

09 JUL 2002



166166

1

Specification



G00002

Title of the Invention:

Error-Detecting Device and Method Thereof

Detailed Description of the Invention:

[0001]

[Field of the invention]

The present invention relates to an error-detecting device and a method thereof which are employed to reproduce data from a storage medium such as a digital video disk (DVD).

[0002]

[Background art]

In the DVD, for example, data is recorded along with an error-correcting code (ECC). For the data read out from the DVD, an error-correcting process is performed in the unit of an ECC block.

Published Unexamined Patent Application Nos. 7-230388, 5-315974, and 7-50595, for example, disclose a device and method for making a correction of data read out from a storage medium.

[0003]

When data is read out from the DVD, for example, an error-correcting process must be performed by employing the error-correcting code, added to data, up to the limit of the ability in order to realize the error-correcting ability standardized with respect to burst errors.

If the error-correcting process is performed by employing the error-correction code up to the limit of the ability, the probability of a miscorrection will become high and therefore there will be a need to detect the occurrence of a miscorrection.

[0004]

To detect the miscorrection, a method called an EDC is employed.

After completion of the error-correction process, the EDC process is performed in the order of reading of an ECC block from a buffer, release of scrambling, calculations of the value of the EDC, comparison of the value of the read EDC and the value of the calculated EDC, and judgement of a miscorrection.

- 3 -

If the EDC process is performed in this order, the ECC block must be read out from a buffer. As a result, the overhead operation and the overhead time will become longer.

[0005]

[Problems to be solved by the invention]

The present invention has been made in view of the above-mentioned problems of background art. Accordingly, it is the object of the present invention to provide an error-detecting device and a method thereof which are capable of reducing overhead operation, detecting a miscorrection in a short processing time, and correcting errors.

[0006]

[Summary of the Invention]

[Error-Detecting Device]

According to one aspect of the present invention there is provided an error-correcting device for detecting errors that have occurred in an ECC block to be corrected, the ECC block being generated by respectively adding error-correcting codes (ECCs) to one or more EDC blocks each added with an error-detecting code (EDC), the error-correcting device comprising:

error-value calculating means for calculating values of said errors by using said error-correcting codes included in each data sequence included in said ECC block to each data sequence having said error-correcting codes (ECCs);

EDC syndrome correcting means for correcting a value of syndrome of said error-detecting code in accordance with the calculated value of the error every time said calculated value of the error in said data sequences is obtained; and

error-detecting means for detecting an occurrence of an error in said EDC block, based on the corrected value of the syndrome of said error-detecting code, wherein

said EDC block is constructed of one or more rows and one or more columns;

said error-correcting code is added to each of the rows or columns of said EDC block;

said data sequences include respectively the row or column of said EDC block and the error-correcting code added to the row or column of said EDC block;

said error-value calculating means processes said data sequence one by one and calculates the value of the error;

said EDC syndrome correcting means sequentially corrects a predetermined initial value of the syndrome of the error-detecting code every time said error value is calculated; and

every time the value of the error in said data sequence corresponding to said EDC block is calculated, said error-detecting means detects said occurrence of an error in said EDC block, based on the corrected value of the syndrome of said error-detecting code.

[0010]

[Operation of the Error-Correcting Device]

The error-correcting device according to the present invention is first employed to detect errors in a plurality of EDC blocks added with EDCs that are employed for error detection, like the data block recorded on the DVD, and furthermore, the error-correcting device is employed for detecting errors in an ECC block added with two kinds of parity codes PI and PO (product codes; error-correcting codes).

As an example, a description will hereinafter be made with respect to the case where the EDC is added at a predetermined location on the EDC block in the same direction of row as the parity code PI.

[0011]

[Error-Value Calculating Means]

The error-value calculating means sequentially takes out data sequence, including one row of data on the EDC block and a parity code PI added to this row, from the ECC block, and performs an error detecting process and an error value calculating process to one row of data on the EDC block on each the data sequence by employing the parity code PI.

[0012]

[EDC syndrome correcting means]

The EDC syndrome correcting means first sets the value of EDC syndrome to an initial value when an error with a fixed size occurs at a fixed location on each row of the EDC block.

Furthermore, every time the error-value calculating means calculates the value of an error in each data sequence, the EDC

syndrome correcting means corrects the above-mentioned initial value in sequence in accordance with the error value.

[0013]

[EDC Calculating Means]

The EDC calculating means calculates the value of the EDC syndrome of the entire EDC block.

[0014]

[Error Detecting Means]

If the error-value detecting means finishes the error detecting process and the error-value calculating process for all the rows included in the EDC block, the error detecting means compares the value of the EDC syndrome calculated by the EDC calculating means with the value of the EDC syndrome corrected by the EDC syndrome correcting means. When these values do not match each other, an error is detected.

The detected error is employed in the error correcting process.

[0015]

[Data Error Detecting Method]

The data error detecting method according to the present invention is a method of detecting errors that have occurred in an ECC block to be corrected. The ECC block is generated by respectively adding error-correcting codes (ECCs) to one or more EDC blocks each added with an error-detecting code (EDC). The method comprises the steps of: calculating a value of an error in each data sequence corresponding to each data sequence by using the error-correcting code included in the each data sequence to each data sequence having the ECCs; correcting a value of syndrome of the error-detecting code in accordance

with the calculated value of the error every time the calculated value of the error in said data sequences is obtained; and detecting an occurrence of an error in the EDC block, based on the corrected value of the syndrome of the error-detecting code.

[0016]

[Preferred embodiment]

The preferred embodiment of the present invention will hereinafter be described.

First, to make easy an understanding of an error-correcting device 1 according to the present invention, a description will be made with respect an ECC block, encoding, judgement of whether there is an error, error location, an error locator equation, chain search, etc.

[0017]

[ECC block]

Figure 1 is a diagram showing the process (S10) of generating data that is recorded on a DVD.

Figure 2 is a diagram showing the data unit 1 (EDC block) shown in Figure 1.

Figure 3 is a diagram showing the ECC block shown in Figure 1.

[0018]

As shown in Figure 1, if 2048 bytes of main data is input from a host system and if 4 bytes of Data ID, 2 bytes of IED, and 6 bytes of REV are input from a DVD controller, in the process in step 200 (S200) the ID, IED, and REV are added to the ID area in the main data.

[0019]

Furthermore, in step 202 (S202), an EDC is added and data unit 1 of 172 bytes $\times$ 12 rows (12 rows $\times$ 172 columns) are formed as shown in Figure 2.

The row and column are used for the convenience of explanation and do not have an meaning essential to the present invention. For clarifying explanation, the row and column will hereinafter be defined as shown in Figures 2 and 3.

[0020]

The data unit 1 shown in Figure 2 will hereinafter be referred to as an EDC block.

As shown in the figure, the EDC block includes 12 rows with 172 bytes per row. The first row of the EDC block includes an EDC with 4 bytes, the first to the twelfth rows include main data with 2048 bytes (main data #0 - main data #2047), and the twelfth row includes the data ID with 4 bytes, the IED with 2 bytes, and the RSV with 6 bytes.

