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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 December 2004 (02.12.2004)

PCT

(10) International Publication Number
WO 2004/105260 A2

(51) International Patent Classification[7]: H04B

(21) International Application Number:
PCT/US2004/009536

(22) International Filing Date: 29 March 2004 (29.03.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/440,330 16 May 2003 (16.05.2003) US

(71) Applicant *(for all designated States except US)*: **ANALOG DEVICES, INC.** [US/US]; Three Technology Way, Norwood, MA 02062 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **STEIN, Yosef** [IL/US]; 4 Turning Mill Road, Sharon, MA 02067 (US). **KABLOTSKY, Joshua, A.** [US/US]; 2191 Bay Road, Sharon, MA 02067 (US).

(74) Agents: **POIRIER, David, W.** et al.; Iandiorio & Teska, 260 Bear Hill Road, Waltham, MA 02451 (US).

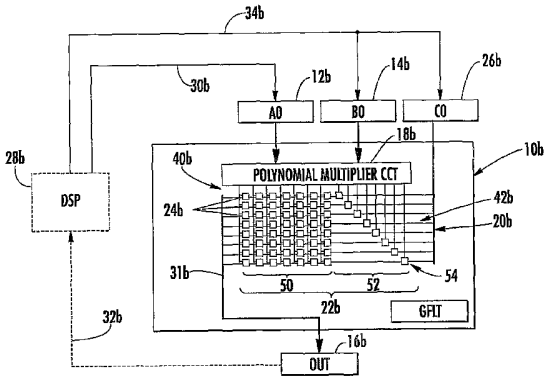
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, 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, 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, IT, LU, 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.

(54) Title: COMPOUND GALOIS FIELD ENGINE AND GALOIS FIELD DIVIDER AND SQUARE ROOT ENGINE AND METHOD



(57) Abstract: A Galois field divider engine and method inputs a 1 and a first Galois field element to a Galois field reciprocal generator to obtain an output, multiplies in a Galois field reciprocal generator a first Galois field element by a first element of the Galois field reciprocal generator for predicting the modulo remainder of the square of the polynomial product of an irreducible polynomial m-2 times where m is the degree of the Galois field to obtain the reciprocal of the first Galois field element, and multiplying in the Galois field reciprocal engine the reciprocal of the first Galois field element by a second Galois field element for predicting the modulo remainder of the polynomial product for an irreducible polynomial to obtain the quotient of the two Galois field elements in m cycles; in a broader sense the invention includes a compound Galois field engine for performing a succession of Galois field linear transforms on a succession of polynomial inputs to obtain an ultimate output where each input except the first is the output of the previous Galois field linear transform; Galois field square root is achieved by inputting a Galois field element to a Galois field square root generator to obtain an output which is squared in the Galois field square root generator to predict the modulo remainder of the square of the polynomial product of an irreducible polynomial m-1 times where m is the degree of the Galois field to obtain the square root of the Galois field to obtain the square root of the Galois field element in (m-1) cycles.

WO 2004/105260 A2

COMPOUND GALOIS FIELD ENGINE AND GALOIS FIELD DIVIDER AND SQUARE ROOT ENGINE AND METHOD

FIELD OF THE INVENTION

This invention relates to a Galois field divider engine and method and more generally to a compound Galois field engine for performing a succession of Galois field transforms in one transform operation.

RELATED APPLICATIONS

This application claims priority of U.S. Provisional application to Stein et al. entitled A COMPACT GALOIS FIELD MULTIPLIER, filed October 9, 2002 (AD-337J), U.S. Provisional application serial no. 60/334,662, filed November 30, 2001 to Stein et al., entitled GF2-ALU (AD-239J); U.S. Provisional application serial no. 60/334,510 filed November 20, 2001 to Stein et al., entitled PARALLEL GALOIS FIELD MULTIPLIER (AD-240J); U.S. Provisional application serial no. 60/341,635, filed December 18, 2001 to Stein et al., entitled GALOIS FIELD MULTIPLY ADD (MPA) USING GF2-ALU (AD-299J); U.S. Provisional application serial no. 60/341,737, filed December 18, 2001, to Stein et al., entitled PROGRAMMABLE GF2-ALU LINEAR FEEDBACK SHIFT REGISTER – INCOMING DATA SELECTION (AD-300J). This application further claims priority of U.S. Patent application serial no. 10/395,620 filed March 24, 2003 to Stein et al., entitled COMPACT GALOIS FIELD MULTIPLIER ENGINE (AD-337J); U.S. Patent application serial no. 10/051,533 filed January 18, 2002 to Stein et al., entitled GALOIS FIELD LINEAR TRANSFORMERER (AD-239J); U.S. Patent application serial no. 10/060,699 filed January 30, 2002 to Stein et al., entitled GALOIS FIELD MULTIPLIER SYSTEM (AD-240J); U.S. Patent application serial no. 10/228,526 filed

