외한 3비트를 입력으로 하는 상기 3비트 출력 오류검출코드 생성수단의 썸 아웃 출력 오류검출코드로 생성하고 최상위 비트는 상기 4비트 출력 데이터의 최상위 비트의 논리 부정 값으로 생성하여 3비트의 썸 아웃 출력 오류검출코드로 생성되며, 상기 캐리 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블의 2비트 오류검출코드 중, 하위 비트로 생성하여, 상기 출력 오류검출코드를 생성하는 4비트 출력 오류검출코드 생성수단;을 더 포함한다.
- [0038] 바람직한 실시예에 있어서, 상기 오류 검출기:는 상기 썸 아웃 데이터가 n 비트 출력 데이터일 경우, 상기 출력 오류검출코드 생성수단들 각각의 오류검출코드나, 상기 출력 오류검출코드 생성수단들 중 어느 하나의 출력 오

류검출코드 생성수단의 오류검출코드와 하나 또는 복수 개의 3비트 출력 오류검출코드의 조합을 상기 출력 오류검출코드로 생성하는 n비트 출력 오류검출코드 생성수단;을 더 포함한다.

[0039] 또한, 본 발명은 상기 오류 검출 장치와 프로그래머블 덧셈/뺄셈 연산 유닛을 포함하는 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛(Self-checking Programmable Add/Subtract Operations Unit)을 더 제공한다.

[0040] 또한, 본 발명은 상기 오류 검출 장치 및 상기 프로그래머블 덧셈/뺄셈 연산 유닛으로 기능하는 프로그래머블 산술/논리 연산유닛을 포함하는 자가검사 프로그래머블 산술/논리 연산유닛(Self-checking Programmable Arithmetic and Logic Unit)을 더 제공한다.

[0041] 또한, 본 발명은 컴퓨터를 상기 오류 검출 장치로 기능하게 하는 오류 검출 프로그램이 저장된 컴퓨터로 읽을 수 있는 기록매체를 더 제공한다.

[0042] 또한, 본 발명은 상기 오류 검출 프로그램이 저장되고 통신망을 통해 상기 오류 검출 프로그램을 전송할 수 있는 서버 시스템을 더 제공한다.

발명의 효과

[0043] 본 발명은 다음과 같은 우수한 효과를 가진다.

[0044] 먼저, 본 발명의 오류 검출 장치 및 자가 검사 프로그래머블 덧셈/뺄셈 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 덧셈/뺄셈 연산의 결함을 검출할 수 있는 장점이 있다.

[0045] 또한, 본 발명의 오류 검출 장치 및 자가 검사 프로그래머블 덧셈/뺄셈 연산 유닛에 의하면, 오류검출코드의 생성이 입력 데이터에 대한 덧셈/뺄셈 연산과는 별개로 이루어지므로 입력 데이터의 연산의 속도나 레이턴시와 같은 성능의 하락 없이 결합 허용 가능 시스템을 구현할 수 있는 장점이 있다.

도면의 간단한 설명

[0046] 도 1은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛을 설명하기 위한 도면,

도 2는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 2비트 오류검출코드 진리테이블을 보여주는 도면,

도 3은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 2비트 오류검출코드를 생성하는 논리회로의 일례를 보여주는 도면,

도 4는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 2비트 오류검출코드를 생성하는 논리회로의 다른 예를 보여주는 도면,

도 5는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 연산종류에 따른 1비트 입력 오류검출코드의 생성 로직을 설명하기 위한 도면,

도 6는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 연산종류에 따른 2비트 입력 오류검출코드의 생성 로직을 설명하기 위한 도면,

도 7는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 연산종류에 따른 3비트 입력 오류검출코드의 생성 로직을 설명하기 위한 도면,

도 8는 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 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 덧셈/뺄셈 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 덧셈/뺄셈 연산 유닛의 2비트

출력 오류검출코드의 진리 테이블을 보여주는 도면,

도 11은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 3비트 출력 오류검출코드의 진리 테이블을 보여주는 도면,

도 12는 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 3비트 출력 오류검출코드 생성수단을 구현한 논리회로,

도 13은 본 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 4비트 출력 오류검출코드의 진리 테이블을 보여주는 도면,

도 14는 발명의 일 실시예에 따른 오류 검출 장치 및 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛의 4비트 출력 오류검출코드 생성수단을 구현한 논리회로이다.

발명을 실시하기 위한 구체적인 내용

[0047] 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.

[0048] 이하, 첨부한 도면에 도시된 바람직한 실시예들을 참조하여 본 발명의 기술적 구성을 상세하게 설명한다.

[0049] 그러나, 본 발명은 여기서 설명되는 실시예에 한정되지 않고 다른 형태로 구체화 될 수도 있다. 명세서 전체에 걸쳐 동일한 참조번호는 동일한 구성요소를 나타낸다.

[0050] 본 발명의 일 실시예에 따른 오류 검출 장치(100)는 덧셈/뺄셈 연산(Add/Substract Operations)을 수행하는 프로그래머블 덧셈/뺄셈 연산 유닛(Programmable Add/Substract Operations Unit, 10)으로 입력되는 이진 입력 데이터들(A,B) 및 캐리 인 비트(Cin)를 포함하는 입력 데이터와 상기 입력 데이터에 대한 덧셈/뺄셈 연산의 결과인 캐리 아웃 데이터(Cout) 및 썸 아웃 데이터(S)를 포함하는 출력 데이터에 대해 각각 오류검출코드를 생성하여 서로 동일성을 비교함으로써 상기 덧셈/뺄셈 연산 유닛(10)의 연산에 결함이 있는지 여부를 판단한다.

[0051] 또한, 상기 오류 검출 장치(100)는 상기 프로그래머블 덧셈/뺄셈 연산 유닛(10)과 함께 하나의 자가검사 프로그래머블 덧셈/뺄셈 연산 유닛(Self-checking Programmable Add/Substract Operations Unit)으로 구성되어 제공될 수 있다.

[0052] 또한, 상기 프로그래머블 덧셈/뺄셈 연산 유닛(10)은 덧셈/뺄셈 연산을 수행하는 프로그래머블 산술/논리 연산 유닛(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.

[0053] 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 덧셈/뺄셈 연산에 대한 오류 검출을 수행할 수 있다.

[0054] 이하에서는 본 발명의 일 실시예에 따른 오류 검출 장치(100)의 구성을 상세히 설명하기로 한다.

[0055] 도 1을 참조하면, 본 발명의 일 실시예에 따른 오류 검출 장치(100)는 입력 오류검출코드 생성기(110) 및 오류 검출기(120)를 포함하여 이루어진다.

[0056] 또한, 상기 오류 검출코드 생성기(110)는 상기 프로그래머블 덧셈/뺄셈 연산 유닛(10)의 입력 데이터인 이진 입력데이터들(A,B)과 캐리 인 비트(Cin)를 입력받고 덧셈/뺄셈 연산결과에 대한 오류검출코드인 입력 오류검출코드를 생성한다.

[0057] 또한, 상기 오류 검출기(120)는 상기 프로그래머블 덧셈/뺄셈 연산 유닛(10)의 출력 데이터들(Add/Substract Result)을 입력받아 출력 데이터에 대한 오류검출코드인 출력 오류검출코드를 생성하며, 상기 입력 오류검출코드와 상기 출력 오류검출코드의 동일성을 판단하여 오류검출결과(Fault Detection Result)를 출력한다.

