



(51) International Patent Classification:

H03M 13/15 (2006.01) *G11C 16/06* (2006.01)
G06F 12/16 (2006.01) *G11C 29/42* (2006.01)
G11C 16/02 (2006.01)

(21) International Application Number:

PCT/JP2008/067577

(22) International Filing Date:

19 September 2008 (19.09.2008)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

2008-051462 1 March 2008 (01.03.2008) JP

(71) Applicant (for all designated States except US):
Kabushiki Kaisha Toshiba [JP/JP]; 1-1, Shibaura 1-
 chome, Minato-ku, Tokyo, 1058001 (JP).

(72) Inventor; and

(75) Inventor/Applicant (for US only): **YAMAGA, Akira**
 [JP/JP]; c/o Intellectual Property Division, Toshiba Cor-
 poration, 1-1, Shibaura 1-chome, Minato-ku, Tokyo,
 1058001 (JP).

(74) Agent: **SAKAI, Hiroaki**; Sakai International Patent Of-
 fice, Kasumigaseki Building, 2-5, Kasumigaseki 3-chome,
 Chiyoda-ku, Tokyo, 1006020 (JP).

(81) Designated States (unless otherwise indicated, for every
 kind of national protection available): AE, AG, AL, AM,
 AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ,
 CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ,
 EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN,
 HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ,
 LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG,
 MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ,
 OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK,
 SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG,
 US, UZ, VC, VN, ZA, ZM, ZW.

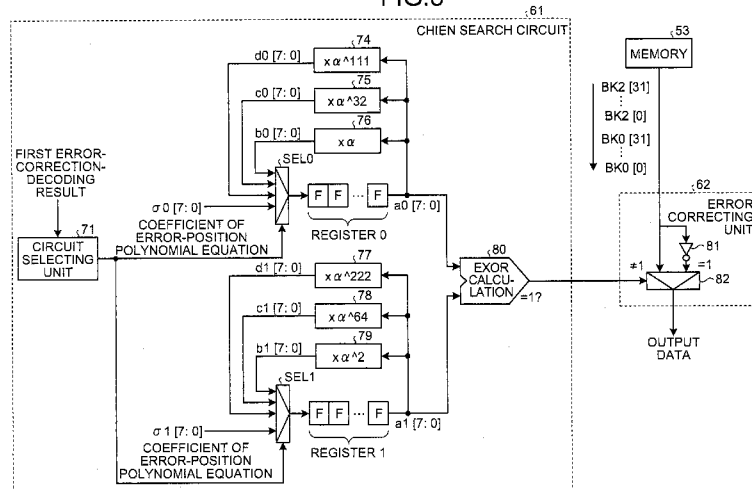
(84) Designated States (unless otherwise indicated, for every
 kind of regional protection available): ARIPO (BW, GH,
 GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,
 ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ,
 TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE,
 ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,
 MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI
 (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR,
 NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) Title: CHIEN SEARCH DEVICE AND CHIEN SEARCH METHOD

FIG.8



(57) Abstract: To provide a Chien search device and a Chien search method capable of performing a Chien search process at a high speed. The Chien search device calculates an error position at the time of correcting an error included in data read from a nonvolatile memory, and includes a first processing unit that performs a search process of an error position in at least one-bit unit to an error-correction area of input data, and a second processing unit that processes at one time plural bits in an non-error-correc- tion-target area of the input data.

DESCRIPTION

CHIEN SEARCH DEVICE AND CHIEN SEARCH METHOD

5 TECHNICAL FIELD

The present invention relates to a Chien search device and a Chien search method, and, more particularly to a Chien search device and a Chien search method capable of making a Chien search to be performed at a higher speed.

10

BACKGROUND ART

Recently, there are widely known semiconductor memory devices such as flash memory that stores information depending on a held charge amount. A multi-value memory technology for storing two bits or more of information by setting a plurality of thresholds of the charge amount is also developed.

In the semiconductor memory devices, electric charge is discharged with passage of time, and thus, if the charge is discharged more than the threshold, an error may occur when information is read. Particularly, a multi-value type of memory device has generally narrow intervals between thresholds, and this causes the probability of occurrence of an error to increase.

25 Japanese Patent Application Laid-Open No. 2007-87464 discloses a storage device, using the semiconductor memory device, which includes an error correction mechanism to correctly recover incorrect information.

A BCH code or an RS (Reed-Solomon) code is often used as the error correction code used to correct errors in the error correction mechanism. In a decoding process using the BCH code or the RS code, it becomes necessary to

30

calculate an error locator. To calculate the error locator, a Chien search method is generally used.

However, a conventional Chien search circuit processes each one bit in an area which is known to have no error.

5 Therefore, when a code length is large, it takes a long time to perform the Chien search.

The present invention has been achieved in view of the above problems, and an object of the present invention is to provide a Chien search device and a Chien search method
10 capable of increasing a speed of performing the Chien search process.

DISCLOSURE OF INVENTION

To solve the problems and achieve the object, a Chien search device that calculates an error position at a time
15 of correcting an error included in data read from a nonvolatile memory according to an aspect of the present invention includes a first processing unit that performs a search process of an error position in at least one-bit unit to an error-correction area of input data; and a
20 second processing unit that processes a plurality of bits at one time in a non-error-correction-target area of the input data.

A Chien search Method that calculates an error position at a time of correcting an error included in data read from
25 a nonvolatile memory according to another aspect of the present invention includes performing a search process of an error position in at least one-bit unit to an error-correction area of input data; and processing a plurality of bits at one time in a non-error-correction-target area
30 of the input data.

BRIEF DESCRIPTION OF DRAWINGS

Fig. 1 is a block diagram of a configuration example of a solid state drive (SSD);

Fig. 2 is a circuit diagram of a configuration example of one block included in a NAND memory chip;

5 Fig. 3 is a data configuration example written into a NAND memory;

Fig. 4-1 is an example of a first error-correction processing result in Fig. 3;

10 Fig. 4-2 is an example of the first error-correction processing result;

Fig. 5-1 is a configuration example of a second error-correction decoding unit;

Fig. 5-2 is a flowchart for explaining an outline of the second error-correction decoding unit;

15 Fig. 6 is a circuit configuration example of a syndrome calculator;

Fig. 7 is a configuration diagram of data of a second error-correction process;

20 Fig. 8 is a configuration example of a Chien search circuit and an error correcting unit of an error-locator-calculating and error-correcting unit;

Fig. 9-1 is a configuration diagram of a $x\alpha$ circuit (a circuit that multiplies by α times);

25 Fig. 9-2 is a configuration diagram of a $x\alpha^{32}$ circuit (a circuit that multiplies by α^{32} times);

Fig. 9-3 is a configuration diagram of a $x\alpha^{111}$ circuit (a circuit that multiplies by α^{111} times);

Fig. 9-4 is a configuration diagram of a $x\alpha^2$ circuit (a circuit that multiplies by α^2 times);

30 Fig. 9-5 is a configuration diagram of a $x\alpha^{64}$ circuit (a circuit that multiplies by α^{64} times);

Fig. 9-6 is a configuration diagram of a $\times \alpha^{222}$ circuit (a circuit that multiplies by α^{222} times);

Fig. 10 is a configuration example of a circuit selecting unit; and

5 Fig. 11 is a flowchart for explaining an operation of the circuit selecting unit.

BEST MODE(S) FOR CARRYING OUT THE INVENTION

The present invention is explained below in detail
10 with reference to the accompanying drawings. Note that the invention is not limited to the following embodiments. In addition, constituent elements in the embodiments include elements that readily occur to those skilled in the art or substantially equivalent elements.

15 Fig. 1 is a schematic block diagram of an SSD1 according to an embodiment of the present invention. Fig. 1 is a schematic block diagram of a configuration of the SSD (Solid State Drive) 1. The SSD1 is connected to a host device (host) 4 via an I/F (interface), and functions as an
20 external memory of the host device 4.

The SSD1 includes a NAND flash memory (hereinafter, "NAND memory") 2 as a nonvolatile memory, and a NAND controller 3 that reads and writes data to and from the NAND memory 2 corresponding to a command from the host
25 device 4. In the present embodiment, while the NAND memory is used as a nonvolatile memory, any storage device can be provided when the nonvolatile memory stores information as nonvolatile information and can generate a change in stored data along a lapse of time.

30 The NAND memory 2 includes plural memory blocks BLKs as a unit of data erasure. A configuration of the memory block BLK is explained with reference to Fig. 2. Fig. 2 is

an equivalent circuit diagram of a configuration of any one of the memory blocks BLKs.

The memory block BLK includes m (m is an integer equal to or larger than 1) NAND strings laid out along a X-direction. Each NAND string includes selection transistors ST1 and ST2, and n (n is an integer equal to or larger than 1) memory cell transistors MTs. The selection transistor ST1 included in each of the m NAND strings has a drain connected to bit lines BL1 to BL m , and has a gate connected in common to a selection gate line SGD. The selection transistor ST2 has a source connected in common to a source line SL, and has a gate connected in common to a selection gate line SGS.

