(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년12월18일
(11) 등록번호 10-1930583
(24) 등록일자 2018년12월12일

(51) 국제특허분류(Int. Cl.)
H03M 13/15 (2015.01) *H03M 13/13* (2006.01)
(21) 출원번호 10-2014-7029911
(22) 출원일자(국제) 2012년11월26일
심사청구일자 2017년11월15일
(85) 번역문제출일자 2014년10월24일
(65) 공개번호 10-2014-0142320
(43) 공개일자 2014년12월11일
(86) 국제출원번호 PCT/US2012/066554
(87) 국제공개번호 WO 2013/147935
국제공개일자 2013년10월03일
(30) 우선권주장
13/430,222 2012년03월26일 미국(US)
(56) 선행기술조사문헌
US4777635 A
US4293951 A

(73) 특허권자
자일링크스 인코포레이티드
미합중국 95124 캘리포니아 산 호세 로직 드라이브 2100
(72) 발명자
크리시난 칼야나
미국 캘리포니아주 95124 세너제이 로직 드라이브 2100
탄 하이-조
미국 캘리포니아주 95124 세너제이 로직 드라이브 2100
(74) 대리인
김태홍

전체 청구항 수 : 총 15 항 심사관 : 조춘근

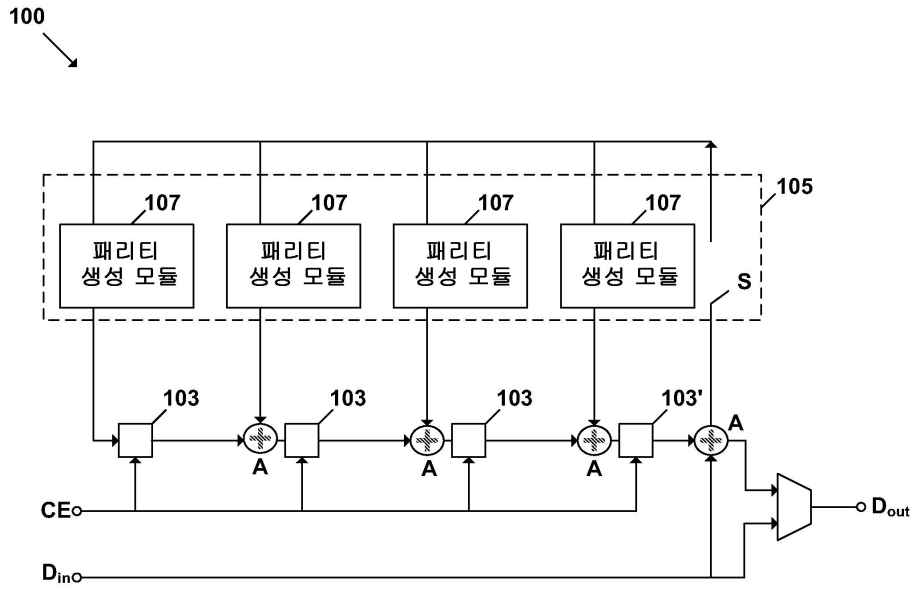
(54) 발명의 명칭 **비이진 선형 블록 코드에 대한 병렬 인코딩**

(57) 요 약

인코더 모듈(400)은 순차적으로 연결되는 P/L개의 패리티 시프트 레지스터(403, 403', 403")를 포함하며, 패리티 시프트 레지스터(403, 403', 403") 중 제1 패리티 시프트 레지스터(403')의 입력은 인코더 모듈(400)의 입력(D_{in})에 연결되고, 패리티 시프트 레지스터(403, 403', 403") 중 마지막 패리티 시프트 레지스터(403")의 출력은

(뒷면에 계속)

대 표 도 - 도1



인코더 모듈(400)의 출력(D_{out})에 연결되며, 패리티 시프트 레지스터(403, 403', 403")의 각각은 L 패리티 자릿수를 저장하도록 구성된다. 인코더 모듈(400)은 또한 P/L개의 패리티 생성 모듈(407)을 포함하는 피드백 모듈(405)을 구비하며, 패리티 생성 모듈(407)의 각각은 스위치(S1, S2, S3, S4)에 의해 패리티 시프트 레지스터(403, 403', 403") 중 대응하는 하나의 출력에 연결되고, 제1 패리티 시프트 레지스터(403')의 입력에 또한 연결되며, 패리티 생성 모듈(407)의 각각은 그 대응하는 스위치(S1, S2, S3, S4)가 폐쇄될 때 제1 패리티 시프트 레지스터(403')의 입력에 송신하기 위한 L 패리티 자릿수를 생성하도록 구성된다.

명 세 서

청구범위

청구항 1

K개 자릿수(digit) 메시지의 L개 자릿수 부분들을 P개 패리티(parity) 자릿수를 갖는 N개 자릿수의 코딩된 메시지로 인코딩하는 인코더 모듈에 있어서,

상기 K개 자릿수 메시지의 L개 자릿수 부분들을 수신하도록 구성되는 입력;

상기 N개 자릿수의 코딩된 메시지를 출력하도록 구성되는 출력;

상기 입력 및 상기 출력에 연결되는 시프트 레지스터(shift register) 회로;

순차적으로 연결되는 P/L개의 패리티 시프트 레지스터로서, 상기 패리티 시프트 레지스터 중 마지막 패리티 시프트 레지스터의 입력이 상기 인코더 모듈의 입력에 연결되고, 상기 마지막 패리티 시프트 레지스터의 출력이 상기 인코더 모듈의 출력에 연결되는, P/L개의 패리티 시프트 레지스터; 및

P/L개의 패리티 생성 모듈을 포함하는 피드백 회로를 포함하며, 상기 패리티 생성 모듈의 각각의 입력은 스위치에 의해 상기 마지막 패리티 시프트 레지스터의 출력에 연결되고, 상기 패리티 생성 모듈의 각각의 출력은 상기 패리티 시프트 레지스터 중 대응하는 하나의 입력에 연결되며, 상기 패리티 생성 모듈 중 하나는 생성 다항식(generator polynomial)에 기초하여 L개 패리티 자릿수를 생성하고, 상기 패리티 생성 모듈 중 하나에 대응하는 상기 패리티 시프트 레지스터 중 하나는 상기 스위치가 폐쇄될 때 L개 패리티 자릿수를 저장하며;

K, L, N 및 P는 양의 정수이고, L은 1보다 크고, P는 N-K이며;

상기 피드백 회로의 상기 패리티 생성 모듈 중 하나는 상기 마지막 패리티 시프트 레지스터로부터 데이터를 수신하고 (K/L) 클록 사이클 중 하나 동안 상기 L개 패리티 자릿수를 생성하며, 상기 패리티 시프트 레지스터 중 하나는 그에 저장된 정보를 (K/L)+2 내지 (N/L)+1 클록 사이클 중 하나 동안 상기 출력에 전송(forward)하도록 구성되고;

상기 스위치는 상기 (K/L) 클록 사이클 동안 폐쇄되며 상기 스위치는 상기 (K/L)+2 내지 (N/L)+1 클록 사이클 동안 개방되는 것인, 인코더 모듈.

청구항 2

제1항에 있어서, 상기 시프트 레지스터 회로는 상기 입력 및 상기 출력에 연결되는 L개 자릿수 입력 시프트 레지스터인 것인, 인코더 모듈.

청구항 3

제2항에 있어서,

각각의 (K/L) 클록 사이클 동안,

상기 피드백 회로의 스위치는 폐쇄되고;

상기 인코더 모듈은 그 입력에서 수신되는 상기 K개 자릿수 메시지의 상기 L개 자릿수 부분들 중 하나를 상기 입력 시프트 레지스터 및 상기 마지막 패리티 시프트 레지스터의 입력에 전송하며;

상기 마지막 패리티 시프트 레지스터는 그 저장된 패리티 자릿수를 상기 대응하는 패리티 시프트 레지스터에 저장될 L개 패리티 자릿수를 생성하는 상기 피드백 회로의 상기 패리티 생성 모듈의 각각에 전송하고;

상기 마지막 패리티 시프트 레지스터 이외의 상기 패리티 시프트 레지스터의 각각은 그 저장된 패리티 자릿수를 상기 패리티 시프트 레지스터 중 후속하는 하나에 전송하며;

상기 입력 시프트 레지스터는 그 저장된 데이터를 상기 인코더 모듈의 출력에 전송하고;

(K/L)+1 클록 사이클 동안:

상기 입력 시프트 레지스터는 그 저장된 데이터를 상기 인코더 모듈의 출력에 전송하며;

상기 인코더 모듈은 그 입력에서 수신되는 제로(0) 입력을 상기 입력 시프트 레지스터 및 상기 마지막 패리티 시프트 레지스터의 입력에 전송하고, 그에 의해 상기 피드백 회로의 상기 스위치를 개방하며;

각각의 (K/L)+2 내지 (N/L)+1 클록 사이클 동안:

상기 마지막 패리티 시프트 레지스터는 그 저장된 패리티 자릿수를 상기 인코더 모듈의 출력에 전송하고;

상기 마지막 패리티 시프트 레지스터 이외의 상기 패리티 시프트 레지스터의 각각은 그 저장된 패리티 자릿수를 상기 패리티 시프트 레지스터 중 후속하는 하나에 전송하는 것인, 인코더 모듈.

청구항 4

제2항 또는 제3항에 있어서, 상기 마지막 패리티 시프트 레지스터의 출력 및 상기 입력 시프트 레지스터의 출력은 멀티플렉서를 통해 상기 인코더 모듈의 출력에 연결되는 것인, 인코더 모듈.

청구항 5

제1항 내지 제3항 중 어느 한 항에 있어서, 각각의 (K/L) 클록 사이클 동안 상기 마지막 패리티 시프트 레지스터의 입력에 전송되는 L개 자릿수 부분은 상기 각각의 (K/L) 클록 사이클에서 상기 마지막 패리티 시프트 레지스터에 저장되는 상기 L개 패리티 자릿수와 결합되는(combined) 것인, 인코더 모듈.

청구항 6

제1항 내지 제3항 중 어느 한 항에 있어서, 각각의 (K/L) 클록 사이클 동안 각각의 패리티 생성 모듈에 의해 생성되는 상기 L개 패리티 자릿수는 상기 각각의 (K/L) 클록 사이클에서 대응하는 패리티 시프트 레지스터에 저장되는 상기 L개 패리티 자릿수와 결합되는 것인, 인코더 모듈.

청구항 7

제3항에 있어서, 상기 (K/L)+1 클록 사이클에서 각각의 상기 패리티 시프트 레지스터에 저장되는 상기 L개 패리티 자릿수는 최종 패리티 자릿수 값인 것인, 인코더 모듈.

청구항 8

제1항 또는 제2항에 있어서, 상기 P개 패리티 자릿수는 (K/L)+2 내지 (N/L)+1 클록 사이클 동안 상기 인코더 모듈에 의해 출력되는 것인, 인코더 모듈.

청구항 9

제8항에 있어서, 상기 P개 패리티 자릿수는 L개 자릿수 병렬 출력으로서 출력되는 것인, 인코더 모듈.

