(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 July 2006 (06.07.2006)

PCT

(10) International Publication Number
WO 2006/071837 A2

(51) International Patent Classification:
G06F 11/10 (2006.01)

(21) International Application Number: PCT/US2005/046996

(22) International Filing Date: 21 December 2005 (21.12.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/021,708 23 December 2004 (23.12.2004) US

(71) Applicant *(for all designated States except US)*: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **EDIRISOORIYA, Samantha** [LK/US]; 640 East Vinedo Lane, Tempe, AZ 85284 (US). **TSE, Gregory** [US/US]; 577 East Kelly Lane, Tempe, AZ 85284 (US). **SCHMISSEUR, Mark** [US/US]; 1402 East Amberwood Drive, Phoenix, AZ 85048 (US). **SHEFFIELD, Robert** [US/US]; 2453 West Sqruce Drive, Chandler, AZ 85248 (US).

(74) Agents: **VINCENT, Lester, J.** et al.; BLAKELY SOKOFOFF TAYLOR & ZAFMAN, 12400 Wilshire Boulevard, 7th Floor, Los Angeles, CA 90025 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, 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, HU, IE, IS, IT, LT, LU, LV, MC, NL, 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:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 2006/071837 A2

(54) **Title:** METHOD AND SYSTEM FOR SYNDROME GENERATION AND DATA RECOVERY

(57) **Abstract:** A method and system for syndrome generation and data recovery is described. The system includes a parity generator coupled to one or more storage devices to generate parity for data recovery. The parity generator includes a first comparator to generate a first parity factor based on data in one or more of the storage devices, a multiplier to multiply data from one or more of the storage devices with a multiplication factor to generate a product, a second comparator coupled to the multiplier to generate a second parity factor based at least in part on the product, and a selector to choose between the first parity factor and the second parity factor.

METHOD AND SYSTEM FOR SYNDROME GENERATION AND DATA RECOVERY

TECHNICAL FIELD

[0001] Embodiments of the invention relate to syndrome generation and data recovery, and more specifically to PQ RAID syndrome generation and data recovery.

BACKGROUND

[0002] With the increase in use of large-scale storage systems, such as with Fiber Channel and Gigabit Ethernet systems, there is an increase in the susceptibility of these systems to multiple disk failures. The rapid growth of disk capacity also prolongs the disk recovery time in the event of disk failures. This prolonged recovery time increases the probability of subsequent disk failures during the reconstruction of user data and parity information stored in a faulty disk. In addition, latent sector failures caused by data that was left unread for a long period of time may prevent data recovery after a disk failure that results in loss of data. The use of less expensive disks, such as ATA (Advanced Technology Attachment) disks, in arrays where high data integrity is required also increases the probability of such disk failures.

[0003] RAID (Redundant Array of Independent Disks) architectures have been developed to allow recovery from disk failures. Typically, the XOR (Exclusive-OR) of data from a number of disks is maintained on a redundant disk. In the event of a disk failure, the data on the failed disk is reconstructed by XORing the data on the surviving disks. The reconstructed data is written to a spare disk. However, data will be lost if the second disk fails before the reconstruction is complete. Traditional disk arrays that protect the loss of no more than one disk are inadequate for data recovery, especially for large-scale storage systems.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

FIG. 1 is a block diagram illustrating a system that allows for recovery from multiple disk failures.

FIG. 2 is a table of example values for a Galois field.

FIG. 3 is a block diagram illustrating a system according to an embodiment of the invention.

FIG. 4 is a flow diagram illustrating a method according to an embodiment of the invention.

DETAILED DESCRIPTION

[0004] Embodiments of a system and method for syndrome generation and data recovery are described. In the following description, numerous specific details are set forth. However, it is understood that embodiments of the invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been shown in detail in order not to obscure the understanding of this description.

[0005] Reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention.

Thus, the appearances of the phrases "in one embodiment" or "in an embodiment" in various places throughout this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

[0006] Referring to Fig. 1, a block diagram illustrates a system 100 that allows for recovery from multiple disk failures. The system 100 includes one or more storage blocks for storing data, such as 102-124, and two or more storage blocks for storing parity or syndrome information, such as 130-144. In one embodiment, the system 100 is a RAID (Redundant Array of Independent Disks) system. In one embodiment, two syndromes are generated and stored: P syndrome and Q syndrome. The P syndrome is generated by computing parity across a stripe. The Q syndrome is generated by using Galois Field multiplication. The regeneration scheme for data recovery uses both Galois Field multiplication and division.

[0007] The following are the equations for generating P and Q for a storage array with n data disks and two check disks:

$$P = D_0 \oplus D_1 \oplus D_2 \;\ldots\ldots \oplus D_{n-1} \quad \text{(Equation 1)}$$

$$Q = g^0{*}D_0 \oplus g^1{*}D_1 \oplus g^2{*}D_2 \ldots\ldots \oplus g^{n-1}{*}D_{n-1} \quad \text{(Equation 2)}$$

[0008] P is the simple parity of data (D) computed across a stripe using ⊕ (XOR) operations. Q requires multiplication (*) using a Galois Field multiplier (g).