Each memory cell transistor MT is a MOSFET (Metal oxide semiconductor field effect transistor) including a stacked gate configuration formed via a gate insulation film on a semiconductor substrate. The stacked gate configuration includes a charge accumulation layer (a floating gate electrode) formed on the gate insulation film, and a control gate electrode formed via the gate insulation film on the charge accumulation layer. In each NAND string, n memory cell transistors MTs are laid out so that respective current paths are connected in series between the source of the selection transistor ST1 and the drain of the selection transistor ST2. That is, the n memory cell transistors MTs are connected in series in a Y-direction so that adjacent memory cell transistors MTs share a source region or a drain region.

Control gate electrodes are connected to word lines WL1 to WL n respectively starting from the memory cell transistor MT positioned at the nearest side of the drain. Therefore, the drain of the memory cell transistor connected to the word line WL1 is connected to the source

of the selection transistor ST1, and the source of the memory cell transistor connected to the word line WLn is connected to the drain of the selection transistor ST2.

The word lines WL1 to WLn have control gate electrodes
5 of the memory cell transistors MTs connected in common between the NAND strings within the memory block BLK. That is, the control gate electrodes of the memory cell transistors MTs in the same row within the memory block BLK are connected to the same word line WL. Plural memory
10 cells connected to the same word lines WLn are handled as one page, and data writing and data reading are performed for each page.

The bit lines BL1 to BLn have the drain of the selection transistor ST1 connected in common between the
15 memory blocks BLKs. That is, the NAND strings in the same column within plural memory blocks BLKs are connected to the same bit line BL.

A threshold voltage of the memory cell transistor MT changes corresponding to a number of electrons accumulated
20 in the floating gate electrode, and the memory cell transistor MT stores information corresponding to a difference between threshold voltages. The memory cell transistor MT can be configured to store information of one bit, or can be configured to store plural bits (multi-
25 value) information. The embodiment of the present invention is particularly effective to a multi-value memory-cell transistor MT having a narrow distance between threshold values. A control circuit (not shown) including a sense amplifier within the NAND memory 2, and a
30 potential-generating circuit is configured to be able to write data supplied to the NAND memory 2 into the memory cell transistor MT, and output data stored in the memory cell transistor MT to the outside.

In Fig. 1, the NAND controller 3 includes a host I/F 10 that performs an interface process with the host device 4, a NAND I/F 20 that performs an interface process with the NAND memory 2 and controls reading and writing of data, and an error-correction decoding unit 40 that performs error detection and error correction of data read from the NAND memory 2.

An error-correction encoding unit 30 includes a first error-correction-code generator 31, and a second error-correction-code generator 32. The first error-correction-code generator 31 generates a first error-correction code to perform error correction in a predetermined block BK unit to writing data. For the first error-correction code, an error correction code capable of correcting errors of one bit or plural bits can be used. In the present embodiment, the first error-correction-code generator 31 generates a six-bit hamming code (first error-correction code) having a one-bit correction capacity in a block BK unit having a 32-bit data configuration.

The second error-correction-code generator 32 generates a second error-correction code to perform error correction in a unit of plural blocks to data written into the NAND memory 2. For the second error-correction code, an error correction code capable of correcting errors of plural bits can be used. For example, a BCH code (Bosechaudhuriocquenghem code) or an RS code can be used. In the present embodiment, a BCH code of 16 bits having a correction capacity of two bits in four-block (128 bit) unit is generated.

Fig. 3 is a data configuration example written into the NAND memory 2. As shown in Fig. 3, a hamming code (first error-correction code) of six bits having a correction capacity of one bit in a block BK unit having a

data configuration of 32 bits is added. Further, BCH code (second error-correction code) of 16 bits having a correction capacity of two bits in a four-block BK (128 bit) unit is added.

5 In Fig. 1, the error-correction decoding unit 40 includes a first error-correction decoding unit 42 and a second error-correction decoding unit 41. The first error-correction decoding unit 42 performs first error correction using a hamming code in a BK unit to the data read from the
10 NAND memory 2. The first error-correction decoding unit 42 performs error detection to the data after the first error correction, and outputs a result of the error correction to the second error-correction decoding unit 41. The second error-correction decoding unit 41 references a result of
15 the error correction, and performs second error correction to the data after the first error correction, and performs second error correction using a BCH code in a four-block BK unit.

 An outline of the operation of the SSD1 having the
20 above configuration is explained. When the host device 4 supplies data (writing data) requested to be written into the SSD1, the host I/F 10 supplies the received writing data to the error-correction encoding unit 30. The error-correction encoding unit 30 generates first and second
25 error-correction codes to the writing data. The NAND I/F 20 writes the writing data added with the first and second error-correction codes into the NAND memory 2.

 When a data reading request is input from the host device 4, the NAND I/F 20 reads the read-requested data
30 (reading data and the first and second error-correction codes added to this data, and supplies the read result to the error-correction decoding unit 40. The error-correction decoding unit 40 performs the first and second

error corrections to the reading data. The host I/F 10 transfers the error-corrected data to the host device 4. [Second error correcting unit]

A configuration and operation of the second error-correction decoding unit 41 shown in Fig. 1 are explained in detail with reference to Fig. 4 to Fig. 11. Fig. 4-1 is a data configuration example after first error correction, and Fig. 4-2 is an example of a result of the first error-correction process. Fig. 5-1 is a configuration example of the second error-correction decoding unit 41. Fig. 5-2 is a flowchart for explaining an outline of the second error-correction decoding unit. Fig. 6 is a circuit configuration example of a syndrome calculator.

In the present embodiment, the Chien search circuit of the second error-correction decoding unit 41 includes a second processing unit that performs a passing process (skipping process) in one cycle (at one time) a predetermined bit-width component in other areas than an error-correction area, in addition to a first processing unit that performs a search process of an error position in one-bit unit in an error-correction area. With this arrangement, the Chien search circuit performs the Chien search at a high speed.

As shown in Fig. 5-1, the second error-correction decoding unit 41 includes a syndrome calculator 50 that calculates a syndrome value of reading data of which first error correction is finished, an error-position polynomial calculator 51 that generates an error-position polynomial equation based on the syndrome value, an error-locator-calculating and error-correcting unit 52 that specifies an error position by calculating a root of the error-position polynomial equation, and corrects data of the error position, and a memory 53 that temporarily stores data.

As shown in Fig. 5-2, in the second error-correction decoding unit 41, the syndrome calculator 50 first calculates a syndrome value of reading data of which first error correction is finished (Step S1). The error-position polynomial calculator 51 generates an error-position polynomial equation based on the syndrome value (Step S2). The error-locator-calculating and error-correcting unit 52 calculates a root of the error-position polynomial equation (Step S3), and corrects data of the error position (Step S4).

The error-correction decoding operations performed by the first error-correction decoding unit 42 and the second error-correction decoding unit 41 are explained in detail below. When the host device 4 requests to read data from the NAND memory 2, the NAND I/F 20 reads the requested data from the NAND memory 2, and outputs the read data to the first error-correction decoding unit 42.

Data (data block, hamming code, BCH code) read from the NAND memory 2 are input to the first error-correction decoding unit 42. The first error-correction decoding unit 42 performs first error correction using the hamming code for each block BK. In the present embodiment, the hamming code has a one-bit error correction capacity. Therefore, two or more bits cannot be corrected. The first error-correction decoding unit 42 detects whether the error correction of the block BK can be done by the first error-correction process. Specifically, for example, a syndrome calculation is performed for the block BK after the first error correction. It can be determined whether the error correction can be performed, based on a result of the calculation ("0"= no error). Alternatively, an error detection code is added to each block BK, and it can be

checked whether there is an error in the data after the first error correction using the error detection code.

The first error-correction decoding unit 42 abandons the hamming code, and outputs the block BK after the first error-correction process, the BCH code, and a result of the first error-correction process of each block BK (a result of the first error-correction decoding) to the second error-correction decoding unit 41. It is explained below that, as shown in Fig. 4-1, there are errors in BK0 and BK2 out of the read data after the first error correction. Fig. 4-2 depicts a result (one bit) of the first error-correction process in Fig. 4-1. In Fig. 4-2, "0" represents that there is no error, and "1" represents that there is an error.

In the second error-correction decoding unit 41, the block BK after the first error-correction process, the BCH code, and the result of the first error correction of each block BK (one bit) are stored in the memory 53. At the same time, the block BK after the first error-correction process and the BCH code are input to the syndrome calculator 50. In the following explanations, when the second error-correction-code generator 32 performs the BCH encoding, a primitive polynomial equation $G(X)$ is used. The primitive polynomial equation

$$G(X)=x^8+x^4+x^3+x^2+1 \quad \dots (1)$$

As shown in Fig. 6, in the syndrome calculator 50, BK0, 1, 2, 3 and data [0] to data [143] of the BCH code are input by each one bit in this order. The syndrome calculator 50 performs syndrome calculation, and outputs syndrome calculation results S0 [7:0] to S3 [7:0] to the error-position polynomial calculator 51.

