

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) Int. Cl.⁶
G11B 20/02

(45) 공고일자 1999년04월 15일

(11) 등록번호 특0182001

(24) 등록일자 1998년12월 10일

(21) 출원번호 특1995-039602

(65) 공개번호 특1997-029555

(22) 출원일자 1995년11월03일

(43) 공개일자 1997년06월26일

(73) 특허권자 삼성전자주식회사 김광호
경기도 수원시 팔달구 매탄동 416번지
(72) 발명자 김중대
서울특별시 중구 신당 2동 432-173번지 9/6
(74) 대리인 김능균

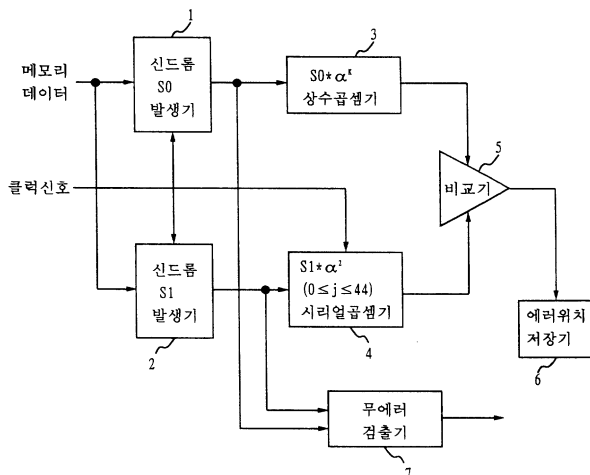
심사관 : 서호선

(54) 시디-롬 디코더에서의 무클럭 에러위치 검출회로

요약

본 발명은 시디-롬(CD-ROM) 디코더에서의 무클럭 에러 위치 검출회로에 관한 것인 바, 그 특징은 메모리로부터 입력된 데이터들을 생성 다항식으로 나눈 나머지값으로 에러의 존재유무와 에러의 위치 및 에러값도 연산을 통해서 알 수 있는 신드롬 S0 및 S1을 발생시키는 신드롬 S0 및 S1발생수단과, 상기 신드롬 S0에 상수 α^K 에 대한 곱셈을 수행하는 상수 곱셈수단과, 상기 신드롬 S1에 알파상수 α^0 부터 α^{n-1} 까지 병렬로 각각의 상수들에 대해서 미리 곱셈연산을 수행하는 알파 상수 곱셈수단과, 상기 알파 상수 곱셈수단의 각각의 출력값을 상수 곱셈수단의 출력값을 비교하여 두개의 입력값이 다른지를 판단하는 병렬 비교수단과, 상기 병렬 비교수단의 출력으로부터 에러 위치를 찾기 위한 디코딩수단으로 구성함에 있다.

대표도



명세서

[발명의 명칭]

시디-롬(CD-ROM) 디코더에서의 무클럭 에러 위치 검출 회로.

[도면의 간단한 설명]

제1도는 종래의 에러 정정 회로의 블록도.

제2도는 제1도에서 하나의 코드 워드에 대해서 에러 정정을 하는 과정을 시간적으로 나타낸 동작 타이밍도.

제3도는 본 발명에 의한 에러 위치 검출 회로의 일 실시예를 보인 블록도.

제4도는 제3도의 알파 상수 곱셈기의 일 실시예도.

제5도는 제3도의 동작 타이밍도.

* 도면의 주요부분에 대한 부호의 설명

1까지의 수)의 각각의 상수 (α^0 부터 α^{n-1} 까지의 상수)들에 대해서 병렬로 알파상수 α^i (단 i 는 0에서 $n-1$ 까지의 수)의 i 값에 해당하는 크기를 미리 곱셈 연산하는 알파 상수 곱셈기(40)로 구성하고, 병렬 비교수단은 상기 알파 상수 곱셈수단의 각각의 출력값에 상수 곱셈수단의 출력값을 비교하여 두개의 입력값이 다른지를 판단하는 병렬 비교기(50)로 구성하며, 디코딩 수단은 상기 병렬 비교수단의 출력으로부터 에러 위치를 찾기 위한 디코더(60)로 구성함이 바람직하다.