[0009] The following equations show the generation of P and Q when updating a data block D_a:

$$P\,(\text{new}) = P(\text{old}) \oplus D_a\,(\text{old}) \oplus D_a\,(\text{new})$$

$$Q\,(\text{new}) = Q(\text{old}) \oplus g^a{*}D_a\,(\text{old}) \oplus g^a{*}D_a\,(\text{new})$$

[0010] There are four cases of multiple disk failure that require recovery. In case one, P and Q fail. In this case, P and Q may be regenerated using Equations 1 and 2 shown above.

[0011] In case two, Q and a data disk (D_a) fail. In this case, D_a may be regenerated using P and the remaining data disks via Equation 1. Q may then be regenerated using Equation 2.

[0012] In case three, P and a data disk (D_a) fail. In this case, D_a may be regenerated using Q, the remaining data disks, and the following equation:

$$D_a = (Q \oplus Q_a) * g^{-a} = (Q \oplus Q_a) * g^{255-a}, \text{ where}$$

$$Q_a = g^0 D_0 \oplus g^1 D_1 \oplus \ldots \oplus g^{a-1} D_{a-1} \oplus g^{a+1} D_{a+1} \ldots \oplus g^{n-1} D_{n-1} .$$

After D_a is regenerated, P may be regenerated using Equation 1.

[0013] In case four, two data disks (D_a and D_b) fail. In this case, D_a and D_b may be regenerated using P and Q, the remaining data disks, and the following equations:

$$D_a = (g^{-a} * (Q \oplus Q_{ab}) \oplus g^{b-a} * (P \oplus P_{ab})) / (g^{b-a} \oplus 0000\ 0001)$$

$$D_b = D_a \oplus (P \oplus P_{ab}), \text{ where}$$

$$P_{ab} = D_0 \oplus D_1 \oplus \ldots \oplus D_{a-1} \oplus D_{a+1} \ldots \oplus D_{b-1} \oplus D_{b+1} \ldots \oplus D_{n-1}$$

$$Q_{ab} = g^0 D_0 \oplus g^1 D_1 \oplus \ldots \oplus g^{a-1} D_{a-1} \oplus g^{a+1} D_{a+1} \ldots \oplus g^{b-1} D_{b-1} \oplus g^{b+1} D_{b+1} \ldots \oplus g^{n-1} D_{n-1} .$$

[0014] The following are examples of recovery from disk failures in the cases described above. In the following examples, the datapath is assumed to be one byte or 8 bits wide. Therefore, a Galois Field, GF (2^8) is used. The invention may be implemented for datapaths that are more or less than one byte wide, and larger or smaller Galois fields may be used.

[0015] The following equations may be used for multiplying two 8-bit elements (b and c) to yield an 8-bit product (a).

b = [b7 b6 b5 b4 b3 b2 b1 b0] and c = [c7 c6 c5 c4 c3 c2 c1 c0].

a0 = b0.c0 ⊕ b7.c1 ⊕ b6.c2 ⊕ b5.c3 ⊕ b4.c4 ⊕ b3.c5 ⊕ b7.c5 ⊕ b2.c6 ⊕ b7.c6 ⊕ b6.c6 ⊕ b1.c7 ⊕ b7.c7 ⊕ b6.c7 ⊕ b5.c7

a1 = b1.c0 ⊕ b0.c1 ⊕ b7.c2 ⊕ b6.c3 ⊕ b5.c4 ⊕ b4.c5 ⊕ b3.c6 ⊕ b7.c6 ⊕ b2.c7 ⊕ b7.c7 ⊕ b6.c7

a2 = b2.c0 ⊕ b1.c1 ⊕ b7.c1 ⊕ b0.c2 ⊕ b6.c2 ⊕ b7.c3 ⊕ b5.c3 ⊕ b6.c4 ⊕ b4.c4 ⊕ b5.c5 ⊕ b3.c5 ⊕ b7.c5 ⊕ b2.c6 ⊕ b7.c6 ⊕ b6.c6 ⊕ b4.c6 ⊕ b1.c7 ⊕ b3.c7 ⊕ b6.c7 ⊕ b5.c7

a3 = b3.c0 ⊕ b2.c1 ⊕ b7.c1 ⊕ b1.c2 ⊕ b7.c2 ⊕ b6.c2 ⊕ b0.c3 ⊕ b6.c3 ⊕ b5.c3 ⊕ b7.c4 ⊕ b5.c4 ⊕ b4.c4 ⊕ b6.c5 ⊕ b4.c5 ⊕ b3.c5 ⊕ b7.c5 ⊕ b2.c6 ⊕ b6.c6 ⊕ b5.c6 ⊕ b3.c6 ⊕ b2.c7 ⊕ b4.c7 ⊕ b1.c7 ⊕ b5.c7