청구항 10

제1항 또는 제2항에 있어서, 상기 패리티 시프트 레지스터는 상기 패리티 시프트 레지스터의 각각에 연결되는 제어 신호에 의해 제어되는 것인, 인코더 모듈.

청구항 11

제1항 또는 제2항에 있어서, 상기 인코더 모듈에 의해 수신되는 상기 K개 자릿수 메시지의 상기 L개 자릿수 부분들은 (K/L)+1 클록 사이클 동안 상기 인코더 모듈에 의해 출력되는 것인, 인코더 모듈.

청구항 12

제1항에 있어서, 상기 입력은 상기 L개 자릿수 부분들을 병렬 입력으로서 수신하도록 구성되는 것인, 인코더 모듈.

청구항 13

제1항에 있어서, 상기 생성 다항식은 q^m차 갈루아 필드(Galois field) 상에 정의되는 것인, 인코더 모듈.

청구항 14

제13항에 있어서, q는 소수(prime number)인 것인, 인코더 모듈.

청구항 15

제13항에 있어서, 상기 P/L개의 패리티 생성 모듈 중 적어도 하나는 mL-1개의 XOR을 이용하여 실현되는 것인, 인코더 모듈.

발명의 설명

기 술 분 야

[0001] 저작물에서의 권리의 보유

[0002] 이 특허 문헌의 개시내용의 일부는 저작권 보호의 대상인 저작물을 포함한다. 저작권 소유자는 특허 문헌이나 특허 개시물 중 어느 것에 의한 복사에 이의가 없는데, 그 이유는 특허 상표청의 특허 파일이나 기록에 나타나 있기 때문이지만, 이와 달리 모든 저작권을 보유한다.

[0003] 발명의 분야

[0004] 본 출원은 일반적으로 비이진 선형 블록 코드를 인코딩하는 것에 관한 것으로, 특히 비이진 선형 블록 코드의 병렬 인코딩를 실현하기 위한 시스템 및 방법에 관한 것이다.

배 경 기 술

[0005] 선형 순방향 에러 정정(FEC: forward error correcting) 코드는 데이터 무결성 및 정확성이 보호되어야 하는 통신 및 대용량 저장 시스템에서의 넓은 애플리케이션을 갖는다. 선형 블록 FEC는 데이터 자릿수의 블록을 취하고 다수의 중복 자릿수(redundant digit)를 가산함으로써 데이터 블록을 보호한다. 일반적으로, 인코더는 이들 중복 자릿수를 계산하는 한편, 디코더는 중복 자릿수를 이용하여 수신된 데이터 블록의 정확도를 결정하고, 가능한 경우, 중복 자릿수를 이용하여 데이터의 오류 블록을 정정한다.

[0006] 선형 블록 코드는 표기법 (n,k)에 의해 표시되며, 여기에서 k 메시지 자릿수는 인코더 모듈에 의해 n 코딩 자릿수로 인코딩된다. 그와 같이, n-k 중복 자릿수(redundant digits)는 인코더에 의해 k 메시지 자릿수(예를 들면, 데이터 블록)에 가산된다. n-k 중복 자릿수는 또한 패리티 자릿수(parity digits)라고 칭해진다. 조직 부호(systematic code)가 실현되는 경우, n 코딩 자릿수는 간단히 n-k 패리티 자릿수가 뒤따르는 k 메시지 자릿수(예를 들면, 데이터 블록)이다.

[0007] 패리티 자릿수는 갈루아 필드(GF: Galois Field) 상에 정의되는 생성 다항식 g(x)를 기초로 하여 계산된다. 이진 선형 블록 코드에 있어서, 생성 다항식 g(x)는 2차(GF(2)) 갈루아 필드 상에 정의된다. 비이진 선형 블록 코드에 있어서, 생성 다항식 g(x)는 q^m차(GF(q^m)) 갈루아 필드 상에 정의되며, 여기에서 q는 소수이다. 리드 솔로몬(RS: Reed Solomon) 인코더는 비이진 선형 블록 코드의 분류에 속한다.

[0008] 10 Gbps 인코더에 있어서, 달성될 수 있는 최대의 선속이 11.09 Gbit/sec이다. 단일의 입력이 한 번에 처리되는 직렬 입력을 사용하면, 그러한 선속을 달성하는 데는 (11.09/m)=1.109 GHz에서 동작하기 위한 클록킹이 필요하게 된다. 패리티 자릿수를 계산하는 데 수반되는 복잡한 로직으로 인해, 직렬 입력을 사용하면서 그러한 선속을 유지하는 것은 달성하기 매우 어렵다.

발명의 내용

[0009] 일부 실시예에 따르면, K 자릿수 메시지의 L 자릿수 부분을 P 패리티 자릿수를 갖는 N 자릿수 코딩된 메시지로 병렬 인코딩하는 인코더 모듈이 상기 K 자릿수 메시지의 L 자릿수 부분을 수신하도록 구성되는 입력 및 상기 N 자릿수 코딩된 메시지를 출력하도록 구성되는 출력을 포함하며, 여기에서 상기 인코더 모듈의 입력이 상기 출력에 연결된다. 상기 인코더 모듈은 또한 순차적으로 연결되는 P/L개의 패리티 시프트 레지스터를 포함하며, 여기에서 상기 패리티 시프트 레지스터 중 제1 패리티 시프트 레지스터의 입력은 상기 인코더 모듈의 입력에 연결되고, 상기 패리티 시프트 레지스터 중 마지막 패리티 시프트 레지스터의 출력은 상기 인코더 모듈의 출력에 연결되며, 상기 패리티 시프트 레지스터의 각각은 L 패리티 자릿수를 저장하도록 구성된다. 인코더 모듈은 또한 P/L개의 패리티 생성 모듈을 포함하는 피드백 모듈을 구비하며, 여기에서 상기 패리티 생성 모듈의 각각은 스위

치에 의해 상기 패리티 시프트 레지스터 중 대응하는 하나의 출력에 연결되고, 상기 제1 패리티 시프트 레지스터의 입력에 또한 연결되며, 상기 패리티 생성 모듈의 각각은 그 대응하는 스위치가 폐쇄될 때 상기 제1 패리티 시프트 레지스터의 입력에 송신하기 위한 L 패리티 자릿수를 생성하도록 구성된다.

[0010] 다른 실시예에 따르면, K 자릿수 메시지의 L 자릿수 부분을 P 패리티 자릿수를 갖는 N 자릿수 코딩된 메시지로 병렬 인코딩하는 인코더 모듈이 상기 K 자릿수 메시지의 L 자릿수 부분을 수신하도록 구성되는 입력 및 상기 N 자릿수 코딩된 메시지를 출력하도록 구성되는 출력을 포함하며, 여기에서 상기 인코더 모듈의 입력이 상기 출력에 연결된다. 상기 인코더 모듈은 또한 상기 입력 및 상기 출력에, 그리고 순차적으로 연결되는 P/L개의 패리티 시프트 레지스터에 연결되는 P/L 단 딜레이(P/L stage delay)를 포함하고, 여기에서 상기 패리티 시프트 레지스터 중 제1 패리티 시프트 레지스터의 입력은 상기 인코더 모듈의 입력에 연결되고, 상기 패리티 시프트 레지스터 중 마지막 패리티 시프트 레지스터의 출력은 상기 인코더 모듈의 출력에 연결되며, 상기 패리티 시프트 레지스터의 각각은 L 패리티 자릿수를 저장하도록 구성된다. 인코더 모듈은 또한 P/L개의 패리티 생성 모듈을 포함하는 피드백 모듈을 구비하며, 여기에서 상기 패리티 생성 모듈의 각각은 스위치에 의해 상기 패리티 시프트 레지스터 중 대응하는 하나의 출력에 연결되고, 상기 제1 패리티 시프트 레지스터의 입력에 또한 연결되며, 상기 패리티 생성 모듈의 각각은 그 대응하는 스위치가 폐쇄될 때 상기 제1 패리티 시프트 레지스터의 입력에 송신하기 위한 L 패리티 자릿수를 생성하도록 구성된다.

[0011] 다른 및 추가의 양태 및 특징들이 아래의 실시예의 상세한 설명을 보면 명확해질 것이다.

[0012] 도면은 실시예의 설계 및 효용성을 예시하며, 여기에서 유사한 구성요소는 공통 참조 번호로 나타낸다. 이들 도면은 반드시 크기 변경하도록 도시된 것은 아니다. 상기 인용된 및 다른 장점 및 목적이 어떻게 획득되는지 더욱 잘 이해하기 위해, 첨부하는 도면에 예시되는 실시예의 더욱 특정한 설명이 제공될 것이다. 이들 도면은 단지 전형적인 실시예를 도시할 뿐이므로, 청구항들의 범위를 제한하는 것으로 생각되어서는 안된다.

도면의 간단한 설명

[0013] 도 1은 비이진 선형 블록 코드의 직렬 인코딩을 위한 직렬 인코더 모듈을 도시한다.

도 2는 비이진 선형 블록 코드의 병렬 인코딩을 실현하는 인코더 모듈을 도시한다.

도 3은 일부 실시예에 따라서 비이진 선형 블록 코드의 병렬 인코딩을 실현하는 인코더 모듈을 도시한다.

도 4는 일부 실시예에 따라서 감소된 팬 아웃을 갖는 비선형 블록 코드의 병렬 인코딩을 실현하기 위한 인코더 모듈을 도시한다.

발명을 실시하기 위한 구체적인 내용

[0014] 이하 도면을 참조하여 여러 가지 실시예를 기술한다. 도면들은 크기 변경하도록 도시된 것이 아니고, 유사한 구성 또는 기능의 구성요소가 전체 도면에 걸쳐 같은 참조 번호로 나타내고 있음을 주의해야 한다. 도면들은 실시예의 설명을 용이하게 하도록만 의도된 것임을 주의해야 한다. 그들 도면은 발명의 완전한 설명으로서 또는 청구되는 발명의 범위에 대한 제한으로서 의도되는 것은 아니다. 또한, 예시된 실시예는 도시된 모든 양태나 장점을 가질 필요는 없다. 특정 실시예와 관련하여 기술되는 양태나 장점은 그 실시예로 반드시 제한되는 것이 아니라 예시되지 않았다고 하더라도 임의의 다른 실시예에서 실시될 수 있다. 또한, 이 명세서 전체에 걸쳐 "일부 실시예" 또는 "다른 실시예"에 대한 기준은 실시예와 관련하여 기술된 특정 특징, 구성, 재료 또는 특성이 적어도 하나의 실시예에 포함되는 것을 의미한다. 그러므로, 이 명세서의 전체에 걸쳐 다양한 장소에서의 "일부 실시예에서" 또는 "다른 실시예에서"의 출현은 반드시 동일한 실시예 또는 실시예들에 관한 것은 아니다.