- [0058] 즉, 본 발명의 일 실시예에 따른 오류 검출 장치(100)는 상기 프로그래머블 덧셈/뺄셈 연산 유닛(10)의 연산에 대해서 관여하지 않으므로 결함 허용 가능한 구조(fault-tolerant reconfigurable architecture)이다.
- [0059] 또한, 상기 입력 오류검출코드 생성기(110)와 상기 오류 검출기(120)는 상기 입력 데이터들(A,B)이나 상기 출력 데이터의 길이에 선형적으로 가변하는 확장형 오류검출코드(SED_C:scalable error detection coding)를 생성한다.
- [0060] 상기 입력 오류검출코드 생성기(100)는 2비트 오류검출코드 진리테이블(truth table, 111) 및 입력 오류검출코드 생성수단을 포함하여 이루어진다.
- [0061] 또한, 상기 입력 오류검출코드 생성기(100)로 입력되는 데이터들은 적어도 두 개의 이진 입력 데이터(A,B), 덧셈 연산시 캐리 비트(carry bit)로 사용되고 뺄셈 연산시 바로우 비트(borrow bit)로 사용되는 캐리 인 비트(Cin), 및 덧셈 또는 뺄셈의 연산 종류를 선택하기 위한 연산 선택 신호(opAS)를 포함한다.
- [0062] 또한, 뺄셈연산의 경우 상기 이진 입력데이터들 중, 제1 이진입력 데이터(A)는 피감수가 되고, 제2 이진 입력데이터(B)는 감수가 된다.
- [0063] 또한, 도 2를 참조하면, 상기 2비트 오류검출코드 진리테이블(111)은 2비트 데이터(A, 111a)에 대응하는 2비트의 오류검출코드인 2비트 오류검출코드들(SED_{C2}, 111b, 111c)이 계산되어 저장된다.
- [0064] 또한, 상기 2비트 오류검출코드 진리테이블(111)의 2비트 데이터(A, 111a)는 상기 이진 입력 데이터(A,B)의 덧셈 또는 뺄셈 연산 결과와 대응하는 데이터로써 상기 이진 입력 데이터(A,B)들과는 구분된다.
- [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 오류검출코드(SED_{C2})는 두 개의 케이스로 계산될 수 있는데, 첫 번째 케이스(Case1, 111b)는 아래의 수학적식에 의해 계산된다.

수학적식 1

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

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

[0067]

- [0068] 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.
- [0069] 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 NOT-exclusive-OR 연산을 수행하는 하나의 XNOR게이트(111-1a)와 2비트 입력에 대해 NOT-AND 연산을 수행하는 하나의 NAND게이트(111-1b)를 이용하여 구현하였다.
- [0070] 그러나, 상기 제1 2비트 오류검출코드 생성 논리회로(111-1)의 구성은 예시일 뿐, 다양한 논리 게이트의 조합으로 구현할 수 있음은 물론이다.
- [0071] 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.

- [0072] 즉, 상기 두 번째 케이스(111c)의 경우, 2비트 오류검출코드(SED_{C₂})의 최상위 비트(C₁)는 최상위 비트는 '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.
- [0073] 또한, 상기 두 번째 케이스(111c)의 경우, 도 4에 도시한 제2 2비트 오류검출코드 생성 논리회로(111-2)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.
- [0074] 또한, 상기 제2 2비트 오류검출코드 생성 논리회로(111-2) 역시, 다양한 논리 게이트의 조합으로 구현될 수 있다.
- [0075] 또한, 상기 입력 오류검출코드 생성수단은 상기 2비트 오류검출코드 진리테이블(111)을 이용하여 상기 이진 입력 데이터(A,B) 및 상기 캐리 인 비트의 덧셈 또는 뺄셈 연산 결과에 대응하는 오류검출코드인 입력 오류검출코드를 생성하고, 상기 입력 오류검출코드는 캐리 아웃 입력 오류검출코드(CCout) 및 썸 아웃 입력 오류검출코드(CS)를 포함한다.
- [0076] 또한, 상기 캐리 아웃 입력 오류검출코드는 상기 프로그래머블 덧셈/뺄셈 연산 유닛(10)에서 출력되는 캐리 아웃 데이터(Cout)에 대응하는 오류검출코드이고, 상기 썸 아웃 입력 오류검출코드는 상기 프로그래머블 덧셈/뺄셈 연산 유닛(10)에서 출력되는 썸 아웃 데이터(S)에 대응하는 오류검출코드이다.
- [0077] 또한, 상기 입력 오류검출코드 생성수단은 상기 이진 입력데이터들이 1비트 데이터일 경우, 1비트 입력 오류검출코드(SED_{C_{1-in}})를 생성하는 1비트 입력 오류검출코드 생성수단, 상기 이진 입력데이터들이 2비트 데이터일 경우, 2비트 입력 오류검출코드(SED_{C_{2-in}})를 생성하는 2비트 입력 오류검출코드 생성수단, 상기 이진 입력데이터들이 3비트 데이터일 경우, 3비트 입력 오류검출코드(SED_{C_{3-in}})를 생성하는 3비트 입력 오류검출코드 생성수단, 상기 이진 입력데이터들이 4비트 데이터일 경우, 4비트 입력 오류검출코드(SED_{C_{4-in}})를 생성하는 4비트 입력 오류검출코드 생성수단 또는 상기 이진 입력데이터들이 n비트 데이터일 경우, 길이가 가변하는 확장형의 n비트 입력 오류검출코드를 생성하는 n비트 입력 오류검출코드(SED_{C_{n-in}})를 생성하는 4비트 입력 오류검출코드 생성수단을 포함한다.
- [0078] 또한, 상기 입력 오류검출코드들은 각각 상기 썸 아웃 입력 오류검출코드(CS) 및 상기 캐리 아웃 입력 오류검출코드(CCout)를 포함한다.
- [0079] 이하에서는 상기 각 입력 오류검출코드 생성수단들이 입력 오류검출코드를 생성하는 로직을 상세히 설명하기로 한다.
- [0080] 상기 1비트 입력 오류검출코드 생성수단은 상기 이진 입력데이터들이 1비트 데이터일 경우, 상기 1비트 입력 오류검출코드(SED_{C_{1-in}})를 생성한다.
- [0081] 즉, 상기 1비트 입력 오류검출코드 생성수단은 상기 1비트 입력 오류검출코드(SED_{C_{1-in}})를 상기 입력 오류검출코드로 생성하는 역할을 한다.
- [0082] 도 5는 상기 1비트 입력 오류검출코드 생성수단이 상기 1비트 입력 오류검출코드(SED_{C_{1-in}})를 생성하는 로직을 설명하기 위한 것으로, 상기 1비트 입력 오류검출코드 생성수단은 상기 연산 선택 신호(opAS)가 덧셈 연산(opAS=0)이고, 상기 캐리 인 비트(Cin)가 '0'일 경우, 상위 비트는 '0'으로 가정하고 하위 비트는 상기 1비트 입력 데이터로 하는 2비트의 입력 데이터들(이하, '2비트 가정 입력 데이터'라 함)의 논리곱 값에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드(CCout)로 생성하고, 상기 2비트 가정 입력 데이터의 배타적 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드의 하위 비트를 상기 썸 아웃 입력 오류검출코드(CS)로 생성한다.
- [0083] 즉, 상기 1비트 입력 오류검출코드 생성수단은 각각 1비트의 캐리 아웃 입력 오류검출코드와 썸 아웃 입력 오류

검출코드를 생성한다.