[0021]

In step 204 (S204), the EDC block is given a scramble process, which takes Exclusive OR operation with the element of a pseudo-random number sequence.

In step 206 (S206), 16 scrambled EDC blocks are collected and parity codes PO (external codes) with 16 bytes are added in the row directions of 16 data units 1 (i.e., to 12 row $\times$ 16 bytes = 192 bytes), respectively.

[0022]

Furthermore, parity codes PI (internal codes) with 10 bytes are added in the column direction of the data block with 172 bytes

$\times (192 + 16)$ rows (16 EDC blocks and parity codes PO) obtained in the aforementioned process.

Thus, the parity codes PO and PI added to the EDC blocks are also referred to as product codes.

[0023]

As described above, the EDC block expanded to 182 bytes $\times$ 13 rows is also referred to as a sector. 16 sectors form an ECC block with $(172 + 10)$ bytes $\times$ $(192 + 16)$ rows shown in Figure 3. This ECC block is the minimum unit of recording and regeneration with respect to a DVD.

[0024]

[Encoding Process]

The encoding process of adding the parity codes PO and PI to the EDC block will be described in further detail.

In the DVD, Reed-Solomon codes defined on the Galois field $GF(2^8)$ are employed as error-correcting codes (ECCs).

For the Reed-Solomon codes, a sequence of data bits is segmented into symbols 8 bits (1 byte) by 8 bits and a predetermined calculation procedure are performed for each symbol to generate a parity code. The generated parity codes are added to the data sequence. The data sequence added with parity codes are referred to as a code word.

[0025]

The calculation procedure is realized as division employing a polynomial. For instance, if 8-bit data 00000010 is taken to be a primitive root α , polynomials of the tenth degree and the sixteenth degree for parity codes PI and PO are expressed with the following Equations 1-1 and 1-2 and are also referred to as

generator polynomials.

Data sequence is multiplied by X^{10} . The result is divided by the generator polynomial shown in the following equation 1-1. The remainder is the parity code PI. Similarly, data sequence is multiplied by X^{16} . The result is divided by the generator polynomial shown in the following equation 1-2. The remainder is the parity code PO.

[0026]

[Equation 1]

$$PI: (x - \alpha^0) \cdot (x - \alpha^1) \cdots (x - \alpha^9) \quad (1-1)$$

$$PO: (x - \alpha^0) \cdot (x - \alpha^1) \cdots (x - \alpha^{15}) \quad (1-2)$$

[0027]

[Error Detection]

As evident in the encoding process described up to here, in the code word (ECC block) the remainder from the division is added to the ECC block as a parity code.

On the Galois field $GF(2^8)$, since addition and subtraction (XOR) are the same, the addition of the remainder from the division is equal to subtracting the remainder from the data sequence. Therefore, if the code word is divided by the polynomial (Equations 1-1 and 1-2), it can be divided without a remainder.

[0028]

Therefore, when the code word cannot be divided by the polynomial without a remainder, it can be judged that errors resulting from noise and the like has occurred in the code word. Thus, whether or not the code word can be divided by the polynomial without a remainder is the sufficient condition for checking whether or not there is an error in the code word.

[0029]

[Syndrome]

Actually, to obtain a calculated value needed for not only error detection but also error correction to be described later, the value of syndrome is calculated which is equivalent to checking that a code can be divided by the generator polynomial without a remainder.

[0030]

The value of the syndrome is calculated by substituting the root of the generator polynomial for a code word F . If the code word F can be divided by the parity-code PI generator polynomial without a remainder ($F(\alpha^0) = F(\alpha^1) = \dots = F(\alpha^9) = 0$), it means that there is no error in the code word F .

If the code word F can be divided by the parity-code PO generator polynomial without a remainder ($F(\alpha^0) = F(\alpha^1) = \dots = F(\alpha^{15}) = 0$), it means that there is no error in the code word F .

[0031]

[Error Location]

The value of the error location is counted so that, in the ECC block shown in Figure 3, the value of the location of a symbol (byte) away from the parity code PI is increased with the last symbol of the parity code PI as zero. Thus the value of the error location indicates the location of a symbol where an error has occurred.

Note that when a symbol is erroneous, the size of the error is expressed by the result of Exclusive OR between the value of the erroneous symbol and the value of a correct symbol. If there is an error at location n (error location = n), for example, the value of the error is 11111111 when the value of

the erroneous symbol at the location n is 01011010 and the value of the correct symbol is 10100101.

[0032]

[Error Locator Equation and the Range of Definition]

With the above-mentioned syndrome, an equation including information about an error location can be derived by a predetermined calculation procedure. This polynomial is called an error locator polynomial and defined like the following Equation 2.

[0033]

[Equation 2]

$$(1 - \alpha^{m_1}x) \cdot (1 - \alpha^{m_2}x) \cdots (1 - \alpha^{m_k}x) \\ = 1 + P_1x + P_2x^2 + \cdots + P_kx^k \quad (2)$$

in which

$m_1 - m_k$ = error location,

α = primitive root.

[0034]

However, practically the following Equation 3-1, which is reciprocity of Equation 2, is often employed and the following Equation 3-2 obtained from Equation 3-1 is employed as the error locator equation.

[0035]

[Equation 3]

$$(z - \alpha^{m_1}) \cdot (z - \alpha^{m_2}) \cdots (z - \alpha^{m_k}) \\ = z^k + P_1 \cdot z^{k-1} + \cdots + P_k \quad (3-1)$$

$$z^k + P_1 \cdot z^{k-1} + \dots + P_k = 0 \quad (3-2)$$

[0036]

When the error locator polynomial is solvable, the roots α^{m_1} , α^{m_2} , ..., and α^{m_k} are obtained and the logarithmic values of these roots represent error locations $m_1 - m_k$, respectively.

The roots of the error locator polynomial are obtained as 8-bit symbols and the logarithmic values become values within a range of 0 - 254.

On the other hand, the values of the parity codes PI and PO are within a range of 0 - 181 (182 - 1) and 0 - 207 (208 - 1), respectively. Thus, the ranges of definition of the error equations for the parity codes PI and PO become $\alpha^0 - \alpha^{181}$ and $\alpha^0 - \alpha^{207}$, respectively.

[0037]

[Chain Search Method]

Figure 4 is a diagram showing the chain search method.

In the correction of a code word including many errors, a method called a chain search is generally employed.

The chain search method sequentially substitutes the elements in the definition range of the error locator equation into the equation and checks whether or not each value, obtained as the result of the substitution, becomes zero, thereby solving the error locator equation.

[0038]

[Direction of the Chain Search in the Present Invention]

The value that is substituted into the error locator equation

is obtained by sequentially multiplying a substituted value by a primitive root α , with a numeric value 1 (α to the 0 power (α^0)) as an initial value. Therefore, as shown in Figure 4, a conventional chain search is generally made from the end of the parity code toward the head of data bits.