[0015] 선형 블록 코드는 표기법 (n,k)로 표시되고, 여기에서 k 메시지 자릿수는 인코더 모듈에 의해 n 코딩 자릿수로 인코딩된다. 그와 같이, n-k 중복 자릿수는 인코더에 의해 k 메시지 자릿수(예를 들면, 데이터 블록)에 가산된다. n-k 중복 자릿수는 또한 패리티 자릿수라고 칭해진다. 조직 부호가 실현되는 경우, n 코딩 자릿수는 간단히 n-k 패리티 자릿수가 뒤따르는 k 메시지 자릿수(예를 들면, 데이터 블록)이다. 패리티 자릿수는 갈루아 필드(GF) 상에 정의되는 생성 다항식 g(x)를 기초로 하여 계산된다. 비이진 선형 블록 코드에 있어서, 생성 다항식 g(x)는 q^m차(GF(q^m)) 갈루아 필드 상에 정의되며, 여기에서 q는 소수이다.

[0016] 예를 들 목적으로, 설명의 나머지는 2^10차(GF(2^10)) 갈루아 필드 상에 정의되는 생성 다항식에 대해 기술된다. 그러나, 비이진 블록 코드의 병렬 인코딩을 위한 방법 및 시스템은 임의의 차수의 갈루아 필드 상에 정의되는 생성 다항식을 포함하도록 확장될 수도 있다.

[0017] 인코딩될 k 자릿수 메시지는 아래의 메시지 다항식으로 표현되며, 여기에서 m_{N-K}는 2^m차 갈루아 필드에서의 자릿수에 대응한다:

$$m(x) = \sum_{i=1}^{K} m_{i-1} x^{K-i} \qquad (1)$$

[0018]

[0019] N은 코딩된 메시지 길이를 나타내고 P는 패리티 자릿수 길이(P=N-K)를 나타낸다. 생성 다항식은 2^m차 갈루아 필드 상에 정의되는 g(x)로 표현된다.

[0020] n 자릿수 코딩된 메시지는 아래의 다항식으로 표현되며, 여기에서 p(x)의 계수는 패리티 자릿수이고 c(x)의 계수는 최종 인코딩된 자릿수이다:

$$p(x) = (x^P m(x))_{g(x)} \qquad (2)$$

$$c(x) = x^P m(x) + p(x) \qquad (3)$$

[0021]

[0022] $(\ )_{g(x)}$는 다항식 g(x)에 대한 모듈로 연산(modulo operation)을 나타낸다.

[0023] p(x)의 계산은 아래의 방식으로 확장될 수 있다:

$$p(x) = \left(x^P \sum_{i=1}^{K} m_{i-1} x^{K-i}\right)_{g(x)} = \left(\sum_{i=1}^{K} x^P m_{i-1} x^{K-i}\right)_{g(x)}$$

(4)

$$= \left(x^{K-1} x^P m_0\right)_{g(x)} + \left(x^{K-2} x^P m_1\right)_{g(x)} + \left(x^{K-3} x^P m_2\right)_{g(x)} + \left(x^{K-4} x^P m_3\right)_{g(x)} + \ldots$$

[0024]

[0025] $[a(x)b(x)]_{g(x)} = ([a(x)]_{g(x)}[b(x)]_{g(x)})_{g(x)}$ 인 사실을 이용하면, 식 (4)는 아래와 같이 반복 방식으로 더 고쳐쓰기 될 수 있다:

$$p(x) = \left(x^{K-2}\left(x\left[x^P m_0\right]_{g(x)} + x^P m_1\right)_{g(x)}\right)_{g(x)} + \left(x^{K-3} x^P m_2\right)_{g(x)} + \left(x^{K-4} x^P m_3\right)_{g(x)} + \ldots \quad (5)$$

$$= \left(x^{K-3}\left[x\left(x\left[x^P m_0\right]_{g(x)} + x^P m_1\right)_{g(x)} + x^P m_2\right]_{g(x)}\right)_{g(x)} + \left(x^{K-4} x^P m_3\right)_{g(x)} + \ldots$$

[0026]

[0027] 상기 반복 식은 아래의 의사 코드에 도시되는 바와 같이 실현될 수 있다:

```
for i = 1:K
    parityv = parityt;% parity corresponds to the array containing the
parity coefficients
    feedback = (m(i) + parityt(n-k)) * g(x);%m(i) are the input
message digits.
    parityt(1) = feedback(1);
    parityt(2:n-k) = feedback(2:n-k) + parityv(1:n-k-1);
end;
```

[0028]

[0029] 기술된 알고리즘은 도 1에 예시된 바와 같은 직렬 인코더 모듈을 사용하여 실현될 수도 있다. 도 1은 비이진 선형 블록 코드의 직렬 인코딩을 위한 직렬 인코더 모듈(100)을 도시한다. 직렬 인코더 모듈(100)은 K 자릿수

메시지를 위한 임의의 P개의 패리티 자릿수를 생성하도록 구성될 수도 있다. 그러나, 예를 들 목적으로, 도 1의 직렬 인코더 모듈(100)은 K 자릿수 메시지를 위한 4 패리티 자릿수를 생성하도록 구성된다.

[0030] 직렬 인코더 모듈(100)은 입력(D_{in}), 출력(D_{out}), 4개의 단 자릿수 패리티 시프트 레지스터(103/103') 및 4개의 패리티 생성 모듈(107)을 포함하는 피드백 회로(105)를 구비한다. 도 1의 직렬 인코더 모듈(100)은 단지 4개의 패리티 생성 모듈(107)(예를 들면, 갈루아 필드 승산기) 및 4개의 패리티 시프트 레지스터(103/103')를 구비하고 있지만, 직렬 인코더 모듈은 임의의 P개의 패리티 시프트 레지스터 및 생성될 P개의 패리티 자릿수에 대응하는 임의의 P개의 패리티 생성 모듈을 구비할 수도 있다.

[0031] 인코더 모듈(100)의 입력(D_{in})은 직렬 입력으로서 K 자릿수 메시지를 수신하도록 구성되고, 직렬 인코더 모듈(100)의 출력(D_{out})은 P 패리티 자릿수를 갖는 N 자릿수 코딩된 메시지를 출력하도록 구성된다. 피드백 회로(105)의 패리티 생성 모듈(107)은 이후에 상세히 논의되는 바와 같이, 패리티 시프트 레지스터(103/103')에 저장되는 패리티 자릿수를 생성하도록 구성된다.

[0032] 4개의 패리티 시프트 레지스터(103/103')는 순차적으로 연결되므로, 패리티 시프트 레지스터(103/103')의 출력이 (후속하는 패리티 시프트 레지스터가 존재하는 경우) 후속하는 패리티 시프트 레지스터(103/103')의 입력에 연결되게 된다. 각 패리티 시프트 레지스터(103/103')는 패리티 생성 모듈(107)에 대응하고, 각 패리티 생성 모듈(107)은 그 대응하는 패리티 시프트 레지스터(103/103')의 입력에 연결된다. 가산기 모듈(A)이 다수의 소스(예를 들면, 패리티 생성 모듈 및 선행하는 패리티 시프트 레지스터)에 연결되는 각 패리티 시프트 레지스터(103/103')의 입력과 연관될 수도 있다. 가산기 모듈(A)은 저장을 위해 패리티 시프트 레지스터(103/103')의 입력에서 수신되는 데이터의 결합을 용이하게 한다. 4개의 패리티 시프트 레지스터(103/103') 중 마지막 패리티 시프트 레지스터(103')의 출력은 인코더 모듈(100)의 입력(D_{in})에 연결되고, 피드백 회로(105)의 각 패리티 생성 모듈(107)에 또한 연결된다. 마지막 패리티 시프트 레지스터(103')의 출력은 멀티플렉서를 통해 인코더 모듈(100)의 출력(D_{out})에 추가로 연결된다. 추가의 가산기 모듈(A)이 마지막 패리티 시프트 레지스터(103')의 출력과 연관될 수도 있다. 추가의 가산기 모듈(A)은 피드백 회로(105)의 패리티 생성 모듈(107)에 송신을 위해 마지막 패리티 시프트 레지스터(103')의 출력에서 수신되는 데이터의 결합을 용이하게 한다. 인코더 모듈(100)의 입력(D_{in})은 멀티플렉서를 통해 인코더 모듈(100)의 출력(D_{out})에 유사하게 연결된다.

[0033] 피드백 회로(105)는 스위치(S)를 더 포함한다. 스위치(S)가 폐쇄될 때, 피드백 회로(105)는 활성이 되고, 패리티 생성 모듈(107)은 패리티 자릿수를 활성적으로 생성하고 있다. 스위치(S)가 개방될 때, 피드백 회로(105)는 비활성화되고 패리티 생성 모듈(107)은 더 이상 패리티 자릿수를 생성하지 않는다.

[0034] 제1의 K(예를 들면, 메시지 자릿수의 수) 클록 사이클 동안, 피드백 회로(105)의 스위치(S)는 폐쇄된다. 제1의 K 클록 사이클의 매 클록 사이클 동안, K 자릿수 메시지의 자릿수가 직렬 입력으로서 인코더 모듈(100)의 입력(D_{in})에서 수신되어, 마지막 패리티 시프트 레지스터(103')의 출력에 전송되며, 그 후에 피드백 회로(105)의 각 패리티 생성 모듈(107)로 전송된다. 피드백 회로(105)의 각 패리티 생성 모듈(107)은 대응하는 패리티 시프트 레지스터(103, 103')에 저장되는 단일 패리티 자릿수를 생성하도록 구성된다. 각 패리티 생성 모듈(107)은 상술한 반복 식(5) 및 의사 코드에 따라서 K 자릿수 메시지의 자릿수마다 패리티 자릿수를 생성하도록 구성된다. 각 클록 사이클 동안, 각 패리티 생성 모듈(107)에 의해 생성되는 패리티 자릿수는 각각의 대응하는 패리티 시프트 레지스터(103/103')에 현재 저장되어 있는 데이터와 결합되어 갱신된 패리티 자릿수의 세트를 형성한다.

[0035] 또한, 제1의 K 클록 사이클의 각 클록 사이클 동안, 각 패리티 시프트 레지스터(103, 103')는 그 저장된 데이터를 (후속하는 패리티 시프트 레지스터가 존재하는 경우) 후속하는 패리티 시프트 레지스터에 송신한다. 마지막 패리티 시프트 레지스터(103')는 그 저장된 데이터를 피드백 회로(105)의 각 패리티 생성 모듈(107)에 송신한다. 패리티 시프트 레지스터(103/103')에 저장된 패리티 자릿수는, K 자릿수 메시지가 수신되고 새로운 패리티 자릿수가 패리티 시프트 레지스터(103/103')에 저장하기 위해 패리티 생성 모듈(107)에 의해 생성되므로, 클록 사이클마다 갱신된다. 상술한 바와 같이, 각 클록 사이클 동안, 새롭게 생성된 패리티 자릿수가 대응하는 패리티 시프트 레지스터(103/103')에 현재 저장되어 있는 데이터와 결합되어 갱신된 패리티 자릿수의 세트를 형성한다.