- [0084] 예를 들어, 상기 연산 선택 신호(opAS)가 덧셈 연산(opAS=0)이고, 상기 1비트 입력 데이터(A,B)가 '0', '1'이며, 상기 캐리 인 비트(Cin)가 '0'인 경우, 상기 2비트 가정 입력 데이터는 '00', '01'이 되고, 상기 2비트 가정 입력 데이터의 논리곱 값은 '00'이 되며, 이진 데이터 '00'에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드는 '11'이 된다.
- [0085] 즉, 상기 연산 선택 신호(opAS)가 덧셈 연산(opAS=0)이고, 상기 1비트 입력 데이터(A,B)가 '0', '1'이며, 상기 캐리 인 비트(Cin)가 '0'인 경우, 상기 캐리 아웃 입력 오류검출코드(C Cout)는 '1'로 생성되는 것이다.
- [0086] 또한, 상기 2비트 가정 입력 데이터의 배타적 논리합 값은 '01'이 되고, 이진 데이터 '01'에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드는 '10'이 되므로 상기 연산 선택 신호(opAS)가 덧셈 연산(opAS=0)이고, 상기 1비트 입력 데이터(A,B)가 '0', '1'이며, 상기 캐리 인 비트(Cin)가 '0'인 경우, 상기 썸 아웃 입력 오류검출코드(CS)는 '0'된다.
- [0087] 또한, 상기 1비트 입력 오류검출코드 생성수단은 상기 연산 선택 신호(opAS)가 덧셈 연산(opAS=0)이고, 상기 캐리 인 비트(Cin)가 '1'일 경우, 상기 캐리 아웃 입력 오류검출코드(C Cout)는 상기 2비트 가정 입력 데이터의 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드의 하위 비트로 생성하고, 상기 썸 아웃 입력 오류검출코드(CS)는 상기 2비트 가정 입력 데이터의 배타적 부정 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드의 하위 비트로 생성한다.
- [0088] 또한, 상기 1비트 입력 오류검출코드 생성수단은 상기 연산 선택 신호(opAS)가 뺄셈 연산(opAS=1)이고, 상기 캐리 인 비트(Cin)가 '1'인 경우(상기 캐리 인 비트(Cin)의 논리 부정 값이 '0'인 경우와 같다), 상기 2비트 가정 입력 데이터들 중, 피감수가 되는 제1 2비트 가정 입력 데이터와 감수가 되는 제2 2비트 가정 입력 데이터의 논리 부정 값의 논리곱 값에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드(C Cout)로 생성하고, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 배타적 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드의 하위 비트를 상기 썸 아웃 입력 오류검출코드(CS)로 생성한다.
- [0089] 예를 들어, 상기 연산 선택 신호(opAS)가 뺄셈 연산(opAS=1)이고, 상기 1비트 입력 데이터(A,B)가 '0', '1'이며, 상기 캐리 인 비트(Cin)가 '1'인 경우, 상기 제1 2비트 가정 입력 데이터는 '00'이 되고, 상기 제2 2비트 가정 입력 데이터는 '01'이 되며, 상기 캐리 인 비트의 논리 부정값은 '0'이 된다.
- [0090] 또한, 상기 제1 2비트 가정 입력 데이터 '00'과 상기 제2 2비트 가정 입력 데이터의 논리 부정 값 '10'의 논리곱 값은 '00'이 되어, 이진 데이터 '00'에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드는 '11'이 된다.
- [0091] 즉, 상기 1비트 입력 데이터(A,B)가 '0', '1'이고 상기 캐리 인 비트(Cin)가 '1'이며, 상기 연산 선택 신호(opAS)가 뺄셈 연산(opAS=1)일 경우, 상기 캐리 아웃 입력 오류검출코드(C Cout)는 '1'이 되는 것이다.
- [0092] 또한, 상기 연산 선택 신호(opAS)가 뺄셈 연산(opAS=1)이고, 상기 1비트 입력 데이터(A,B)가 '0', '1'이며, 상기 캐리 인 비트(Cin)가 '0'인 경우, 상기 캐리 인 비트의 논리 부정값은 '1'이 되고, 상기 제1 2비트 가정 입력 데이터 '00'과 상기 제2 2비트 가정 입력 데이터 '01'의 논리 부정 값 '10'의 배타적 논리 부정합 값은 '01'이 되며, 이진 데이터 '01'에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드는 '10'이 된다.
- [0093] 즉, 상기 1비트 입력 데이터(A,B)가 '0', '1'이고 상기 캐리 인 비트(Cin)가 '0'이며, 상기 연산 선택 신호(opAS)가 뺄셈 연산(opAS=1)일 경우, 상기 썸 아웃 입력 오류검출코드(CS)는 '0'이 되는 것이다.
- [0094] 또한, 상기 1비트 입력 오류검출코드 생성수단은 상기 연산 선택 신호(opAS)가 뺄셈 연산(opAS=1)이고, 상기 캐리 인 비트(Cin)가 '0'일 경우(상기 캐리 인 비트(Cin)의 논리 부정값이 '1'인 경우와 같다), 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드의 하위 비트를 상기 캐리 아웃 입력 오류검출코드(C Cout)로 생성하고, 상기 제1 2비트 가정 입력 데이터와 상기 제2 2비트 가정 입력 데이터의 논리 부정 값의 배타적 부정 논리합 값에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드의 하위 비트를 상기 썸 아웃 입력 오류검출코드(CS)로 생성한다.

