



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0008263
(43) 공개일자 2009년01월21일

- | | |
|--|--|
| <p>(51) Int. Cl.⁹
<i>H03M 13/09</i> (2006.01)</p> <p>(21) 출원번호 10-2008-7025838</p> <p>(22) 출원일자 2008년10월22일
심사청구일자 없음
번역문제출일자 2008년10월22일</p> <p>(86) 국제출원번호 PCT/GB2007/001371
국제출원일자 2007년04월13일</p> <p>(87) 국제공개번호 WO 2007/122384
국제공개일자 2007년11월01일</p> <p>(30) 우선권주장
0607976.8 2006년04월22일 영국(GB)</p> | <p>(71) 출원인
더 퀸즈 유니버시티 오브 벨파스트
영국 벨파스트 유니버시티 로드 (우편번호 비티7 1엔엔)</p> <p>(72) 발명자
세제르, 사키르
영국, 노던 아일랜드, 비티8 6지에이치, 벨파스트, 핀스버리 크레센트 40
토알, 씨아란
영국, 노던 아일랜드, 비티36 6이에스, 벨파스트, 처치 크레센트 31</p> <p>(74) 대리인
김학수, 문경진</p> |
|--|--|

전체 청구항 수 : 총 33 항

(54) 순환 중복 검사 코드의 구성 가능한 병렬 계산

(57) 요약

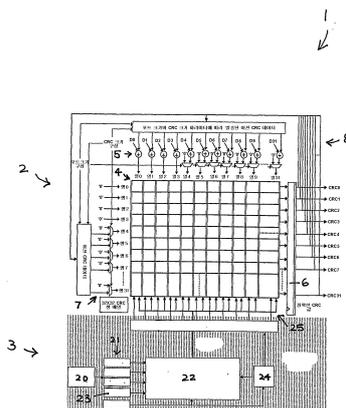
순환 중복 검사(CRC) 에러 검출 방법론을 구현하기 위한 장치(1)로서, 상기 장치는 상기 방법론에 따라 데이터에 관한 CRC 에러 검출 코드를 계산하기 위한 것이고,

CRC 에러 검출 코드를 계산하기 위해 병렬 계산(4)을 사용하는 계산 수단(2)과,

상기 CRC 에러 검출 코드를 계산하기 위해 요구되는 상기 계산 수단의 구성을 결정하기 위해 상기 CRC 에러 검출 방법론을 사용하고, 그에 따라 상기 계산 수단을 구성하는(25) 구성기 수단(3)을 포함하며,

상기 구성기 수단은 각각의 방법론에 따라 CRC 에러 검출 코드의 병렬 계산을 위해 요구되는 상기 계산 수단의 구성을 결정하기 위해 복수의 CRC 에러 검출 방법론 각각을 사용할 수 있고, 상기 계산 수단은 각각의 CRC 에러 검출 코드의 병렬 계산을 위한 구성을 허용하도록 구성 가능한, CRC 에러 검출 방법론을 구현하기 위한 장치.

대표도 - 도1



특허청구의 범위

청구항 1

순환 중복 검사(CRC: Cyclic Redundancy Check) 에러 검출 방법론에 따라 데이터에 관한 CRC 에러 검출 코드를 계산하기 위해 상기 CRC 에러 검출 방법론을 구현하기 위한 장치로서,

CRC 에러 검출 코드를 계산하기 위해 병렬 계산을 사용하는 계산 수단과,

상기 CRC 에러 검출 코드를 계산하기 위해 요구되는 상기 계산 수단의 구성을 결정하기 위해 상기 CRC 에러 검출 방법론을 사용하고, 그에 따라 상기 계산 수단을 구성하는 구성기(configurator) 수단을

포함하며,

상기 구성기 수단은 각각의 방법론에 따라 CRC 에러 검출 코드의 병렬 계산을 위해 요구되는 상기 계산 수단의 구성을 결정하기 위해 복수의 CRC 에러 검출 방법론 각각을 사용할 수 있고, 상기 계산 수단은 각각의 CRC 에러 검출 코드의 병렬 계산을 위한 구성을 허용하도록 구성 가능한, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 2

제 1항에 있어서, 상기 계산 수단은 복수의 구성 가능한 요소를 포함하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 3

제 2항에 있어서, 상기 구성 가능한 요소 중 적어도 일부의 구성은, 상기 CRC 에러 검출 방법론 중 하나를 사용하는 상기 구성기 수단에 의해 결정되는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 4

제 2항 또는 제 3항에 있어서, 구성 가능한 요소들 중 적어도 일부에 관해서는, 각각의 구성 가능한 요소가 데이터 부분을 수신하고, CRC 에러 검출 코드의 계산에서 상기 데이터 부분을 사용하거나 상기 CRC 에러 검출 코드의 계산시 상기 데이터 부분을 사용하지 않도록 구성 가능한, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 5

제 2항 내지 제 4항 중 어느 한 항에 있어서, 구성 가능한 요소들 중 적어도 일부에 관해, 각각의 구성 가능한 요소는 데이터의 부분을 수신하는 데이터-경로 회로를 포함하고, 상기 데이터-경로 회로가 CRC 에러 검출 코드의 계산시 상기 데이터의 부분을 사용하거나 상기 CRC 에러 검출 코드의 계산시 상기 데이터의 부분을 사용하지 않도록 구성 가능한, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 6

제 5항에 있어서, 상기 데이터 경로-회로의 적어도 일부는 XOR 게이트와 구성 가능한 디바이스를 포함하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 7

제 6항에 있어서, 상기 데이터-경로 회로 중 적어도 일부의 각각에 관해서는, 상기 XOR 게이트가 데이터의 부분을 수신하고 XOR 함수에서 상기 데이터를 사용하며, 상기 데이터-경로 회로는 상기 데이터의 부분이 상기 XOR 게이트의 XOR 함수로부터 생기는 신호를 출력하기 위해 상기 구성 가능한 디바이스를 구성함으로써, CRC 에러 검출 코드의 계산시 사용되도록 구성 가능한, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 8

제 6항에 있어서, 상기 데이터-경로 회로들 중 적어도 일부의 각각에 관해서는, 상기 XOR 게이트가 상기 데이터의 부분을 수신하고 XOR 함수에서 상기 데이터를 사용하며, 상기 데이터-경로 회로는 상기 데이터의 부분이 상기 XOR 게이트의 XOR 함수로부터 생기는 신호를 출력하지 않도록 하기 위해 상기 구성 가능한 디바이스를 구성함으로써, CRC 에러 검출 코드의 계산시 사용되지 않도록 구성 가능한, CRC 에러 검출 방법론을 구현하기 위한

장치.

청구항 9

제 8항에 있어서, 상기 구성 가능한 디바이스는 상기 디바이스에 의해 수신된 신호를 출력하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 10

제 2항 내지 제 9항 중 어느 한 항에 있어서, 상기 구성 가능한 요소들 중 적어도 일부는 각각 제어-경로 회로를 포함하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 11

제 10항에 있어서, 상기 제어-경로 회로들 중 적어도 일부는 구성 가능한 디바이스를 포함하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 12

제 10항 또는 제 11항에 있어서, 상기 제어-경로 회로들 중 적어도 일부는 구성 레지스터를 포함하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 13

제 5항 내지 제 9항 중 어느 한 항에 따른 제 10항 내지 제 12항 중 어느 한 항에 있어서, 상기 구성 가능한 요소들 중 적어도 일부 각각에 관해서는, 한 요소의 제어-경로 회로는 그 요소의 데이터-경로 회로의 구성을 제어하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 14

제 13항에 있어서, 상기 구성 가능한 요소들 중 적어도 일부의 각각에 관해서는, 한 요소의 제어-경로 회로가 그 요소의 데이터-경로 회로의 구성 가능한 디바이스의 구성을 제어하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 15

제 14항에 있어서, 상기 구성 가능한 요소들 중 적어도 일부의 각각에 관해서는, 한 요소의 제어-경로 회로의 구성 가능한 디바이스가 상기 요소의 데이터-경로 회로의 구성 가능한 디바이스의 구성을 제어하기 위해, 상기 요소의 제어-경로 회로의 구성 레지스터의 동작을 제어하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 16