In contrast to this, in the error-correcting device 1 according to the present invention the chain search is performed from the head of data bits toward the last of the parity code, as shown in Figure 4. The chain search is also characterized in that a value to be substituted into the error locator polynomial is obtained by sequentially dividing the primitive root α , with the initial value as α^{181} or α^{207} .

[0039]

[Syndrome and its Correction Difference]

For instance, consider the case where three errors e_{m1} , e_{m2} , and e_{m3} are added to code words c_n , c_{n-1} , ..., and c_0 and words d_n , d_{n-1} , ..., and d_0 are received, as shown in the following Equation 4.

[0040]

[Equation 4]

$$\begin{aligned} & (c_n, c_{n-1} \dots c_{m1}, \cdot c_{m2}, \cdot c_{m3}, \dots, c_0) \\ & + (0, \dots 0 \dots e_{m1}, \cdot e_{m2}, \cdot e_{m3}, \dots, 0) \\ & = (d_n, d_{n-1} \dots d_{m1}, \cdot d_{m2}, \cdot d_{m3}, \dots, d_0) \end{aligned} \quad (4)$$

[0041]

Here, when $D(x) = d_n x^n + d_{n-1} x^{n-1} + \dots + d_0$ and $C(x) = c_n x^n + c_{n-1} x^{n-1} + \dots + c_0$, the values of the syndromes $S_0 - S_k$ are given by the following Equation 5.

[0042]

[Equation 5]

$$S_0 = D(\alpha^0) = \sum d_i = \sum c_i + e_{m1} + e_{m2} + e_{m3}$$

$$S_0 = D(\alpha^1) = \sum d_i \alpha^i$$

$$= \sum c_i \alpha^i$$

$$+ e_{m1} \alpha^{m1} + e_{m2} \alpha^{m2} + e_{m3} \alpha^{m3}$$

.

.

$$S_k = D(\alpha^k) = \sum d_i \alpha^{ki}$$

$$= \sum c_i \alpha^{ki}$$

$$+ e_{m1} \alpha^{km1} + e_{m2} \alpha^{km2} + e_{m3} \alpha^{km3}$$

(5)

where $i = 1 \sim n$.

[0043]

Here, in Equation 5 the value of $\sum c_i \sim \sum c_i \alpha^k$ becomes 0 by the definition of the code word, so the values of the syndromes $S_0 \sim S_k$ are obtained as shown in the following Equation 6.

[0044]

[Equation 6]

$$S_0 = e_{m1} + e_{m2} + e_{m3}$$

$$S_0 = e_{m1} \alpha^{m1} + e_{m2} \alpha^{m2} + e_{m3} \alpha^{m3}$$

.

.

$$S_k = e_{m1} \alpha^{km1} + e_{m2} \alpha^{km2} + e_{m3} \alpha^{km3}$$

(6)

[0045]

For data sequence $(0, 0, \dots, e_m, \dots 0)$ where, each time a set of an error position and an error value are found in the decoding process, a value includes the symbol of the error value only at the error position and the remaining symbols are 0, if a

syndrome with respect to the data sequence is defined as a correction difference, correction differences S_0' , S_1' , ..., S_k' are obtained as shown in the following Equations 7-1 ~ 7-k. The value of the syndrome equals the sum of all the correction differences ($S_i = \sum S_i \alpha^{mj}$ where Σ represents the sum total with respect to j).

[0046]

[Equation 7]

$$\begin{aligned} S_0^{m1'} &= e_{m1} \\ S_1^{m1'} &= e_{m1} \alpha^{m1} \\ &\vdots \\ S_k^{m1'} &= e_{m1} \alpha^{km1} \end{aligned} \quad (7-1)$$

$$\begin{aligned} S_0^{m2'} &= e_{m2} \\ S_1^{m2'} &= e_{m2} \alpha^{m2} \\ &\vdots \\ S_k^{m2'} &= e_{m2} \alpha^{km2} \end{aligned} \quad (7-2)$$

$$\begin{aligned} S_0^{mk'} &= e_{mk} \\ S_1^{mk'} &= e_{mk} \alpha^{mk} \\ &\vdots \\ S_k^{mk'} &= e_{mk} \alpha_{kmk} \end{aligned} \quad (7-k)$$

[0047]

[EDC and Correction Difference]

The correction difference with respect to the EDC can be defined in the same way as the above-mentioned correction difference with respect to the syndrome.

However, when the error-correcting process employing the parity code PO is performed, the positions of objected to be evaluated by the chain search are dispersed on a code sequence defining

the EDC, so it is difficult to calculate the correction difference for the EDC.

[0048]

[Correction Difference for the EDC in the Present Invention
(Standard Correction Difference)]

In contrast to this, in the calculation of the correction difference for the EDC in the present invention (hereinafter also stated as an EDC correction difference), the value (standard correction difference) of the EDC is derived with respect to an error whose value in the domain of an EDC to be evaluated is 00000001 (01h) and whose value in the domain of another EDC is 0. The value is multiplied by the size of the error, whereby the correction difference is calculated.

[0049]

Here, the EDC correction-difference calculating method in the present invention will be described with respect to the case where an error with an error value of e occurs only at the row 1 of the rows 0 - 11 of sector N (see the Figure 2).

In the EDC correction difference calculation according to the present invention, when an object of evaluation is in the row 0, no error has occurred in the row 0. Therefore, at the boundary between sector N and sector N-1, the EDC correction difference for the sector N-1 is saved and the correction difference for the sector N is loaded.

[0050]

When the object of evaluation is in the row 1, an error has occurred in the row 1. Therefore, the correction difference is calculated from the standard correction difference #1 for the EDC and the error value e . The calculated value is added to a work register for correction difference calculation (this

register does not always correspond to the following description).

When the object of evaluation is between the rows 2 and 10, there is no error in the rows and therefore the process of adding a calculated correction difference to the work register is not performed.

When the object of evaluation is in row 11, it is at the boundary between sector N and sector N+1. Therefore, the numeric value within the work register is stored in the correction difference register of the sector N (this register does not always correspond to the following description).

[0051]

[Error-Correcting Device 1]

Figure 5 is a diagram showing the construction of the error-correcting device 1 according to the present invention.

The error-correcting device 1 according to the present invention will hereinafter be described in reference to Figure 5.

As shown in the figure, the error-correcting device 1 is constructed of an error-correcting block 10, a miscorrection detecting block 12, and a control section 20.

[0052]

The error-correcting block 10 is constructed of a storage 100, a read/write control section 102, a syndrome/EDC calculating section 104, an error-locator-polynomial/error-evaluator-polynomial deriving section 106, an erasure error location control section 108, a location counter 110, a chain search section 112, an error-value calculating section 114, and a correction control

section 116.