In Fig. 5-1, the error-position polynomial calculator 51 calculates coefficients σ_1 and σ_2 of an error-position

polynomial equation $\sigma(Z)=1+\sigma_0Z+\sigma_1Z^2$, using the syndrome calculation results S_0 [7:0] to S_3 [7:0]. This is explained using the Peterson's algorithm. The Berlekamp-Massey (BM) algorithm and the Euclidean algorithm can be also used. According to the Peterson's algorithm, a matrix of the following equation (2) is calculated using the syndrome calculation values S_0 to S_3 , to obtain the coefficients σ_0 and σ_1 of the error-position polynomial equation. Coefficients σ_0 [7:0] and σ_1 [7:0] of the error-position polynomial equation calculated by the error-position polynomial calculator 51 are output to the error-locator-calculating and error-correcting unit 52.

$$\begin{pmatrix} S_1 & S_0 \\ S_2 & S_1 \end{pmatrix} \begin{pmatrix} \sigma_0 \\ \sigma_1 \end{pmatrix} = - \begin{pmatrix} S_2 \\ S_3 \end{pmatrix} \quad \dots(2)$$

15

The error-locator-calculating and error-correcting unit 52 specifies an error position by performing a Chien search based on the coefficients σ_0 [7:0] and σ_1 [7:0] of the error-position polynomial equation received from the error-position polynomial calculator 51, and performs second error correction to the data of which the first error correction is finished.

As shown in Fig. 7, the error-locator-calculating and error-correcting unit 52 processes header 111 bits by zero padding to the data to be processed, to set a code length to 2^N-1 (where $N=8$)= 255 bits (fixed length)).

Fig. 8 is a configuration example of a Chien search circuit 61 and an error correcting unit 62 of the error-locator-calculating and error-correcting unit 52. Fig. 9-1 is a configuration diagram of a α circuit (a circuit that multiplies by α times (a multiplier circuit)) of the Chien

30

search circuit 61. Fig. 9-2 is a configuration diagram of a $x\alpha^{32}$ circuit (a circuit that multiplies by α^{32} times) of the Chien search circuit 61. Fig. 9-3 is a configuration diagram of a $x\alpha^{111}$ circuit (a circuit that multiplies by α^{111} times) of the Chien search circuit 61. Fig. 9-4 is a configuration diagram of a $x\alpha^2$ circuit (a circuit that multiplies by α^2 times) of the Chien search circuit 61. Fig. 9-5 is a configuration diagram of a $x\alpha^{64}$ circuit (a circuit that multiplies by α^{64} times) of the Chien search circuit 61. Fig. 9-6 is a configuration diagram of a $x\alpha^{222}$ circuit (a circuit that multiplies by α^{222} times) of the Chien search circuit 61. Fig. 10 is a configuration example of a circuit selecting unit of the Chien search circuit.

The Chien search algorithm is a method of checking whether $\sigma(\alpha^i)$ is zero, by sequentially substituting a power α^i ($i=0, 1, \dots, n-1$) of the algorithm α into $\sigma(z)$. The search of a root according to this method is called the Chien search.

As shown in Fig. 8, the Chien search circuit 61 includes a circuit selecting unit 71 that outputs a changeover signal to selectors SEL0 and SEL1, the selector SEL0 that selectively outputs an output of a $x\alpha$ circuit 76, a $x\alpha^{32}$ circuit 75, and a $x\alpha^{111}$ circuit 74, the selector SEL1 that selectively outputs an output of a $x\alpha^2$ circuit 79, a $x\alpha^{64}$ circuit 78, and a $x\alpha^{222}$ circuit 77, registers 0 and 1, and an EXOR calculating unit 80.

The Chien search circuit 61 includes a one-bit processing unit (the $x\alpha$ circuit 76, the $x\alpha^2$ circuit 79), a 32-bit processing unit (the $x\alpha^{32}$ circuit 75, the $x\alpha^{64}$

circuit 78), and a 111-bit processing unit (the $x\alpha^{111}$ circuit 74, the $x\alpha^{222}$ circuit 77).

In the Chien search circuit 61 according to the present embodiment, the 32-bit processing unit (the $x\alpha^{32}$ circuit 75, the $x\alpha^{64}$ circuit 78), and the 111-bit processing unit (the $x\alpha^{111}$ circuit 74, the $x\alpha^{222}$ circuit 77) are provided in addition to the one-bit processing unit (the $x\alpha$ circuit 76, the $x\alpha^2$ circuit 79). With this arrangement, a speed of the Chien search is increased.

The one-bit processing unit (the $x\alpha$ circuit 76, the $x\alpha^2$ circuit 79) is used to process in one-bit unit a block in the error-correction area. The 32-bit processing unit (the $x\alpha^{32}$ circuit 75, the $x\alpha^{64}$ circuit 78) is used to process a block (32 bits) in which it is determined that no error is detected in a non-error-correction-target area. The 32-bit processing unit can process 32-bit data in one cycle (one degree). The 111-bit processing unit (the $x\alpha^{111}$ circuit 74, the $x\alpha^{222}$ circuit 77) is used to process a padding area (111 bits) that is padded by zero in the non-error-correction-target area. The 111-bit processing unit can process 111-bit data in one cycle (one degree).

The error correcting circuit 62 includes an inverter 81 that outputs data in inversion, and a selector 82 that outputs an inverted value of input data when a result of the EXOR calculation performed by the EXOR calculating unit 80 is "1", and outputs the input data straight when a result of the EXOR calculation is other than "1".

A flow of the process performed by the Chien search circuit 61 is explained. First, the Chien search circuit 61 takes in the coefficients σ_0 [7:0] and σ_1 [7:0]

calculated by the error-position polynomial equation calculation.

(1) First, the process proceeds to the process of the header zero padding area. Because it is clear that there
5 is no error in this area, the circuit selecting unit 71 outputs a selection signal to select the $x\alpha^{111}$ circuit 74 and the $x\alpha^{222}$ circuit 77, to the selectors SEL0 and SEL1. The selectors SEL0 and SEL1 select an output (d0 [7:0]) of the $x\alpha^{111}$ circuit 74 and an output (d1 [7:0]) of the
10 $x\alpha^{222}$ circuit 77 respectively, and take the outputs into the register 0 and the register 1. With this arrangement, the process of the zero padding area (111 bits) can be performed in one cycle (at one time), and the zero padding area (111 bits) can be processed at a high speed.

15 (2) The Chien search circuit 61 processes the block BK0. The block BK0 is an error-correction block BK. The error correcting unit 62 takes out the header data BK0 [0] of the block BK0 from the memory 53. Next, the selectors SEL0 and SEL1 select an output (b0 [7:0]) of the $x\alpha$ circuit
20 76, and an output (b1 [7:0]) of the $x\alpha^2$ circuit 79, and take the outputs into the register 0 and the register 1. When a result of the EXOR calculation performed by the EXOR calculating unit 80 of the register 0 and the register 1 is "1", a selector 92 of the error correcting unit 62 outputs
25 an inverted value of the BK0 [0]. When a result of the EXOR calculation is other than "1", the selector 92 of the error correcting unit 62 outputs the value of BK0 [0] straight. The error correcting unit 62 takes out second data BK0 [1] of the block BK0 from the memory 53. The
30 error correcting unit 62 selects the output (b0 [7:0]) of the $x\alpha$ circuit 76, and the output (b1 [7:0]) of the $x\alpha^2$ circuit 79, and takes the outputs into the register 0 and

the register 1. When a result of the EXOR calculation performed by the EXOR calculating unit 80 of the register 0 and the register 1 is "1", the selector 92 of the error correcting unit 62 outputs an inverted value of the BK0 [1].

- 5 When a result of the EXOR calculation is other than "1", the selector 92 of the error correcting unit 62 outputs the value of BK0 [1] straight. This operation is repeated to the last data BK0 [31] of the block BK0 in a similar manner. In this way, BK0 is processed in one-bit unit.

- 10 (3) The Chien search circuit 61 processes the block BK1. Because the block BK1 is a non-error-correction-target block, an output (c0 [7:0]) of the α^{32} circuit 75 and an output (c1 [7:0]) of the α^{64} circuit 78 are taken into the register 0 and the register 1. As a result, the
15 non-error-correction-target block (32 bits) can be processed at once, and a speed of processing the non-error-correction-target block (32 bits) can be increased.

- (4) The Chien search circuit 61 processes the block BK2. The block BK2 is an error-correction block. The
20 header data from BK2 [0] to the last data BK2 [31] of the block BK2 are taken out from the memory 53, and are processed, in a similar manner to that in (3).

- (5) The Chien search circuit 61 processes the block BK3. Because the block B3 is a non-error-correction-target
25 block, the output (c0 [7:0]) of the α^{32} circuit 75 and the output (c1 [7:0]) of the α^{64} circuit 78 are taken into the register 0 and the register 1.

- While the error-correction block is processed in one-bit unit in the above embodiment, the error-correction
30 blocks of two or more bits can be processed in parallel, and the non-error-correction-target area can be processed in one cycle. For example, in the case of a parallel

process of two bits, providing the Chien search circuit 61 and the error correcting unit 62 for odd number bits as well as for even number bits is sufficient. Further, in the case of a parallel process of three bits, providing
5 three pairs of the Chien search circuit 61 and the error correcting unit 62 is sufficient.

Fig. 10 is a configuration example of the circuit selecting unit 71. The circuit selecting unit 71 includes a first error-correction-decoding-result storage unit 91,
10 the selector 92, a block counter unit 93, a bit counter unit 95, a bit comparing unit 97, and a determining unit 98.