제 2항 내지 제 15항 중 어느 한 항에 있어서, 상기 구성 가능한 요소는 요소들의 상호 연결된 배열을 포함하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 17

제 16항에 있어서, 요소들의 상기 배열의 각 행에 관해서는, 한 행에서의 요소들이 CRC 에러 검출 코드의 부분을 집합적으로(collectively) 계산하기 위해 상호연결되는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 18

제 16항 또는 제 17항에 있어서, 요소들의 배열의 각각의 행에 관해서는, CRC 에러 검출 코드의 계산시 데이터를 사용하도록 구성된 한 행에서의 각 요소에 의해 수신된 데이터는 CRC 에러 검출 코드의 부분을 계산하기 위해 결합되는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 19

제 16항 내지 제 18항 중 어느 한 항에 있어서, 상기 배열의 각각의 행에 관해서는, 한 행에 있는 요소들이 구성기 수단으로부터 구성 데이터를 수신하기 위해 상호연결되는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 20

제 16항 내지 제 19항 중 어느 한 항에 있어서, 구성 가능한 요소들의 배열에서의 각각의 열은 CRC 에러 검출 코드의 병렬 계산을 위해 데이터의 부분을 수신하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 21

제 16항 내지 제 20항 중 어느 한 항에 있어서, 요소들의 배열에서 각각의 열에 관해서는, 한 열에서의 요소들이 구성기 수단으로부터 구성 제어 신호들을 수신하기 위해 상호연결되는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 22

제 1항 내지 제 21항 중 어느 한 항에 있어서, 상기 계산 수단은 2개 이상의 입력 디바이스를 사용하여 데이터의 하나 이상의 블록에서 데이터를 수신하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 23

제 22항에 있어서, 계산 수단에 의해 수신된 블록들의 크기는 상기 계산 수단의 입력 디바이스의 개수 이하인, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 24

제 23항에 있어서, 데이터를 수신하지 않는 입력 디바이스 또는 각각의 입력 디바이스는 낮은 신호를 출력하기 위해 프로그래밍 가능한, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 25

제 1항 내지 제 24항 중 어느 한 항에 있어서, 상기 계산 수단은 하나 이상의 피드백 디바이스를 포함하고, 이들 중 적어도 일부는 계산된 CRC 에러 검출 코드의 부분을 상기 계산 수단으로 피드백하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 26

제 25항에 있어서, 상기 계산 수단은 데이터의 제 1 블록을 사용하여 제 1 CRC 에러 검출 코드를 계산하고, 제 1 CRC 에러 검출 코드를 상기 계산 수단으로 피드백하며, 상기 제 1 CRC 에러 검출 코드를 데이터의 제 2 블록과 결합하고, 데이터의 결합된 제 2 블록과 제 1 CRC 에러 검출 코드를 사용하여 제 2 CRC 에러 검출 코드를 계산하며, 최종 CRC 에러 검출 코드를 계산하여, 최종 CRC 에러 검출 코드를 출력하기 위해 모든 데이터가 사용될 때까지 이러한 프로세스를 계속하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 27

제 25항 또는 제 26항에 있어서, 상기 계산 수단은 입력 디바이스 중 적합한 디바이스를 프로그래밍하고, 상기 계산 수단의 피드백 디바이스 중 적합한 디바이스를 프로그래밍함으로써, 개수가 변할 수 있는 블록들을 포함하는 데이터를 수신하도록 프로그래밍 가능한, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 28

제 1항 내지 제 27항 중 어느 한 항에 있어서, 각각 이 미리 결정된 크기의 CRC 발생기 다항식을 사용하는, 복수의 CRC 에러 검출 방법론을 구현하도록 구성 가능한, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 29

제 28항에 있어서, 구성기 수단은 CRC D 매트릭스를 계산하기 위해 CRC 에러 검출 방법론의 CRC 발생기 다항식을 사용하고, 상기 CRC D 매트릭스는 CRC 에러 검출 코드를 계산하기 위해 요구되는 계산 수단의 구성을 결정하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 30

제 29항에 있어서, 상기 D 매트릭스는 0들과 1들의 배열을 포함하고, 1들의 위치는 CRC 에러 검출 코드의 계산

시 데이터를 사용하는 계산 수단의 구성 가능한 요소들의 요구되는 위치를 표시하며, 0들의 위치는 CRC 에러 검출 코드의 계산시 데이터를 사용하지 않는 계산 수단의 구성 가능한 요소들의 요구되는 위치를 표시하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 31

제 1항 내지 제 30항 중 어느 한 항에 있어서, 하드웨어 디바이스를 포함하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 32

제 1항 내지 제 31항 중 어느 한 항에 있어서, 프로토콜 프로세서의 부분을 포함하는, CRC 에러 검출 방법론을 구현하기 위한 장치.

청구항 33

CRC 에러 검출 방법론에 따라 데이터에 관한 CRC 에러 검출 코드를 계산하는 방법으로서, 상기 CRC 에러 검출 방법론을 구현하고 상기 CRC 에러 검출 코드를 계산하기 위해, 제 1항 내지 제 32항 중 어느 한 항에 따른 장치를 사용하는 단계를 포함하는, CRC 에러 검출 방법론에 따라 데이터에 관한 CRC 에러 검출 코드를 계산하는 방법.

명세서

기술분야

<1> 본 발명은, 예컨대 통신망에서 데이터에 관한 에러 검출 코드를 계산하기 위한 장치와 방법에 관한 것이다.

배경기술

<2> 통신망에서는 데이터 무결성(data integrity)이 가장 중요하다. 데이터가 정확하게 송신되고 수신되는 것이 필수 불가결하다. 하지만, 예컨대 운반 매체에서의 잡음이 송신된 데이터에 영향을 미칠 수 있어, 그것에 에러를 더할 수 있다. 그러므로 송신된 데이터에서 에러를 검출하기 위해 다수의 방법론이 개발돼 왔다.

<3> 그러한 방법론 중 하나는 순환 중복 검사(CRC: Cyclic Redundancy Check)라고 알려져 있다. 이는 공지되고 널리 사용된 에러 검출 방법이고, 데이터-링크 계층 프로토콜을 사용하여 송신된 데이터 페이로드 내에서 비트 에러를 검출하기 위해 특별히 사용된다. CRC 에러 검출 방법론은 데이터에 관한 에러 검출 코드를 계산하는데 사용되고, 에러 검출 코드는 일반적으로 데이터에 첨부되고, 코드 및 데이터는 함께 송신된다. 데이터와 에러 검출 코드를 수신하면, 에러 검출 코드는 송신하는 동안 데이터에서 에러가 발생하였는지를 결정하기 위해 사용된다.

<4> 다수의 에러 검출 코드 계산 장치가 제안되었고, 이들은 더 크거나 더 작은 효율을 가진다. 하지만, 통신망의 많은 요소들을 가지고, 그러한 망들의 사용이 계속해서 성장함에 따라, 에러 검출 코드 계산의 개선이 필요해진다.

발명의 상세한 설명

<5> 본 발명의 제 1 양상에 따르면, CRC 에러 검출 방법론을 구현하기 위한 장치가 제공되고, 이 장치는 상기 방법론에 따라 데이터에 관한 CRC 에러 검출 코드를 계산하기 위한 것이며,

<6> CRC 에러 검출 코드를 계산하기 위해 병렬 계산을 사용하는 계산 수단과,

<7> 상기 CRC 에러 검출 코드를 계산하기 위해 요구되는 상기 계산 수단의 구성을 결정하기 위해 상기 CRC 에러 검출 방법론을 사용하고, 그에 따라 상기 계산 수단을 구성하는 구성기(configurator) 수단을 포함하고,

<8> 상기 구성기 수단은 각각의 방법론에 따라 CRC 에러 검출 코드의 병렬 계산을 위해 요구되는 상기 계산 수단의 구성을 결정하기 위해 복수의 CRC 에러 검출 방법론 각각을 사용할 수 있고, 상기 계산 수단은 각각의 CRC 에러 검출 코드의 병렬 계산을 위한 구성을 허용하도록 구성 가능하다.