The miscorrection detecting block 12 is constructed of EDC correction difference calculating sections 14a and 14b, an EDC standard correction difference calculating section 16, a syndrome correction difference calculating section 18, comparing circuits 120 and 124, a miscorrection detection section 122, and selectors 126 and 128.

With these components, the error-correcting device 1 performs the error-correcting process on the ECC block (Figure 3) read out from a DVD and supplied in sequence to the error-correcting device 1 by a DVD player (not shown), for example.

[0053]

[Control Section 20]

The control section 20 controls the calculating operation of each component of the error-correcting device 1.

[0054]

[Error-Correcting Block 10]

The components of the error-correcting block 10 will hereinafter be described.

[0055]

[Storage 100 and Read/Write Control Section 102]

In the error-correcting block 10, the storage 100 operates according to control of the read/write control section 102. This storage 100 stores ECC blocks (Figure 3) supplied in sequence, reads out the stored ECC blocks in the order of the 0 to the 207th rows in the unit of a row or in order of the 181st to the 0 columns in the unit of a column, and supplied the read

data to the read/write control section 102.

Also, the storage 100 corrects an error occurring in the ECC block and supplies the corrected data to an information processing unit and a storage (not shown) in accordance with the correction data supplied from the read/write control section 102.

[0056]

In addition to controlling the operation of the storage 100, the read/write control section 102 outputs the read data input from the storage 100 to the syndrome/EDC calculating section 104 and outputs the corrected data input from the error-correcting section 116 to the storage 100, thereby correcting the data where an error has occurred.

[0057]

[Syndrome/EDC Calculating Section 104]

The syndrome/EDC calculating section 104 calculates the value of a syndrome and the value of error-detecting codes from the read data input from the read/write control section 102, each time 12 rows of data are input from the storage 100 in the unit of a row.

The syndrome/EDC calculating section 104 outputs the calculated value of the syndrome to both the error-locator-polynomial/error-evaluator-polynomial deriving section 106 of the error-correcting block 10 and the comparing circuit 124 of the miscorrection detecting block 12 and also outputs the calculated value of the error-correcting codes to the comparing circuit 120 of the miscorrection detecting block 12.

[0058]

[Error-Locator-Polynomial/Error-Evaluator-Polynomial Deriving
Section 106]

The error-locator-polynomial/error-evaluator-polynomial
deriving section 106 derives an error locator polynomial and an
error evaluation polynomial shown in the following Equations
8-1 and 8-2 from the values of the syndromes input from the
syndrome/EDC calculating section 104. Furthermore, the
error-locator-polynomial/error-evaluator-polynomial deriving
section 106 generates uncorrectable data.

[0059]

[Equation 8]

$$\sigma(x) = (x - \alpha^{m1}) \cdots (x - \alpha^{mk}) \quad (8-1)$$

$$\omega(x) = \sum e_{mi} \prod_{j \neq i} (x - \alpha^{mj}) \quad (8-2)$$

[0060]

Also, the error-locator-polynomial/error-evaluator-polynomial
deriving section 106 outputs the derived error locator
polynomial to the chain search section 112, outputs the
generated uncorrectable data to the erasure error location
control section 108, and outputs the error locator polynomial
and the error evaluation polynomial to the error-value
calculating section 114.

[0061]

Note that erasure-location data indicates the location of a
symbol where the occurrence of the error is known but the size
of the error is unclear. When the parity code PI or the parity
code PO is processed or when an error cannot be calculated at a
certain row or column, the row or column number indicates the
erasure location with respect to the parity code PO or PI.

Also, the uncorrectable data indicates that an error locator polynomial and an error evaluation polynomial cannot be calculated at a certain row or column.

In addition, the error-locator-polynomial/error-evaluator-polynomial deriving section 106 outputs to the location counter 110 a start signal indicating that calculations of the two polynomials have ended.

[0062]

[Erasure error Location Control Section 108]

The erasure error location control section 108 generates erasure-location data from the uncorrectable data input from the error-locator-polynomial/error-evaluator-polynomial deriving section 106 and outputs the generated data to the error-locator-polynomial/error-evaluator-polynomial deriving section 106.

[0063]

[Location Counter 110]

The location counter 110 generates a clock signal needed for sequentially calculating $\sigma(\alpha_i)$ in the chain search in accordance with the start signal input from the error-locator-polynomial/error-evaluator-polynomial deriving section 106 and holds the number of generations as i (which assumes a value of $181 - 0$ in the case of the parity code PI and a value of $207 - 0$ in the case of the parity code PO).

The location counter 110 outputs the held i to the EDC standard correction difference calculating section 16 of the miscorrection detecting block 12, the chain search section 112 of the error-correcting block 10, the selector 126 of the miscorrection detecting block 12, and the correction control section 116 of the error-correcting block 10. In the case of

the parity code PO, the location counter 110 also outputs to the EDC standard correction difference calculating section 16 a signal that requests updating of the EDC standard correction difference when processing has been finished up to the last of a column.

[0064]

[Chain Search Section 112]

Based on the error locator polynomial input from the error-locator-polynomial/error-evaluator-polynomial deriving section 106 and the clock signal and substitution data input from the location counter 110, the chain search section 112 calculates $\sigma(\alpha^i)$ in order of $i = 181, 180, \dots$ in the case of the parity code PI and in order of $i = 207, 206, \dots$ in the case of the parity code PO and detects root α^i that satisfies $\sigma(\alpha^i) = 0$. The detected data (α^i) indicating an error location is output to the error-value calculating section 114 and the correction control section 116.

Also, the chain search section 112 outputs the detected data to the selectors 126 and 128 and controls data selection of each of the selectors 126 and 128.

[0065]

[Error-Value Calculating Section 114]

The error-value calculating section 114 calculates error value from the detected data input from the chain search section 112 and outputs the calculated value data indicating error value to the correction control section 16.

[0066]

[Correction Control Section 116]

The correction control section 116 generates correction data from the location data (α^1) input from the location counter 110, the location data input from the chain search section 112, the size data input from the error-value calculating section 114, and the error detection/correction data input from the miscorrection detecting section 122 of the error-detecting/correction block 12 and outputs the generated correction data to the read/write control section 102.

[0067]

[Miscorrection Detecting block 12]

The components of the miscorrection detecting block 12 will hereinafter be described.

[0068]

[EDC Standard Correction Difference Calculating Section 16]

Figure 6 is a diagram showing the construction of the EDC standard correction difference calculating section 16 shown in Figure 5. The EDC standard correction difference calculating section 16 calculates an EDC standard correction difference with the updated data input from the location counter 110, by the components shown in Figure 16 and outputs the EDC standard correction difference to the EDC correction difference calculating sections 14a and 14b.

[0069]

[Selector 126]

The selector 126 operates according to control of the chain

search section 112 of the error-correcting block 10. When the detected data indicates that there is no error, the selector 126 selects a numeric value of 0. In the case other than that, i.e., in the case where the detected data indicates that there is an error, the selector 126 selects the root input from the location counter 110 and outputs the selected root to the EDC correction difference calculating sections 14a and 14b and the syndrome correction difference calculating section 18.