- [0095] 상기 2비트 입력 오류검출코드 생성수단은 상기 이진 입력데이터들이 2비트 데이터일 경우, 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)를 생성한다.
- [0096] 즉, 상기 2비트 입력 오류검출코드 생성수단은 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)를 상기 입력 오류검출코드로 생성하는 역할을 한다.
- [0097] 도 6는 상기 2비트 입력 오류검출코드 생성수단이 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)를 생성하는 로직을 설명하기 위한 것으로, 상기 2비트 입력 오류검출코드 생성수단은 상기 이진 입력 데이터들(A,B)이 2비트 입력 데이터($A(A_1, A_0), B(B_1, B_0)$)일 경우, 상기 2비트 입력데이터들의 덧셈/뺄셈 연산결과인 캐리 아웃 데이터(Cout)에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드를 상기 캐리 아웃 입력 오류검출코드(CCout)로 생성하고, 썸 아웃 데이터(S)에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드를 상기 썸 아웃 입력 오류검출코드(CS)로 생성한다.
- [0098] 예를 들어, 상기 2비트 입력 데이터(A,B)가 '01', '10'이고 상기 캐리 인 비트(Cin)가 '0'일 경우, 상기 2비트 입력 데이터(A,B) 및 상기 캐리 인 비트(Cin)의 덧셈 연산 결과 중, 캐리 아웃 데이터(Cout)는 '0'이 되고 썸 아웃 데이터(S)는 '11'이 된다. 여기서 상기 캐리 아웃 데이터(Cout)는 1비트 데이터이므로 상위 비트를 '0'으로 가정한 2비트 데이터 '00'에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드 '11'의 하위 비트 '1'을 상기 캐리 아웃 입력 오류검출코드(CCout)로 생성하고, 썸 아웃 데이터(S)는 '11'에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드 '01'을 썸 아웃 입력 오류검출코드(CS)로 생성한다.
- [0099] 또한, 상기 2비트 입력 데이터(A,B)가 '01', '10'이고 상기 캐리 인 비트(Cin)가 '1'일 경우, 상기 2비트 입력 데이터(A,B) 및 상기 캐리 인 비트(Cin)의 덧셈 연산 결과 중, 캐리 아웃 데이터(Cout)는 '1'이 되고 썸 아웃 데이터(S)는 '00'이 된다. 여기서, 상기 캐리 아웃 데이터(Cout)는 상위 비트를 '0'으로 가정한 2비트 데이터 '01'에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드 '10'의 하위 비트 '0'을 상기 캐리 아웃 입력 오류검출코드(CCout)로 생성하고, 썸 아웃 데이터(S)는 '00'에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드 '11'을 썸 아웃 입력 오류검출코드(CS)로 생성하는 것이다.
- [0100] 또한, 본 발명에서는 도 6에서 도시한 바와 같이 상기 입력 오류검출코드 생성기(111)가 상기 2비트 입력데이터들의 덧셈/뺄셈 연산을 수행하기 위한 별도의 덧셈/뺄셈 연산 유닛을 구비하지 않고, 상기 2비트 입력 데이터들($A(A_1, A_0), B(B_1, B_0)$)이 조합가능한 경우에 따른 캐리 아웃 입력 오류검출코드(CCout)와 썸 아웃 입력 오류검출코드(CS)를 진리테이블로 저장하여 계산되도록 하였다.
- [0101] 또한, 도 6에 도시한 상기 2비트 입력 오류검출코드($SEDC_{2-in}$)의 출력을 위한 진리테이블은 다양한 논리 게이트들을 조합한 디지털 논리 회로로 설계할 수 있다.
- [0102] 상기 3비트 입력 오류검출코드 생성수단은 상기 이진 입력데이터들이 3비트 데이터일 경우, 상기 3비트 입력 오류검출코드($SEDC_{3-in}$)를 생성한다.
- [0103] 즉, 상기 3비트 입력 오류검출코드 생성수단은 상기 3비트 입력 오류검출코드($SEDC_{3-in}$)를 상기 입력 오류검출코드로 생성하는 역할을 한다.
- [0104] 도 7는 상기 3비트 입력 오류검출코드 생성수단이 상기 3비트 입력 오류검출코드($SEDC_{3-in}$)를 생성하는 로직을 설명하기 위한 것으로, 먼저, 상기 3비트 입력 오류검출코드 생성수단은 상기 이진 입력 데이터들(A,B)이 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)일 경우, 하위 2비트 입력 데이터($(A_1, A_0), (B_1, B_0)$)를 입력으로 하는 상기 2비트 입력 오류검출 코드 생성수단의 캐리 아웃 입력 오류검출코드를 과도 캐리 아웃 입력 오류검출코드(t_c)로 생성하고, 상기 2비트 입력 오류검출 코드 생성수단의 썸 아웃 입력 오류검출코드를 과도 썸 아웃 입력 오류검출코드(t_{s1}, t_{s0})로 생성한다.
- [0105] 다음, 상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 모두 '0'이거나, 상기 3

비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 서로 다르고 상기 과도 캐리 아웃 입력 오류검출코드(t_c)가 '1'인 경우, 상기 캐리 아웃 입력 오류 검출코드($CCout$)(C_c)를 '1'로 생성하고, 그 외의 경우에는 상기 캐리 아웃 입력 오류검출코드(C_c)를 '0'으로 생성한다.

[0106] 또한, 상기 썸 아웃 입력 오류검출코드(CS)(C_{s1}, C_{s0})는 상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 서로 동일하고 상기 과도 캐리 아웃 입력 오류검출코드(t_c)가 '0'인 경우나, 상기 3비트 입력 데이터($A(A_2, A_1, A_0), B(B_2, B_1, B_0)$)의 최상위 비트들(A_2, B_2)의 값이 서로 다르고 상기 과도 캐리 아웃 입력 오류검출코드(t_c)가 '1'인 경우, 상기 과도 썸 아웃 입력 오류검출코드(t_{s1}, t_{s0})에 '1'을 산술차(-)한 코드를 상기 썸 아웃 입력 오류검출코드(C_{s1}, C_{s0})로 생성하고, 그 외의 경우에는 상기 과도 썸 아웃 입력 오류검출코드(t_{s1}, t_{s0})를 그대로 상기 썸 아웃 입력 오류검출코드(C_{s1}, C_{s0})로 생성한다.

[0107] 상기 4비트 입력 오류검출코드 생성수단은 상기 이전 입력데이터들이 4비트 데이터일 경우, 상기 4비트 입력 오류검출코드($SEDC_{4-in}$)를 생성한다.

[0108] 즉, 상기 4비트 입력 오류검출코드 생성수단은 상기 4비트 입력 오류검출코드($SEDC_{4-in}$)를 상기 입력 오류검출코드로 생성하는 역할을 한다.

[0109] 도 8는 상기 4비트 입력 오류검출코드 생성수단이 상기 4비트 입력 오류검출코드($SEDC_{4-in}$)를 생성하는 로직을 설명하기 위한 것으로, 먼저, 상기 4비트 입력 오류검출코드 생성수단은 상기 이전 입력 데이터들(A, B)이 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)일 경우, 하위 3비트 입력 데이터($(A_2, A_1, A_0), (B_2, B_1, B_0)$)를 입력으로 하는 상기 3비트 입력 오류검출 코드 생성수단의 캐리 아웃 입력 오류검출코드를 과도 캐리 아웃 입력 오류검출코드(t_c)로 생성하고, 상기 3비트 입력 오류검출 코드 생성수단의 썸 아웃 입력 오류검출코드를 과도 썸 아웃 입력 오류검출코드(t_{s1}, t_{s0})로 생성한다.

[0110] 다음, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)의 최상위 비트들(A_3, B_3)의 값이 모두 '0'이거나, 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)의 최상위 비트들(A_3, B_3)의 값이 서로 다르고 상기 과도 캐리 아웃 입력 오류검출코드(t_c)가 '1'인 경우, 상기 캐리 아웃 입력 오류 검출코드($CCout$)(C_c)를 '1'로 생성하고, 그 외의 경우에는 상기 캐리 아웃 입력 오류검출코드(C_c)를 '0'으로 생성한다.

[0111] 또한, 상기 4비트 입력 오류검출코드 생성수단은 상기 과도 썸 아웃 입력 오류검출코드(t_{s1}, t_{s0})를 상기 썸 아웃 입력 오류검출코드(CS)(C_{s2}, C_{s1}, C_{s0})의 하위 두 비트(C_{s1}, C_{s0})로 생성하고, 상기 썸 아웃 입력 오류검출코드(C_{s2}, C_{s1}, C_{s0})의 최상위 비트(C_{s2})는 상기 4비트 입력 데이터($A(A_3, A_2, A_1, A_0), B(B_4, B_2, B_1, B_0)$)의 최상위 비트들(A_3, B_3)의 서로 다를 경우, 상기 과도 캐리 아웃 입력 오류검출코드(t_c)의 논리 부정 값($NOT\ t_c$)으로 계산하고, 그 외의 경우에는 상기 과도 캐리 아웃 입력 오류검출코드(t_c)를 상기 썸 아웃 입력 오류검출코드(C_{s2}, C_{s1}, C_{s0})의 최상위 비트(C_{s2})로 계산하여 3비트의 썸 아웃 입력 오류검출코드(C_{s2}, C_{s1}, C_{s0})를 생성한다.