<9> 다양한 CRC 에러 검출 방법론을 구현하도록 장치가 구성될 수 있으므로, 다양한 응용예에 사용될 수 있다는 장점을 가지고, 이 경우 상이한 CRC 에러 검출 방법론이 요구된다. 예컨대, 통신망은 다수의 송신 프로토콜을 사

용하고, 이들은 일반적으로 에러 검출을 수반한다. 프로토콜이 크게 변화하고, 새로운 프로토콜이 계속해서 개발되며, 레거시(legacy) 프로토콜을 유지할 필요성으로 인해, 통신망의 프로토콜 적용 범위를 최대화할 필요가 존재하게 된다. 프로토콜 적용 범위를 최대화하는 한 가지 방식은 프로토콜의 에러 검출시 구성 능력을 통합하는 것이다.

- <10> 장치는 하드웨어 디바이스를 포함할 수 있다. 계산 수단과 구성기 수단은 하드웨어 디바이스 상에서 구현될 수 있다. 계산 수단은 복수의 구성 가능한 요소를 포함할 수 있다. 각 요소의 구성은 CRC 에러 검출 방법론 중 하나를 사용하여 구성기 수단에 의해 결정될 수 있다. 각 요소는 데이터의 부분을 수신할 수 있고, CRC 에러 검출 코드의 계산시 데이터의 부분을 사용하거나 CRC 에러 검출 코드의 계산시 데이터의 부분을 사용하지 않도록 구성 가능할 수 있다.
- <11> 구성 가능한 요소의 적어도 일부의 각각에 관해, 각각의 구성 가능한 요소는 데이터의 부분을 수신할 수 있는 데이터-경로 회로를 포함할 수 있고, 회로가 CRC 에러 검출 코드의 계산시 데이터의 부분을 사용하거나 CRC 에러 검출 코드의 계산시 데이터의 부분을 사용하지 않도록 구성 가능할 수 있다. 데이터-경로 회로들 중 적어도 일부는 XOR 게이트를 포함할 수 있다. 데이터-경로 회로들 중 적어도 일부는 구성 가능한 멀티플렉서와 같은 구성 가능한 디바이스를 포함할 수 있다. 데이터-경로 회로들 중 적어도 일부의 각각에 관해, XOR 게이트는 데이터의 부분을 수신할 수 있고, XOR 함수에서 데이터를 사용할 수 있으며, 데이터-경로 회로는 XOR 게이트의 XOR 함수로부터의 신호 결과를 출력하도록 구성 가능한 디바이스를 구성함으로써, CRC 에러 검출 코드의 계산시 데이터의 부분이 사용되도록 구성 가능할 수 있다. 데이터-경로 회로들 중 적어도 일부의 각각에 관해, XOR 게이트는 데이터의 부분을 수신할 수 있고, XOR 함수에서 데이터를 사용할 수 있으며, 데이터-경로 회로는 XOR 게이트의 XOR 함수로부터의 신호 결과를 출력하지 않도록 구성 가능한 디바이스를 구성함으로써, CRC 에러 검출 코드의 계산시 데이터의 부분이 사용되지 않도록 구성 가능할 수 있다. 구성 가능한 디바이스가 XOR 게이트의 XOR 함수로부터의 신호 결과를 출력하지 않을 때, 디바이스는 그 디바이스에 의해 수신된 신호를 출력할 수 있다. 이 경우, 디바이스를 포함하는 구성 가능한 요소는 입력 출력 연결부로서 기능을 할 수 있다.
- <12> 구성 가능한 요소들 중 적어도 일부는 각각 제어-경로 회로를 포함할 수 있다. 그러한 제어-경로 회로들 중 적어도 일부는 구성 가능한 멀티플렉서와 같은 구성 가능한 디바이스를 포함할 수 있다. 제어-경로 회로들 중 적어도 일부는 구성 레지스터를 포함할 수 있다. 구성 가능한 요소들 중 적어도 일부의 각각에 관해서는, 한 요소의 제어-경로 회로는 그 요소의 데이터-경로 회로의 구성을 제어할 수 있다. 구성 가능한 요소들 중 적어도 일부의 각각에 관해서는, 한 요소의 제어-경로 회로가 그 요소의 데이터-경로 회로의 구성 가능한 디바이스의 구성을 제어할 수 있다. 요소들 중 적어도 일부의 각각에 관해서, 한 요소의 제어-경로 회로의 구성 가능한 디바이스는, 그 요소의 데이터-경로 회로의 구성 가능한 디바이스의 구성을 제어하기 위해, 그 요소의 제어-경로 회로의 구성 레지스터의 동작을 제어할 수 있다. 구성 가능한 요소들 중 적어도 일부의 각각에 관해서, 한 요소의 제어-경로 회로의 구성 가능한 디바이스는 구성 레지스터의 동작을 제어하는 것이 가능하게 되거나 가능하지 않게 될 수 있다.
- <13> 구성 가능한 요소들은 상호연결된 요소들의 배열을 포함할 수 있다. 요소들의 배열의 각 행에 관해, 한 행에서의 요소들은 CRC 에러 검출 코드의 부분을 집합적으로 계산하기 위해 상호연결될 수 있다. 요소들의 배열의 각 행에 관해, CRC 에러 검출 코드의 계산시 데이터를 사용하도록 구성된 한 행에서의 각 요소에 의해 수신된 데이터는 CRC 에러 검출 코드의 부분을 계산하기 위해 결합될 수 있다. 그 배열에서의 각 행에 관해, 한 행에서의 요소들은 구성기 수단으로부터 구성 데이터를 수신하기 위해 상호연결될 수 있다. 구성 가능한 요소들의 배열에서의 각 열은 CRC 에러 검출 코드의 병렬 계산을 위해 데이터의 부분을 수신할 수 있다. 요소들의 배열에서의 각 열에 관해, 한 열에서의 요소들은 데이터의 부분을 각각 수신하기 위해 상호연결될 수 있다. 요소들의 배열에서의 각 열에 관해, 한 열에서의 요소들은 구성기 수단으로부터 구성 제어 신호들을 수신하기 위해 상호연결될 수 있다.
- <14> 계산 수단은 데이터의 하나 이상의 블록에서 데이터를 수신할 수 있다. 계산 수단은 데이터의 블록들을 수신하기 위해 2개 이상의 입력 디바이스를 포함할 수 있다. 계산 수단에 의해 수신된 블록들의 크기는 계산 수단의 입력 디바이스들의 개수와 같을 수 있다. 예컨대, 계산 수단은 2 내지 N의 범위에서 입력 디바이스들의 개수를 제공할 수 있다. 계산 수단에 의해 수신된 블록들의 크기는 계산 수단이 입력 디바이스들의 개수보다 작을 수 있다. 데이터를 수신하지 않는 입력 디바이스나 각각의 입력 디바이스는 로우 신호를 출력하도록 프로그래밍될 수 있다. 프로그래밍 가능한 입력 디바이스들은 프로그래밍 가능한 멀티플렉서들을 포함할 수 있다. 프로그래밍 가능한 입력 디바이스들은 장치의 실행시간시 프로그래밍될 수 있다. 계산 수단은 프로그래밍 가능한 입력 디바이스들의 프로그래밍을 제어하는 하나 이상의 제어 디바이스를 포함할 수 있다. 프로그래밍 가능한 입력 디바이

스들을 제공하는 것은, 가변적인 크기를 가지는 데이터의 블록들이 계산 수단에 의해 수신될 수 있다는 것을 의미한다. 그러므로, 계산 수단은 가변 입력 포트 크기를 가지게 된다. 특히, 장치는 여러 포트 크기를 제공할 수 있고, 따라서 예컨대 4, 8, 16, 24, 32, 64, 128 및 그 이상의 비트들 중 임의의 것과 같이 2 내지 N비트의 범위에 있는 데이터 블록들을 수신할 수 있다.