[0070]

[Selector 128]

The selector 128 operates according to control of the chain search section 112 of the error-correcting block 10. When the detected data indicates that there is no error, the selector 128 selects a numeric value of 0. In the case other than that, i.e., in the case where the detected data indicates that there is an error, the selector 128 selects the size data input from the error-value calculating section 114 and outputs the selected size data to the EDC correction difference calculating sections 14a and 14b and the syndrome correction difference calculating section 18.

[0071]

[EDC Correction Difference Calculating Section 14a for a Direction of PO]

Figure 7 is a diagram showing the construction of the first EDC correction difference calculating section 14a shown in Figure 5.

Figure 8 is a diagram showing the direction of calculation in which an EDC correction difference is calculated by the first EDC correction difference calculating section 14a shown in Figure 5.

The EDC correction difference calculating section 14a makes calculations in the direction indicated by an arrow in Figure 8 with respect to the EDC block shown in Figure 2, with the standard correction difference data input from the EDC standard correction difference calculating section 16 and the location data, size data, or numeric value 0 input from the selectors 126 and 128, by the components shown in Figure 7 and calculates an EDC standard correction difference.

[0072]

[EDC Correction Difference Calculating Section 14b for the direction of PI]

Figure 9 is a diagram showing the construction of the second EDC correction difference calculating section 14b shown in Figure 5.

The EDC correction difference calculating section 14b makes calculations in the direction of the parity code PI (row direction) with respect to the EDC block shown in Figure 2, with the standard correction difference data input from the EDC standard correction difference calculating section 16 and the location data, size data, or numeric value 0 input from the selectors 126 and 128, by the components shown in Figure 9 and calculates an EDC standard correction difference.

[0073]

[Syndrome Correction Difference Calculating Section 18]

Figure 10 is a diagram showing the syndrome correction difference calculating section 18 shown in Figure 5.

The syndrome correction difference calculating section 18 calculates a correction difference with the location data, size data, or numeric value 0 input from the selectors 126 and 128, by the components shown in Figure 10 and outputs the calculated

correction difference to the comparing circuit 124.

[0074]

[Comparing Circuit 120]

The comparing circuit 120 compares the EDC input from the syndrome/EDC calculating section 104 with the correction difference input from the EDC correction difference calculating sections 14a and 14b, by the Exclusive OR therebetween and outputs the result of the comparison to the miscorrection detecting section 122.

[0075]

[Comparing Circuit 124]

The comparing circuit 124 compares the value of the syndrome input from the syndrome/EDC calculating section 104 with the correction difference input from the syndrome correction difference calculating section 18, by the Exclusive OR therebetween and outputs the result of the comparison to the miscorrection detecting section 122.

[0076]

[Miscorrection detecting Section 122]

The miscorrection detecting section 122 detects an error in correction (miscorrection) by using the comparison results input from the comparing circuits 120 and 124, generates miscorrection data indicating that the value of the syndrome and the value of the correction difference do not match each other, and outputs the miscorrection data to the correction control section 116.

[0077]

[Processing in the Error-Correcting Device 1 in the Direction of PI (horizontal direction)]

The processing in the error-correcting device 1 in the direction of PI will hereinafter be described in reference to Figures 11 - 19.

Figure 11 is a diagram showing the error-correcting device 1 performs processing in the direction of PI.

Figure 12 is a diagram showing the processing in step 3 in the error-correcting device 1 shown in Figure 11.

Figure 13 is a diagram showing the processing in step 5 in the error-correcting device 1 shown in Figure 11.

Figure 14 is a diagram showing the processing in step 6 in the error-correcting device 1 shown in Figure 11.

Figures 15 - 17 are diagrams showing the processing in steps 7 and 8 in the error-correcting device 1 shown in Figure 11.

Figure 18 is a diagram showing the processing in step 9 in the error-correcting device 1 shown in Figure 11.

Figure 19 is a diagram showing the processing in step 11 in the error-correcting device 1 shown in Figure 11.

[0078]

As shown in Figure 11, in step 0 the read/write control section 102 of the error-correcting device 1 (Figure 5) substitutes an initial value (0) for a variable i and goes to step 1.

[0079]

In step 1, the read/write control section 102 clears the register for storing the value of the EDC syndrome and goes to step 2.

[0080]

In step 2, the syndrome correction difference calculating section 18 performs an initialization process and returns to the initial state.

[0081]

In step 3, the syndrome/EDC calculating section 104 reads out the i th data of the ECC block (Figure 3) from the read/write control section 102, then calculates the value of the syndrome (error-correcting code), and goes to step 4, as shown in Figure 12.

[0082]

In step 4, the syndrome/EDC calculating section 104 judges whether or not the value of the syndrome calculated in step 3 is zero. When it is not zero, the section 104 goes to step 5. In the case other than that, the syndrome/EDC calculating section 104 adds a numeric value of 1 to the variable i (i.e., $i = i + 1$) and returns back to step 2.

[0083]

In step 5, as shown in Figure 13 the error-locator-polynomial/error-evaluator-polynomial deriving section 106 calculates an error polynomial from the value of the syndrome input from the syndrome/EDC calculating section 104 and substitutes a numeric value of α^{181} ($L = \alpha^{181}$) for a variable L (α^i used in the chain search).

[0084]

In step 6, as shown in Figure 14, the initial value of the value L given by the location counter and the chain search section 112 performs the chain search with the initial value of the variable L and the clock signal given by the location counter 110 in order to judge whether or not the variable L, i.e., α^i is a root of the error locator polynomial. When it is the root of the error locator polynomial, the error-value calculating section 114 calculates error value at that location and goes to step 7.

[0085]

In step 7, when the variable L is the root of the error locator polynomial, the selector 128 selects the error value input from the error-value calculating section 114 and outputs the selected error value to the syndrome correction difference calculating section 18 and the EDC correction difference calculating section 14b. When it is not the root, the selector 128 outputs zero to the syndrome correction difference calculating section 18 and the EDC correction difference calculating section 14b.

The syndrome correction difference calculating section 18 and the EDC correction difference calculating section 14b calculate the contribution to respective correction differences by using the input values and goes to step 8a.

[0086]

In step 8, as shown in Figure 17, the selector 128 selects the error value input from the error-value calculating section 114 and outputs the selected error value to the EDC correction difference calculating section 14b, which in turn calculates a correction difference.

If the chain search has come to the end of the i th row, the EDC

correction difference calculating section 14b goes to step 9. In the case other than that, the EDC correction difference calculating section 14b updates the value of the variable L to $L \times \alpha^{-1}$ and goes to step 6 (step 8a).

[0087]

In step 9, as shown in Figure 18, the comparing circuit 124 compares the value of the syndrome input from the syndrome/EDC calculating section 104 with the value of the correction difference input from the syndrome correction difference calculating section 18.