a4 = b4.c0 ⊕ b3.c1 ⊕ b7.c1 ⊕ b2.c2 ⊕ b7.c2 ⊕ b6.c2 ⊕ b1.c3 ⊕ b7.c3 ⊕ b6.c3 ⊕ b5.c3 ⊕ b0.c4 ⊕ b6.c4 ⊕ b5.c4 ⊕ b4.c4 ⊕ b5.c5 ⊕ b4.c5 ⊕ b3.c5 ⊕ b2.c6 ⊕ b4.c6 ⊕ b3.c6 ⊕ b1.c7 ⊕ b7.c7 ⊕ b2.c7 ⊕ b3.c7

a5 = b5.c0 ⊕ b4.c1 ⊕ b3.c2 ⊕ b7.c2 ⊕ b2.c3 ⊕ b7.c3 ⊕ b6.c3 ⊕ b1.c4 ⊕ b7.c4 ⊕ b6.c4 ⊕ b5.c4 ⊕ b0.c5 ⊕ b6.c5 ⊕ b5.c5 ⊕ b4.c5 ⊕ b5.c6 ⊕ b4.c6 ⊕ b3.c6 ⊕ b2.c7 ⊕ b4.c7 ⊕ b3.c7

a6 = b6.c0 ⊕ b5.c1 ⊕ b4.c2 ⊕ b3.c3 ⊕ b7.c3 ⊕ b2.c4 ⊕ b7.c4 ⊕ b6.c4 ⊕ b1.c5 ⊕ b7.c5 ⊕ b6.c5 ⊕ b5.c5 ⊕ b0.c6 ⊕ b6.c6 ⊕ b5.c6 ⊕ b4.c6 ⊕ b5.c7 ⊕ b4.c7 ⊕ b3.c7

a7 = b7.c0 ⊕ b6.c1 ⊕ b5.c2 ⊕ b4.c3 ⊕ b3.c4 ⊕ b7.c4 ⊕ b2.c5 ⊕ b7.c5 ⊕ b6.c5 ⊕ b1.c6 ⊕ b7.c6 ⊕ b6.c6 ⊕ b5.c6 ⊕ b0.c7 ⊕ b6.c7 ⊕ b5.c7 ⊕ b4.c7

[0016] Fig. 2 shows a table 200 providing example values of the Galois field multiplier g^a for g = 0000 0010. The negative power of a generator for $GF(2^8)$ can be computed using the following equation:

$g^{-a} = g^{255-a}$.

[0017] The following example shows how to generate P and Q parity for a disk array with four data disks and two parity disks. Assume that each data block contains one data byte. Let D_i be the data contain in disk I (i=0,1,2,3). Consider the following data stripe:

$D_0 = 1011\ 0100$, $D_1 = 0010\ 1100$, $D_2 = 1100\ 0110$ and $D_3 = 1101\ 0101$.

Then, P may be generated using Equation 1 as follows:

$$\begin{aligned} P &= D_0 \oplus D_1 \oplus D_2 \oplus D_3 \\ &= 1011\ 0100 \oplus 0010\ 1100 \oplus 1100\ 0110 \oplus 1101\ 0101 \\ &= 1000\ 1011 \ . \end{aligned}$$

Q may be generated using Equation 2 as follows:

$$Q = g^0 D_0 \oplus g^1 D_1 \oplus g^2 D_2 \oplus g^3 D_3 \ .$$

From the table in Fig. 2, $g^0 = 0000\ 0001$, $g^1 = 0000\ 0010$, $g^2 = 0000\ 0100$, and $g^3 = 0000\ 1000$.

$$\begin{aligned} \text{Therefore, } Q &= 0000\ 0001 * 1011\ 0100 \oplus 0000\ 0010 * 0010\ 1100 \oplus \\ &\quad 0000\ 0100 * 1100\ 0110 \oplus 0000\ 1000 * 1101\ 0101 \\ &= 1011\ 0100 \oplus 0101\ 1000 \oplus 0011\ 1111 \oplus 1110\ 0110 \\ &= 0011\ 0101 \ . \end{aligned}$$

[0018] The following example shows how to recover from two disk failures using the array generated above. In the first case, P and Q fail. In this case, P and Q are regenerated using Equations 1 and 2 as shown above. In the second case, Q and a data disk (D_a) fail. In this case, D_a may be regenerated using P and the remaining data disks via Equation 1. Q may then be regenerated using Equation 2. In the third case, P and a data disk (D_a) fail. In this case, D_a may be regenerated using Q, the remaining data disks, and the following equation:

$$D_a = (Q \oplus Q_a) * g^{-a} = (Q \oplus Q_a) * g^{255-a}, \text{ where}$$

$$Q_a = g^0 D_0 \oplus g^1 D_1 \oplus \ldots \oplus g^{a-1} D_{a-1} \oplus g^{a+1} D_{a+1} \ldots \oplus g^{n-1} D_{n-1} \ .$$