- <15> 계산 수단은 출력 레지스터와 같은 하나 이상의 출력 저장 디바이스를 포함할 수 있다. 출력 디바이스 또는 각각의 출력 디바이스는 계산된 에러 검출 코드를 수신하고 저장할 수 있다.
- <16> 계산 수단은 하나 이상의 피드백 디바이스를 포함할 수 있다. 피드백 디바이스들 중 적어도 일부는 계산된 CRC 에러 검출 코드의 부분을 계산 수단으로 공급할 수 있다. 예컨대, 피드백 디바이스들 중 적어도 일부는 계산 수단의 입력 디바이스들 중 적어도 일부로 계산된 CRC 에러 검출 코드의 부분을 피드백시킬 수 있다. 피드백 디바이스들 중 적어도 일부는 계산된 CRC 에러 검출 코드의 부분을 계산 수단의 구성 가능한 요소들 중 적어도 일부에 피드백시킬 수 있다. 계산 수단은 계산 수단으로 CRC 에러 검출 코드의 부분을 피드백시키는 것을 제어하기 위해 사용된 하나 이상의 제어 디바이스를 포함할 수 있다.
- <17> 바람직한 일 실시예에서, 계산 수단은 데이터의 제 1 블록을 사용하여 제 1 CRC 에러 검출 코드를 계산하고, 제 1 CRC 에러 검출 코드를 계산 수단으로 피드백시키며, 제 1 CRC 에러 검출 코드를 데이터의 제 2 블록과 결합하고, 데이터의 결합된 제 2 블록과 제 1 CRC 에러 검출 코드를 사용하여 제 2 CRC 에러 검출 코드를 계산하며, 모든 데이터가 최종 CRC 에러 검출 코드를 계산하기 위해 사용될 때까지 이러한 프로세스를 계속하고, 최종 CRC 에러 검출 코드를 출력한다. 이러한 식으로, 모든 데이터가 CRC 에러 검출 코드를 계산하기 위해 사용된다.
- <18> 계산 수단은 다양한 개수의 블록을 포함하는 데이터를 수신하기 위해 프로그래밍 가능할 수 있다. 이는 계산 수단의 피드백 디바이스들 중 적합한 디바이스를 프로그래밍하고 입력 디바이스들 중 적합한 디바이스를 프로그래밍함으로써 달성될 수 있다. 예컨대, 입력 디바이스들에 의해 수신된 데이터의 마지막 블록은 이전에 수신된 블록들보다 작을 수 있다. 계산 수단은 장치의 실행시간시 프로그래밍될 수 있다. 장치는 입력 데이터의 블록들의 바이트 크기나 비트들의 개수에 있어서의 변화를 다룰 수 있도록 프로그래밍될 수 있다.
- <19> 구성기 수단은 계산 수단의 구성을 결정하기 위해 사용된 CRC 에러 검출 방법론의 하나 이상의 파라미터를 수신하기 위해 사용되는 인터페이스를 포함할 수 있다. 구성기 수단은 파라미터들을 저장하기 위해 사용되는 레지스터와 같은 하나 이상의 저장 디바이스를 포함할 수 있다. 구성기 수단은 계산 수단의 구성을 결정하고 CRC 에러 검출 방법론의 파라미터들을 수신하는 구성 데이터 계산기를 포함할 수 있다. 구성기 수단은 구성 신호와 구성 제어 신호를 그것의 구성을 위한 계산 수단에 공급하는 구성 회로를 포함할 수 있다.
- <20> 구성기 수단은 CRC 에러 검출 코드를 계산하기 위한 계산 수단의 구성 가능한 요소들 전부에 관해 요구된 구성을 결정하고, 계산 수단의 모든 요소들을 함께 구성할 수 있다. 대안적으로, 구성기 수단은 CRC 에러 검출 코드를 계산하기 위해 계산 수단의 구성 가능한 요소들 중 적어도 일부에 관해 요구된 구성을 결정하고, 계산 수단의 이들 요소들을 함께 구성할 수 있다. 요소들이 하나의 배열을 포함할 때, 구성기 수단은 CRC 에러 검출 코드를 계산하기 위해 하나의 열에서의 요소들에 관해 요구된 구성을 결정하고, 그 열에서의 요소들 전부를 함께 구성할 수 있다. 구성기 수단은 열 단위로(column by column) 계산 수단의 요소들을 구성할 수 있다.
- <21> 장치는 각각 예컨대 4 내지 M의 범위에 있는 미리 결정된 크기의 CRC 발생기 다항식을 사용하는 복수의 CRC 에러 검출 방법론을 구현하도록 구성 가능할 수 있다.
- <22> 구성기 수단은 CRC D 매트릭스를 계산하기 위해 CRC 에러 검출 방법론의 CRC 발생기 다항식을 사용할 수 있다. D 매트릭스는 CRC 에러 검출 코드를 계산하기 위해 요구된 계산 수단의 구성을 결정할 수 있다. 구성기 수단은 D 매트릭스의 행을 계산할 수 있고, 이러한 D 매트릭스의 행을 계산 수단이 구성 가능한 요소들의 배열의 대응하는 열을 구성하기 위해 사용할 수 있다. 이는 D 매트릭스의 각각의 행의 계산(제 1 행은 제외하고)이 이전 행의 계산의 결과에 기초하기 때문에 구성기 수단에서 메모리를 절약한다. D 매트릭스는 0들과 1들의 배열을 포함할 수 있고, 1들의 위치는 CRC 에러 검출 코드의 계산시 데이터를 사용하는 계산 수단의 구성 가능한 요소들의 요구된 위치를 나타낼 수 있고, 0들의 위치는 CRC 에러 검출 코드의 계산시 데이터를 사용하지 않는 계산 수단의 구성 가능한 요소들의 요구된 위치를 나타낼 수 있다. D 매트릭스의 크기와 그로 인한 구성 가능한 요소들의 배열의 크기는, 장치에 의해 사용되는 CRC 에러 검출 방법론의 CRC 발생기 다항식의 크기 또는 전력과 계산 수단에 의해 제공된 입력 디바이스들의 개수에 의해 결정될 수 있다. 입력 디바이스들의 개수는 요소들의 배열에서의 열들의 개수를 결정할 수 있고, CRC 발생기 다항식의 크기는 요소들의 배열에서의 행들의 개수를 결정할 수 있다.

- <23> 장치는 또한 추가 피드백 디바이스를 제공함으로써, 장치의 포트 크기보다 큰 크기들을 가지는 CRC 발생기 다항식들을 사용하여 CRC 에러 검출 코드들을 계산할 수 있다. 이러한 추가 피드백 디바이스는 계산 수단의 피드백 디바이스들의 적합한 디바이스들로 계산된 CRC 에러 검출 코드의 부분을 발송할 수 있고, 그러한 피드백 디바이스들은 구성 가능한 요소들로 CRC 에러 검출 코드의 계산된 부분을 발송할 수 있다.
- <24> 장치는 송신될 데이터에 관한 CRC 에러 검출 코드를 계산하기 위해 사용될 수 있다. 이 장치는 계산된 CRC 에러 검출 코드를 데이터에 첨부할 수 있고, 그 데이터와 CRC 에러 검출 코드를 함께 송신할 수 있다.
- <25> 장치는 수신된 데이터에 관한 CRC 에러 검출 코드를 계산하기 위해 사용될 수 있다. 이 장치는 수신된 데이터에 관한 CRC 에러 검출 코드를 계산하고 계산된 CRC 에러 검출 코드를 데이터를 가지고 수신된 CRC 에러 검출 코드와 비교할 수 있다. CRC 에러 검출 코드가 동일하지 않다면, 이는 데이터 송신기 에러가 발생하였음을 나타내고, 장치는 이를 나타내기 위해 하나 이상의 신호를 출력할 수 있다.
- <26> 장치는 프로토콜 프로세서의 부분을 포함할 수 있다.
- <27> 본 발명의 제 2 양상에 따르면, CRC 에러 검출 방법론에 따라 데이터에 관한 CRC 에러 검출 코드를 계산하는 방법이 제공되고, 이 방법은 CRC 에러 검출 코드 방법론을 구현하고 CRC 에러 검출 코드를 계산하기 위해 본 발명의 제 1 양상의 장치를 사용하는 단계를 포함한다.
- <28> 이제 본 발명의 일 실시예가 첨부 도면을 참조하여, 오직 예에 의해서 설명된다.