The first error-correction-decoding-result storage unit 91 stores a result of first error-correction decoding of each of the blocks BK0 to BK3. "0" represents a BK that
15 can be corrected (second non-error-correction-target-decoding block). "1" represents a BK that cannot be corrected (second error-correction-decoding block).

The bit counter unit 95 counts a bit counter value as a current processing position in one block BK, and outputs
20 a count result to the bit comparing unit 97 (bit counter value =0 to 31). The block counter unit 93 counts a block counter value as a current processing block in four blocks BKs, and outputs a count result to the selector 92 (block counter value =0 to 3). The bit comparing unit 97
25 determines whether a bit counter value is equal to a block size. When a bit counter value is equal to a block size, the bit comparing unit 97 causes the block counter unit 93 to increment the block counter value.

The selector 92 outputs a first error-correction-decoding result stored in the first error-correction-decoding-result storage unit 91 of the block indicated by
30 the block counter value to the determining unit 98. The determining unit 98 selects one of the one-bit processing

unit (the α circuit 76, the α^2 circuit 79), the 32-bit processing unit (the α^{32} circuit 75, the α^{64} circuit 78), and the 111-bit processing unit (the α^{111} circuit 74, the α^{222} circuit 77), and outputs a selection signal to the selectors SEL0 and SEL1. Immediately after starting the process, the determining unit 98 selects the 111-bit processing unit (the α^{111} circuit 74, the α^{222} circuit 77) to process the padding area. When the current processing block is the non-error-correction-target block, the determining unit 98 selects the 32-bit processing unit (the α^{32} circuit 75, the α^{64} circuit 78). When the current processing block is the error-correction block, the determining unit 98 selects the one-bit processing unit (the α circuit 76, the α^2 circuit 79).

Fig. 11 is a flowchart for explaining the operation of the circuit selecting unit 71. In Fig. 11, the determining unit 98 first selects the 111-bit processing unit (the α^{111} circuit 74, the α^{222} circuit 77) (Step S11). Next, the determining unit 98 determines whether a value of the first error-correction-decoding-result storage unit 91 indicated by the block counter value is "1" (Step S12). When a value of the first error-correction-decoding-result storage unit 91 indicated by the block counter value is "1" (YES at Step S12), the determining unit 98 determines the one-bit processing unit (the α circuit 76, the α^2 circuit 79) (Step S16). The bit count unit 95 increments the bit counter value (Step S17), and the bit comparing unit 97 determines whether a bit counter value is equal to a block size "32" (Step S18). When a bit counter value is equal to a block size (YES at Step S18), the process proceeds to Step S14. On the other hand, when a bit

counter value is not equal to a block size (NO at Step S18), the process returns to Step S17.

When a value of the first error-correction-decoding-result storage unit 91 indicated by the block counter value is not "1" at Step S12 (NO at Step S12), the determining unit 98 selects the 32-bit processing unit (the $x\alpha^{32}$ circuit 75, the $x\alpha^{64}$ circuit 78) (Step S13). The block count unit 95 increments the block counter (Step S14). When the block counter value is "4" (YES at Step S15), the flow is finished. On the other hand, when the block counter value is not "4" (NO at Step S15), the process returns to Step S12.

As explained above, according to the present embodiment, in addition to the processing unit that performs a search process of the error position in one-bit unit in the error-correction area, the Chien search circuit 61 includes the processing unit that processes in one cycle (at one time) plural bits in the non-error-correction-target area. Therefore, the Chien search can be performed at a high speed.

The Chien search circuit 61 processes in one cycle (at one time) the non-error-correction-target block which is determined to have no error by the error detection, by the 32-bit processing unit (the $x\alpha^{32}$ circuit 75, the $x\alpha^{64}$ circuit 78). Therefore, the processing time required for the block which is determined to have no error by the error detection can be decreased, and the Chien search can be performed at a high speed.

In the Chien search circuit 61, the 111-bit processing unit (the $x\alpha^{111}$ circuit 74, the $x\alpha^{222}$ circuit 77) processes in one cycle (at one time) the data in the padding area. Therefore, the processing time required for

the padding area can be decreased, and the Chien search can be performed at a high speed.

In the above embodiment, a size of the block is set as 32 bits. The BCH code has a 16-bit size having two-bit
5 correction capacity. The padding area has 111 bits.
However, in the present invention, sizes are not limited to the above. A size of the block, correction capacity and a size of the BCH code, and a size of the padding area can take any values. While the BCH code is explained as an
10 example for the encoding system that performs a Chien search, the code is not limited to this in the present invention. Other encoding system using a Chien search such as the RS code can be also used.

In the above embodiment, while the present invention
15 is applied to the SSD having a NAND memory, the present invention can be also applied to the SSD having other flash EEPROM such as a NOR type.

In each embodiment of the present invention, each functional block can be achieved as one of hardware and
20 computer software or a combination of both. Therefore, each block is explained as follows to identify any one of hardware, computer softer or both, from the viewpoint of these functions. Whether these functions are achieved as hardware or software depends on a design constraint applied
25 to a detailed embodiment or the entire system. Those skilled in the art can achieve these functions in various ways for each specific embodiment, and determining such achievement is also included in the scope of the present invention.

30 According to the present invention, a Chien search device and a Chien search method capable of performing the Chien process at a high speed can be provided.

CLAIMS

1. A Chien search device that calculates an error position at a time of correcting an error included in data
5 read from a nonvolatile memory, the Chien search device comprising:
a first processing unit that performs a search process of an error position in at least one-bit unit to an error-correction area of input data; and
10 a second processing unit that processes a plurality of bits at one time in a non-error-correction-target area of the input data.
2. The Chien search device according to claim 1, wherein
15 the second processing unit includes
a first plural-bit processing unit that process first plural bits at one time; and
a second plural-bit processing unit that process second plural bits at one time.
- 20 3. The Chien search device according to claim 2, wherein
the first plural-bit processing unit processes at one time an area which is determined to have no error as a result of an error detection in the non-error-correction-
25 target area, and
the second plural-bit processing unit processes at one time a padding area in the non-error-correction-target area.
4. The Chien search device according to claim 1, wherein
30 the nonvolatile memory is a NAND flash memory.
5. A Chien search method that calculates an error position at a time of correcting an error included in data

read from a nonvolatile memory, the Chien search method comprising:

performing a search process of an error position in at least one-bit unit to an error-correction area of input

5 data; and

processing a plurality of bits at one time in a non-error-correction-target area of the input data.

10.