[0112] 상기 n비트 입력 오류검출코드 생성수단은 상기 이전 입력데이터들이 n비트 데이터($n \geq 2$)일 경우, 상기 n비트 입력 오류검출코드($SEDC_{n-in}$)를 생성한다.

[0113] 즉, 상기 n비트 입력 오류검출코드 생성수단은 상기 n비트 입력 오류검출코드($SEDC_{n-in}$)를 상기 입력 오류검출코드로 생성하는 역할을 한다.

[0114] 도 9는 상기 n비트 입력 오류검출코드 생성수단이 상기 n비트 입력 오류검출코드($SEDC_{n-in}$)를 생성하는 로직을 설명하기 위한 것으로, 먼저, 상기 n비트 입력 오류검출코드 생성수단은 아래의 수학식 2와 같이 입력되는 입력 데이터의 길이에 따라 생성될 입력 오류검출코드의 길이를 계산한다.

- [0115] 또한, 여기서 입력 오류검출코드의 길이는 상기 썸 아웃 입력 오류검출코드(CS)의 길이를 의미하며, 상기 캐리 아웃 입력 오류검출코드(CCout)의 길이는 예를 들면 1비트로 일정하다.

수학식 2

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

[0116]

- [0117] 여기서, m은 상기 n비트 입력 오류검출코드(SED_{C_{n-in}})의 길이이고, 파라미터 'a'는 '0'에서부터 '∞'까지의 정수 중 어느 하나이며, 상기 파라미터 'a'는 아래의 수학식 3에 의해 계산된다.

수학식 3

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

[0118]

- [0119] 여기서, n은 상기 이진 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.

- [0120] 예를 들어, 상기 이진 입력데이터(n)의 길이가 '8비트'일 경우, 상기 수학식 3의 max(b)는 정수 '2'로 선택되고, 상기 파라미터 'a'는 정수 '2'로 계산된다.

- [0121] 또한, 상기 파라미터 'a'가 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 입력 입력데이터의 길이가 '8비트'일 경우, 상기 썸 아웃 입력 오류검출코드(CS)의 길이는 '6비트'로 계산되는 것이다.

- [0122] 즉, 상기 n비트 입력 오류검출코드 생성수단은 상기 입력 데이터(A,B)가 '8비트'일 경우, 상기 2비트 입력 오류검출코드(SED_{C_{2-in}}) 하나와 상기 3비트 입력 오류검출코드(SED_{C_{3-in}}) 두 개를 조합하여 '6비트'의 썸 아웃 입력 오류검출코드(CS)와 '1비트'의 캐리 아웃 입력 오류검출코드(CCout)를 생성하는 기능을 한다.

- [0123] 다시 말해서, 상기 n비트 입력 오류검출코드 생성수단은 상기 2비트 입력 오류검출코드(SED_{C_{2-in}}), 상기 3비트 입력 오류검출코드(SED_{C_{3-in}}) 및 상기 4비트 입력 오류검출코드(SED_{C_{4-in}}) 중, 상기 파라미터 'b'의 숫자와 대응하는 어느 하나의 입력 오류검출코드와 상기 파라미터 'a'의 숫자만큼의 3비트 입력 오류검출코드(SED_{C_{3-in}}) 조합을 상기 n비트 입력 오류검출코드로 생성하는 것이다.

- [0124] 따라서, 본 발명의 일 실시예에 따른 오류 검출 장치(100)는 입력 데이터의 길이에 선형적으로 오류검출코드를 확장할 수 있는 확장형 오류검출 기반의 오류 검출 장치이다.

- [0125] 더욱 자세하게는, 상기 n비트 입력 오류검출코드 생성수단은 상기 n비트 입력 데이터들(A,B)이 각각 5비트 이상의 입력 데이터일 경우, 상기 각 n비트 입력 데이터(A,B)를 각각 하나의 b비트 입력 데이터와 하나 또는 복수 개의 3비트 입력 데이터 세트에 구분하고, 상기 b비트 입력 데이터에 대한 오류검출코드인 b비트 입력 오류검출코드와 상기 각 3비트 입력 데이터 세트에 대한 오류검출코드인 3비트 입력 오류검출코드를 생성하여, 상기 b비트 입력 오류검출코드와 상기 3비트 입력 오류검출코드를 하나의 입력 오류검출코드로 생성한다.

- [0126] 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}, ..., 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.

- [0127] 또한, 본 발명에서는 상기 파라미터 'k'가 '0'인 3비트 데이터 세트가 상기 b비트 데이터와 연결되는 최상위 데이터 세트이고 나머지 3비트 데이터 세트는 상기 파라미터 'k'가 증가하며 순차적으로 하위로 연결되는 데이터

세트인 것으로 정의한다.