[0036] 동시에, 제1의 K 클록 사이클 동안, 인코더 모듈(100)의 입력(D_{in})에서 수신되는 K 자릿수 메시지의 각 자릿수는 멀티플렉서를 통해 인코더 모듈(100)의 출력(D_{out})에 전송되므로, 인코더 모듈(100)에 의해 출력되는 제1의 K 자

릿수가 K 자릿수 메시지가 된다.

[0037] K+1 내지 N 클록 사이클 동안, 피드백 회로(105)의 스위치(S)는 개방되고 패리티 시프트 레지스터(103/103')에 저장되는 최종 패리티 자릿수 값이 인코더 모듈(100)의 출력(D_{out})에 송신된다. 패리티 시프트 레지스터(103/103')에 저장되는 최종 패리티 자릿수 값은 각 패리티 시프트 레지스터(103/103')에 연결되는 제어 신호(CE)에 응답하여 출력된다. 예를 들면, K+1 클록 사이클 동안, 마지막 패리티 시프트 레지스터(103')에 저장된 패리티 자릿수가 멀티플렉서를 통해 인코더 모듈(100)의 출력(D_{out})에 송신될 수도 있고, 각각의 나머지 패리티 시프트 레지스터(103)에 저장된 패리티 자릿수는 각각의 가산기 모듈을 통해 후속하는 패리티 시프트 레지스터(103)에 송신될 수도 있다. 이것은 모든 패리티 자릿수가 인코더 모듈(100)에 의해 출력되었을 때, 제N 클록 사이클에 거쳐 지속한다. 이러한 방식으로, 인코더 모듈(100)은 N 자릿수 코딩된 메시지를 출력하며, 여기에서 제1의 K 자릿수는 K 메시지 자릿수이고 마지막 P 자릿수는 패리티 자릿수이다.

[0038] 도 1의 인코더 모듈이 K 자릿수 메시지를 P 패리티 자릿수를 갖는 N 자릿수 코딩된 메시지로 인코드하도록 동작하는 동안, 여러 가지 결점이 발생한다. 인코더 모듈이 직렬 방식으로 동작하기 때문에, 높은 선속을 달성하기 위해, 인코더 모듈은 매우 높은 주파수에서 동작해야 한다. 그러나, 패리티 생성 모듈을 구성하는 데 수반되는 로직의 복잡성으로 인해, 그러한 높은 주파수는 인코딩 프로세서 내로 여러 가지 에러 및 추가의 비용을 도입함 없이 달성될 수 없다.

[0039] 비이진 선형 블록 코드의 병렬 인코딩을 실현하는 인코더 모듈을 위한 한 가지 방법은 도 2에 도시된다. 도 1의 직렬 인코더 모듈(100)에서는, K 자릿수 메시지의 단 자릿수만 클록 사이클마다 처리된다. 그러나, 도 2의 인코더 모듈(200)은 다수의 자릿수(L)가 클록 사이클마다 처리될 수 있게 한다. 인코더 모듈(200)은 K 자릿수 메시지를 위한 임의의 P개의 패리티 자릿수를 생성하도록 구성될 수도 있고, 여기에서 K 자릿수 메시지는 L 자릿수 병렬 입력으로서 수신되고 N 자릿수 코딩된 메시지는 L 자릿수 병렬 출력으로서 출력된다. 그러나, 예를 들 목적으로, 도 2의 인코더 모듈(200)은 K 자릿수 메시지를 위한 16 패리티 자릿수(예를 들면, P=16)를 생성하여 N 자릿수 코딩된 메시지를 형성하도록 구성되며, 여기에서 K 자릿수 메시지는 4 자릿수 병렬 입력(예를 들면, L=4)으로서 수신되고, N 자릿수 코딩된 메시지는 4 자릿수 병렬 출력으로서 출력된다. K가 L의 정수배가 아닌 경우, L의 배수인 전체 메시지 길이를 달성하기 위해 메시지의 시작부에 제로(0)들이 삽입된다.

[0040] 인코더 모듈(200)은 입력(D_{in}), 출력(D_{out}), 4개의 4 자릿수 패리티 시프트 레지스터(203/203') 및 4개의 패리티 생성 모듈(207)을 포함하는 피드백 회로(205)를 구비한다. 도 2의 인코더 모듈(200)은 단지 4개의 패리티 생성 모듈(207) 및 4개의 4 자릿수 패리티 시프트 레지스터(203/203')를 구비하고 있지만, 인코더 모듈(200)은 임의의 P/L개의 L 자릿수 패리티 시프트 레지스터 및 L 자릿수 병렬 입력이 수신되고 생성될 P개의 패리티 자릿수에 대응하는 임의의 P/L개의 패리티 생성 모듈을 구비할 수도 있다.

[0041] 인코더 모듈(200)의 입력(D_{in})은 4 자릿수 병렬 입력으로서 K 자릿수 메시지를 수신하도록 구성되고, 인코더 모듈(200)의 출력(D_{out})은 16 패리티 자릿수를 갖는 N 자릿수 코딩된 메시지를 4 자릿수 병렬 출력으로서 출력하도록 구성된다. 피드백 회로(205)의 패리티 생성 모듈(207)은 이후에 상세히 논의되는 바와 같이, 4 자릿수 패리티 시프트 레지스터(203/203')에 저장되는 4 패리티 자릿수를 생성하도록 각각 구성된다.

[0042] 4개의 4 자릿수 패리티 시프트 레지스터(203/203')는 순차적으로 연결되므로, 패리티 시프트 레지스터(203/203')의 출력이 (후속하는 패리티 시프트 레지스터가 존재하는 경우) 후속하는 패리티 시프트 레지스터(203/203')의 입력에 연결되게 된다. 각 4 자릿수 패리티 시프트 레지스터(203/203')는 패리티 생성 모듈(207)에 대응하고, 각 패리티 생성 모듈(207)은 그 대응하는 패리티 시프트 레지스터(203/203')의 입력에 연결된다. 가산기 모듈(A)이 다수의 소스(예를 들면, 패리티 생성 모듈 및 선행하는 패리티 시프트 레지스터)에 연결되는 각 패리티 시프트 레지스터(203/203')의 입력과 연관될 수도 있다. 가산기 모듈(A)은 저장을 위해 패리티 시프트 레지스터(203/203')의 입력에서 수신되는 데이터의 결합을 용이하게 한다. 4개의 4 자릿수 패리티 시프트 레지스터(203/203') 중 마지막 패리티 시프트 레지스터(203')의 출력은 인코더 모듈의 입력(D_{in})에 연결되고, 피드백 회로(205)의 각 패리티 생성 모듈(207)에 또한 연결된다. 마지막 패리티 시프트 레지스터(203')의 출력은 멀티플렉서를 통해 인코더 모듈(200)의 출력(D_{out})에 추가로 연결된다. 추가의 가산기 모듈(A)이 마지막 패리티 시프트 레지스터(203')의 출력과 연관될 수도 있다. 추가의 가산기 모듈(A)은 피드백 회로(205)의 패리티 생성 모듈(207)에 송신을 위해 마지막 패리티 시프트 레지스터(203')의 출력에서 수신되는 데이터의 결합을 용이하게 한다. 인코더 모듈(200)의 입력(D_{in})은 멀티플렉서를 통해 인코더 모듈(200)의 출력

(D_{out})에 또한 연결된다.

[0043] 피드백 회로(205)는 스위치(S)를 더 포함한다. 스위치(S)가 폐쇄될 때, 피드백 회로(205)는 활성이 되고, 패리티 생성 모듈(207)은 패리티 자릿수를 활성적으로 생성하고 있다. 스위치(S)가 개방될 때, 피드백 회로(205)는 비활성화되고 패리티 생성 모듈(207)은 더 이상 패리티 자릿수를 생성하지 않는다.

[0044] 도 1의 직렬 인코더 모듈(100)은 단 자릿수를 각각 저장한 패리티 시프트 레지스터(103/103')를 포함한 반면에, 도 2의 인코더 모듈(200)은 한 번에 4 자릿수를 저장하는 4 자릿수 패리티 시프트 레지스터(203/203')를 포함한다. 유사하게, 도 1의 피드백 회로(105)의 패리티 생성 모듈(107)은 한 번에 단일 메시지 자릿수를 처리한 반면에, 도 2의 피드백 회로의 패리티 생성 모듈은 한 번에 4 메시지 자릿수를 처리한다.

[0045] 비선형 블록 코드의 병렬 인코딩을 지원하기 위해서, 패리티 자릿수를 생성하기 위한 반복 식 (5)는 동시에 L 자릿수의 처리를 지원하는 것으로 확장될 수도 있다. 변형된 반복 식은 아래에 제공된다:

$$p(x) = \left(x^P \sum_{i=1}^{K} m_{i-1} x^{K-i}\right)_{g(x)} \quad (6)$$

$$= \left(x^P \sum_{i=0}^{\frac{K}{L}-1} \sum_{r=1}^{L} m_{i*L+r-1} x^{K-i*L-r}\right)_{g(x)}$$

$$= \left(\sum_{i=0}^{\frac{K}{L}-1} x^{K-(i+1)*L} \left[x^P \sum_{r=1}^{L} m_{i*L+r-1} x^{L-r}\right]_{g(x)}\right)_{g(x)}$$

$$= \left(x^{K-L} \left[x^P \sum_{r=1}^{L} m_{r-1} x^{L-r}\right]_{g(x)}\right)_{g(x)} +$$

$$\left(x^{K-2L} \left[x^P \sum_{r=1}^{L} m_{L+r-1} x^{L-r}\right]_{g(x)}\right)_{g(x)} +$$

[0046]

$$\left(x^{K-3L} \left[x^P \sum_{r=1}^{L} m_{2L+r-1} x^{L-r}\right]_{g(x)}\right)_{g(x)} +$$

$$\left(x^{K-4L} \left[x^P \sum_{r=1}^{L} m_{3L+r-1} x^{L-r}\right]_{g(x)}\right)_{g(x)} + \cdots$$

$$= \left(x^{K-2L} \left[x^L \left[x^P \sum_{r=1}^{L} m_{r-1} x^{L-r}\right]_{g(x)} +\right.\right.$$

$$xPr=1Lm L+r-1xL-rg(x)\,g(x) +$$

$$\left(x^{K-3L} \left[x^P \sum_{r=1}^{L} m_{2L+r-1} x^{L-r}\right]_{g(x)}\right)_{g(x)} +$$

$$\left(x^{K-4L} \left[x^P \sum_{r=1}^{L} m_{3L+r-1} x^{L-r}\right]_{g(x)}\right)_{g(x)} + \cdots$$

$$= \left[x^{K-3L} \left(x^L \left[x^L \left[x^P \sum_{r=1}^{L} m_{r-1} x^{L-r}\right]_{g(x)} + x^P \sum_{r=1}^{L} m_{L+r-1} x^{L-r}\right]_{g(x)} +\right.\right.$$

$$xPr=1Lm2L+r-1xL-rg(x)\mathrm{g(x)} + xK-4LxPr=1Lm3L+r-1xL-rg(x)\mathrm{g(x)} + \ldots \quad (7)$$

[0047]