FIG.1

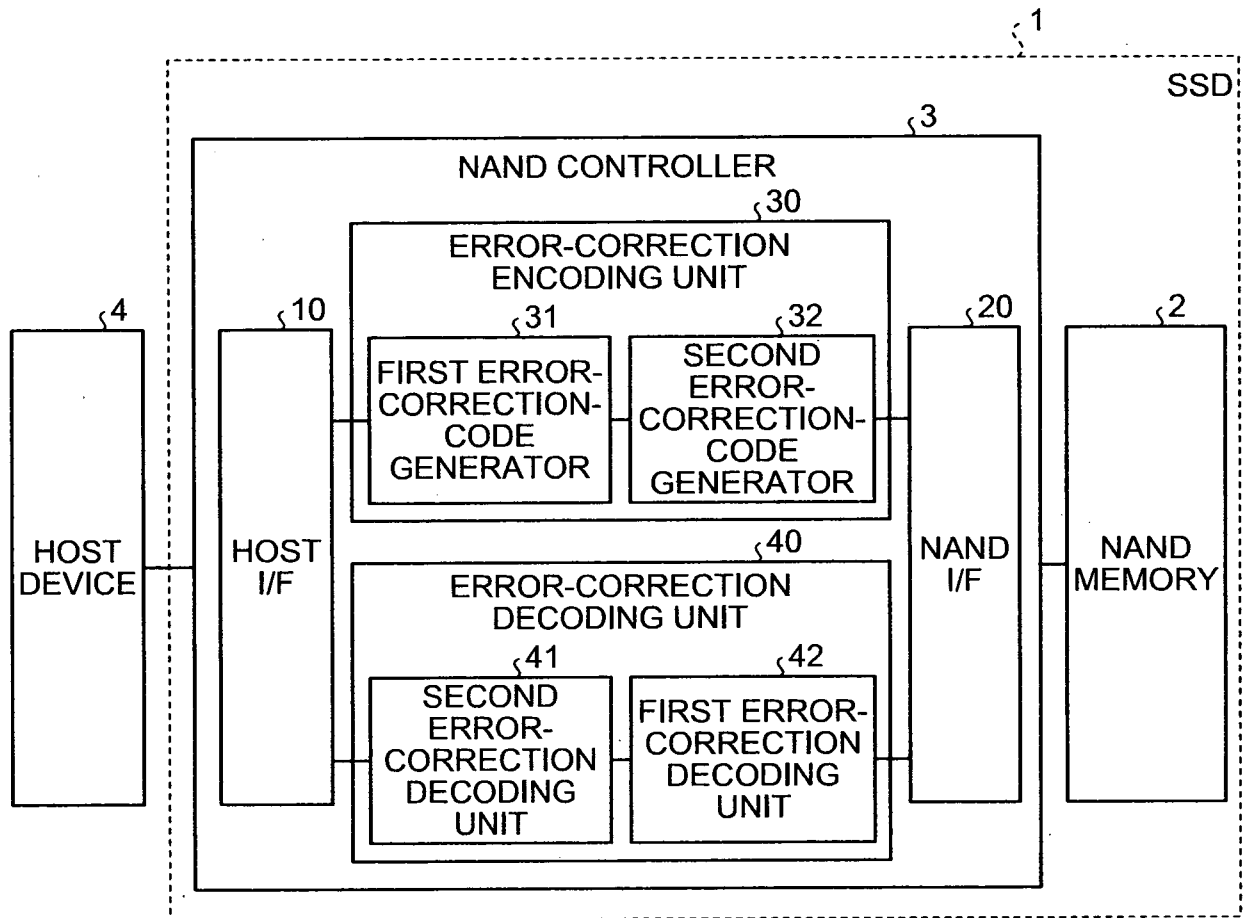


FIG.2

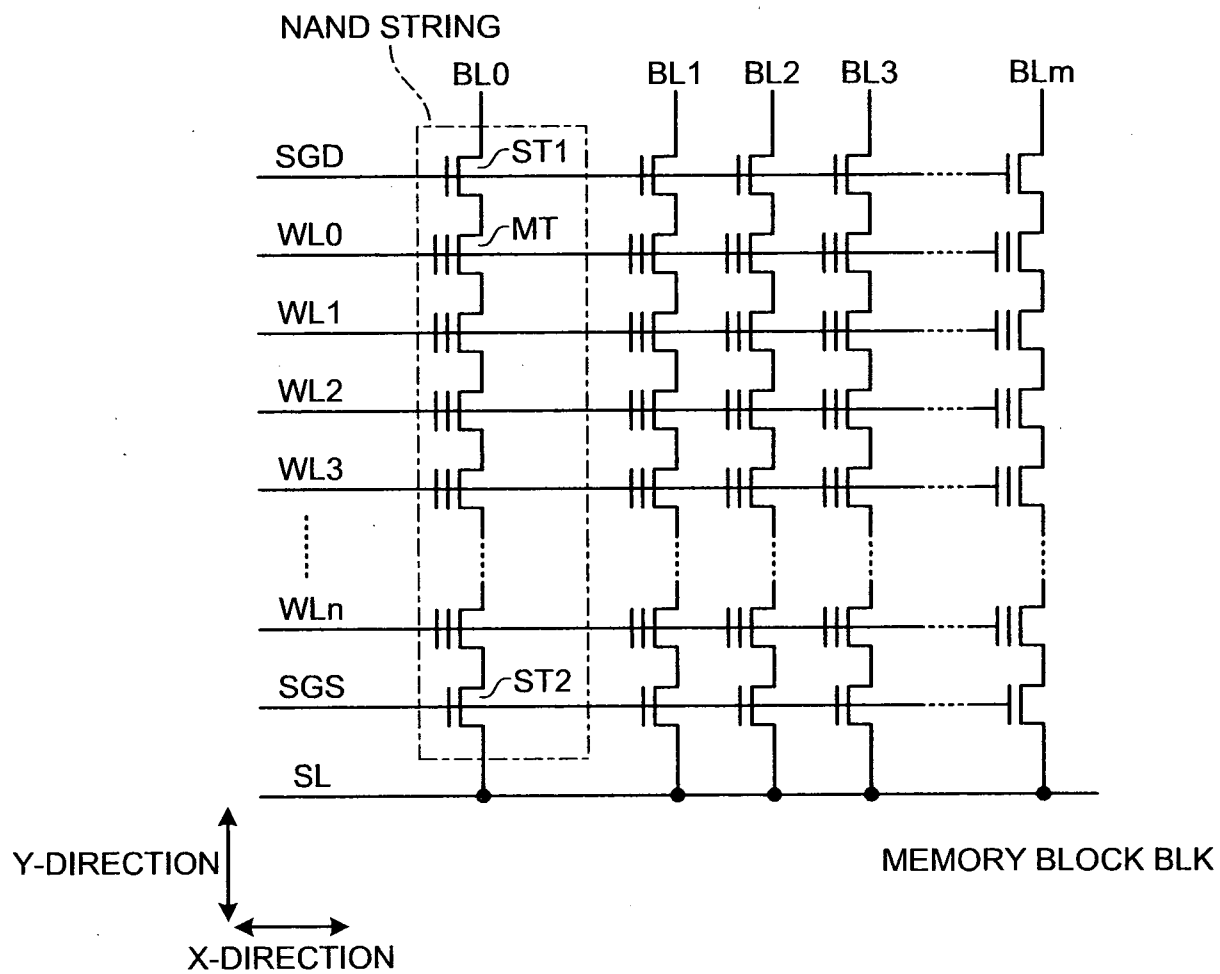


FIG.3

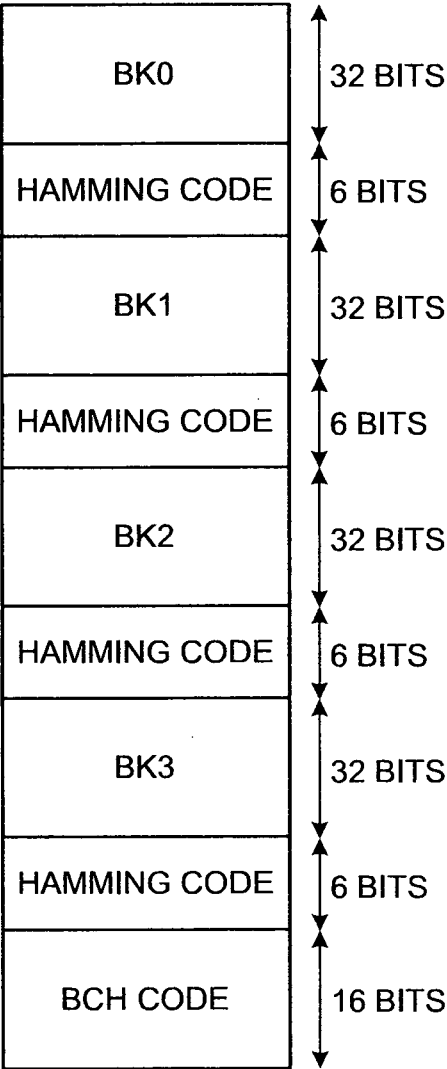


FIG.4 - 1

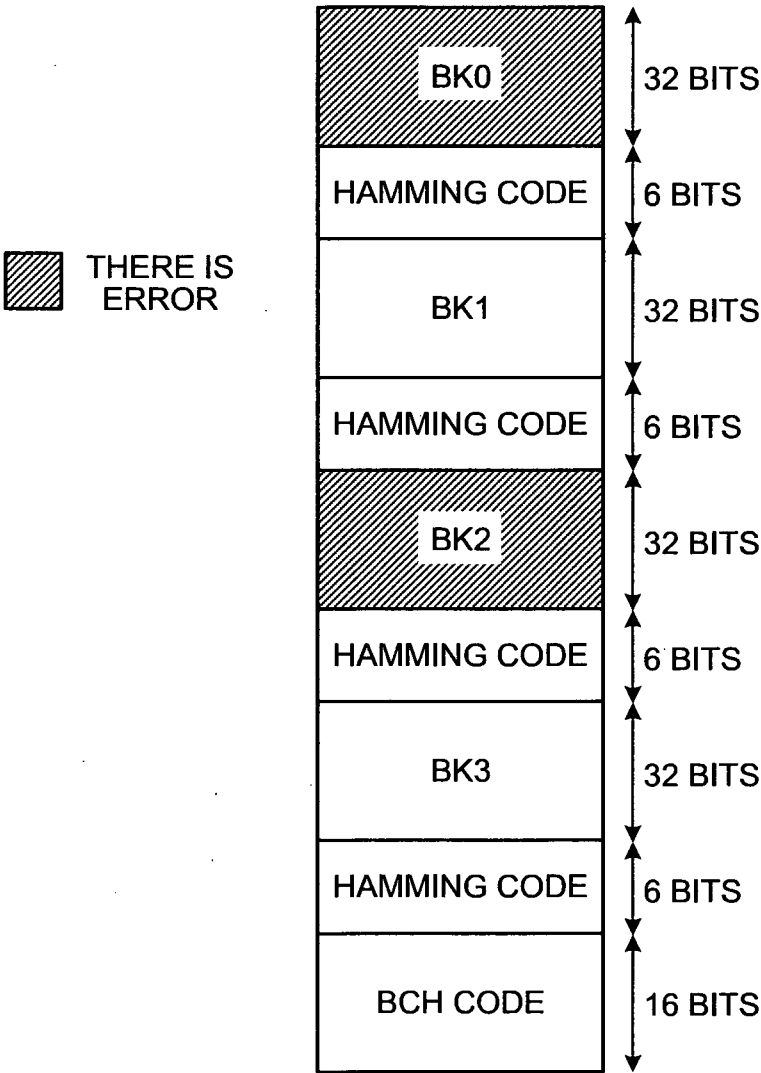


FIG.4 - 2

RESULT OF FIRST
ERROR-CORRECTION
PROCESS

0	1	2	3
1	0	1	0
BK0	BK1	BK2	BK3

BK: BLOCK

FIG. 5 - 1

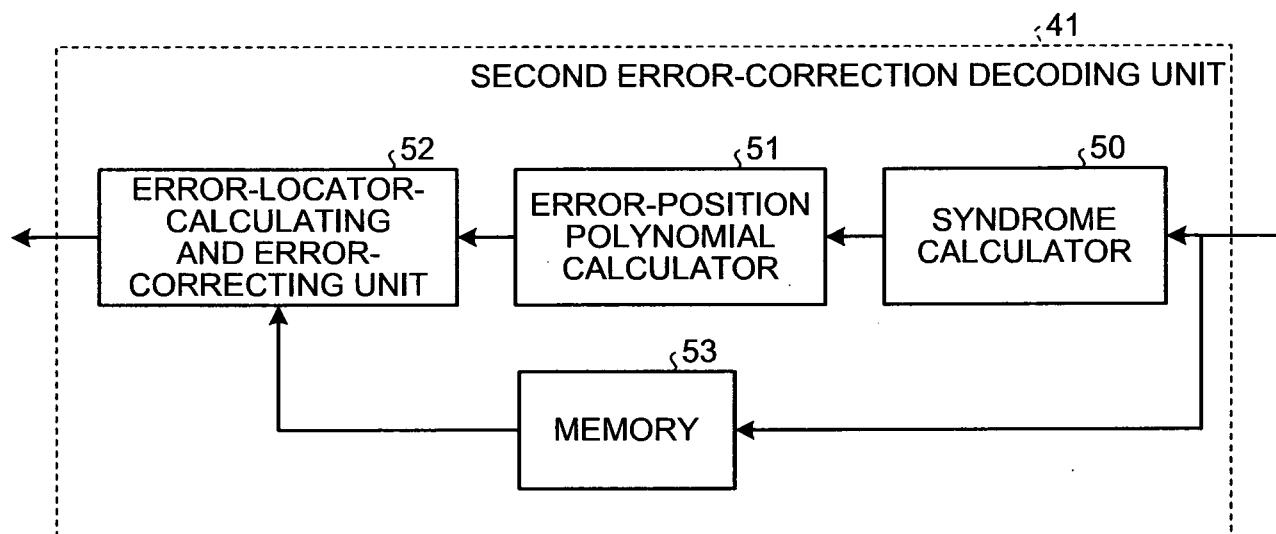


FIG. 5 - 2

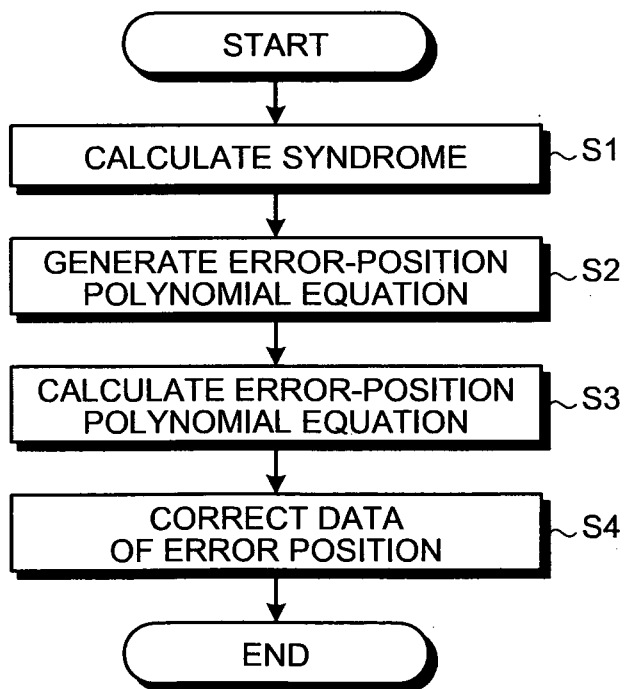


FIG.6

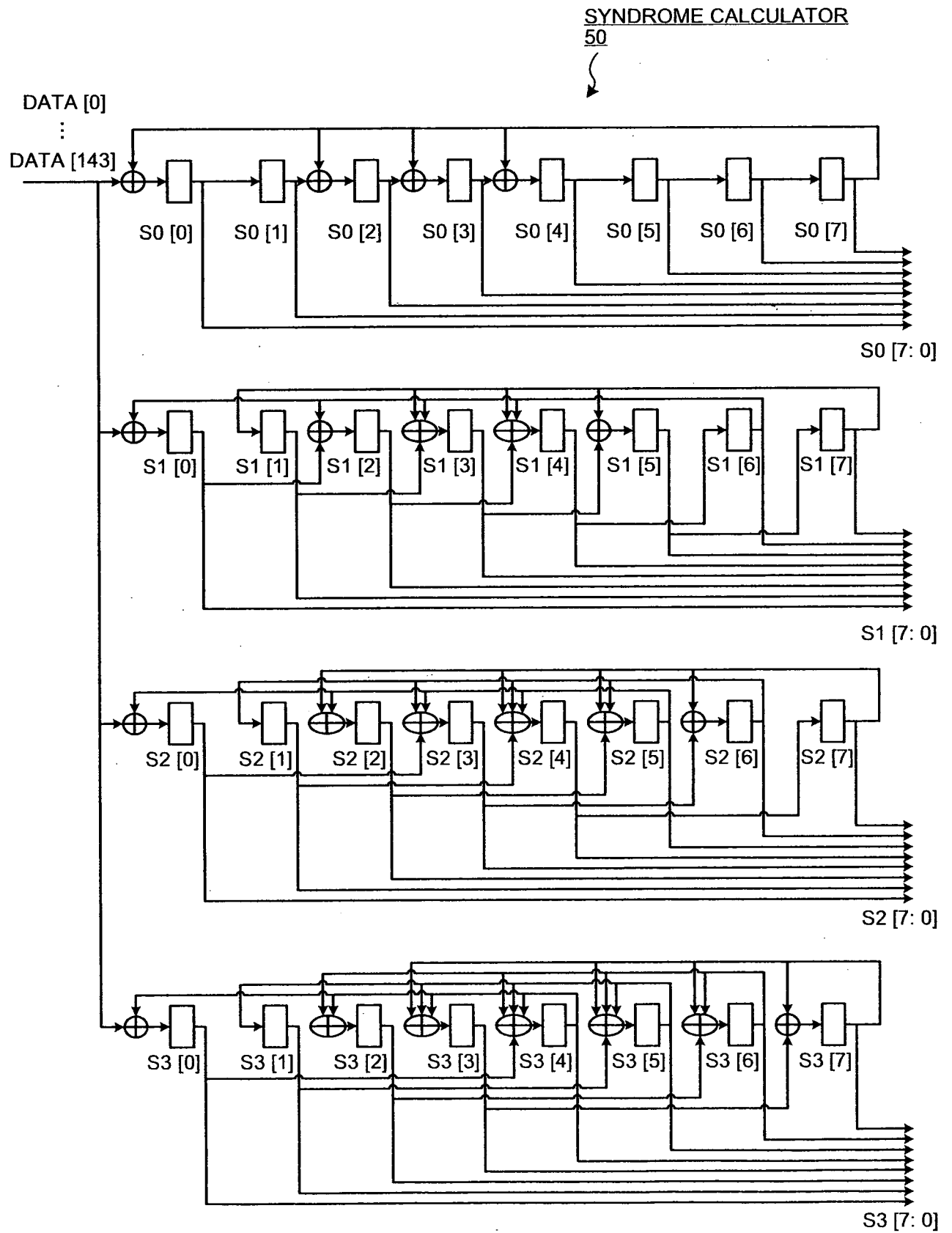


FIG.7

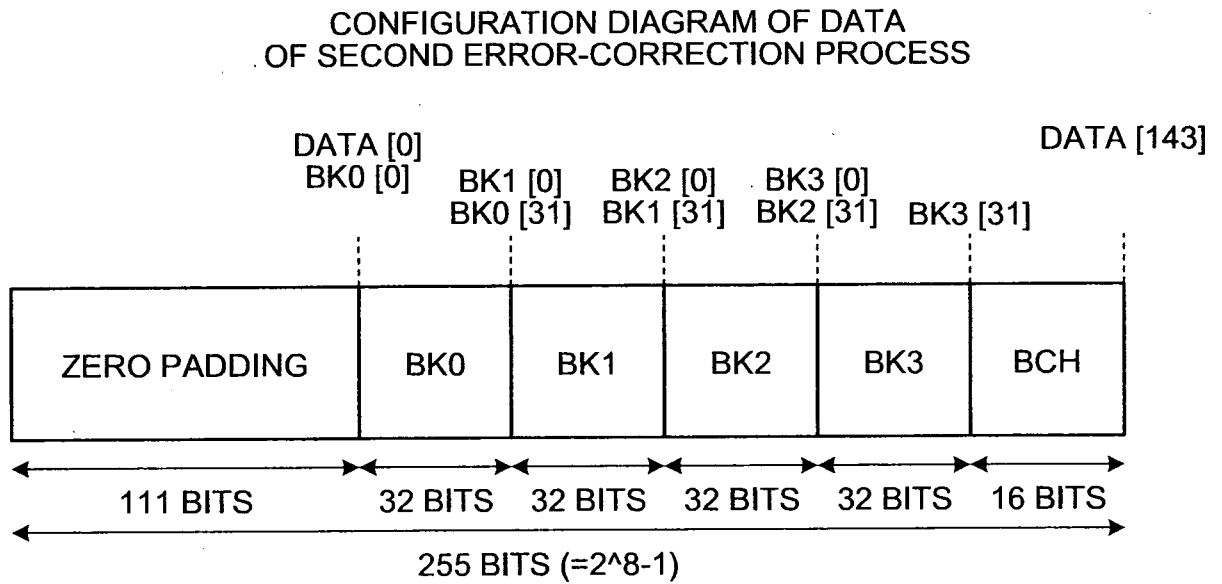
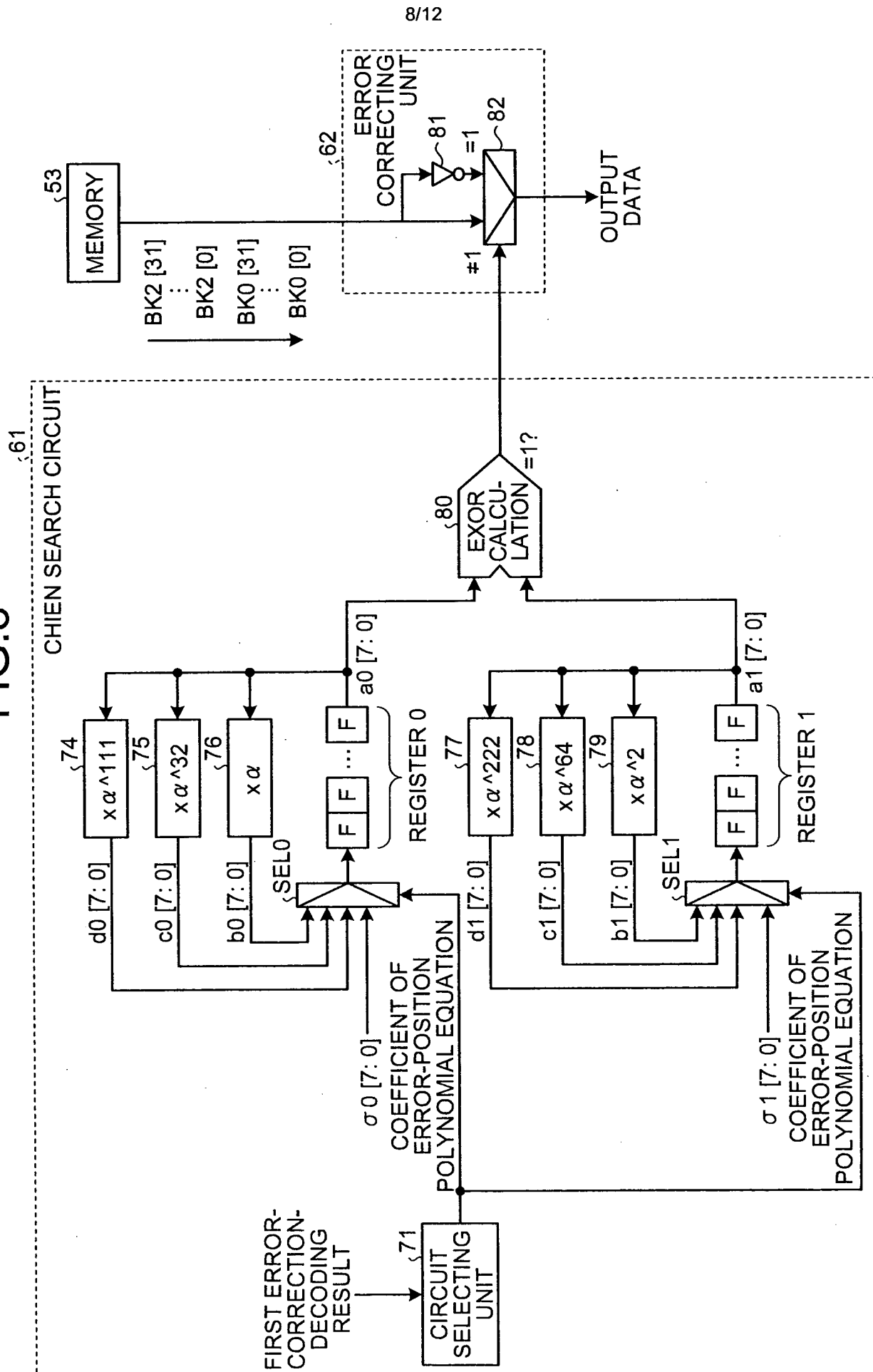


FIG.8



9/12

FIG.9 - 1**CONFIGURATION DIAGRAM OF $x\alpha$ CIRCUIT**

$$\begin{aligned}
b0[0] &= a0[7] \\
b0[1] &= a0[0] \\
b0[2] &= a0[1] \oplus a0[7] \\
b0[3] &= a0[2] \oplus a0[7] \\
b0[4] &= a0[3] \oplus a0[7] \\
b0[5] &= a0[4] \\
b0[6] &= a0[5] \\
b0[7] &= a0[6]
\end{aligned}$$

 $\oplus$ REPRESENTS EXOR CALCULATION**FIG.9 - 2****CONFIGURATION DIAGRAM OF $x\alpha^{32}$ CIRCUIT**

$$\begin{aligned}
c0[0] &= a0[0] \oplus a0[1] \oplus a0[4] \oplus a0[7] \\