When the value of the syndrome and the correction difference match each other, the miscorrection detecting circuit 122 outputs the miscorrection detecting data to the correction control section 116 as a value indicating that a miscorrection has not occurred. In the case other than that, the miscorrection detecting circuit 122 outputs the miscorrection detecting data to the correction control section 116 as a value indicating that a miscorrection has occurred.

When the miscorrection detecting data indicates that a miscorrection has occurred, the correction control section 116 makes an error correction on the buffer (not shown) of the read/write control section 102. In the case other than that, the correction control section 116 makes no error correction.

[0088]

In step 10, the read/write control section 102 judges whether or not it has processed the last row of the EDC block (Figure 2). When it has processed the last row, it goes to step 11. In the case other than that, it goes to step 3.

[0089]

In step 11, the comparing circuit 120 compares the value of the

EDC input from the syndrome/EDC calculating section 104 with the value of the correction difference input from the EDC correction difference calculating section 14b. When the value of the EDC and the value of the correction difference match each other, miscorrection detecting data is output to the correction control section 116 as a value indicating correction is impossible. In the case other than that, the comparing circuit 120 outputs data, which is detected and corrected, to the correction control section 116 as a value indicating that correction is possible and goes to step 12.

[0090]

In step 12, the comparing circuit 120 judges whether or not it has processed the last row of the EDC block (Figure 2). When it has processed the last row, it ends processing. In the case other than that, it returns to step 1.

[0091]

As described above, in the error-correcting device 1 the chain search for error correction in the direction of PI is executed from the head of a row.

That is, in the chain search in the PI direction, numeric values of α^{181} , α^{180} , ..., and α^0 are first substituted into the error locator polynomial to judge whether or not they are roots of the polynomial. When they are the roots, the error value is calculated.

[0092]

Also, the syndrome/EDC calculating section 104 processes the data of the ECC block from the head of the column. In addition, the value of the correction difference can be calculated by calculating the sum of the influences of errors on a syndrome.

Therefore, when in the chain search an error is found and the error value is calculated, the value is input to the syndrome correction difference calculating section 18 and the EDC correction difference calculating section 14b. In the case other than that, a numeric value of 0 is input. With this, the correction difference for the syndrome can be calculated.

Therefore, the processing in the error-correcting device 1 in the PI direction can be performed without a standard correction difference.

[0093]

[Processing in the Error-Correcting Device 1 in the PO (vertical) Direction]

For instance, the correction difference of the syndrome in the first EDC block (Figure 2), in which an error 0x94 occurs at the third row of the first EDC block and the 179th column of the ECC block, is given by $e(x) \bmod g(x)$, with $e(x)$ shown in the following Equation 10. To calculate this, it is necessary to calculate $e(x) \bmod g(x)$ from $e(x)$ shown in the following Equation 10.

[0094]

[Equation 10]

$$e(x) = (x^7 + x^4 + x^2) * x^{((2*172+169)*8)} \quad (10)$$

[0095]

For this reason, standard correction difference N_j is first defined as shown in the following Equation 11.

In the case where an error with a size of 0x01 has occurred at the 170th of the EDC block, the standard correction difference N_j indicates the EDC's contribution to the syndrome.

[0096]

[Equation 11]

$$N_{12} = X^{((169+172*11)*8)} \bmod g(x)$$

$$N_{11} = X^{((169+172*10)*8)} \bmod g(x)$$

$$N_3 = X^{((169+172*2)*8)} \bmod g(x)$$

$$N_2 = X^{((169+172*1)*8)} \bmod g(x)$$

$$N_1 = X^{((169+172*0)*8)} \bmod g(x)$$

where N_x represents the x th standard correction difference of the EDC. (11)

[0097]

The chain search section 112 and the location counter 110 of Figure 5 output error location 198 and error value 0x94 in the process of the 179th column.

The error 0x94, which has occurred at the third row of the first EDC block and the 179th column of the ECC block, is located at the 198th row of the ECC block because $12 * 15 + 16 - 1 = 198$ (each EDC block has 12 rows, the EDC blocks are 16 in total, the row begins at the 0th row, and the parity code P0 has 16 rows). Therefore, it is found that the error position calculated by the error-correcting circuit is at 198 (the 199th row).

It is also found that the error position on the EDC block is at the third row on the first EDC block (Figure 2) of the ECC block (Figure 3) because $(199 - 1) = 12 * 15 + 3$.

Because it is also found that the error value is 0x94, $e(x) = (x^7 + x^4 + x^2) * x^{((2*172+169)*8)}$.

At this time, the correction difference $S(x)$ can be expressed

like the following Equation 12 by using the standard correction difference N_3 shown in Equation 11, because an error has occurred at the third row on the EDC block.

[0098]

[Equation 12]

$$\begin{aligned}
 S(x) &= (x^7 + x^4 + x^2) * x^{((2*172+169)*8)} \bmod g(x) \\
 &= x^7 * x^{((2*172+169)*8)} \bmod g(x) \\
 &\quad + x^4 * x^{((2*172+169)*8)} \bmod g(x) \\
 &\quad + x^2 * x^{((2*172+169)*8)} \bmod g(x) \\
 &= [x^7 * N_3] \bmod g(x) \\
 &\quad + [x^4 * N_3] \bmod g(x) \\
 &\quad + [x^2 * N_3] \bmod g(x)
 \end{aligned} \tag{12}$$

in which

$$\begin{aligned}
 (A + B) \bmod g &= A \bmod g + B \bmod g, \\
 (A * B) \bmod g &= \{B * (A \bmod g)\} \bmod g.
 \end{aligned}$$

[0099]

At this time, a circuit for calculating $[x^7 * N_3] \bmod g(x)$ can easily be constructed by hardware such as a random logic. Or, the equivalent calculation process is easily realizable by software.

[0100]

In the EDC correction difference calculating section 14a shown in Figure 7, if $B_7 = 1$, $B_4 = 1$, $B_2 = 1$, and the others are 0 so as to correspond to the error value 0x94, the output of the adder of the EDC correction difference calculating section 14a becomes the correction difference $S(x)$.

[0101]

The foregoing description is generalized and the first to the twelfth standard correction differences of the EDC block in the i th column are defined like the following Equation 13. When there is an error in the 180th column, the standard correction differences are given by the following Equation 14, which is obtained by substituting $i = 180$ into the following Equation 13.

[0102]

[Equation 13]

$$N_{12} = X^{((i-10+172*11)*8)} \bmod g(x)$$

$$N_{11} = X^{((i-10+172*10)*8)} \bmod g(x)$$

.

.

$$N_3 = X^{((i-10+172*2)*8)} \bmod g(x)$$

$$N_2 = X^{((i-10+172*1)*8)} \bmod g(x)$$

$$N_1 = X^{((i-10+172*0)*8)} \bmod g(x)$$

(13)

[0103]

[Equation 14]

$$N_{11} = X^{((170+172*10)*8)} \bmod g(x)$$

.