For example, suppose disk 2 fails. Then,

$$D_2 = (Q \oplus Q_2) \cdot g^{253} = (Q \oplus g^0 D_0 \oplus g^1 D_1 \oplus g^3 D_3) \cdot g^{253}$$

$= (0011\ 0101 \oplus 0000\ 0001 * 1011\ 0100 \oplus 0000\ 0010 * 0010\ 1100 \oplus 0000\ 1000) * g^{253}$

Using the table in Fig. 2, $g^{253} = 0100\ 0111$. Therefore,

$$D_2 = (0011\ 0101 \oplus 1011\ 0100 \oplus 0101\ 1000 \oplus 1110\ 0110) * 0100\ 0111$$
$$= 0011\ 1111 * 0100\ 0111$$
$$= 1100\ 0110\ .$$

P may then be regenerated using Equation 1, since all data blocks are now available.

[0019] In the fourth case, two data disks (D_a and D_b) fail. In this case, D_a and D_b may be regenerated using P and Q, the remaining data disks, and the following equations:

$$D_a = (g^{-a} * (Q \oplus Q_{ab}) \oplus g^{b-a} * (P \oplus P_{ab})) / (g^{b-a} \oplus 0000\ 0001)$$

$$D_b = D_a \oplus (P \oplus P_{ab}),\ \text{where}$$

$$P_{ab} = D_0 \oplus D_1 \oplus \ldots \oplus D_{a-1} \oplus D_{a+1} \ldots \oplus D_{b-1} \oplus D_{b+1} \ldots \oplus D_{n-1}$$

$$Q_{ab} = g^0 D_0 \oplus g^1 D_1 \oplus \ldots \oplus g^{a-1} D_{a-1} \oplus g^{a+1} D_{a+1} \ldots \oplus g^{b-1} D_{b-1} \oplus g^{b+1} D_{b+1} \ldots \oplus g^{n-1} D_{n-1}\ .$$

For example, assume that disks 1 and 3 failed. Then,

$$D_1 = (g^{-1} . (Q \oplus Q_{13}) \oplus g^{3-1} * (P \oplus P_{13})) / (g^{3-1} \oplus 0000\ 0001)$$
$$= (g^{254} * (Q \oplus Q_{13}) \oplus g^2 * (P \oplus P_{13})) / (g^2 \oplus 0000\ 0001)$$

$$Q \oplus Q_{13} = 0011\ 0101 \oplus 0000\ 0001 * 1011\ 0100 \oplus 0000\ 0100 * 1100\ 0110$$
$$= 0011\ 0101 \oplus 1011\ 0100 \oplus 0011\ 1111 = 1011\ 1110$$

$$P \oplus P_{13} = 1000\ 1011 \oplus 1011\ 0100 \oplus 1100\ 0110 = 1111\ 1001$$

From the table in Fig. 2, $g^{254} = 1000\ 1110$ and $g^2 = 0000\ 0100$. Therefore,

$$D_1 = ((1000\ 1110 * 1011\ 1110) \oplus (0000\ 0100 * 1111\ 1001)) / (0000\ 0100 \oplus 0000\ 0001)$$
$$= (0101\ 1111 \oplus 1100\ 0011) / (0000\ 0101)$$
$$= (1011\ 1100) / (0000\ 0101)$$
$$= 0010\ 1100\ .$$

$$D_3 = D_1 \oplus (P \oplus P_{13})$$

$$= 0010\ 1100 \oplus 1111\ 1001$$

$$= 1101\ 0101 .$$

[0020] Fig. 3 is a block diagram of a system 300 according to an embodiment of the invention. System 300 includes a multiplier 302 and one or more comparators, such as 304 and 306. In one embodiment, one or more of the comparators are XOR (Exclusive-OR) gates. System 300 may also include one or more buffers, such as 308 and 310. The buffer 308 stores the output from the comparator 306. The comparator 306 compares the data 320 read from the storage blocks shown in Fig. 1 with the output of buffer 308. In this way, the comparator 306 may be used to compute the P syndrome described above, which is the parity across a stripe.

[0021] The multiplier 302 multiplies the multiplicand 330 with the data 320 read from the storage blocks shown in Fig. 1. In one embodiment, the multiplicand 330 is a Galois field, such as shown in Table 200 of Fig. 2. The output of the multiplier 302 is compared to the output of the buffer 310 by comparator 304. In this way, a Galois field multiplication may be performed and the Q syndrome may be computed. The multiplier 302 may also be used to perform the various multiplication operations for the equations described above with respect to the four cases in which multiple disks fail. Data may be allowed to pass through the multiplier by setting the multiplicand 330 equal to one. A selector 312 may be used to select between the output of the multiplier 302 and the output of the comparator 304. In one embodiment, the selector 312 is a multiplexer (MUX).