[0048] 이것은 아래의 의사 코드에 의해 정의되는 P×L 패리티 매트릭스(PMAT)를 사용하여 한 번에 L 패리티 자릿수를 생성하는 피드백 회로의 각 패리티 생성 모듈을 결과적으로 초래한다. PMAT의 모든 L 행들은 하나의 패리티 생성 모듈을 형성한다.

```
%determine the remainder for the cases eye(L)
msg = gf(eye(L),m);
residue_mat = [msg zeros(L,P)];
for row_index = 1:L
   for col_index = 1:L
      residue_mat(row_index,:) = residue_mat(row_index,:) +...
         [gf(zeros(1,col_index-1),m)
residue_mat(row_index,col_index)*g(x) ...
         gf(zeros(1,L-col_index),m) ];
   end
end
PMAT = (residue_mat(:,L+1:end)).';
```

[0049]

[0050] 상기 반복 식 (7)은 그 후 아래의 의사 코드에 도시된 바와 같이 실현될 수 있다:

```
msg padded = [gf(zeros(1,num_zeros_to_pad),m) msg]; %the zero
padding ensures that the message length is a multiple of the block length
L.
for i = 1:(k /L)
    %combine the parity with the next incoming messages
   u(i) = msg_padded((i-1)*L + (1:L));
feedback = (PMAT*( (fliplr(lfsr_state(P-L+1:P))  +u(i)).')).';
   lfsr_state = [gf(zeros(1,L),m) lfsr_state(1:P-L)]+...
      fliplr(feedback);
end
```

[0051]

[0052] 제1의 K/L 클록 사이클 동안, 피드백 회로(205)의 스위치(S)가 폐쇄된다. K 자릿수 메시지의 각 4 자릿수 부분은 인코더 모듈(200)의 입력(D_{in})에서 병렬 입력으로서 수신되어, 마지막 패리티 시프트 레지스터(203')의 출력에 전송되며, 그 후 피드백 회로(205)의 각 패리티 생성 모듈(207)에 전송된다. 피드백 회로(205)의 각 패리티 생성 모듈(207)은 대응하는 4 자릿수 패리티 시프트 레지스터(203/203')에 저장되는 4 패리티 자릿수(예를 들면, L 패리티 자릿수)를 생성하도록 구성된다. 각 패리티 생성 모듈(207)은 상술한 반복 식 (7), PMAT 매트릭스 및 의사 코드에 따라서 K 자릿수 메시지의 각 L 자릿수 부분을 위한 패리티 자릿수를 생성하도록 구성된다. 각 클록 사이클 동안, 각 패리티 생성 모듈(207)에 의해 생성되는 패리티 자릿수는 각각의 대응하는 패리티 시프트 레지스터(203/203')에 현재 저장되어 있는 데이터와 결합되어 갱신된 패리티 자릿수의 세트를 형성한다.

[0053] 또한, 제1의 K/L 클록 사이클의 각 클록 사이클 동안, 각 패리티 시프트 레지스터(203, 203')는 그 저장된 데이터를 (후속하는 패리티 시프트 레지스터가 존재하는 경우에) 후속하는 패리티 시프트 레지스터(203/203')에 송신한다. 마지막 패리티 시프트 레지스터(203/203')는 그 저장된 데이터를 피드백 회로(205)의 각 패리티 생성 모듈(207)에 송신한다. 패리티 시프트 레지스터(203/203')에 저장된 패리티 자릿수는, K 자릿수 메시지가 수신되고 새로운 패리티 자릿수가 대응하는 패리티 시프트 레지스터(203/203')에 저장하기 위해 패리티 생성 모듈(207)에 의해 생성되므로, 클록 사이클마다 갱신된다. 상술한 바와 같이, 각 클록 사이클 동안, 새롭게 생성된 패리티 자릿수가 대응하는 패리티 시프트 레지스터(203/203')에 현재 저장되어 있는 데이터와 결합되어 갱신된 패리티 자릿수의 세트를 형성한다.

[0054] 동시에, 제1의 K/L 클록 사이클 동안, 인코더 모듈(200)의 입력(D_{in})에서 수신되는 K 자릿수 메시지의 각 4 자릿수 부분은 멀티플렉서를 통해 인코더 모듈(200)의 출력(D_{out})에 전송되므로, 인코더 모듈(200)에 의해 출력되는 제1의 K 자릿수가 K 자릿수 메시지의 4 자릿수 병렬 출력이 된다.

[0055] (K/L)+1 내지 N/L 클록 사이클 동안, 피드백 회로(205)의 스위치(S)는 개방되고 패리티 시프트 레지스터(203/203')에 저장되는 최종 패리티 자릿수 값이 인코더 모듈(200)의 출력(D_{out})에 송신된다. 패리티 시프트 레지스터(203/203')에 저장되는 최종 패리티 자릿수 값은 각 패리티 시프트 레지스터(203/203')에 연결되는 제어 신호(CE)에 응답하여 출력된다. 예를 들면, (K/L)+1 클록 사이클 동안, 마지막 패리티 시프트 레지스터(203')에 저장된 4 패리티 자릿수가 멀티플렉서를 통해 인코더 모듈(200)의 출력(D_{out})에 송신될 수도 있고, 각각의 나머지 패리티 시프트 레지스터(203)에 저장된 4 패리티 자릿수는 각각의 가산기 모듈을 통해 후속하는 패리티 시프트 레지스터(203)에 송신될 수도 있다. 이것은 모든 패리티 자릿수가 인코더 모듈(200)에 의해 출력되었을 때, 제(N/L) 클록 사이클에 거쳐 지속한다. 이러한 방식으로, 인코더 모듈(200)은 N 자릿수 코딩된 메시지를 출력하며, 여기에서 제1의 K 자릿수는 K 메시지 자릿수이고 마지막 P 자릿수는 패리티 자릿수이다.

[0056] PMAT 매트릭스를 사용한 패리티 자릿수의 생성은 XOR의 합으로서 효율적으로 계산될 수 있다. 단일 메시지 자릿수가 패리티 생성 모듈(예를 들면, 직렬 인코딩)에 의해 처리되고 패리티 자릿수를 생성하는 데 GF(q^m) 갈루아 필드 상에 정의된 생성 다항식 g(x)가 사용될 때, 그러한 매트릭스 승산은 최대 m-1 XOR을 필요로 할 수도 있다. L 메시지 자릿수가 패리티 생성 모듈에 의해 동시에 처리되고 패리티 자릿수를 생성하는 데 GF(q^m) 갈루아 필드 상에 정의된 생성 다항식 g(x)가 사용될 때, 그러한 매트릭스 승산은 mL-1 XOR을 필요로 할 수도 있다.

[0057] 그러나, 도 2의 인코더 모듈(200)에 있어서, 각 패리티 생성 모듈(207)이 마지막 패리티 시프트 레지스터(203')의 출력에 연결되는 가산기 모듈을 통해 인코더 모듈(200)의 입력(D_{in}) 및 마지막 패리티 시프트 레지스터(203')의 출력에 연결되므로, 피드백 회로(205)의 각 패리티 생성 모듈(207)은 수신 중인 K 자릿수 메시지의 4 자릿수 부분에서 매트릭스 승산을 실행할 뿐만 아니라, 마지막 패리티 시프트 레지스터(203')의 출력에서 매트릭스 승산을 추가로 실행하고 있다. 그러므로, 각 패리티 생성 모듈(207)을 실현하기 위해 mL-1 XOR을 필요로 하기보다는, 각 패리티 생성 모듈(207)을 실현하는 데 2mL-1 XOR이 필요하다. 피드백 회로가 P 및 m에 의해 빠르게 성장하는 P*m의 그러한 경로를 수반하는 것을 주의하는 것이 또한 중요하다. 결과적으로, 각 패리티 생성 모듈에 의한 패리티 자릿수의 생성은 K 자릿수 메시지가 N 자릿수 코딩된 메시지로 인코딩될 수도 있는 속도를 제한하는 임계적인 경로를 형성할 수도 있다. 더욱이, 도 2에서의 각 패리티 생성 모듈(207)을 실현하는 데 필요한 추가의 계산 로직이 또한 하드웨어 자원 요건의 증가를 결과적으로 초래할 수도 있다.

[0058] 도 2의 인코더 모듈(200)은 패리티 생성 모듈을 실현하는 데 필요한 계산 로직을 감소시키도록 변형될 수도 있다. 도 3은 일부 실시예에 따르는 비이진 선형 블록 코드의 병렬 인코딩을 실현하는 인코더 모듈(300)을 도시한다. 도 3의 인코더 모듈(300)은 각 패리티 생성 모듈을 실현하는 데 필요한 계산 로직을 감소시키므로, 도 2의 인코더 모듈(200)에 필요한 2mL-1 XOR과는 대조적으로 패리티 생성 모듈을 실현하는 데 단지 mL-1 XOR만이 필요하게 된다.

[0059] 도 3의 인코더 모듈(300)은 도 2의 인코더 모듈과 매우 유사하게, 클록 사이클마다 K 자릿수 메시지의 다수의 자릿수(L)가 처리될 수 있게 한다. 인코더 모듈(300)은 K 자릿수 메시지를 위한 임의의 P개의 패리티 자릿수를 생성하여 N 자릿수 코딩된 메시지를 형성하도록 구성될 수도 있고, 여기에서 K 자릿수 메시지는 L 자릿수 병렬 입력으로서 수신되고 N 자릿수 코딩된 메시지는 L 자릿수 병렬 출력으로서 출력된다. 그러나, 예를 들 목적으로, 도 3의 인코더 모듈(300)은 K 자릿수 메시지를 위한 16 패리티 자릿수(예를 들면, P=16)를 생성하여 N 자릿수 코딩된 메시지를 형성하도록 구성되며, 여기에서 K 자릿수 메시지는 4 자릿수 병렬 입력(예를 들면, L=4)으로서 수신되고, N 자릿수 코딩된 메시지는 4 자릿수 병렬 출력으로서 출력된다. K가 L의 정수배가 아닌 경우, L의 배수인 전체 메시지 길이를 달성하기 위해 메시지의 시작부에 제로(0)들이 삽입된다.

[0060] 인코더 모듈(300)은 입력(D_{in}), 출력(D_{out}), 4 자릿수 입력 시프트 레지스터(301), 4개의 4 자릿수 패리티 시프트 레지스터(303/303') 및 4개의 패리티 생성 모듈(307)을 포함하는 피드백 회로(305)를 구비한다. 도 3의 인코더 모듈(300)은 단지 4개의 패리티 생성 모듈(307) 및 4개의 4 자릿수 패리티 시프트 레지스터(303/303')를 구비하고 있지만, 인코더 모듈(300)은 임의의 P/L개의 L 자릿수 패리티 시프트 레지스터 및 L 자릿수 병렬 입력이 수신되고 생성될 P개의 패리티 자릿수에 대응하는 임의의 P/L개의 패리티 생성 모듈을 구비할 수도 있다.