실시예

- <32> 도 1을 참조하면, 장치(1)는 에러 검출 코드 계산 수단(2)과 구성기(configurator) 수단(3)을 포함한다. 계산 수단(2)은 구성 가능한 요소들의 배열(4)을 포함한다. 그러한 요소들은 32개의 열과 32개의 행으로 이루어진 매트릭스를 형성한다. 하지만, 다른 개수의 열들과 행들도 이용될 수 있다는 것이 이해된다. 계산 수단은 또한 각각 XOR 게이트를 포함하는 32개의 입력 포트의 행(5)을 포함한다. 입력 포트 5번부터 32번까지는 또한 멀티플렉서를 포함하고 따라서 프로그래밍 가능하다. 입력 포트(5)는 에러 검출 코드가 계산될 데이터를 수신하고, 그 데이터를 구성 가능한 요소들의 배열(4)에 공급한다. 계산 수단은 또한 출력 레지스터(6)를 포함한다. 구성 가능한 요소들의 배열(4)의 각각의 행은 에러 검출 코드의 부분을 계산하고, 계산한 결과를 출력 레지스터(6)에 출력한다. 계산 수단은 또한 프로그래밍 가능한 피드백 멀티플렉서(7)의 한 열과 피드백 회로(8)를 포함한다. 피드백 회로(8)는 출력 레지스터(6)를 입력 포트(5)와 피드백 멀티플렉서(7)에 연결하고, 요구된 대로 계산된 에러 검출 코드의 부분을 입력 포트(5)와 피드백 멀티플렉서(7)로 공급한다.
- <33> 도 2와 도 3을 참조하여, 구성 가능한 요소(4)의 배열의 구성이 더 상세히 설명된다. 각각의 구성 가능한 요소(4)는 데이터-경로 회로(10)와 제어-경로 회로(11)를 포함한다(도 2). 데이터-경로 회로(10)는 XOR 게이트(12)와 멀티플렉서(13)를 포함한다. XOR 게이트(12)는 도시된 바와 같이, 2개의 입력인 입력 0(에러 검출 코드가 계산되어야 할 데이터의 부분을 수신하는)과 입력 1 및 출력을 포함한다. 멀티플렉서(13)는 또한 도시된 바와 같이 2개의 입력과 1개의 출력을 포함한다. 멀티플렉서(13)는 또한 제어-경로 회로(11)에 연결되는 제어 입력(14)을 포함한다. XOR 게이트(12)의 출력은 멀티플렉서(13)의 제 1 입력에 연결되고, XOR 게이트(12)의 입력 1은 멀티플렉서(13)의 제 2 입력에 연결된다.
- <34> 제어-경로 회로(11)는 멀티플렉서(15)와 구성 레지스터(16)를 포함한다. 멀티플렉서(15)는 도시된 바와 같이 2개의 입력과 1개의 출력을 포함한다. 멀티플렉서(15)는 또한 제어 입력(17)을 포함한다. 구성 레지스터는 도시된 바와 같이 1개의 입력과 1개의 출력을 포함한다. 멀티플렉서(15)의 출력은 레지스터(16)의 입력에 연결되고, 레지스터(16)의 출력은 데이터-경로 회로(10)의 멀티플렉서(13)의 제어 입력(14)에 연결된다. 레지스터(16)의 출력은 또한 멀티플렉서(15)의 입력들 중 하나에 연결된다.
- <35> 각각의 구성 가능한 요소(4)에 관해서는, 제어-경로 회로(11)의 멀티플렉서(15)가 Config Data라고 이름이 붙여진 입력으로 구성 신호를 수신하고, 멀티플렉서(15)의 제어 입력(17)(Config Enable이라고 이름이 붙여진)이 높은 구성 제어 신호를 수신할 때에는 그 구성 신호가 구성 레지스터(16)에 공급된다. 이는 요소(4)의 데이터-경로 회로(10)의 멀티플렉서(13)의 제어 입력(14)에 구성 신호(낮은 신호 또는 높은 신호)를 출력한다. 높은 구성 신호는 멀티플렉서(13)의 출력이 XOR 게이트의 입력 0에서 수신된 데이터의 부분과 입력 1에서 수신된 신호 사이의 XOR 함수의 결과가 되게, 데이터-경로 회로(10)가 구성되도록 결정한다. 이 경우, 데이터의 부분은 에러 검출 코드의 계산시 사용된다. 낮은 구성 신호는 멀티플렉서(13)의 출력이 간단하게 XOR 게이트(12)의 입력 1에서 수신된 신호가 되게 데이터-경로 회로(10)가 구성되도록 결정한다. 이 경우, XOR 게이트(12)의 입력 0에서

수신된 데이터의 부분은 에러 검출 신호의 계산시 사용되지 않는다.

- <36> 구성 가능한 요소들(4)은 상호연결된 배열로 배치된다. 도 3은 요소들(4) 중 16개가 어떻게 상호연결되는지를 보여준다. 각각의 행에 관해 그 행에서의 요소들(4) 각각이 상호연결되는 것을 볼 수 있다. 각각의 행에 관해, 제 1 요소(4)의 데이터-경로 회로(10)의 멀티플렉서(13)의 출력은 제 2 요소(4)의 데이터-경로 회로(10)의 XOR 게이트(12)의 입력 1에 연결되고 제 2 요소(4)의 데이터-경로 회로(10)의 멀티플렉서(13)의 출력은 제 3 요소(4)의 데이터-경로 회로(10)의 XOR 게이트(12)의 입력 1에 연결되는 식이다. 이러한 연결 배치는 그 행에서의 마지막 요소가까지 반복된다. 각각의 행에 관해, 요소들(4)의 제어-경로 회로들(11)의 멀티플렉서(15)의 입력들(Config Data) 중 하나는, 구성 신호들을 수신을 위해 함께 연결된다. 각 열에 관해서는 열에서의 요소들(4) 각각이 상호연결되는 것을 볼 수 있다. 각각의 열에 관해서는, 요소들(4)의 데이터-경로 회로들(10)의 XOR 게이트(12)의 입력 0이, 에러 검출 코드가 계산될 데이터의 수신을 위해, 서로 연결된다. 각각의 열에 관해서는, 요소들(4)의 제어-경로 회로들(11)의 멀티플렉서(15)의 제어 입력(17)이 또한 구성 제어 신호들의 수신을 위해, 서로 연결된다.
- <37> 도 1을 다시 참조하면, 장치(1)의 구성기 수단(3)은 다수의 레지스터(21) 각각의 입력에 연결되는 마이크로프로세서 인터페이스(20)를 포함한다. 구성기 수단(3)은 또한 레지스터(21)의 각각의 출력에 연결되는 구성 데이터 계산기(22)를 포함한다. 구성기 수단(3)은 또한 구성 데이터 계산기(22)에 연결된 출력을 가지는 프로세서 제어 신호 발생기(23)를 포함한다. 구성기 수단(3)은 구성 계산기(22)에 연결되는 카운터(24)를 포함한다. 구성기 수단(3)은 또한 구성 회로(25)를 포함한다. 이 구성 회로(25)는 구성 데이터 계산기(22)에 연결되는 1개의 입력과, 계산 수단(2)에 연결된 복수의 출력을 포함한다.
- <38> 이제 특정 CRC 에러 검출 방법론을 참조하여 장치(1)의 동작이 설명된다. CRC는 모든 통신망 주요 층(major layer) 2개의 처리 프로토콜에서 사용된 주된 에러 검출 방법론이다. CRC는 데이터에서 에러를 검출하기 위한 다항식-기반의 방법이다. 에러 검출 코드는 그 데이터에 관해 계산된다. 이 에러 검출 코드는 이후 송신된 본래의 데이터와 에러 검출 코드를 포함하는 데이터 및 메시지에 첨부된다. 수신기에서, 메시지는 에러를 검사하기 위해 분석된다.
- <39> CRC 에러 검출 방법론은 선택된 CRC 발생기 다항식으로 본래의 데이터를 모듈로-2 나누기(modulo-2 division)하는 것에 기초한다. 이 CRC 방법론은 D 매트릭스라고 알려진 매트릭스를 발생시키기 위해 선택된 CRC 발생기 다항식을 사용하는 것을 수반한다. D 매트릭스는 에러 검출 코드 계산 수단의 구성 가능한 요소들의 배열을 위해 요구되는 구성을 결정한다. D 매트릭스는 0들과 1들의 배열을 포함하고, 1들의 위치는 CRC 에러 검출 코드의 계산시 데이터를 사용하는 배열에서의 구성 가능한 요소들의 요구된 위치를 표시하며, 0들의 위치는 CRC 에러 검출 코드의 계산시 데이터를 사용하지 않는 배열에서의 구성 가능한 요소들의 요구된 위치를 표시한다. 그러므로 D 매트릭스의 크기와 구성 가능한 요소들의 배열의 크기는, 선택된 CRC 발생기 다항식의 크기 또는 거듭제곱(power)과 계산 수단에 의해 제공된 입력 디바이스의 개수에 의해 결정된다. 입력 디바이스의 개수는 그 배열에서의 열들의 개수를 결정하고, CRC 발생기 다항식의 거듭제곱은 그 배열에서의 행들의 개수를 결정한다.
- <40> 본 실시예의 장치는 재구성 가능한 CRC 에러 검출 장치를 포함하고, 이 검출 장치에서는 구성 가능한 요소들의 배열이 32개의 열과 32개의 행을 가진다. 그러므로 본 실시예의 장치는 크기가 4,8,16 및 32비트인 32개의 비트까지의 블록들에서 데이터를 수신할 수 있고, 32의 거듭제곱까지 임의의 CRC 발생기 다항식에 관한 에러 검출 코드 계산을 지원할 수 있다.
- <41> 32의 거듭제곱을 가지는 CRC 발생기 다항식을 사용하는 32개의 비트를 포함하는 블록들로 나누어진 데이터에 관한 에러 검출 코드의 계산이 먼저 설명된다. CRC 에러 검출 방법론의 파라미터들은 구성기 수단(3)의 마이크로프로세서 인터페이스(20)로 입력된다. 이들은 크기, 즉 선택된 CRC 발생기 다항식의 거듭제곱인 크기(32개), 입력 디바이스들의 개수(32개) 및 CRC 발생기 다항식을 포함한다. 이들은 레지스터(21)에 저장되고, 구성 데이터 계산기(22)에 출력된다. 신호는 프로세서 제어 신호 발생기(23)로부터 출력되고, 구성 데이터를 계산하고 계산 수단(2)의 구성 가능한 요소들(4)의 배열을 구성하는 프로세스가 개시된다. 구성기 수단(3)은 우선 레지스터(21)에 저장된 파라미터들을 사용하여 CRC 에러 검출 방법론 D 매트릭스를 계산한다. D 매트릭스는 한번에 한 행씩 계산된다. 이것은 요구되는 메모리를 줄이는데, 이는 그 매트릭스의 각 행의 계산(제 1 행을 제외한)이 이전 행의 계산의 결과에 기초하기 때문이다. 이후 구성기 수단(3)은 계산 수단(2)의 구성 가능한 요소들(4)의 배열의 각각의 대응하는 열에 관한 구성 데이터를 결정하기 위해, D 매트릭스의 각각의 행의 계산을 사용한다. 예컨대, 구성기 수단(3)은 계산 수단(2)의 구성 가능한 요소들(4)의 배열의 제 1 열에 관한 구성 데이터를 결정하기 위해 D 매트릭스의 제 1 행의 계산을 사용한다. D 매트릭스의 각각의 행은 일련의 0과 1을 포함하게 된다.