- [0128] 예를 들어, 상기 n 비트 입력 데이터($A(A_{n-1}, A_{n-2}, \dots, A_{n-b}, \dots)$)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 b 비트 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.
- [0129] 또한, 상기 n 비트 입력 오류검출코드 생성수단은 연산의 종류가 덧셈 연산일 경우, 상기 b 비트 입력 오류검출코드는 상기 캐리 인 비트(Cin)를 상기 3비트 입력 데이터 세트들 중, 파라미터 'k' 값이 '0'인 최상위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드(C_k)로 대체하여 계산하고, 상기 각 3비트 입력 오류검출코드는 상기 캐리 인 비트를 해당 3비트 입력 데이터 세트의 바로 하위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드(C_{k+1})로 대체되어 계산하되, 해당 3비트 입력 데이터 세트가 최하위 3비트 입력 데이터 세트일 경우(파라미터 'k' 값이 $(a-1)$ 인 경우이다), 입력되는 캐리 인 비트(Cin)를 그대로 이용하여 계산한다.
- [0130] 그러나, 상기 3비트 입력 데이터 세트가 하나만 존재할 경우에는 상기 b 비트 입력 오류검출코드는 상기 캐리 인 비트(Cin)를 하나의 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드(C_k)로 대체하여 계산되고, 상기 3비트 입력 오류검출코드는 입력되는 캐리 인 비트(Cin)를 그대로 이용하여 계산한다.
- [0131] 또한, 상기 파라미터 'a'가 '0'인 경우에는 상기 3비트 입력 데이터 세트가 존재하지 않는 것을 의미하므로 상기 2비트 입력 오류검출코드 생성수단, 상기 3비트 입력 오류검출코드 생성수단 및 상기 4비트 입력 오류검출코드 생성수단 중, 상기 파라미터 'b'와 대응하는 하나의 입력 오류검출코드 생성수단을 이용하여 상기 썸 아웃 입력 오류검출코드 및 캐리 아웃 입력 오류검출코드를 생성한다.
- [0132] 또한, 상기 n 비트 입력 오류검출코드 생성수단은 연산의 종류가 뺄셈 연산일 경우, 상기 b 비트 입력 오류검출코드는 상기 캐리 인 비트(Cin)를 상기 최상위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드(C_k)의 논리 부정 값($\text{NOT } C_k$)으로 대체하고, 상기 b 비트 입력 데이터들 중, 감수가 되는 b 비트 입력 데이터는 감수가 되는 b 비트 입력 데이터의 논리 부정 값으로 대체하여 덧셈 연산을 수행함으로써 계산한다.
- [0133] 또한, 상기 n 비트 입력 오류검출코드 생성수단은 연산의 종류가 뺄셈 연산일 경우, 상기 3비트 입력 오류검출코드는 계산되는 해당 3비트 입력 데이터 세트가 상기 최하위 3비트 입력 데이터 세트($k \neq (a-1)$ 인 경우)가 아닐 경우, 상기 캐리 인 비트(Cin)를 해당 3비트 입력 데이터 세트의 바로 하위 3비트 입력 데이터 세트의 캐리 아웃 입력 오류검출코드(C_{k+1})의 논리 부정 값($\text{NOT } C_{k+1}$)으로 대체하고, 해당 3비트 입력 데이터 세트가 최하위 3비트 입력 데이터 세트($k=(a-1)$ 인 경우)일 경우, 상기 캐리 인 비트(Cin)를 입력되는 캐리 인 비트의 논리 부정 값(NOT Cin)으로 대체하며, 상기 3비트 입력 데이터 세트들 중, 감수가 되는 3비트 입력 데이터 세트는 감수가 되는 3비트 입력 데이터 세트의 논리 부정 값(NOT B)으로 대체하여 덧셈 연산을 수행함으로써 상기 3비트 입력 데이터 세트들에 대한 오류검출코드들을 계산한다.
- [0134] 상기 오류 검출기(120)는 상기 프로그래머블 덧셈/뺄셈 연산 유닛(10)에서 출력되는 연산결과인 캐리 아웃 데이터(Cout)에 대한 오류검출코드인 캐리 아웃 출력 오류검출코드와 썸 아웃 데이터(S)에 대한 오류검출코드인 썸 아웃 출력 오류검출코드를 상기 출력 오류검출코드로 생성하며, 상기 입력 오류검출코드 생성기(110)에서 생성된 입력 오류검출코드와 상기 출력 오류검출코드를 서로 비교하여 동일하지 않을 경우 오류 발생 신호를 출력한다.
- [0135] 또한, 상기 오류 검출기(120)는 상기 썸 아웃 데이터(S)가 1비트일 경우, 1비트 출력 오류검출코드를 출력하는 1비트 출력 오류검출코드 생성수단, 상기 썸 아웃 데이터(S)가 2비트일 경우, 2비트 출력 오류검출코드를 출력하는 2비트 출력 오류검출코드 생성수단, 상기 썸 아웃 데이터(S)가 3비트일 경우, 3비트 출력 오류검출코드를 출력하는 3비트 출력 오류검출코드 생성수단, 상기 썸 아웃 데이터(S)가 4비트일 경우, 4비트 출력 오류검출코드를 출력하는 4비트 출력 오류검출코드 생성수단 또는 상기 썸 아웃 데이터(S)가 n 비트일 경우, n 비트 출력 오류검출코드를 출력하는 n 비트 출력 오류검출코드 생성수단과 상기 입력 오류검출코드와 상기 출력 오류검출코드의 동일성을 비교하는 코드비교수단을 포함한다.
- [0136] 또한, 상기 출력 오류검출코드들은 각각 상기 캐리 아웃 데이터(Cout)에 대한 오류검출코드인 캐리 아웃 출력

오류검출코드 및 상기 썸 아웃 데이터(S)에 대한 오류검출코드인 썸 아웃 출력 오류검출코드를 포함한다.

[0137] 상기 1비트 출력 오류검출코드 생성수단은 상기 썸 아웃 데이터(S)가 1비트 출력 데이터일 경우, 상위 비트는 '0'으로 가정되고, 하위 비트는 상기 1비트 출력 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드 중, 하위 비트를 상기 썸 아웃 출력 오류검출코드로 생성한다.

[0138] 또한, 상기 1비트 출력 오류검출코드 생성수단은 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드 중, 하위 비트를 상기 캐리 아웃 출력 오류 검출코드로 생성한다.

[0139] 상기 2비트 출력 오류검출코드 생성수단은 상기 썸 아웃 데이터(S)가 2비트 출력 데이터일 경우, 상기 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드를 상기 썸 아웃 출력 오류검출코드로 생성하고, 상기 캐리 아웃 출력 오류검출코드는 상위 비트는 '0'으로 가정하고, 하위 비트는 상기 캐리 아웃 데이터(Cout)로 하는 2비트 출력 데이터에 대응하는 상기 2비트 오류검출코드 진리테이블(111)의 2비트 오류검출코드 중, 하위 비트로 생성한다.

[0140] 또한, 상기 2비트 출력 오류검출코드 생성수단은 상기 입력 오류검출코드 생성수단에 구비된 2비트 오류검출코드 진리테이블(111)을 이용하여 상기 출력 오류검출코드를 생성할 수 있으나 도 10에 도시한 2비트 출력 오류검출코드 진리테이블(121)을 이용하여 상기 출력 오류 검출코드를 생성할 수도 있다.

[0141] 또한, 도 10의 데이터 'F'($F_{1(2bit)}, F_{0(2bit)}$)는 상기 썸 아웃 데이터(S) 또는 2비트로 가정된 캐리 아웃 데이터를 의미한다.

[0142] 또한, 상기 2비트 출력 오류검출코드 진리테이블(121)은 상기 2비트 오류검출코드 진리테이블(111)과 실질적으로 동일하며, 도 3 내지 도 4에 도시한 논리회로로 구성될 수 있다.

[0143] 상기 3비트 출력 오류검출코드 생성수단은 상기 썸 아웃 데이터(S)가 3비트 출력 데이터일 경우, 상기 3비트 출력 데이터의 최상위 비트가 '0'일 경우에는, 중간비트 및 최하위 비트를 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 썸 아웃 출력 오류검출코드와 동일한 썸 아웃 출력 오류검출코드를 생성하고, 상기 3비트 출력 데이터의 최상위 비트가 '1'일 경우에는 중간비트 및 최하위 비트의 논리 부정값을 입력으로 하는 상기 2비트 출력 오류검출코드 생성수단의 썸 아웃 출력 오류검출코드와 동일한 코드로 상기 썸 아웃 출력 오류검출코드를 생성한다.

[0144] 또한, 도 11은 상기 3비트 출력 오류검출코드의 진리테이블(122)을 보여주는 것으로 도 11을 참조하면, 상기 3비트 출력 오류검출코드 생성수단은 상기 썸 아웃 데이터(S)가 3비트 출력 데이터($F_{2(2bit)}, F_{1(2bit)}, F_{0(2bit)}$)일 경우, 상기 3비트 출력 오류검출코드 진리테이블(122)을 이용하여 2비트의 썸 아웃 출력 오류검출코드($C_{1'(3bit)}, C_{0'(3bit)}$)를 생성한다.

[0145] 또한, 상기 3비트 출력 오류검출코드 진리테이블(122)은 아래의 수학적식 4를 이용하여 계산된다.