[0022] System 300 may also include a divider 314 to be used to perform the division operations for the equations described above with respect to the four cases in which multiple disks fail. For example, in case four, the computation for regeneration of D_a has a division operation, which may be performed by divider 314. Data may be allowed to pass through the divider 314 by setting the divisor 340 equal to one. This may be desired when no division operation is required to be performed.

[0023] As shown in Fig. 3, the system 300 performs the generation of the P and Q syndromes in parallel. Other multiplication and division operations that are required may also be performed by system 300. A selector 316 may be used to select the desired output of the system. In one embodiment, the selector 316 is a multiplexer (MUX).

[0024] Fig. 4 illustrates a method for generating parity to aid in the recovery of data in one or more storage blocks according to one embodiment of the invention. At 400, a first parity factor is computed based on comparing data from one or more of the storage blocks. At 402, the data from one or more of the storage blocks is multiplied with a multiplication factor to generate a product. At 404, a second parity factor is computed based at least in part on the product. At 406, a selection is made between the first parity factor and the second parity factor. In one embodiment, the first parity factor is a P syndrome and the second parity factor is a Q syndrome as described above. In one embodiment, the second parity factor is further divided by a divisor. In one embodiment, the first parity factor and the second parity factor are buffered. In one embodiment, the first parity factor and the second parity factor are computed in parallel.

[0025] While the invention has been described in terms of several embodiments, those of ordinary skill in the art will recognize that the invention is not limited to the embodiments described, but can be practiced with modification and alteration within the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting.

CLAIMS

What is claimed is:

1. An apparatus comprising:

a first comparator to generate a first parity factor based on data from one or more storage blocks;

a multiplier to multiply the data from one or more of the storage blocks with a multiplication factor to generate a product;

a second comparator coupled to the multiplier to generate a second parity factor based at least in part on the product; and

a selector coupled to the first comparator and the second comparator to choose between the first parity factor and the second parity factor.

2. The apparatus of claim 1, further comprising a divider coupled to the second comparator to perform division operations on the second parity factor.

3. The apparatus of claim 1, further comprising a first buffer coupled to the first comparator to store the first parity factor.

4. The apparatus of claim 1, further comprising a second buffer coupled to the second comparator to store the second parity factor.

5. The apparatus of claim 1, wherein the first comparator is an XOR (Exclusive OR) gate.

6. The apparatus of claim 1, wherein the second comparator is an XOR (Exclusive OR) gate.

7. The apparatus of claim 1, wherein the multiplier performs Galois Field multiplication.

8. The apparatus of claim 1, wherein the first comparator generates a RAID (Redundant Array of Independent Disks) syndrome based on data from one or more of the storage blocks.

9. The apparatus of claim 8, wherein the second comparator generates a RAID (Redundant Array of Independent Disks) syndrome using an output from the multiplier.

10. The apparatus of claim 9, wherein the first comparator generates a RAID P-syndrome and the second comparator generates a RAID Q-syndrome.

11. A method comprising:

computing a first parity factor based on comparing data from one or more storage blocks;

multiplying the data from one or more of the storage blocks with a multiplication factor to generate a product;

computing a second parity factor based at least in part on the product; and selecting between the first parity factor and the second parity factor.

12. The method of claim 11, wherein computing a first parity factor comprises computing a RAID (Redundant Array of Independent Disks) syndrome based on data from one or more of the storage blocks.

13. The method of claim 11, wherein computing a second parity factor comprises computing a RAID (Redundant Array of Independent Disks) syndrome based at least in part on the product.

14. The method of claim 11, wherein multiplying the data from one or more of the storage blocks with a multiplication factor comprises multiplying the data from one or more of the storage blocks with a Galois field multiplicand.

15. The method of claim 11, further comprising dividing the second parity factor by a divisor.

16. The method of claim 11, further comprising storing the first parity factor in a first buffer.

17. The method of claim 11, further comprising storing the second parity factor in a second buffer.

18. The method of claim 11, wherein the first parity factor and the second parity factor are computed in parallel.

19. A system comprising:

one or more storage devices to store data; and

a parity generator coupled to the one or more storage devices to generate parity for the recovery of data, the parity generator including:

a first comparator to generate a first parity factor based on data in one or more of the storage devices;

a multiplier to multiply data from one or more of the storage devices with a multiplication factor to generate a product;

a second comparator coupled to the multiplier to generate a second parity factor based at least in part on the product; and

a selector coupled to the first comparator and the second comparator to choose between the first parity factor and the second parity factor.

20. The system of claim 19, further comprising a divider coupled to the second comparator to perform division operations on the second parity factor.

21. The system of claim 19, wherein the multiplier performs Galois Field multiplication.

22. The system of claim 19, wherein the first comparator and the second comparator operate in parallel.

23. The system of claim 19, wherein the first comparator generates a RAID (Redundant Array of Independent Disks) syndrome based on data from one or more of the storage devices.

24. The system of claim 19, wherein the second comparator generates a RAID (Redundant Array of Independent Disks) syndrome based at least in part on the product.