제4도는 제3도의 알파 상수 곱셈기의 일 실시예도로서, (a)는 신드롬 S_0 에 알파 상수($\alpha^0=1$)를 곱셈함으로써, 출력이 S_0 가 되는 알파상수 곱셈기(40A)를 보인 것이고, (b)는 다수의 배타적 오어게이트(EX-OR1~EX-OR4)를 이용하여 신드롬 S_0 에 알파상수($\alpha^1=x$)를 곱셈함으로써, 출력이 입력과 동일하게 되는 알파상수 곱셈기(40B)의 실시예를 보인 것이다. 이때 (a)의 경우 $S_0 * \alpha^0$ 는 $\alpha^0=1$ 이므로 입력과 출력이 동일하게 되는 것이고, (b)의 경우 $S_0 * \alpha^1$ 는 $\alpha^1=x$ 이므로

$$\begin{aligned} S_0 * \alpha^1 &= \{s(0) + s(1) * x + s(2) * x^2 + s(3) * x^3 + s(4) * x^4 + s(5) * x^5 \\ &\quad + s(6) * x^6 + s(7) * x^7\} * x \\ &= s(0) * x + s(1) * x^2 + s(2) * x^3 + s(3) * x^4 + s(4) * x^5 \\ &\quad + s(5) * x^6 + s(6) * x^7 + s(7) * x^8 \end{aligned}$$

여기서 $x^8 = 1 + x^2 + x^3 + x^4$ 이므로

$$\begin{aligned} S_0 * \alpha^1 &= \{s(0) + s(7)\} + s(0) * x + \{s(1) + s(7)\} * x^2 + \{s(2) + \\ &\quad s(7)\} * x^3 + \{s(3) + s(7)\} * x^4 + s(4) * x^5 + s(5) * x^6 + s(6) * x^7 \text{로 된다.} \end{aligned}$$

제5도는 본 발명에서 하나의 코드 워드에 대해서 에러 정정을 하는 과정을 시간적으로 나타낸 제3도의 동작 타이밍도로서, 메모리로부터 데이터를 읽는 기간($t=a$)과 메모리로부터 에러 데이터를 읽는 기간($t=c$)과 에러정정이 완료된 데이터의 메모리 기록기간($t=d$)을 나타낸 것이며, 이 도면에서와 같이 메모리로부터 데이터를 읽는 기간($t=a$)과 에러 데이터를 읽는 시간($t=c$)과 에러 기록시간($t=d$)은 종래의 경우와 동일하나 에러검출에 필요한 시간은 거의 소요되지 않음을 알 수 있다.

이상에서와 같은 구성을 참고하여 본 발명에 따른 시디-롬 디코더에서의 무클럭 에러 위치 검출 회로의 동작을 설명하면 다음과 같다.

먼저, 신드롬 S_0 발생기(1) 및 신드롬 S_1 발생기(2)에서 메모리로부터 코드 워드의 각 기호를 읽어들이는 시간($t=a$)이후 최종적으로 마지막 기호를 읽어들이는 순간에 신드롬 S_0, S_1 이 결정되면, 상수곱셈기(3)에서는 상기 신드롬 S_0 발생기(1)에서 만들어진 신드롬 S_0 에 상수(α^k)에 대한 상수 곱셈을 수행하여 출력($S_0 * \alpha^k$)을 만들어내고, 알파 상수 곱셈기(40)에서는 신드롬 S_1 발생기(2)에서 만들어진 신드롬 S_1 에 α^0 부터 α^{n-1} 까지 각각의 상수들에 대해서 병렬로 미리 상수 곱셈을 수행하여 알파 상수 α^i 가 각각 곱해진 각각의 i 값에 해당하는 크기의 곱셈 출력($S_1 * \alpha^0, S_1 * \alpha^1, S_1 * \alpha^2, \dots, S_1 * \alpha^{44}$)을 얻을 수 있으며, 병렬 비교기(50)에서는 상기 알파 상수 곱셈기(40)의 각각의 곱셈 출력값을 상수 곱셈기(3)의 출력값에 비교하여 두개의 입력값이 다른지를 판단하며, 디코더(60)에서는 상기 병렬 비교기(50)의 출력을 이용하여 빠르게 에러 위치를 찾을 수 있게 된다.