수학적식 4

$$(C_{1'(3bit)}, C_{0'(3bit)}) = \begin{cases} SEDC_{2-out}(F_{1(3bit)}, F_{0(3bit)}), & \text{if } F_{2(3bit)}=0 \\ 1's \text{ complement}(SEDC_{2-out}(\overline{F_{1(3bit)}}), \overline{F_{0(3bit)}}), & \text{if } F_{2(3bit)}=1 \end{cases}$$

[0147] 또한, 상기 3비트 출력 오류 검출코드 생성수단은 상기 2비트 출력 오류 검출코드 생성수단이 캐리 아웃 출력 오류검출코드를 생성하는 로직과 동일하게 캐리 아웃 출력 오류검출코드를 생성하므로 자세한 설명을 생략하기로 한다.

[0148] 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.

[0149] 그러나, 상기 3비트 출력 오류검출코드 생성수단은 3비트의 출력 데이터에 생성되는 오류검출코드가 도 11의 진리 테이블을 만족할 수 있다면, 어떠한 논리 게이트들의 조합으로도 설계가 가능하다.

[0150] 상기 4비트 출력 오류검출코드 생성수단은 상기 썸 아웃 데이터(S)가 4비트 출력 데이터일 경우, 상기 4비트 출력 데이터들 중, 최상위 비트를 제외한 3비트를 입력으로 하는 상기 3비트 출력 오류검출코드 생성수단의 썸 아웃 출력 오류검출코드를 하위 2비트로 생성하고, 최상위 비트는 상기 4비트 출력 데이터의 최상위 비트의 논리 부정 값으로 하는 3비트의 썸 아웃 출력 오류검출코드를 생성한다.

[0151] 또한, 도 13은 상기 4비트 출력 오류검출코드의 진리테이블(123)을 보여주는 것으로 도 13을 참조하면, 상기 4비트 출력 오류검출코드 생성수단은 상기 썸 아웃 데이터(S)가 4비트 출력 데이터($F_{3(2bit)}, F_{2(2bit)}, F_{1(2bit)}, F_{0(2bit)}$)일 경우, 상기 4비트 출력 오류검출코드 진리테이블(123)을 이용하여 3비트의 썸 아웃 출력 오류검출코드($C_2'(3bit), C_1'(3bit), C_0'(3bit)$)를 생성한다.

[0152] 또한, 상기 4비트 출력 오류검출코드 진리테이블(123)은 아래의 수학적식 5를 이용하여 계산된다.

수학적식 5

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

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

[0154] 또한, 상기 4비트 출력 오류검출코드 생성수단은 상기 캐리 아웃 출력 오류검출코드에 대해서는 상기 2비트 출력 오류 검출코드 생성수단이 캐리 아웃 출력 오류검출코드를 생성하는 로직과 동일한 방법으로 생성하므로 자세한 설명은 생략하기로 한다.

[0155] 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.

[0156] 상기 n비트 출력 오류검출코드 생성수단은 상기 썸 아웃 데이터(S)가 n비트 출력 데이터일 경우, 길이가 가변하는 확장형의 n비트 출력 오류검출코드를 생성한다.

[0157] 또한, 상기 n비트 출력 오류검출코드 생성수단은 상기 썸 아웃 데이터(S)를 하나의 b비트의 출력 데이터와 하나 또는 복수 개의 3비트 데이터 세트로 구분하여 각각 출력 오류검출코드를 계산한 후 서로 연결하여 하나의 출력 오류검출코드를 생성한다.

[0158] 또한, b비트의 데이터와 3비트의 데이터 세트에 대해서는 상기 n비트 입력 오류검출코드 생성수단의 설명에서 보인바 있으므로 설명을 생략하기로 한다.

[0159] 단, 여기서 주의할 점은 상기 n비트 출력 데이터를 b비트 및 3비트 데이터 세트로 구분하는 방법과 상기 n비트 입력 데이터를 b비트 및 3비트 데이터 세트로 구분하는 방법은 서로 동일하여야 하며, b비트 및 각 3비트 데이터 세트에 대해 생성된 오류검출코드를 동일한 방법으로 연결하여 서로 동일성 여부를 판단하여야 한다.

[0160] 또한, 상기 n비트 출력 오류검출코드 생성수단은 상기 캐리 아웃 출력 오류검출코드에 대해서는 상기 2비트 출력 오류 검출코드 생성수단이 캐리 아웃 출력 오류검출코드를 생성하는 로직과 동일한 방법으로 생성하므로 자세한 설명은 생략하기로 한다.

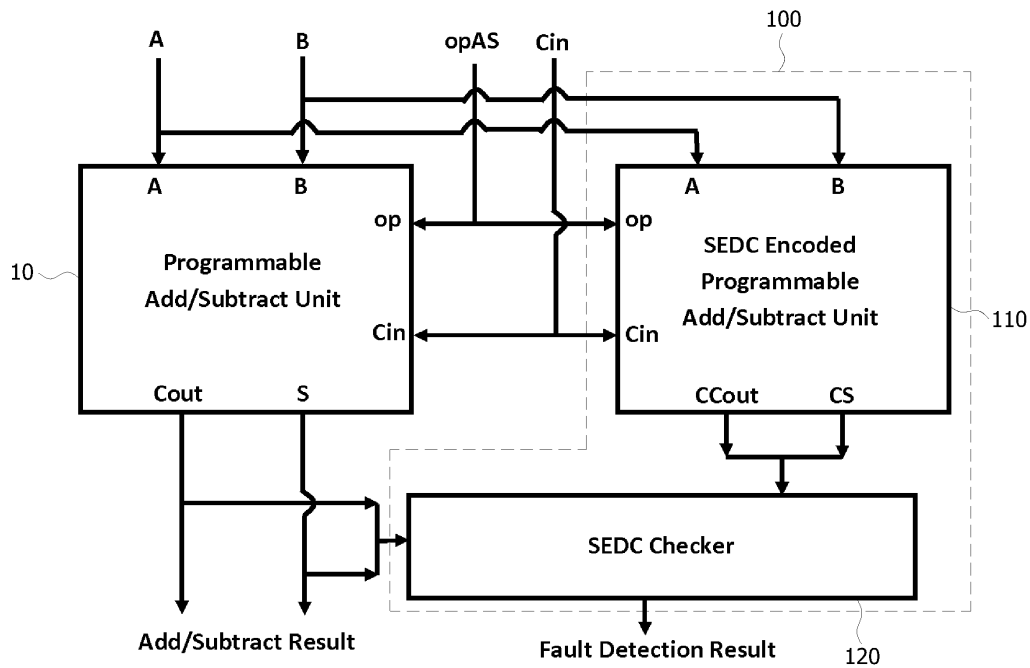
[0161] 이상에서 살펴본 바와 같이 본 발명은 바람직한 실시예를 들어 도시하고 설명하였으나, 상기한 실시예에 한정되지 아니하며 본 발명의 정신을 벗어나지 않는 범위 내에서 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변경과 수정이 가능할 것이다.