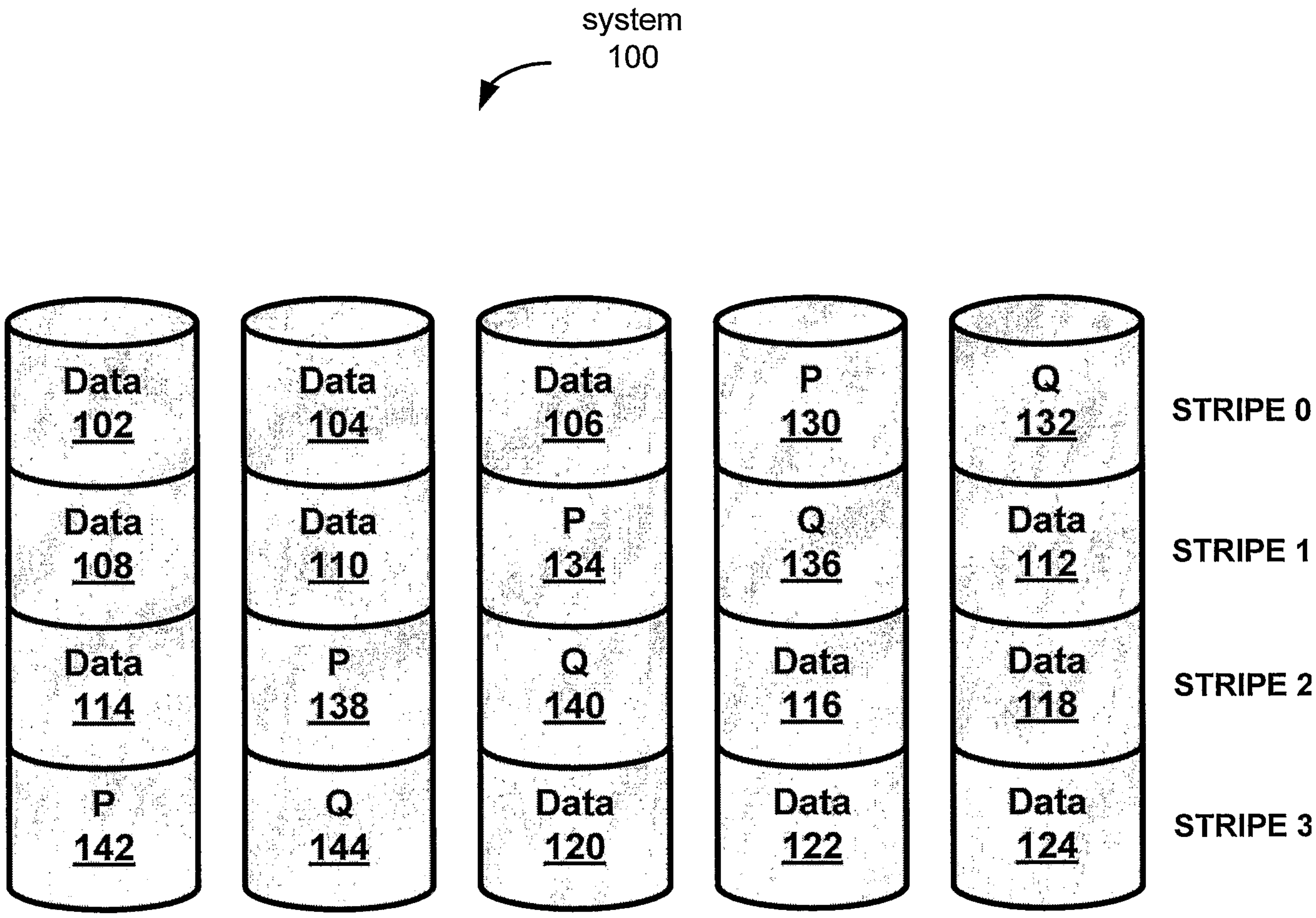
system
100
Data
102
Data
104
Data
106
P
130
Q
132
STRIPE 0
Data
108
Data
110
P
134
Q
136
Data
112
STRIPE 1
Data
114
P
138
Q
140
Data
116
Data
118
STRIPE 2
P
142
Q
144
Data
120
Data
122
Data
124
STRIPE 3