[0061] 인코더 모듈(300)의 입력(D_{in})은 4 자릿수 병렬 입력으로서 K 자릿수 메시지를 수신하도록 구성되고, 인코더 모듈(300)의 출력(D_{out})은 16 패리티 자릿수를 갖는 N 자릿수 코딩된 메시지를 4 자릿수 병렬 출력으로서 출력하도록 구성된다. 피드백 회로(305)의 패리티 생성 모듈(307)은 이후에 상세히 논의되는 바와 같이, 4 자릿수 패리

티 시프트 레지스터(303/303')에 저장되는 4 패리티 자릿수를 한 번에 생성하도록 각각 구성된다.

[0062] 입력 시프트 레지스터(301)의 입력은 인코더 모듈(300)의 입력(D_{in})에 연결되고, 입력 시프트 레지스터(301)의 출력은 멀티플렉서를 통해 인코더 모듈(300)의 출력(D_{out})에 연결된다.

[0063] 4개의 4 자릿수 패리티 시프트 레지스터(303/303')는 순차적으로 연결되므로, 패리티 시프트 레지스터(303/303')의 출력이 (후속하는 패리티 시프트 레지스터가 존재하는 경우) 후속하는 패리티 시프트 레지스터(303/303')의 입력에 연결되게 된다. 각 4 자릿수 패리티 시프트 레지스터(303/303')는 패리티 생성 모듈(307)에 대응하고, 각 패리티 생성 모듈(307)은 그 대응하는 패리티 시프트 레지스터(303/303')의 입력에 연결된다. 4개의 4 자릿수 패리티 시프트 레지스터(303/303') 중 마지막 패리티 시프트 레지스터(303')의 입력은 인코더 모듈의 입력(D_{in})에 연결되고, 마지막 패리티 시프트 레지스터(303')의 출력은 피드백 회로(305)의 각 패리티 생성 모듈(307)에 또한 연결된다. 가산기 모듈(A)이 다수의 소스(예를 들면, 패리티 생성 모듈(307), 선행하는 패리티 시프트 레지스터(303), 인코더 모듈(300)의 입력(D_{in}))에 연결되는 각 패리티 시프트 레지스터(303/303')의 입력과 연관될 수도 있다. 가산기 모듈(A)은 저장을 위해 패리티 시프트 레지스터(303/303')의 입력에서 수신되는 데이터의 결합을 용이하게 한다. 추가의 가산기 모듈(A)이 마지막 패리티 시프트 레지스터(303')의 출력과 연관될 수도 있다. 추가의 가산기 모듈(A)은 피드백 회로(305)의 패리티 생성 모듈(307)에 송신을 위해 마지막 패리티 시프트 레지스터(303')의 출력에서 수신되는 데이터의 결합을 용이하게 한다. 마지막 패리티 시프트 레지스터(303')의 출력은 멀티플렉서를 통해 인코더 모듈(300)의 출력(D_{out})에 추가로 연결된다.

[0064] 피드백 회로(305)는 스위치(S)를 더 포함한다. 스위치(S)가 폐쇄될 때, 피드백 회로(305)는 활성이 되고, 패리티 생성 모듈(307)은 패리티 자릿수를 활성적으로 생성하고 있다. 스위치(S)가 개방될 때, 피드백 회로(305)는 비활성화되고 패리티 생성 모듈(307)은 더 이상 패리티 자릿수를 생성하지 않는다.

[0065] 제1의 K/L 클록 사이클 동안, 피드백 회로(305)의 스위치(S)는 폐쇄된다. K 자릿수 메시지의 각각의 4 자릿수 부분은 인코더 모듈(300)의 입력(D_{in})에서 병렬 입력으로서 수신되고, 제1 클록 사이클 동안 마지막 패리티 시프트 레지스터(303')의 입력과 입력 시프트 레지스터(301)의 입력의 양자에 전송된다. 제2 클록 사이클 동안, 마지막 패리티 시프트 레지스터(303')로부터 K 자릿수 메시지의 4 자릿수 부분이 피드백 회로(305)의 각 패리티 생성 모듈(307)에 출력되고, 그에 따라 각 패리티 생성 모듈(307)이 그 대응하는 패리티 시프트 레지스터(303/303')에 송신되어 저장되는 4 패리티 자릿수를 생성한다. 각 패리티 생성 모듈(307)은 아래의 의사 코드에 따라서 K 자릿수 메시지의 각 자릿수를 위한 패리티 자릿수를 생성하도록 구성된다.

```
    msg_padded = [gf(zeros(1,num_zeros_to_pad),m)  msg]; %the
zero padding ensures that the message length is a multiple of the block
length L.
    msg_padded = [msg_padded  gf(zeros(1,L),m)]; %A zero is
appended to the message to calculate the parity digits.
    for i = 1:(k /L)
        feedback = (PMAT*((fliplr(lfsr_state(P-L+1:P))).')).';
        %combine the parity with the next incoming messages
        lfsr_state = [gf(zeros(1,L),m) lfsr_state(1:P-L)]+...
            fliplr(feedback);
        u(i) = msg_padded((i-1)*L + (1:L));
        lfsr_state(P-L+1:P)=lfsr_state(P-L+1:P)+fliplr(u(i));
    end
```

[0066]

[0067] 상기 의사 코드에 대한 식은 식 (8)을 산출하도록 아래에 나타내는 바와 같이 상기로부터 식 (4)를 재배열함으로써 산출될 수도 있다.

$$p(x) = 0 + (m_0 x^{k-1} x^p)_{g(x)} + (m_1 x^{k-2} x^p)_{g(x)} + (m_2 x^{k-3} x^p)_{g(x)} + \ldots + (m_{k-1} x^p)_{g(x)} + 0$$

$$= x^{k-1} x(0 + m_0 x^{p-1})_{g(x)} + (m_1 x^{k-2} x^p)_{g(x)} + (m_2 x^{k-3} x^p)_{g(x)} + \ldots + (m_{k-1} x^p)_{g(x)} + 0$$

$$= x^{k-2} x\left[x(0 + m_0 x^{p-1})_{g(x)} + m_1 x^{p-1}\right]_{g(x)} + (m_2 x^{k-3} x^p)_{g(x)} + \ldots + (m_{k-1} x^p)_{g(x)} + 0$$

$$= x^{k-3} x\left\{x\left[x(0 + m_0 x^{p-1})_{g(x)} + m_1 x^{p-1}\right]_{g(x)} + m_2 x^{p-1}\right\} + \ldots + (m_{k-1} x^p)_{g(x)}$$

$$p(x) = \left[x\left\{x[\ldots]_{g(x)} + m_{k-1} x^{p-1}\right\}_{g(x)} + 0\right]_{g(x)} \qquad (8)$$

[0068]

[0069] (4)에 도시된 식을 직렬 인코더 모듈에 적용하는 동안, 동일한 조작이 인코더 모듈의 병렬 실행에 대해 참(true)을 유지한다. 이것은, 인코더 모듈의 입력에서 수신되고 있는 K 자릿수 메시지의 L 자릿수 부분이 마지막 패리티 시프트 레지스터의 입력에 입력되어야 함과, K 자릿수 메시지가 그에 부가되는 제로(0)를 가져야 함을 의미한다.

[0070] 또한, 제2 클록 사이클 동안, K 자릿수 메시지의 4 자릿수 부분은 입력 시프트 레지스터(301)로부터 멀티플렉서를 통해 인코더 모듈(300)의 출력(D_{out})에 출력된다. 이것은 K 자릿수 메시지의 모든 4 자릿수 부분이 수신될 때까지 지속한다.

[0071] 또한, 제1의 K/L 클록 사이클의 각 클록 사이클 동안, 각 패리티 시프트 레지스터(303/303')는 그 저장된 데이터를 (후속하는 패리티 시프트 레지스터가 존재하는 경우) 후속하는 패리티 시프트 레지스터(303/303')에 송신한다. 마지막 패리티 시프트 레지스터(303')는 그 저장된 데이터를 피드백 회로의 각 패리티 생성 모듈(307)에 송신한다. 패리티 시프트 레지스터(303/303')에 저장된 패리티 자릿수는, K 자릿수 메시지가 수신되고 새로운 패리티 자릿수가 패리티 시프트 레지스터(303/303')에 저장하기 위해 패리티 생성 모듈(307)에 의해 생성되므로, 클록 사이클마다 갱신된다. 각 클록 사이클 동안, 새롭게 생성된 패리티 자릿수가 대응하는 패리티 시프트 레지스터(303/303')에 현재 저장되어 있는 데이터와 결합되어 갱신된 패리티 자릿수의 세트를 형성한다.

[0072] (K/L)+1 클록 사이클 동안, 제로 입력이 인코더 모듈(300)의 입력(D_{in})에 의해 수신된다. 제로 입력은 인코더 모듈(300)이 피드백 회로(305)의 스위치(S)를 개방하게 한다. 이 클록 사이클 동안, 입력 시프트 레지스터(301) 내의 K 자릿수 메시지의 마지막 4 자릿수 부분이 인코더 모듈의 출력(D_{out})에 출력되고, 마지막 패리티 시프트 레지스터(303') 내의 K 자릿수 메시지의 마지막 4 자릿수 부분은, 피드백 회로(305)의 스위치(S)가 개방하기 전에 대응하는 패리티 자릿수를 생성하도록 피드백 회로(305)의 각 패리티 생성 모듈(307)로 출력된다.

[0073] ((K/L)+2) 클록 사이클 내지 ((N/L)+1) 클록 사이클 동안, 피드백 회로(305)의 스위치(S)는 개방 상태를 유지하고, 패리티 시프트 레지스터(303/303')에 저장되는 최종 패리티 자릿수 값이 인코더 모듈(300)의 출력(D_{out})에 출력된다. 패리티 자릿수는 각각의 패리티 시프트 레지스터(303/303')에 연결되는 제어 신호(CE)에 응답하여 출력된다. 예를 들면, (K/L)+2 클록 사이클 동안, 마지막 패리티 시프트 레지스터(303')에 저장되는 4 패리티 자릿수는 멀티플렉서를 통해 인코더 모듈(300)의 출력(D_{out})에 송신될 수도 있고, 각각의 나머지 패리티 시프트 레지스터(303)에 저장되는 4 패리티 자릿수는 후속하는 패리티 시프트 레지스터(303/303')에 송신될 수도 있다.

[0074] 마지막 패리티 시프트 레지스터(303')의 출력을 인코더 모듈(300)의 입력(D_{in})에 연결하기보다는 마지막 패리티 시프트 레지스터(303')의 입력을 인코더 모듈(300)의 입력(D_{in})에 연결함으로써, 각 패리티 생성 모듈(307)에 의해 실행되는 매트릭스 승산의 복잡성이 감소된다. 수신 중인 K 자릿수 메시지의 4 자릿수 부분과 주어진 클록 사이클에서 마지막 패리티 시프트 레지스터(303')의 출력의 양자에 대해 매트릭스 승산을 실행해야 하는 것보다는, 매트릭스 승산은 주어진 클록 사이클에서 마지막 패리티 시프트 레지스터(303')의 출력에 대해서만 실행된다. 그러므로, 각 패리티 생성 모듈(307)을 실현하는 데 2mL-1 XOR을 필요로 하기보다는, 각 패리티 생성 모듈(307)을 실현하는 데 2mL-1 XOR만이 필요하다. 이것은, K 자릿수 메시지가 N 자릿수 코딩된 메시지로 인코딩될 수도 있고 하드웨어 자원 요건의 감소를 또한 결과적으로 초래할 수도 있는 속도를 감소시킨다.