즉, 본 발명에서는 에러 위치를 찾는 시간을 최단시간내에 간단한 연산기능을 추가시켜서 클럭이 필요없이 신드롬 S_0, S_1 이 계산되자마자 그 위치를 알 수 있도록 하였다. 아래에 설명한 간단한 식들은 이와 같은 방식을 유도하기 위한 것을 수식적으로 표현한 것이다.

1에러 정정 코드인 (45,43) RS코드를 예로 들어 설명하면 다음과 같다.

1에러의 경우 신드롬 S_0, S_1 은 다음과 같이 구해진다.

$$S_0 = e(i), S_1 = e(i) * \alpha^i = S_0 * \alpha^i$$

위 수식에서 $S_1 = S_0 * \alpha^i$ 이므로 α^i 를 좌변으로 옮기고 상수를 α^k 를 곱하였을 경우 다음과 같다.

$$S_1 * \alpha^{k-i} = S_0 * \alpha^k, \text{ 여기서 변수 } i \text{는 에러가 발생한 위치로서 (45,43)RS 코드이므로 0부터 44까지의 크기를 가지며, } k \text{는 } i \text{의 최대값인 44이다.}$$

위의 수식에서 가장 먼저 메모리로부터 입력된 데이터를 최상위 기호라 하면 수식을 만족하는 실제 에러의 위치는 $(k-i+1)$ 이 된다. 따라서 에러 위치를 찾는 방식으로 i 를 0부터 44까지 모두 대입을 하여서 만족하는 i 값을 찾으면 실제 에러 위치는 $(k-i+1)$ 이 된다. 따라서 본 발명에서는 알파상수 곱셈기(40)에서 클럭을 사용하지 않고 병렬로 각각의 i 값에 해당하는 크기($S_1 * \alpha^{44}, S_1 * \alpha^{43}, S_1 * \alpha^{42}, \dots, S_1 * \alpha^0$)를 미리 계산함으로써 병렬 비교기(50)에서도 클럭을 사용하지 않고 상기 상수 곱셈기(3)의 출력과 알파 상수 곱셈기(40)의 출력을 각각 비교하여 곧 바로 에러 위치를 찾아내어 빠른 에러 정정을 수행할 수 있게 된다.

위의 수식으로부터 메모리에서 신드롬을 계산하기 위해서 읽은 순서대로 $k-i+1$ 의 크기가 현재의 에러 위치를 나타내는데, 1, 2배속에서는 각각 $k-i+1$ 의 값을 1부터 45까지 시간적으로 매 클럭마다 비교하므로 45개의 클럭이 필요하였지만, 본 발명에서는 1개의 에러가 발생하였을 경우 미리 계산된 각각의 결과값들

즉 $S1 \cdot \alpha^0, S1 \cdot \alpha^1, S1 \cdot \alpha^2, \dots, S1 \cdot \alpha^{44}$ 값들중에서 어느 하나의 값이 신드롬($S0 \cdot \alpha^k$)과 반드시 같을 것이므로 따라서 병렬 비교기(50)에서의 비교 결과에 따라 그 위치에 해당하는 값을 이용하여 최종적으로 에러 위치를 구할 수 있게 된다.

이상에서와 같이 본 발명에 따른 무클럭 에러 위치 검출 회로에 의하면 하드웨어적으로도 구조가 단순하고 클럭을 필요로 하지 않는 상태에서 에러 위치를 찾을 수 있을 뿐만 아니라 각각의 알파 상수 곱셈기도 제4도와 같이 8~12개 이내의 배타적 오어게이트만을 필요로 하므로 하드웨어적으로도 부담이 없어 용이하게 적용시킬 수 있는 유용함이 있다.