FIG. 1

Table
200

g^{i+8j}	i=	i=	i=	i=	i=	i=	i=	i=
j=	0000001	0000010	0000100	0000000	0010000	0100000	1000000	0000110
j=0	0011101	0111010	1110100	1100110	1000011	0001001	0010011	0100110
j=1	1001100	0010110	0101101	1011010	0111010	1110101	1100100	1000111
j=2	0000001	0000011	0000110	0001100	0011000	0110000	1100000	1001110
j=3	0010011	0100111	1001110	0010010	0100101	1001010	0011010	0110101
j=4	1101010	1011010	0111011	1110111	1100000	1001111	0010001	0100011
j=5	10001 0	0000010	0000101	0001010	0010100	0101000	1010000	0101110
j=6	1011101	0110100	1101001	1011100	0110111	1101111	1010000	0101111
j=7	1011111	0110000	1100001	1001100	0010111	0101111	1011110	0110010
j=8	1100101	1000100	0000111	0001111	0011110	0111100	1111000	1111110
j=9 0	1110011	1101001	1011101	0110101	1101011	1011000	0111111	1111111
j=0 1	1110000	1101111	1010001	0101101	1011011	0111000	1110001	1101100
j=1 1	1010111	0100001	1000011	0001000	0010001	010001 0	1000100	0000110
j=2 1	0001101	0011010	0110100	1101000	1011110	0110011	1100111	1000000
j=3 1	0001111	0011111	0111110	1111100	1110110	1100011	1001001	0011101
j=4 1	0111011	1110110	1100010	1001011	0011001	0110011	1100110	1000010
j=5 1	0001011	0010111	0101110	1011100	0110110	1101101	1010100	0100111
j=6 1	1001111	0010000	0100001	1000010	0001010	0010101	0101010	1010100
j=7 1	0100110	1001101	0010100	0101001	1010010	0101010	1010101	0100100
j=8 1	1001001	0011100	0111001	1110010	1101010	1011011	0111001	1110011
j=9 2	1101000	1011111	0110001	1100011	1001000	0011111	0111111	1111110
j=0 2	1110010	1101011	1011001	0111101	1111011	1111000	1111111	1110001
j=1 2	1101101	1010101	0100101	1001011	0010000	0110001	1100010	1001010
j=2 2	0011011	0110111	1101110	1010010	0101011	1010111	0100000	1000001
j=3 2	0001100	0011001	0110010	1100100	1000110	0000011	0000111	0001110
j=4 2	0011100	0111000	1110000	1101110	1010011	0101001	1010011	0101000
j=5 2	1010001	0101100	1011001	0111100	1111001	1111100	1110111	1100001
j=6 2	1001101	0010101	0101011	1010110	0100010	1000101	0000100	0001001
j=7 2	0010010	0100100	1001000	0011110	0111101	1111010	1111010	1111011
j=8 2	11110 1	1111101	1110101	1100101	1000101	0000101	0001011	0010110
j=9 3	0101100	1011000	0111110	1111101	1110100	1100111	1000001	0001101
j=0 3	0011011	0110110	1101100	1010110	0100011	1000111	0000000	1101110
1	0	0	0	1	1	0	1	1

FIG. 2

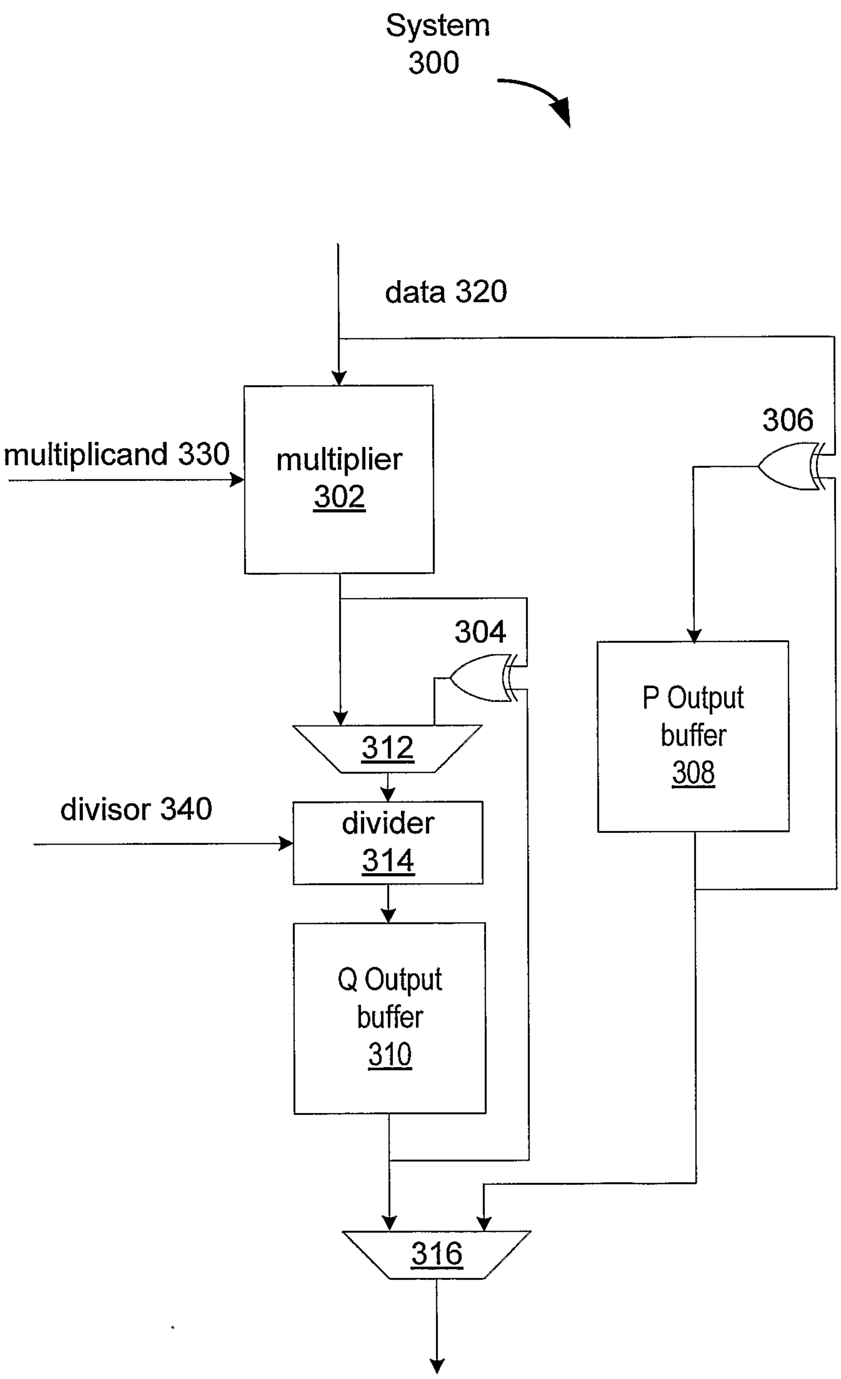
System
300
data 320
multiplicand 330
multiplier
302
306
304
P Output
buffer
308
312
divisor 340
divider
314
Q Output
buffer
310
316