.

$$N_3 = X^{((170+172*2)*8)} \bmod g(x)$$

$$N_2 = X^{((170+172*1)*8)} \bmod g(x)$$

$$N_1 = X^{((170+172*0)*8)} \bmod g(x)$$

(14)

[0104]

From Equations 13 and 14, a new standard correction difference N' , can be expressed like the following Equation 15. With this value, similar calculations can be made on an error at the

181st column.

Finally, since a register for storing 16 correction differences and another register for storing en route result are required, an EDC correction difference calculating section 14a is configured as shown in Figure 7.

[0105]

[Equation 15]

$$N'_j = (x^8 * N_j) \bmod g(x) \quad (15)$$

[0106]

The processing in the error-correcting device 1 in the direction of P0 will hereinafter be described in reference to Figures 20 - 30.

Figure 20 is a diagram showing how the error-correcting device 1 (Figure 5) performs processing in the direction of P0.

Figure 21 is a diagram showing the processing in step 100 in the EDC standard correction difference calculating section 16 (Figure 6) of the error-correcting device 1 (Figure 5) shown in Figure 20.

Figure 22 is a diagram showing the processing in step 101 of the error-correcting device 1 shown in Figure 20.

Figure 23 is a diagram showing the processing in step 103 of the error-correcting device 1 shown in Figure 20.

Figure 24 is a diagram showing the processing in step 104 of the error-correcting device 1 shown in Figure 20.

Figure 25 is a diagram showing the processing in step 107 of the error-correcting device 1 shown in Figure 20.

Figure 26 is a diagram showing the processing in step 107 of the EDC correction difference calculating section 14a (Figure 7) of the error-correcting device 1 shown in Figure 20.

Figure 27 is a diagram showing the processing in step 108 of the EDC correction difference calculating section 14a (Figure 7) of the error-correcting device 1 shown in Figure 20.

Figure 28 is a diagram showing the processing in step 110 of the error-correcting device 1 shown in Figure 20.

Figure 29 is a diagram showing the processing in step 111 of the EDC standard correction difference calculating section 16 (Figure 6) of the error-correcting device 1 shown in Figure 20.

Figure 30 is a diagram showing the processing in step 112 of the error-correcting device 1 shown in Figure 20.

[0107]

As shown in Figure 20, in step 100, the EDC standard correction difference calculating section 106 substitutes an initial value 10 for the variable i (the reason why the initial value of the variable i becomes 10 is that the parity code PI is not included in the EDC block) in order to calculate the initial standard correction difference N_j ($j = 1, \dots, \text{and } 12$) shown in the following Equation 16, then clears the register (not shown) within the syndrome correction difference calculating section 14a that stores the syndrome of the EDC, initializes the syndrome/EDC calculating section 104 (syndrome generating circuit for error-correcting codes), and goes to step 101.

[0108]

[Equation 16]

$$N_{12} = X^{((i-10+172 \cdot 11) \cdot 8)} \bmod g(x)$$

$$N_{11} = X^{((i-10+172 \cdot 10) \cdot 8)} \bmod g(x)$$

$$\begin{aligned}
 N_3 &= X^{((i-10+172*2)*8)} \bmod g(x) \\
 N_2 &= X^{((i-10+172*1)*8)} \bmod g(x) \\
 N_1 &= X^{((i-10+172*0)*8)} \bmod g(x)
 \end{aligned}
 \tag{16}$$

[0109]

In step 101, as shown in Figure 22, the syndrome/EDC calculating section 104 (Figure 5) calculates the value of the syndrome of the error-correcting code at the i th column of the EDC block (Figure 2) and goes to step 102.

[0110]

In step 102, when the value of the syndrome calculated in step 101 is not 0, the error-locator-polynomial/error-evaluator-polynomial deriving section 106 goes to step 103. In the case other than that, the error-locator-polynomial/error-evaluator-polynomial deriving section 106 increments the variable i by 1, then updates the value of the standard correction difference by the following Equation 17 and returns to step 101.

[0111]

[Equation 17]

$$N_j = (N_j * x^8) \bmod g(x) \tag{17}$$

[0112]

In step 103, as shown in Figure 23, the error-locator-polynomial/error-evaluator-polynomial deriving section 106 calculates the error locator polynomial from the value of the syndrome calculated in step 101, substitutes α^{207} for the value of the variable L , and goes to step 104.

[0113]

In step 104, as shown in Figure 24, the chain search section 112 performs the chain search and judges whether or not the value of the variable L is the root of the error locator polynomial.

When the value of the variable L is the root of the error locator polynomial, the error-value calculating section 114 calculates error value. In the case other than that, the value of the error value is made 0.

[0114]

In step 105, the syndrome correction difference calculating section 18 calculates a correction difference from the value of the error value calculated in step 104 and goes to step 106.

[0115]

In step 106, the syndrome correction difference calculating section 18 judges which EDC block (Figure 2) in the ECC block (Figure 3) and which row in the EDC block an error has occurred in, from the location of the error calculated by the chain search section 112 and the location counter 110, and goes to step 107.

[0116]

In step 107, as shown in Figures 25 and 26, the EDC correction difference calculating section 14a selects the standard correction difference N_j corresponding to the j th row on the EDC block where in step 106 it has judged that an error has occurred, then makes a calculation of the following Equation 18 by using error value B_k (where $k = 0 - 7$ and $B_k = 1$ or 0), and goes to step 108.

[0117]

[Equation 18]

$$\Sigma B_k * (N_j^k \bmod g(x)) \quad (18)$$

where Σ represents the sum total with respect to $k = 0 - 7$.

[0118]

In step 108, as shown in Figure 27, the EDC correction difference calculating section 14a updates the correction difference for the syndrome corresponding to the EDC block obtained in step 106, by using the result of the calculation (Equation 18) obtained in step 107 and goes to step 109.

[0119]

In step 109, when the chain search has been finished up to the last of the i th column on the EDC block, the chain search section 112 goes to step 110. In the case other than that, the chain search section 112 multiplies the variable L by a numeric value α^{-1} and returns to step 104.

[0120]

In step 110, as shown in Figure 28, the comparing circuit 124 compares the value of the syndrome input from the syndrome/EDC calculating section 104 with the value of the correction difference calculated in step 109 by the syndrome correction difference calculating section 18. When these values match each other, the miscorrection detecting circuit 122 judges that the correction difference of the syndrome is right, then makes an error correction on the buffer (not shown) of the read/write control section 102, and goes to step 111. In the case other than that, the miscorrection detecting circuit 122 sets an uncorrectable flag to data to be detected and corrected and the correction control section 116 goes to step 111 without making

an error correction.

[0121]

In step 111, as shown in Figure 29, the EDC standard correction difference calculating section 16 judges whether or not error location and error value have been calculated for up to the last row of the EDC block (Figure 2). When the EDC standard correction difference calculating section 16 has calculated the last row, it goes to step 112. In the case other than that, it returns to step 101.