c0[1] &= a0[1] \oplus a0[2] \oplus a0[5] \oplus \\
c0[2] &= a0[0] \oplus a0[1] \oplus a0[2] \oplus a0[3] \oplus a0[4] \oplus a0[6] \oplus a0[7] \\
c0[3] &= a0[0] \oplus a0[2] \oplus a0[3] \oplus a0[5] \\
c0[4] &= a0[0] \oplus a0[3] \oplus a0[6] \oplus a0[7] \\
c0[5] &= a0[1] \oplus a0[4] \oplus a0[7] \\
c0[6] &= a0[2] \oplus a0[5] \oplus \\
c0[7] &= a0[0] \oplus a0[3] \oplus a0[6]
\end{aligned}$$

 $\oplus$ REPRESENTS EXOR CALCULATION**FIG.9 - 3****CONFIGURATION DIAGRAM OF $x\alpha^{111}$ CIRCUIT**

$$\begin{aligned}
d0[0] &= a0[1] \oplus a0[2] \oplus a0[6] \oplus a0[7] \\
d0[1] &= a0[0] \oplus a0[2] \oplus a0[3] \oplus a0[7] \\
d0[2] &= a0[0] \oplus a0[2] \oplus a0[3] \oplus a0[4] \oplus a0[6] \oplus a0[7] \\
d0[3] &= a0[0] \oplus a0[2] \oplus a0[3] \oplus a0[4] \oplus a0[5] \oplus a0[6] \\
d0[4] &= a0[2] \oplus a0[3] \oplus a0[4] \oplus a0[5] \\
d0[5] &= a0[3] \oplus a0[4] \oplus a0[5] \oplus a0[6] \\
d0[6] &= a0[0] \oplus a0[4] \oplus a0[5] \oplus a0[6] \oplus a0[7] \\