FIG. 3

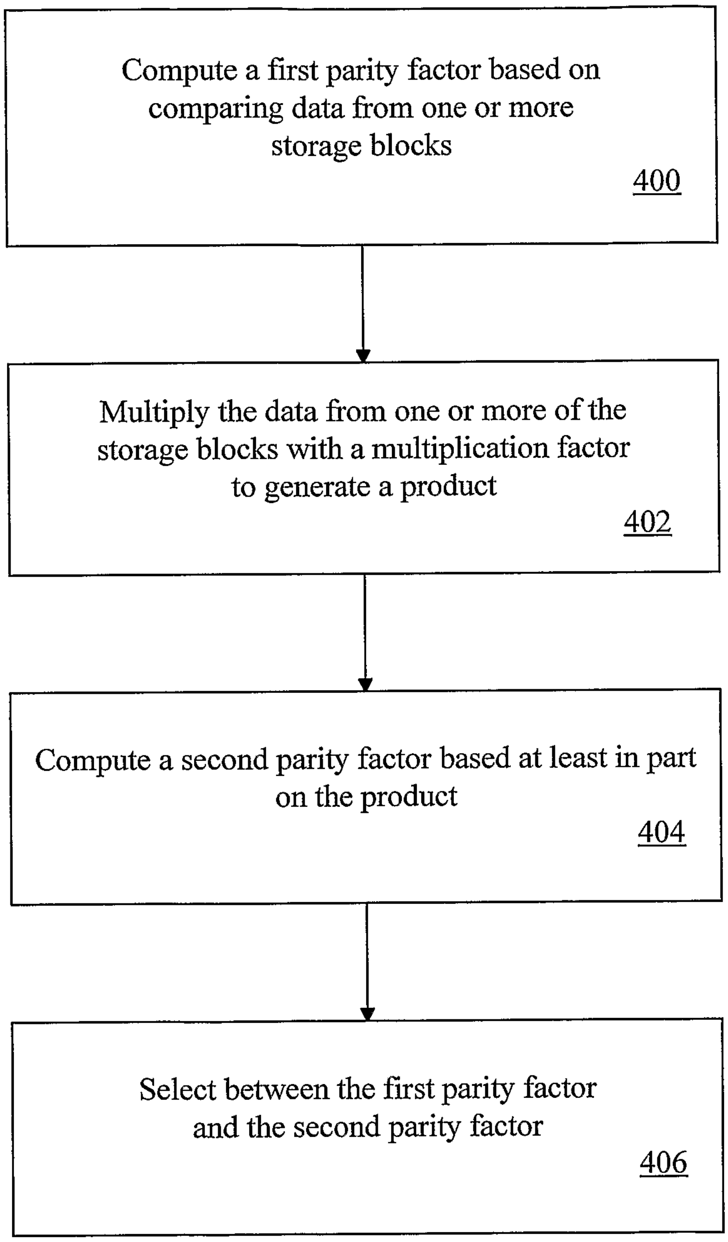
Compute a first parity factor based on comparing data from one or more storage blocks
400
Multiply the data from one or more of the storage blocks with a multiplication factor to generate a product
402
Compute a second parity factor based at least in part on the product
404
Select between the first parity factor and the second parity factor
406

FIG. 4