[0122]

In step 112, the comparing circuit 120 compares the value of the EDC input from the syndrome/EDC calculating section 104 with the value of the correction difference calculated in steps 101 - 112 by the EDC correction difference calculating section 14a. When these values match each other, the error-detecting/correcting circuit 122 ends the process. In the case other than that, the comparing circuit 120 sets an uncorrectable flag to data to be detected and corrected and ends the process.

[0123]

As has been described above, the error-correcting device according to the present invention can perform the calculations of the location and size of an error and the calculations of the value of the syndrome of the EDC that varies by a miscorrection, at the same time. Therefore, the probability of a miscorrection can be reduced.

In addition, the number of times for reading out data from a buffer can be reduced compared with prior art, so the time needed for the error-correcting process can be reduced.

[0124]

[Advantages of the invention]

According to the error-correcting device and method of the present invention, as described above, overhead operation can be reduced. As a result, the present invention is capable of detecting a miscorrection in a short processing time and correcting errors.

Brief Description of the Drawings:

Figure 1 is a diagram showing the process (S10) of generating data that is recorded on a DVD;

Figure 2 is a diagram showing the data unit 1 (EDC block) shown in Figure 1;

Figure 3 is a diagram showing the ECC block shown in Figure 1;

Figure 4 is a diagram showing a chain search method;

Figure 5 is a diagram showing the construction of an error-correcting device according to the present invention;

Figure 6 is a diagram showing the construction of an EDC standard correction difference calculating section shown in Figure 5;

Figure 7 is a diagram showing the construction of the first EDC correction difference calculating section (14a) shown in Figure 5;

Figure 8 is a diagram showing the direction of calculation in which an EDC correction difference is calculated by the first EDC correction difference calculating section (14a) shown in Figure 5;

Figure 9 is a diagram showing the construction of the second EDC correction difference calculating section (14b) shown in Figure 5;

Figure 10 is a diagram showing the syndrome correction difference calculating section shown in Figure 5;

Figure 11 is a diagram showing the error-correcting device 1 performs processing in the direction of PI;

Figure 12 is a diagram showing the processing in step 3 of the error-correcting device 1 shown in Figure 11;

Figure 13 is a diagram showing the processing in step 5 of the error-correcting device 1 shown in Figure 11;

Figure 14 is a diagram showing the processing in step 6 of the error-correcting device 1 shown in Figure 11;

Figure 15 is a first diagram showing the processing in steps 7 and 8 of the error-correcting device 1 shown in Figure 11;

Figure 16 is a second diagram showing the processing in steps 7 and 8 of the error-correcting device 1 shown in Figure 11;

Figure 17 is a third diagram showing the processing in steps 7 and 8 of the error-correcting device 1 shown in Figure 11;

Figure 18 is a diagram showing the processing in step 9 of the error-correcting device 1 shown in Figure 11;

Figure 19 is a diagram showing the processing in step 11 of the error-correcting device 1 shown in Figure 11;

Figure 20 is a diagram showing how the error-correcting device (Figure 5) performs processing in the direction of PO;

Figure 21 is a diagram showing the processing in step 100 in

the EDC standard correction difference calculating section (Figure 6) of the error-correcting device 1 (Figure 5) shown in Figure 20;

Figure 22 is a diagram showing the processing in step 101 of the error-correcting device 1 shown in Figure 20;

Figure 23 is a diagram showing the processing in step 103 of the error-correcting device 1 shown in Figure 20;

Figure 24 is a diagram showing the processing in step 104 of the error-correcting device 1 shown in Figure 20;

Figure 25 is a diagram showing the processing in step 107 of the error-correcting device 1 shown in Figure 20;

Figure 26 is a diagram showing the processing in step 107 of the first EDC correction difference calculating section (14a in Figure 7) of the error-correcting device 1 shown in Figure 20;

Figure 27 is a diagram showing the processing in step 108 of the first EDC correction difference calculating section (14a in Figure 7) of the error-correcting device 1 shown in Figure 20;

Figure 28 is a diagram showing the processing in step 110 of the error-correcting device 1 shown in Figure 20;

Figure 29 is a diagram showing the processing in step 111 of the EDC standard correction difference calculating section (Figure 6) of the error-correcting device 1 shown in Figure 20; and

Figure 30 is a diagram showing the processing in step 112 of the error-correcting device 1 shown in Figure 20.

[Description of symbols]

1 ... Error-correcting device

10 ... Error-correcting block

100 ... Storage
102 ... Read/write control section
104 ... Syndrome/EDC calculating section
106 ... Error-locator-polynomial/error-evaluator-polynomial
deriving section
108 ... Erasure-error-location control section
110 ... Location counter
112 ... Chain search section
114 ... Error-value calculating section
116 ... Correction control section
12 ... Error-detecting/correcting block
120 ... Comparing circuit
122 ... Miscorrection detecting circuit
124 ... Comparing circuit
126 ... Selector
128 ... Selector
14a, 14b ... EDC correction difference calculating section
16 ... EDC standard correction difference calculating section
18 ... Syndrome correction difference calculating section
20 ... Control section

Claims:

1. An error-correcting device for detecting errors that have occurred in an ECC block to be corrected, the ECC block being generated by respectively adding error-correcting codes (ECCs) to one or more EDC blocks each added with an error-detecting code (EDC), the error-correcting device comprising:

error-value calculating means for calculating values of said errors by using said error-correcting codes included in each data sequence included in said ECC block to each data sequence having said error-correcting codes (ECCs);

EDC syndrome correcting means for correcting a value of syndrome of said error-detecting code in accordance with the calculated value of the error every time said calculated value of the error in said data sequences is obtained; and

error-detecting means for detecting an occurrence of an error in said EDC block, based on the corrected value of the syndrome of said error-detecting code, wherein

said EDC block is constructed of one or more rows and one or more columns;

said error-correcting code is added to each of the rows or columns of said EDC block;

said data sequences include respectively the row or column of said EDC block and the error-correcting code added to the row or column of said EDC block;

said error-value calculating means processes said data sequence one by one and calculates the value of the error;

said EDC syndrome correcting means sequentially corrects a predetermined initial value of the syndrome of the error-detecting code every time said error value is calculated; and

every time the value of the error in said data sequence corresponding to said EDC block is calculated, said error-detecting means detects said occurrence of an error in

said EDC block, based on the corrected value of the syndrome of said error-detecting code.

2. The error-detecting device as set forth in claim 1, wherein when a fixed error occurs at a fixed position on each of said data sequences, said EDC syndrome correcting means uses the syndrome of the error-detecting code as said initial value and sequentially corrects said initial value in accordance with the calculated error value.

3. The error-detecting device as set forth in claim 2, further comprising EDC calculating means for calculating a value of the error-detecting code of each of said EDC blocks; and

wherein, every time a value of an error in said data sequence corresponding to said one EDC block is calculated, said error-detecting means compares said calculated value of the error-detecting code with the corrected value of the syndrome of the error-detecting code and detects the occurrence of an error in said EDC block.