각각의 행에 관해, 0들과 1들의 위치는 CRC 에러 검출 방법론에 의해 지시를 받게 된다. 0들의 위치는 계산 수단(2)의 구성 가능한 요소들(4)의 배열의 대응하는 열에서 구성 가능한 요소들(4)의 위치를 표시하고, 이때 그 계산 수단(2)은 에러 검출 코드의 계산시 데이터의 부분을 사용하지 않는다. 1들의 위치는 계산 수단(2)의 구성 가능한 요소들(4)의 배열의 대응하는 열에서 구성 가능한 요소들(4)의 위치를 표시하고, 이때 그 계산 수단(2)은 에러 검출 코드의 계산시 데이터의 부분을 사용한다.

<42> 배열의 열에 관한 구성 데이터가 결정되었을 때, 이 데이터는 구성기 수단(3)의 구성 회로(25)에 공급된다. 이 회로는 도 3의 라인 C0 내지 C3와 같은 구성 라인을 경유하여 구성 가능한 요소들(4)의 배열의 한 열을 구성하기 위해 구성 데이터를 사용한다. 그 열에 관한 구성 데이터는 그 배열에서의 모든 열에 방송되지만, 도 3의 라인들(C0 내지 C3)과 같은 구성 제어 신호 라인들 상에 방송되는 구성 제어 신호들은 그 배열의 바라는 열만이 구성되는 것을 보장하기 위해, 한꺼번에 하이(high) 1로 설정된다. 그 열에서의 각각의 구성 가능한 요소(4)에 관해서는, 제어-경로 회로(11)의 멀티플렉서(15)의 제어 입력(17)(Config Enable)이 하이 구성 제어 신호를 수신하고, 멀티플렉서(15)는 Config Data라고 이름이 붙여진 그것의 입력 상에서 구성 신호를 수신한다. 그 구성 신호는 구성 레지스터(16)에 공급되고, 그 구성 레지스터(16)는 요소(4)의 데이터-경로 회로(10)의 멀티플렉서(13)의 제어 입력(14)에 구성 신호(low이거나 high인 신호)를 출력한다. 구성 신호는 D 매트릭스의 대응하는 행에서의 대응하는 엔트리(entry)가 1 또는 0인지에 따라 로우 또는 하이이다. 하이인 구성 신호는 요소(4)의 멀티플렉서(13)가 요소(4)의 XOR 게이트(12)의 입력 1에서 수신된 신호와 입력 0에서 수신된 데이터 신호 사이의 XOR 함수의 결과를 출력하는 것을 결정한다. 로우인 구성 신호는 요소(4)의 멀티플렉서(13)가 요소(4)의 XOR 게이트(12)의 입력 1에서 수신된 신호를 단순히 출력하는 것을 결정한다.

<43> 계산 수단(2)의 배열에서의 각각의 구성 가능한 요소(4)는 위에서와 같이 구성된다. 배열의 각 노드에서 XOR 게이트를 가지거나 XOR 게이트를 가지지 않는 대신, CRC 에러 검출 방법론에 의해 결정된 것처럼, XOR 게이트를 포함하고, XOR 게이트의 XOR 함수의 결과를 출력하거나 출력하지 않도록 구성될 수 있는 재구성 가능한 요소가 제공된다. 이러한 식으로, 임의의 CRC 에러 검출 방법론이 그 배열의 적합한 요소들의 XOR 게이트를 사용하거나 사용하지 않음으로써 요구에 응할 수 있다.

<44> 일단 계산 수단(2)의 구성 가능한 요소들(4)의 배열이 선택된 CRC 에러 검출 방법론에 따라 구성되었다면, 계산 수단(2)이 계산 수단에 의해 수신된 데이터에 관해 에러 검출 코드를 계산하기 위해 사용될 수 있다. 데이터는 입력 포트(5)들을 경유하여 수신된다. 데이터는 1개의 비트가 각 입력 포트(5)에 의해 수신되는 것처럼, 32개의 비트로 이루어진 블록에서 수신된다. 각 블록에 관해, 들어오는 데이터의 각각의 비트가 요소들의 배열의 모든 행에 대해 하나의 구성 가능한 요소(4)로 공급된다. 예컨대, 제 1 입력 포트(5)에 의해 수신된 데이터의 비트는 그 배열의 행들의 각 행의 제 1 요소(4)로 공급되고, 제 2 입력 포트(5)에 의해 수신된 데이터의 비트는 그 배열의 행들의 각각의 제 2 요소(4)로 공급되는 식이다. 데이터의 각 블록에 관해, 데이터의 각 비트는 요소(4)의 데이터-경로 회로(10)의 XOR 게이트(12)의 입력 0에 공급된다.