[0075] 도 3의 인코더 모듈(300)은 팬 아웃을 향상시키도록 더욱 변형될 수도 있다. 팬 아웃은 로직 게이트 출력이 접속되는 게이트 입력의 수를 지칭한다. 팬 아웃은 로직 게이트 출력이 접속되는 게이트 입력의 수가 감소되는 경우 감소된다. 도 4는 일부 실시예에 따라서 감소된 팬 아웃을 갖는 비선형 블록 코드의 병렬 인코딩을 실현하기 위한 인코더 모듈(400)을 도시한다.

[0076] 도 4의 인코더 모듈(400)은 도 3의 인코더 모듈과 매우 유사하게, 클록 사이클마다 K 자릿수 메시지의 다수의 자릿수(L)가 처리될 수 있게 한다. 인코더 모듈(400)은 K 자릿수 메시지를 위한 임의의 P개의 패리티 자릿수를 생성하여 N 자릿수 코딩된 메시지를 형성하도록 구성될 수도 있고, 여기에서 K 자릿수 메시지는 L 자릿수 병렬 입력으로서 수신되고 N 자릿수 코딩된 메시지는 L 자릿수 병렬 출력으로서 출력된다. 그러나, 예를 들 목적으로, 도 4의 인코더 모듈(400)은 K 자릿수 메시지를 위한 16 패리티 자릿수(예를 들면, P=16)를 생성하여 N 자릿수 코딩된 메시지를 형성하도록 구성되며, 여기에서 K 자릿수 메시지는 4 자릿수 병렬 입력(예를 들면, L=4)으로서 수신되고, N 자릿수 코딩된 메시지는 4 자릿수 병렬 출력으로서 출력된다. K가 L의 정수배가 아닌 경우, L의 배수인 전체 메시지 길이를 달성하기 위해 메시지의 시작부에 제로(0)들이 삽입된다.

[0077] 인코더 모듈(400)은 입력(D_{in}), 출력(D_{out}), 4단 딜레이(4 stage delay)(409), 4개의 4 자릿수 패리티 시프트 레지스터(403/403'/403") 및 4개의 패리티 생성 모듈(407)을 포함하는 피드백 회로(405)를 구비한다. 도 4의 인코더 모듈(400)은 단지 4개의 패리티 생성 모듈(407) 및 4개의 4 자릿수 패리티 시프트 레지스터(403/403'/403")를 구비하고 있지만, 인코더 모듈(400)은 임의의 P/L개의 L 자릿수 패리티 시프트 레지스터 및 L 자릿수 병렬 입력이 수신되고 생성될 P개의 패리티 자릿수에 대응하는 임의의 P/L개의 패리티 생성 모듈을 구비할 수도 있다. 유사하게, 도 4의 인코더 모듈(400)은 오직 4단 딜레이(409)만 구비하고 있지만, 인코더 모듈(400)은 L 자릿수 병렬 입력이 수신되고 생성될 P개의 패리티 자릿수에 대응하는 임의의 (P/L)단 딜레이를 구비할 수도 있다.

[0078] 인코더 모듈(400)의 입력(D_{in})은 4 자릿수 병렬 입력으로서 K 자릿수 메시지를 수신하도록 구성되고, 인코더 모듈(400)의 출력(D_{out})은 16 패리티 자릿수를 갖는 N 자릿수 코딩된 메시지를 4 자릿수 병렬 출력으로서 출력하도록 구성된다. 피드백 회로(405)의 패리티 생성 모듈(407)은 이후에 상세히 논의되는 바와 같이, 4 자릿수 패리티 시프트 레지스터에 저장되는 패리티 자릿수를 생성하도록 각각 구성된다.

[0079] 4개의 4 자릿수 패리티 시프트 레지스터(403/403'/403")는 순차적으로 연결되므로, 패리티 시프트 레지스터(403/403'/403")의 출력이 (후속하는 패리티 시프트 레지스터가 존재하는) 후속하는 패리티 시프트 레지스터(403/403'/403")의 입력에 연결되게 된다. 각 4 자릿수 패리티 시프트 레지스터(403/403'/403")는 패리티 생성 모듈(407)에 대응하고, 각 패리티 생성 모듈(407)은 그 대응하는 패리티 시프트 레지스터(403/403'/403")의 입력에 연결된다. 제1 패리티 시프트 레지스터(403')의 입력은 인코더 모듈(400)의 입력(D_{in})뿐만 아니라 피드백 회로(405)의 각 패리티 생성 모듈(407)에 또한 연결된다. 가산기 모듈(A)이 다수의 소스(예를 들면, 각 패리티 생성 모듈(407)의 출력 및 인코더 모듈(400)의 입력(D_{in}))에 연결되는 제1 패리티 시프트 레지스터의 입력과 연관될 수도 있다. 가산기 모듈(A)은 저장을 위해 제1 패리티 시프트 레지스터(403')의 입력에서 수신되는 데이터의 결합을 용이하게 한다. 4개의 4 자릿수 패리티 시프트 레지스터(403/403'/403") 중의 마지막 패리티 시프트 레지스터(403")의 출력은 멀티플렉서를 통해 인코더 모듈(400)의 출력(D_{out})에 추가로 연결된다.

[0080] 4단 딜레이(409)는 인코더 모듈(400)의 입력(D_{in})에 연결되고, 멀티플렉서를 통해 인코더 모듈(400)의 출력(D_{out})에 또한 연결된다.

[0081] 피드백 회로(405)는 4개의 스위치(S1, S2, S3, S4)를 더 포함하며, 여기에서 각 스위치(S1, S2, S3, S4)는 패리티 생성 모듈(407)에 대응한다. 스위치(S1, S2, S3, S4)가 폐쇄될 때, 피드백 회로(405)의 그 대응하는 패리티 생성 모듈(407)이 활성이 되어, 패리티 자릿수를 생성하고 있다. 스위치(S1, S2, S3, S4)가 개방될 때, 피드백 회로(405)의 그 대응하는 패리티 생성 모듈(407)은 비활성화되고 더 이상 패리티 자릿수를 생성하지 않는다.

[0082] 제1의 K/L 클록 사이클 동안, 피드백 회로(405)의 각 스위치(S1, S2, S3, S4)는 폐쇄된다. 제1의 (K/L) 클록 사이클의 각 클록 사이클 동안, K 자릿수 메시지의 각각의 4 자릿수 부분은 인코더 모듈(400)의 입력(D_{in})에서 수신되어, 제1 패리티 시프트 레지스터(403')에 전송된다. 동시에, 각 패리티 시프트 레지스터(403/403'/403")는 그 저장된 데이터를 대응하는 패리티 생성 모듈(407)에 및 (후속하는 패리티 시프트 레지스터가 존재하는

경우) 또한 후속하는 패리티 시프트 레지스터(403/403'/403")에 송신한다. 각 패리티 생성 모듈(407)은 제1 패리티 시프트 레지스터(403')로 되돌려 출력되는 패리티 자릿수를 생성한다. 각 패리티 시프트 레지스터(403/403'/403")에 저장되는 데이터는, 피드백 회로(405)의 패리티 생성 모듈(407)에 의해 생성되고 있는 새로운 패리티 자릿수가 제1 패리티 시프트 레지스터(403')로 피드백되기 때문에 클록 사이클마다 갱신된다. 각 패리티 생성 모듈(407)은 (L=4, P=16에 대해 도시된) 아래의 의사 코드에 따라서 K 자릿수 메시지의 각 자릿수를 위한 패리티 자릿수를 생성하도록 구성된다.

```
for i = 1:(k /L)
u(i) = msg_padded((i-1)*L + (1:L));
parity_temp = PMAT (1:4,:)*(shift_reg_state_0)+...
      PMAT (5:8,:)*(shift_reg_state_1)+...
      PMAT (9:12,:)*(shift_reg_state_2)+...
      PMAT (13:16,:)*(shift_reg_state_3);
shift_reg_state_3 = shift_reg_state_2;
   shift_reg_state_2 = shift_reg_state_1;
   shift_reg_state_1 = shift_reg_state_0;
   shift_reg_state_0 = parity_temp+u(i);
end
```

[0083]

[0084] %는 패리티 비트를 계산하기 위해 제로 삽입을 행한다. % 이하 동작의 종료 시에 시프트 레지스터는 패리티 자릿수를 포함할 것이다.

```
clr = [0 0 0 0];%these are same as the switches S1,S2,S3,S4
for i = 1:L
u(i) = gf(zeros(1,L),m);
parity_temp = PMAT (1:4,:)*(shift_reg_state_0)*(1-clr(1))+...
      PMAT (5:8,:)*(shift_reg_state_1)*(1-clr(2))+...
      PMAT (9:12,:)*(shift_reg_state_2)*(1-clr(3))+...
      PMAT (13:16,:)*(shift_reg_state_3)*(1-clr(4));
shift_reg_state_3 = shift_reg_state_2;
   shift_reg_state_2 = shift_reg_state_1;
   shift_reg_state_1 = shift_reg_state_0;
shift_reg_state_0 = parity_temp+u(i);
clr = [1 clr(1:3)];
end
```

[0085]

[0086] 패리티 시프트 레지스터(403/403'/403") 내의 데이터는 패리티 자릿수를 형성한다. 의사 코드는 상기 (8)의 도출과 유사한 방식으로 도출될 수도 있다. 임의의 주어진 사이클에서, 전체 P 패리티 자릿수보다는 단지 L 패리티 자릿수가 생성될 것이다. 그 후, 총 P 패리티 자릿수가 P/L 클록 사이클 동안 생성될 것이다.

[0087] 동시에 제1의 K/L 클록 사이클의 각 클록 사이클 동안, 인코더 모듈(400)의 입력(D_{in})에서 수신된 K 자릿수 메시

지의 4 자릿수 부분이 멀티플렉서를 통해 인코더 모듈(400)의 출력(D_{out})에 송신되기 전에 4단 딜레이(409)를 통과하게 되므로, 인코더 모듈(400)에 의해 출력되는 제1의 K 자릿수가 K 자릿수 메시지의 4 자릿수 병렬 출력이 된다.

[0088] ((K/L)+1) 내지 ((K/L)+(P/L)) 클록 사이클의 각 클록 사이클 동안, 제로 입력이 인코더 모듈(400)의 입력(D_{in})에 의해 수신된다. 각각의 제로 입력으로 인해 인코더 모듈(400)이 피드백 회로(405)의 스위치(S1, S2, S3, S4)를 점진적으로 개방시킨다. 또한, 각 클록 사이클 동안, 각 패리티 시프트 레지스터(403/403')는 그 저장된 데이터를 (대응하는 스위치가 폐쇄되는 경우) 대응하는 패리티 생성 모듈(407)에 송신한다. 대응하는 패리티 생성 모듈(407)은 제1 입력 시프트 레지스터(403')에 전송되는 패리티 자릿수를 생성한다. 이것은 피드백 회로(405)의 모든 스위치(S1, S2, S3, S4)가 개방될 때까지 계속된다.