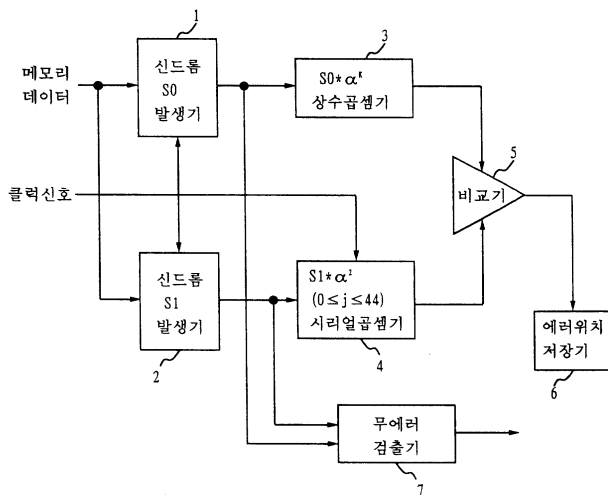
(57) 청구의 범위

청구항 1

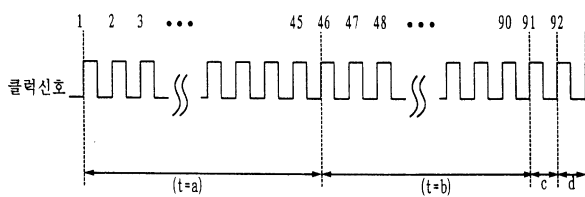
메모리로부터 입력된 데이터들을 생성 다항식으로 나눈 나머지값으로 에러의 존재유무와 에러의 위치 및 에러값도 연산을 통해서 알 수 있는 신드롬 S0 및 S1을 발생시키는 신드롬 S0 및 S1발생수단과, 상기 신드롬 S0에 상수 α^k 에 대한 곱셈을 수행하는 상수 곱셈수단과, 상기 신드롬 S1에 알파상수 α^0 부터 α^{n-1} 까지 병렬로 각각의 상수들에 대해서 미리 곱셈연산을 수행하는 알파 상수 곱셈수단과, 상기 알파 상수 곱셈수단의 각각의 출력값을 상수 곱셈수단의 출력값을 비교하여 두개의 입력값이 다른지를 판단하는 병렬 비교수단과, 상기 병렬 비교수단의 출력으로부터 에러 위치를 찾기 위한 디코딩수단으로 구성함을 특징으로 하는 무클럭 에러위치 검출회로.

도면

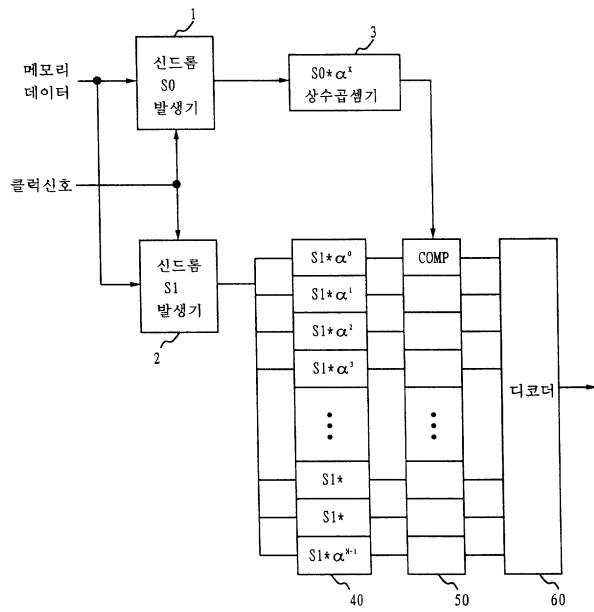
도면1



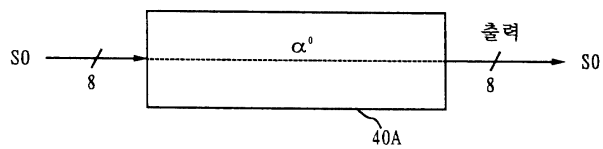
도면2



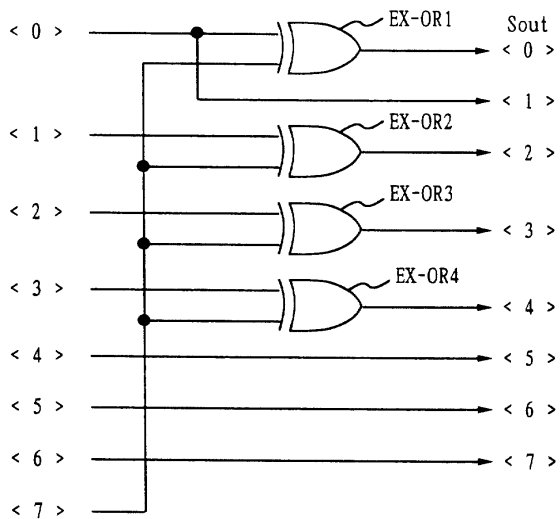
도면3



도면4a



도면4b



도면5