<45> 각 요소(4)는 에러 검출 코드의 계산시 입력 데이터 비트를 사용하거나 사용하지 않도록 구성되었다. 예컨대, 도 3에 도시된 요소들(4)의 배열의 제 1 행을 고려하고, 마지막 2개의 요소들(4)(좌측에서 우측으로의)만이 그 XOR 게이트의 XOR 함수의 결과를 출력하도록 구성되었다고 가정한다. 행에서의 모든 요소들(4)에 관해, 데이터의 제 1 비트는 그 행에서의 제 1 요소(4)의 데이터-경로 회로(10)의 XOR 게이트(12)의 입력 0으로 공급되고, 유사하게 데이터의 제 2 비트는 그 행에서의 제 2 요소(4)의 데이터-경로 회로(10)의 XOR 게이트(12)의 입력 0으로 공급되는 식이다. 로우 신호(0)는 그 행에서의 제 1 요소(4)의 데이터-경로 회로(10)의 XOR 게이트(12)의 입력 1로 공급될 수 있다. 제 1 요소(4)의 출력은 제 2 요소(4)의 데이터-경로 회로(10)의 XOR 게이트(12)의 입력 1로 공급되고, 제 2 요소(4)의 출력은 제 3 요소(4)의 데이터-경로 회로(10)의 XOR 게이트(12)의 입력 1로 공급되며, 제 3 요소(4)의 출력은 제 4 요소(4)의 데이터-경로 회로(10)의 XOR 게이트(12)의 입력 1로 공급된다. 제 4 요소(4)의 출력은 데이터에 관해 계산된 에러 검출 코드의 제 1 부분의 제 1 비트를 포함한다. 행에서의 제 1 요소(4)는 XOR 게이트(12)의 입력 1에서 수신된 신호를 출력하도록, 즉 입력 출력 연결로서 작용하도록 구성되어, 이 요소의 XOR 게이트(12)의 입력(1)에 의해 수신된 로우 신호가 그 행의 제 2 요소(4)에 공급된다. 그러므로, 제 1 요소(4)의 XOR 게이트(12)의 입력 0에 의해 수신된 제 1 데이터 비트는 에러 검출 코드의 계산시 사용되지 않는다. 제 2 요소(4)는 또한 XOR 게이트(12)의 입력 1에서 수신된 신호를 출력하도록, 즉 입력 출력 연결로서 작용하도록 구성되어 로우 신호가 그 행에서의 제 3 요소(4)에 공급된다. 그러므로 제 2 요소(4)의 XOR 게이트(12)의 입력 0에 의해 수신된 제 2 데이터는 에러 검출 코드의 계산시 사용되지 않는다. 행에서의 제 3 요소(4)는 그 XOR 게이트(12)의 XOR 함수의 결과를 출력하도록 구성되어, 이 요소에 관해서 XOR 게이트(12)의 입력 0에 의해 수신된 제 3 데이터 비트와 XOR 게이트(12)의 입력 1에 의해 수신된 로우 신호가 함께 XOR 처리

되고 그러한 XOR 함수의 결과인 제 3 데이터 비트가 그 행의 제 4 요소(4)에 공급된다. 행에서의 제 4 요소(4)는 또한 그 XOR 게이트의 XOR 함수의 결과를 출력하도록 구성되어, 이 요소에 관해서 XOR 게이트(12)의 입력 0에 의해 수신된 제 4 데이터 비트와 XOR 게이트(12)의 입력(1)에 의해 수신된 제 3 데이터 비트가 함께 XOR 처리되고 에러 검출 코드의 제 1 부분의 제 1 비트로서 그 행의 제 4 요소(4)로부터 출력된다. 그러므로 데이터의 제 1 블록의 제 3 데이터 비트와 제 4 데이터 비트는 제 1 에러 검출 코드의 계산시 사용된다.

<46> 유사한 프로세스가 각 행에서의 요소들(4)의 구성에 따라, 도 1의 계산 수단(2)의 구성 가능한 요소들(4)의 배열의 각 행에 관해 수행된다. 계산 수단(2)으로의 제 1 데이터 블록 입력에 관해서는, 각 행의 출력이 제 1 에러 검출 코드의 한 비트를 형성한다. 이는 레지스터(6)에 저장된다. 제 1 에러 검출 코드는 입력 포트(5)로 피드백된다. 데이터의 제 2 블록은 입력 포트(5)로 공급된다. 제 1 에러 검출 코드와 데이터의 제 2 블록이 입력 포트(5)에서 XOR 처리된다. 그 결과는 구성 가능한 요소들(4)의 배열로 공급되고, 이전과 같이 제 2 에러 검출 코드를 계산하기 위해 사용된다. 제 2 에러 검출 코드는 출력 레지스터(6)에 저장되고, 입력 포트(5)로 피드백된다. 데이터의 제 3 블록은 입력 포트(5)로 공급된다. 제 2 에러 검출 코드와 데이터의 제 3 블록이 입력 포트(5)에서 XOR 처리된다. 그 결과는 구성 가능한 요소(4)의 배열로 공급되고, 제 3 에러 검출 코드를 계산하기 위해 사용된다. 제 3 에러 검출 코드는 출력 레지스터(6)에 저장되고, 입력 포트(5)로 피드백된다. 이 프로세스는 데이터 모두가 입력 포트(5)로 공급될 때까지 계속되고, 에러 검출 코드를 계산하기 위해 사용된다. 최종 에러 검출 코드는 출력 레지스터(6)에 저장된다. 이러한 식으로, 모든 데이터가 최종 에러 검출 코드를 계산하기 위해 사용된다.

<47> 위의 내용은 장치(1)의 포트 크기가 32비트이고 그 데이터가 32비트의 블록들로 나누어질 수 있는 실시예를 설명한다. 장치(1)는 다른 크기의 블록들에서 데이터를 수신하기 위해 다른 포트 크기들을 제공할 수 있다. 이는 5번째 입력 포트(5)부터 32번째 입력 포트(5) 중 적합한 입력 포트의 멀티플렉서들을 프로그래밍함으로써 달성된다. 입력 포트(5)의 멀티플렉서들을 프로그래밍하는 것은 장치(1)의 실행 시간 동안 발생할 수 있다. 특히, 장치(1)는 그 포트 크기를 제공할 수 있고 따라서 4, 8, 12 또는 24비트의 데이터 블록을 수신할 수 있다. 예컨대, 오직 8비트의 블록들에서 데이터를 수신하는 것이 요구된다면, 즉 포트의 크기를 8비트로 변경한다면, 장치(1)의 9번째 입력 포트(5)부터 32번째 입력 포트(5)의 멀티플렉서들은 데이터 비트를 배열로 공급하는 대신, 구성 가능한 요소들(4)의 배열로 로우 신호(0)를 공급하도록 프로그래밍된다. 포트 크기를 제어하기 위해 장치(1)에 의해, 입력 포트(5)의 멀티플렉서들의 프로그래밍인 포트 크기 구성(Port Size Configure) 신호가 발생된다.

<48> 또한, 장치(1)는 바이트의 수가 가변적인 데이터를 수신하기 위해 실행시(runtime)에 프로그래밍될 수 있다. 이는 입력 포트(5)의 적합한 멀티플렉서를 프로그래밍하고 장치(1)의 피드백 멀티플렉서들(7)의 적합한 멀티플렉서들을 프로그래밍함으로써 달성된다. 장치(1)에 의해 발생된 포트 크기 구성 신호는 또한 가변적인 입력 데이터 바이트 크기들을 처리하기 위해, 입력 포트(5)의 멀티플렉서들과 피드백 멀티플렉서들(7)을 프로그래밍하는 책임을 진다. 예컨대, 장치는 32의 거듭제곱을 가지는 CRC 발생기 다항식을 가지는 CRC 에러 검출 방법론을 사용하여 에러 검출 코드를 계산하도록 구성될 수 있고, 장치의 포트 크기는 32비트가 되도록 프로그래밍될 수 있다. 데이터의 마지막 블록이 처리될 16비트만을 포함한다면, 포트 크기 구성 신호는 장치(1)로 하여금 입력 데이터의 바이트 크기 또는 비트의 개수에 있어서의 이러한 변화를 다루도록 프로그래밍되게 한다. 특히, 17번째 입력 포트(5)로부터 32번째 입력 포트(5)까지의 멀티플렉서는 장치(1)의 요소들(4)의 배열로 로우 신호(0)를 공급하도록 프로그래밍되고, 배열의 좌측면에서는 하부의 16개의 피드백 멀티플렉서(7)가 이전에 계산된 에러 검출 코드를 수신하도록 프로그래밍되며, 배열의 17번째 행으로부터 32번째 행의 이전에 계산된 에러 검출 코드는 하부의 16개의 피드백 멀티플렉서(7)로 발송된다(routed). 개수가 가변적인 바이트를 포함하는 데이터를 수신하는 장치(1)의 능력은, 알려진 에러 검출 디바이스들에 비해 이 장치가 가지는 중요한 장점인데, 이는 대부분의 통신망이 크기가 가변적인 바이트 지향의(orientated) 프로토콜을 사용하기 때문이다.