부호의 설명

[0162] 10:프로그래머블 덧셈/뺄셈 연산 유닛 100:오류 검출 장치
110:입력 오류검출코드 생성기 120:오류 검출기

도면

도면1



도면2

111

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

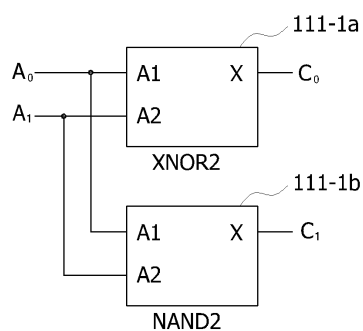
111a

111b

111c

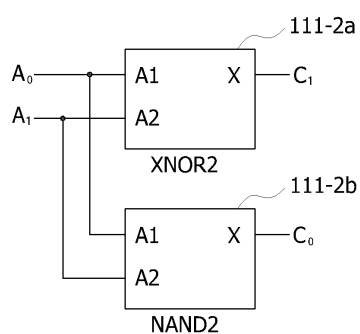
도면3

111-1



도면4

111-2



도면5

Operations	opAS	1-bit SEDC Encoded Add/Subtract Operations
A + B + Cin	0	$CCout = \text{lsb} \left\{ \text{SEDC}_2 \left(\begin{cases} A \text{ AND } B, & \text{if } Cin = 0 \\ A \text{ OR } B, & \text{otherwise} \end{cases} \right) \right\}$ $CS = \text{lsb} \left\{ \text{SEDC}_2 \left(\begin{cases} A \text{ XOR } B, & \text{if } Cin = 0 \\ A \text{ XNOR } B, & \text{otherwise} \end{cases} \right) \right\}$
A - B - Cin	1	$CCout = \text{lsb} \left\{ \text{SEDC}_2 \left(\begin{cases} A \text{ AND } \overline{B}, & \text{if } \overline{Cin} = 0 \\ A \text{ OR } \overline{B}, & \text{otherwise} \end{cases} \right) \right\}$ $CS = \text{lsb} \left\{ \text{SEDC}_2 \left(\begin{cases} A \text{ XOR } \overline{B}, & \text{if } \overline{Cin} = 0 \\ A \text{ XNOR } \overline{B}, & \text{otherwise} \end{cases} \right) \right\}$

도면6

2-bit SEDC Encoded Add Operation							
Operations	opAS	A	B	SEDC ₂ Cout		SEDC ₂ Sum	
				Cin = 0	Cin = 1	Cin = 0	Cin = 1
A + B + Cin	0	00	00	1	1	11	10
		00	01	1	1	10	10
		00	10	1	1	10	01
		00	11	1	0	01	11
		01	00	1	1	10	10
		01	01	1	1	10	01
		01	10	1	0	01	11
		01	11	0	0	11	10
		10	00	1	1	10	01
		10	01	1	0	01	11
		10	10	0	0	11	10
		10	11	0	0	10	10
		11	00	1	0	01	11
		11	01	0	0	11	10
		11	10	0	0	10	10
		11	11	0	0	10	01

도면7

3-bit SEDC Encoded Add Operation						
$(C_c, C_{s1}, C_{s0}) = 3\text{-bit SEDC Encoded Add Operation}((A_2, A_1, A_0), (B_2, B_1, B_0), \text{Cin}, \text{opAS})$						
$(t_c, t_{s1}, t_{s0}) = 2\text{-bit SEDC Encoded Add Operation}((A_1, A_0), (B_1, B_0), \text{Cin}, \text{opAS})$						
$C_c = S_{14}$						
$(C_{s1}, C_{s0}) = \begin{cases} (t_{s1}, t_{s0}) - 1, & \text{if } S_{15} = 1 \\ (t_{s1}, t_{s0}), & \text{otherwise} \end{cases}$						
Operations	opAS	A ₂	B ₂	t _c	S ₁₄	S ₁₅
A + B + Cin	0	0	0	0	1	1
		0	0	1	1	0
		0	1	0	0	0
		0	1	1	1	1
		1	0	0	0	0
		1	0	1	1	1
		1	1	0	0	1
		1	1	1	0	0

도면8

4-bit SEDC Encoded Add Operation						
$(C_c, C_{s2}, C_{s1}, C_{s0}) = 4\text{-bit SEDC Encoded Add Operation}((A_3, A_2, A_1, A_0), (B_3, B_2, B_1, B_0), \text{Cin}, \text{opAS})$						
$(t_c, t_{s1}, t_{s0}) = 3\text{-bit SEDC Encoded Add Operation}((A_2, A_1, A_0), (B_2, B_1, B_0), \text{Cin}, \text{opAS})$						
$C_c = S_{16}$ $C_{s2} = \begin{cases} \text{NOT } t_c & \text{if } S_{17} = 1 \\ t_c & \text{otherwise} \end{cases}$ $(C_{s1}, C_{s0}) = (t_{s1}, t_{s0})$						
Operations	opAS	A ₃	B ₃	t _c	S ₁₆	S ₁₇
A + B + Cin	0	0	0	0	1	0
		0	0	1	1	0
		0	1	0	0	1
		0	1	1	1	1
		1	0	0	0	1
		1	0	1	1	1
		1	1	0	0	0
		1	1	1	0	0

도면9

n-bit SEDC Encoded Add/Subtract Operations (n ≥ 2)				
b-bit SEDC Encoded Add/Subtract Operations((A _{n-1} , A _{n-2} ,..., A _{n-b}), (B _{n-1} , B _{n-2} ,..., B _{n-b}), Cin1, opAS)				
Operations	opAS	a	Cin1	B
A + B + Cin	0	0	Cin	B
		≠ 0	C _K	
A - B - Cin	1	0	$\overline{\text{Cin}}$	$\overline{B}$
		≠ 0	$\overline{C_K}$	
(C _K , CS _{m-L-1} ,..., CS _{m-(2×k)-L-2}) = 3-bit SEDC Encoded Add/Subtract Operations((A _{nn-1} , A _{nn-2} ,..., A ₀), (B _{nn-1} , B _{nn-2} ,..., B ₀), Cin2, opAS)				
Operations	opSRbits	K k = 0 to (a-1)	Cin2	B
A + B + Cin	0	= (a-1)	Cin	B
		≠ (a-1)	C _{K+1}	
A - B - Cin	1	= (a-1)	$\overline{\text{Cin}}$	$\overline{B}$
		≠ (a-1)	$\overline{C_{K+1}}$	
$a = \frac{n - \max(b)}{3}$ $nn = n - (3 \times k) - b$ $k = 0 \text{ to } (a-1)$ $L = \lceil \log_2 [n+1-(3 \times a)] \rceil$				

도면10

121

F		SEDC _{2-out} (F) Case1		SEDC _{2-out} (F) Case2	
F ₁ (2bit)	F ₀ (2bit)	C ₁ '(2bit)	C ₀ '(2bit)	C ₁ '(2bit)	C ₀ '(2bit)
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

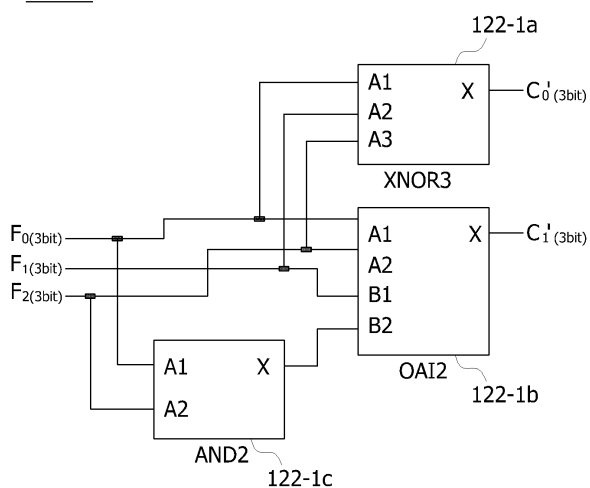
도면11

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