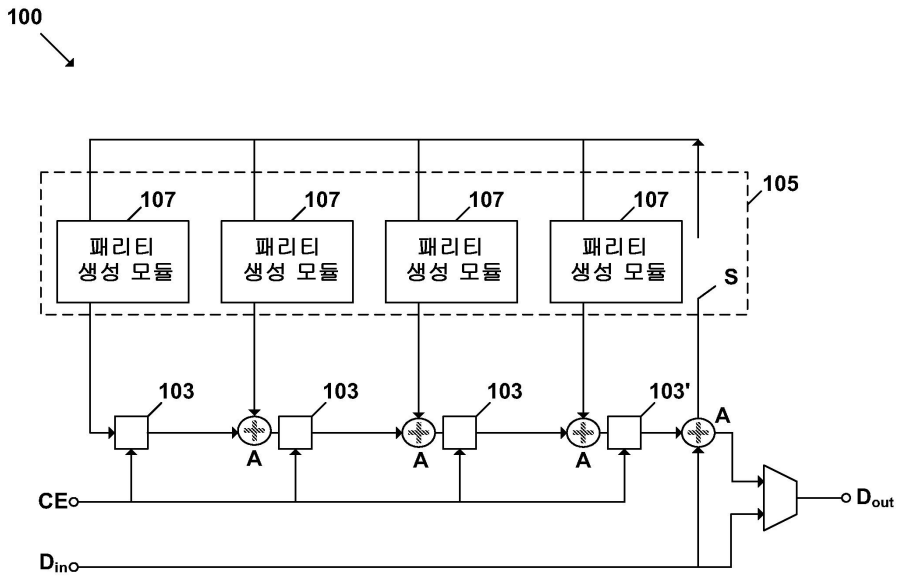
[0089] ((K/L)+(P/L)) 내지 ((N/L)+(P/L)) 클록 사이클의 각 클록 사이클 동안, 모든 스위치(S1, S2, S3, S4)가 개방된 상태를 유지하고, 패리티 시프트 레지스터(403/403'/403")에 저장된 패리티 자릿수가 각각의 패리티 시프트 레지스터(403/403'/403")에 연결되는 제어 신호(CE)에 응답하여 출력된다. 예를 들면, (K/L)+(P/L) 클록 사이클 동안, 마지막 패리티 시프트 레지스터(403")에 저장되는 4 패리티 자릿수는 멀티플렉서를 통해 인코더 모듈(400)의 출력(D_{out})에 송신될 수도 있고, 각각의 나머지 패리티 시프트 레지스터(403/403')에 저장되는 4 패리티 자릿수는 후속하는 패리티 시프트 레지스터(403/403")에 송신될 수도 있다.

[0090] K 자릿수 메시지의 각 L 자릿수 부분을 각 패리티 생성 모듈(407)에 전달하는 것보다는, K 자릿수 메시지의 각 L 자릿수 부분을 제1 패리티 시프트 레지스터(403')에 전달함으로써, 인코더 모듈(400)의 입력(D_{in})에서의 높은 팬 아웃이 회피될 수도 있다. 또한, 패리티 생성 모듈(407)이 (P/L)개의 상이한 시프트 레지스터(403/403'/403")로부터 공급되기 때문에, 임의의 시프트 레지스터(403/403'/403") 상에서의 팬 아웃이 도 2의 인코더 모듈(200) 및 도 3의 인코더 모듈(300)에 비해 (P/L)의 비율만큼 감소된다. 팬 아웃을 감소시키면 라우팅 알고리즘에 대한 과도한 스트레스를 감소시키는 동시에 높은 최대 주파수가 달성될 수 있게 한다.

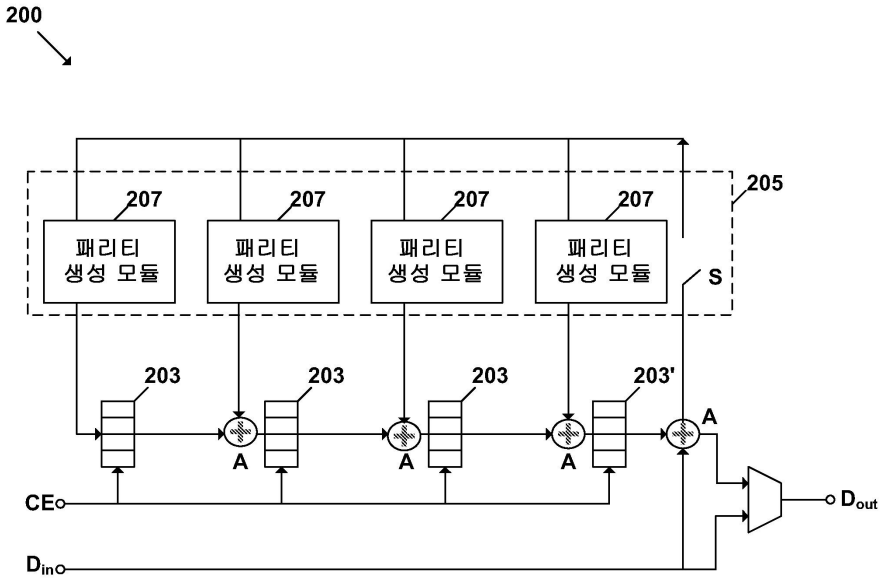
[0091] 특정 실시예가 도시 및 기술되어 있지만, 그 실시예들은 청구하는 발명을 제한하고자 의도된 것이 아님을 이해할 것이고, 당업자에게는 여러 가지 변형 및 수정이 청구하는 발명의 사상 및 범위를 벗어남 없이 이루어질 수도 있음이 명백해질 것이다. 명세서 및 도면은 따라서, 제한적인 의미가 아니라 예시적인 의미로 생각될 것이다. 청구하는 발명은 변경, 수정 및 등가물을 커버하도록 의도된다.

도면

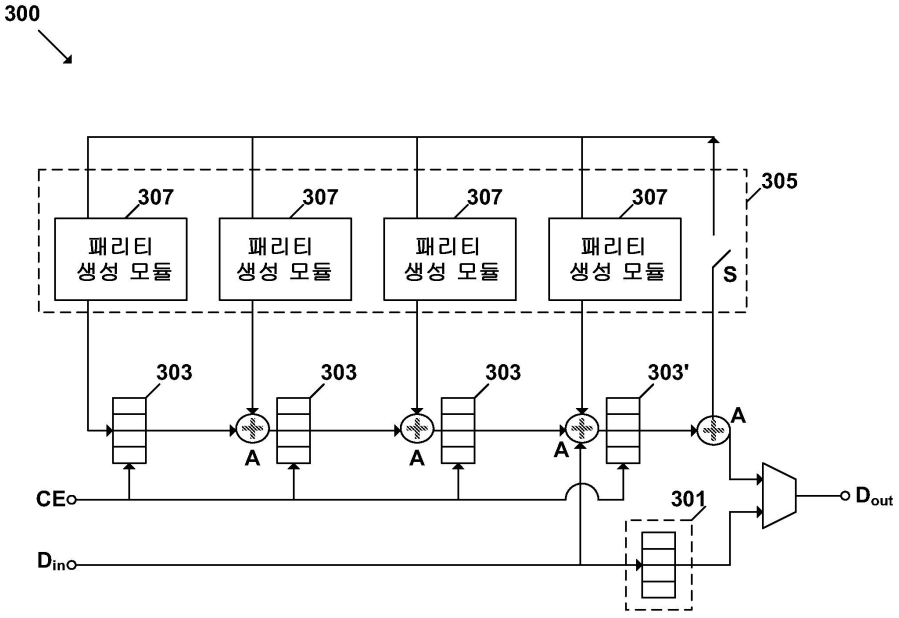
도면1



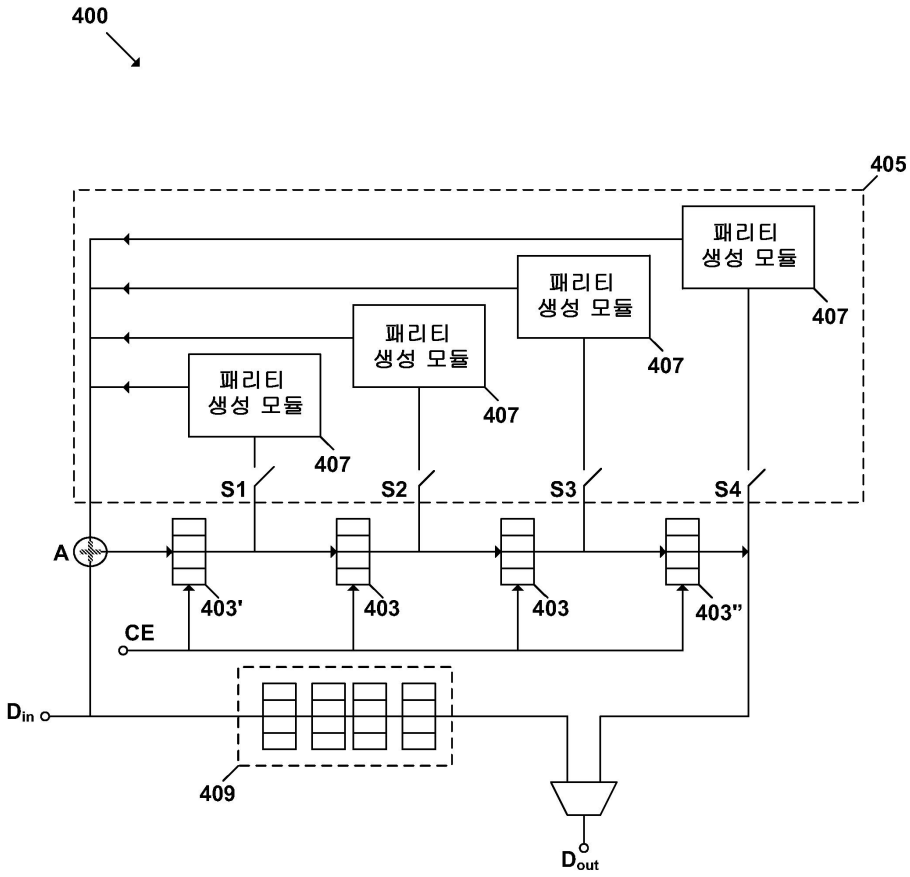
도면2


200
205
207
패리티
생성 모듈
S
203
203'
A
CE
Din
Dout

도면3


300
305
307
패리티
생성 모듈
S
303
303'
A
CE
Din
Dout
301

도면4


400
405
패리티
생성 모듈
407
패리티
생성 모듈
407
패리티
생성 모듈
407
패리티
생성 모듈
407
S1
S2
S3
S4
A
403'
403
403
403"
CE
Din
409
Dout