d0[7] &= a0[0] \oplus a0[1] \oplus a0[5] \oplus a0[6] \oplus a0[7]
\end{aligned}$$

 $\oplus$ REPRESENTS EXOR CALCULATION

10/12

FIG. 9 - 4

CONFIGURATION DIAGRAM OF $x\alpha^2$ CIRCUIT

$b1[0] = a1[6]$
 $b1[1] = a1[7]$
 $b1[2] = a1[0] \oplus a1[6]$
 $b1[3] = a1[1] \oplus a1[6] \oplus a1[7]$
 $b1[4] = a1[2] \oplus a1[6] \oplus a1[7]$
 $b1[5] = a1[3] \oplus a1[7]$
 $b1[6] = a1[4]$
 $b1[7] = a1[5]$

$\oplus$ REPRESENTS EXOR CALCULATION

FIG. 9 - 5

CONFIGURATION DIAGRAM OF $x\alpha^{64}$ CIRCUIT

$c1[0] = a1[0] \oplus a1[2] \oplus a1[4] \oplus a1[5]$
 $c1[1] = a1[0] \oplus a1[1] \oplus a1[3] \oplus a1[5] \oplus a1[6]$
 $c1[2] = a1[0] \oplus a1[1] \oplus a1[5] \oplus a1[6] \oplus a1[7]$
 $c1[3] = a1[0] \oplus a1[1] \oplus a1[4] \oplus a1[5] \oplus a1[6] \oplus a1[7]$
 $c1[4] = a1[0] \oplus a1[1] \oplus a1[4] \oplus a1[6] \oplus a1[7]$
 $c1[5] = a1[1] \oplus a1[2] \oplus a1[5] \oplus a1[7]$
 $c1[6] = a1[0] \oplus a1[2] \oplus a1[3] \oplus a1[6]$
 $c1[7] = a1[1] \oplus a1[3] \oplus a1[4] \oplus a1[7]$