<49> 위에서 상세히 설명된 실시예는 32의 크기 또는 거듭제곱을 가지는 CRC 발생기 다항식을 사용하는 에러 검출 코드를 계산한다. 이 장치는 또한 그 크기가 장치의 포트 크기보다 크거나 작을 수 있는 다른 CRC 발생기 다항식 크기에 관한 에러 검출 코드들을 계산할 수 있다. CRC 발생기 다항식 크기가 장치의 포트 크기보다 클 때에는, 여분의 피드백 버스가 제공된다. 이 버스는 계산된 에러 검출 코드의 부분을 장치(1)의 구성 가능한 요소들(4)의 배열의 좌측면에서 피드백 멀티플렉서(7)들 중 적합한 멀티플렉서로 발송한다. 어떠한 피드백도 요구되지 않은 배열의 임의의 행에서 제 1 요소(4)의 입력 1로 로우 신호가 공급된다. CRC 다항식의 요구된 크기에 응하기 위해, 피드백 멀티플렉서(7)의 프로그래밍을 제어하도록, 장치(1)에 의해 CRC 크기 구성 신호가 발생된다.

<50> 이후 장치(1)는, 예컨대 거듭제곱이 16인 CRC 발생기 다항식을 사용하는 CRC 에러 검출 코드와 같은 상이한 CRC 에러 검출 방법론에 관한 에러 검출 코드를 계산하기 위해 사용될 수 있다. 새로운 CRC 에러 검출 방법론이 선

택되고, 이러한 CRC 방법론을 위한 구성 데이터를 계산하기 위해 구성기 수단(3)이 사용된다. 이후 계산 수단(2)이 재구성되고, 새로운 에러 검출 코드를 계산하기 위해 사용된다. 장치(1)의 재구성 능력은 상이한 CRC 에러 검출 방법론이 오직 1개의 장치를 사용하기 위해 응해지는 것을 허용한다.

- <51> 다수의 가능한 CRC 발생기 다항식 크기를 지원할 프로그래밍 가능한 장치와 그로 인한 다수의 가능한 CRC 에러 검출 방법론은, 본 발명의 중요한 장점이고, 데이터-링크 층 프로토콜에서의 에러 검출을 지원하는 것뿐만 아니라, 상부 층 프로토콜과 데이터 암호화(encryption)를 위해 바람직하다.
- <52> 본 발명의 장치는 고정된 하드웨어 엔진으로서 구현된다. 이는 구성 가능한 요소들의 배열을 구성하기 위해 사용되는데, 즉 소프트웨어를 사용하여 이전에 실행되어야 했던 함수를 수행한다.
- <53> 장치(1)의 일 구현예가 Altera Stratix II FPGA 기술을 사용하여 만들어졌다. 구현된 장치의 최대 속도는 117Mhz이다. 이 속도에서는 완전한 포트 크기 이용(32 비트)을 통해 이러한 아키텍처가 3.77Gbps를 처리할 수 있다. 이는 FPGA 구현예에 있어서는 크게 인상적인 것이다. 장치의 이러한 설계는 기술 수축(shrink)이 완전한 포트 이용으로 10Gbps의 수행 능력을 쉽게 가능하게 할 ASIC 상에서 구현된 대부분의 값을 가지게 된다. 하지만, 장치(1)에 관한 다른 구현예가 사용될 수 있다는 것이 이해된다.
- <54> 장치(1)는 수신된 데이터 또는 송신될 데이터에 관한 에러 검출 코드를 계산하기 위해 사용될 수 있다. 송신 데이터에 관해서는, 장치가 계산된 에러 검출 코드를 데이터의 끝에 추가할 수 있고, 결합된 데이터와 코드를 송신할 수 있다. 수신된 데이터에 관해서는 장치가 그 데이터에 관한 에러 검출 코드를 계산하고, 계산한 것을 데이터에 첨부된 에러 검출 코드와 비교할 수 있다. 이들 결과가 동일한 값을 가진다면, 이는 데이터의 송신시 어떠한 에러도 발생하지 않음을 나타낸다. 이들 결과가 동일한 값을 가지지 않는다면, 장치는 버려질 데이터와 다시 보내져야 할 데이터를 나타내는 신호를 발생시킬 수 있다.

산업상 이용 가능성

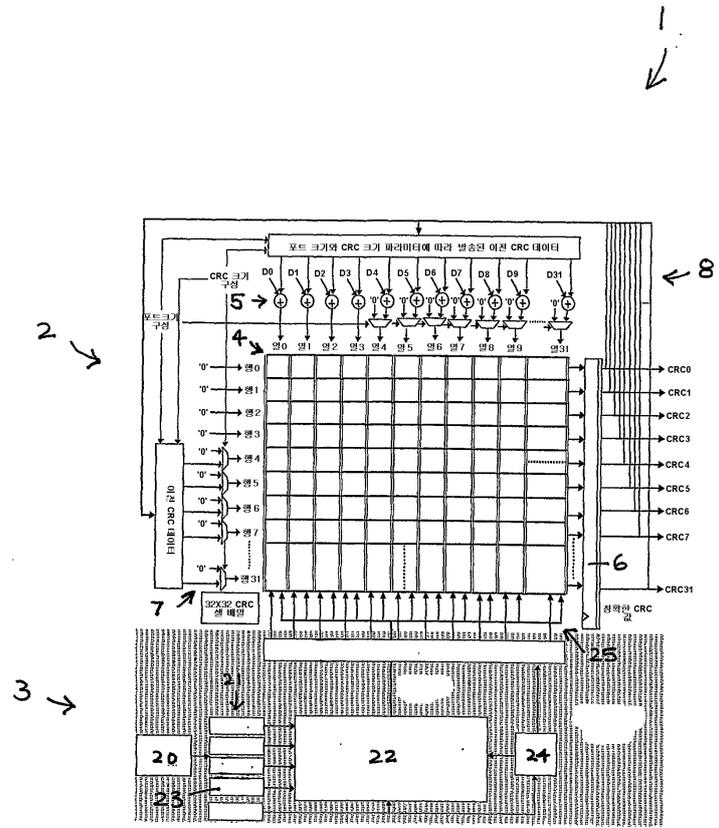
- <55> 전술한 바와 같이, 본 발명은 통신망에서 데이터에 관한 에러 검출 코드를 계산하는 것이 필요한 분야에서 이용 가능하다.

도면의 간단한 설명

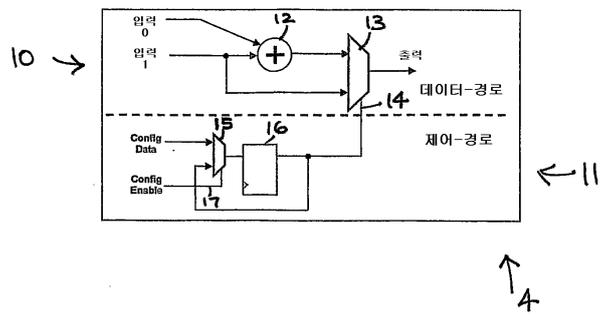
- <29> 도 1은 본 발명의 제 1 양상에 따른 장치의 개략적인 표현을 도시하는 도면.
- <30> 도 2는 도 1의 장치의 구성 가능한 요소들 중 하나의 개략적인 표현을 도시하는 도면.
- <31> 도 3은 도 1의 장치의 구성 가능한 요소들의 배열 부분의 개략적인 표현을 도시하는 도면.

도면

도면1



도면2



도면3