$\oplus$ REPRESENTS EXOR CALCULATION

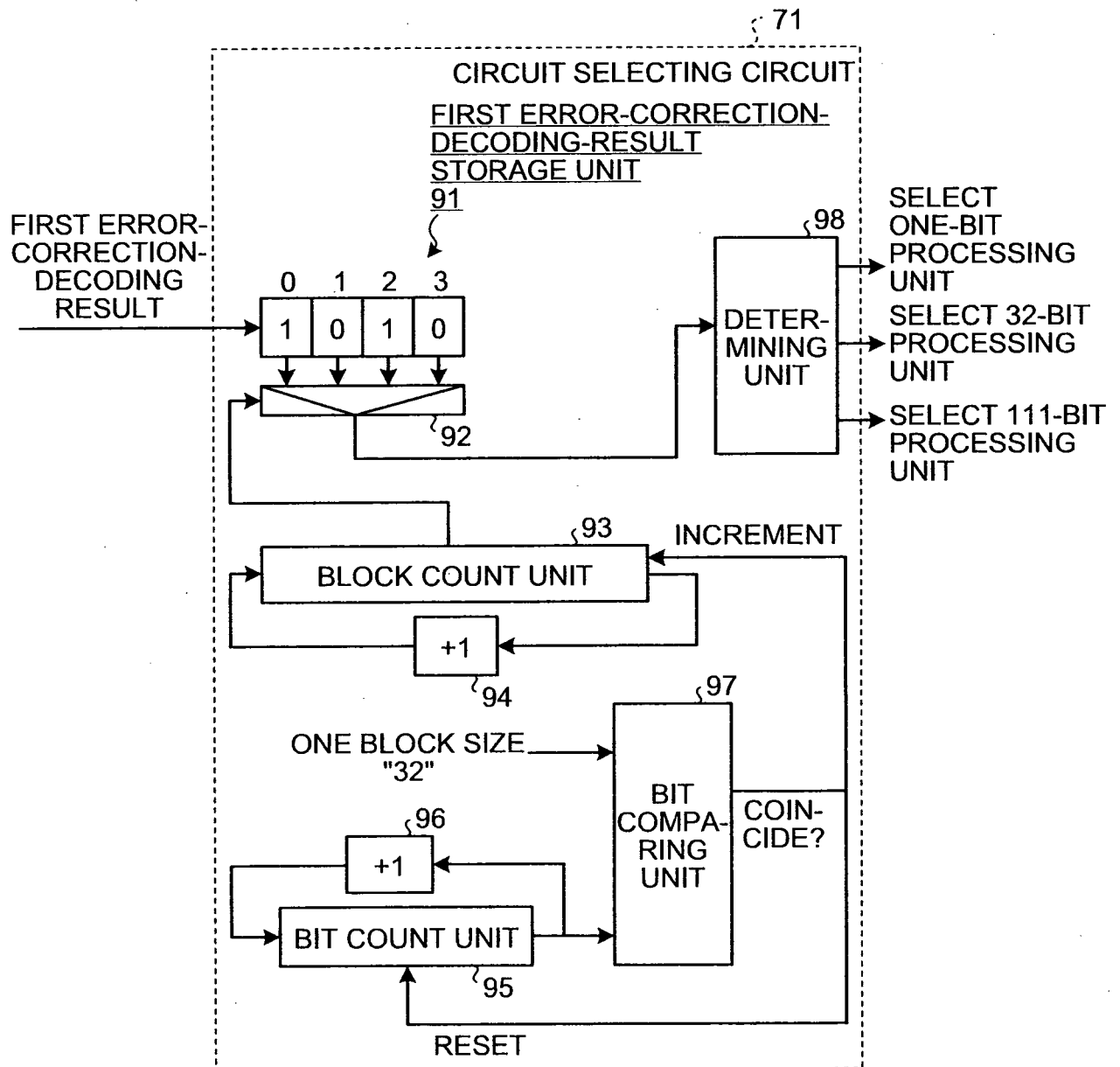
FIG. 9 - 6

CONFIGURATION DIAGRAM OF $x\alpha^{222}$ CIRCUIT

$d1[0] = a1[1] \oplus a1[6]$
 $d1[1] = a1[0] \oplus a1[2] \oplus a1[7]$
 $d1[2] = a1[3] \oplus a1[6]$
 $d1[3] = a1[0] \oplus a1[1] \oplus a1[4] \oplus a1[6] \oplus a1[7]$
 $d1[4] = a1[2] \oplus a1[5] \oplus a1[6] \oplus a1[7]$
 $d1[5] = a1[3] \oplus a1[6] \oplus a1[7]$
 $d1[6] = a1[4] \oplus a1[7]$
 $d1[7] = a1[0] \oplus a1[5]$

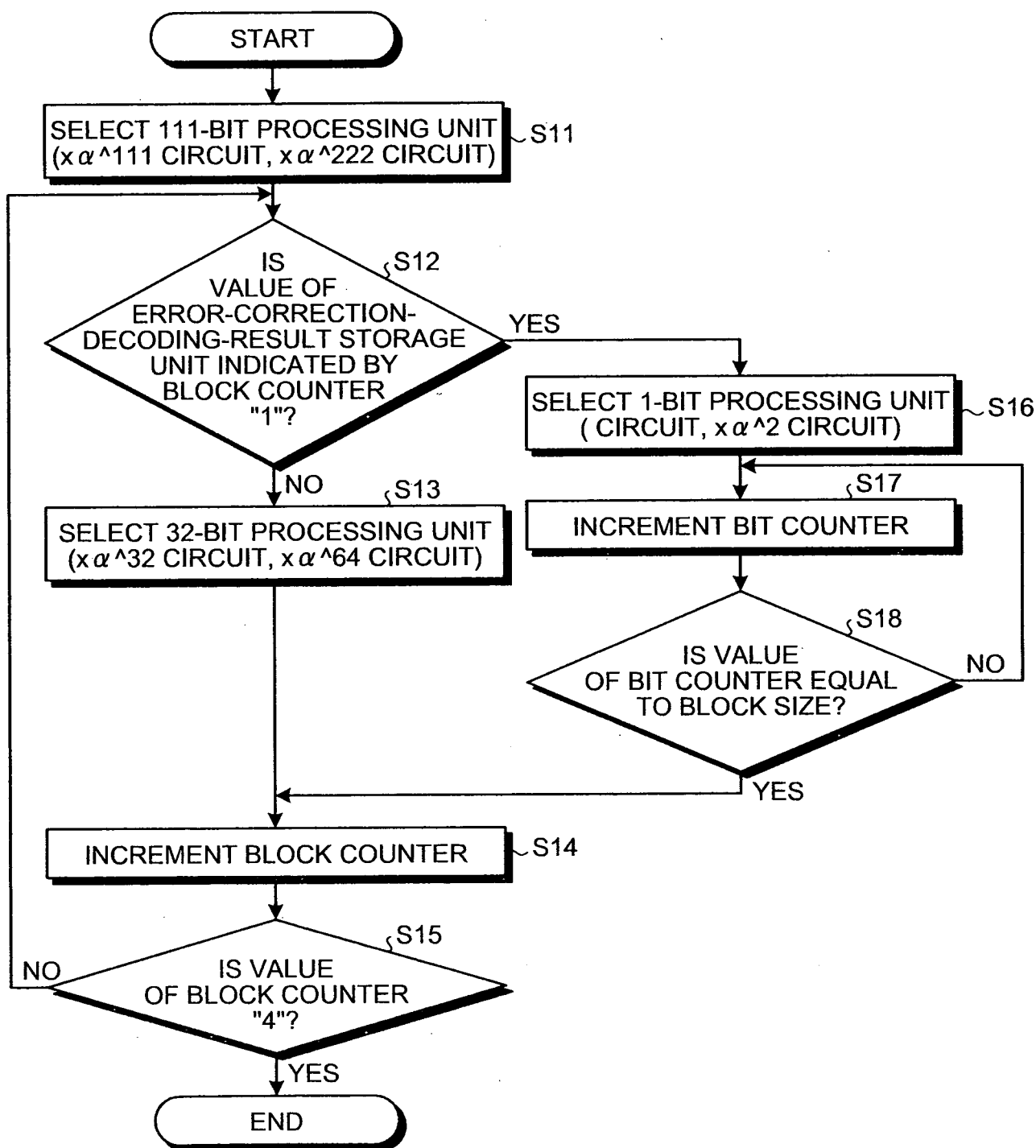
$\oplus$ REPRESENTS EXOR CALCULATION

FIG.10



12/12

FIG.11



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2008/067577

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. H03M13/15 (2006.01) i, G06F12/16 (2006.01) i, G11C16/02 (2006.01) i,
G11C16/06 (2006.01) i, G11C29/42 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl. H03M13/15, G06F12/16, G11C16/02, G11C16/06, G11C29/42

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2008
Registered utility model specifications of Japan 1996-2008
Published registered utility model applications of Japan 1994-2008

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 07-202717 A (Mitsubishi Electric) 1995.08.04, Figs 3, & FR 2714498 A1 & NL 9402204 A & CN 1122025 A & US 5684810 A & JP 3154607 B2 & CN 1055586 C	1-5

☐ Further documents are listed in the continuation of Box C.

☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

13.11.2008

Date of mailing of the international search report

25.11.2008

Name and mailing address of the ISA/JP

Japan Patent Office

3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan

Authorized officer

Naoyuki Yazu

5X

8838

Telephone No. +81-3-3581-1101 Ext. 3593