August 26, 2002 to Stein et al., entitled GALOIS FIELD MULTIPLY/MULTIPLY – ADD/MULTIPLY ACCUMULATE (AD-299J); and U.S. Patent application serial no. 10/136,170, filed May 1, 2002 to Stein et al., entitled RECONFIG.URABLE INPUT GALOIS FIELD LINEAR TRANSFORMERER SYSTEM (AD-300J).

BACKGROUND OF THE INVENTION

In certain applications such as encryption and error control coding, it is necessary to perform arithmetic operations, e.g., add, subtract, square root, multiply, and divide over Galois fields. Any such operation between any two members in a Galois field will result in an output (sum, difference, square root, product, quotient) which is another value in the same Galois field. The number of elements in a Galois field is 2^m where m is the degree of the field. For example, $GF(2^4)$ would have sixteen different elements in it; $GF(2^8)$ would have 256. A Galois field is generated from an irreducible polynomial in a particular power. Each Galois field of a particular degree will have a number of irreducible polynomials form each of which may be devised a different field using the same terms but in a different order.

Division over a Galois field is done by multiplying the dividend by the reciprocal of the divisor. This divisor reciprocal can be generated in a number of ways. One way is to have a stored look-up table of reciprocals where the divisor is the address for the table. One problem with this approach is that for each field of each irreducible polynomial there must be stored a separate table. In addition, the tables can only be accessed in serial: if parallel operations are required a copy of each table must be provided for each parallel operation. Another approach is to multiply each of the stored Galois field elements by the particular divisor. The value that produces a product of one is then the reciprocal of the

particular divisor. Once again all of the values have to be stored and in multiple copies if parallel operation is contemplated. And, a Galois field multiplier is required just to accomplish the retrieval. A third approach uses two linear feedback shift registers (LFSR) each configured to generate a selected Galois field of a particular irreducible polynomial. The first is initialized to the divisor; the second is initialized to "1". Starting from the divisor value the two are clocked synchronously. When the product of the first LFSR equals "1" the divisor has been multiplied by its reciprocal. The product of the second LFSR at that moment is the Galois field element that is the reciprocal of the divisor. One problem with this approach is that for each Galois field of each irreducible polynomial for each degree a different pair of LFSRs is required. In both, the second look-up table approach, above, and the LFSR approach the search for the reciprocal requires up to 2^m-1 iterations.

BRIEF SUMMARY OF THE INVENTION

It is therefore an object of this invention to provide an improved Galois field divider engine and method.

It is a further object of this invention to provide such an improved Galois field divider engine which can complete the search for the divisor reciprocal in m-1 iterations.

It is a further object of this invention to provide such an improved Galois field divider engine which can be easily reconfigured to accommodate different irreducible polynomial Galois fields of different degrees.

It is a further object of this invention to provide such an improved Galois field divider engine which can function to generate both the divisor reciprocal and multiply it by the dividend.

It is a further object of this invention to provide such an improved Galois field divider engine which requires less power and less area.

It is a further object of this invention to provide more generally an improved, compound Galois field engine for performing a succession of Galois field transforms in one transform operation.

The invention results from the realization that such an improved Galois field division engine and method which is smaller, faster, and more efficient can be achieved with a Galois field reciprocal generator and an input selection circuit for initially inputting a 1 and a first Galois field element to the Galois field reciprocal generator to obtain an output, subsequently multiplying in the Galois field reciprocal generator a first Galois field element by the output of the Galois field reciprocal generator for predicting the modulo remainder of the square of the polynomial product of an irreducible polynomial m-2 times where m is the degree of the Galois field, to obtain the reciprocal of the first Galois field element and multiplying in the Galois field reciprocal engine the reciprocal of the first Galois field element by a second Galois field element for predicting the modulo reminder of the polynomial product for an irreducible polynomial to obtain the quotient of the two Galois field elements in m cycles.

It was also realized, more generally, that an improved compound Galois field engine for performing a succession of Galois field linear transforms on a succession of polynomial inputs to obtain an ultimate output where each input, except the first, is the output of the previous Galois field linear transform can be accomplished with an input circuit for providing a first input and a Galois field linear transformer having a matrix of cells responsive to the first input and configured to, in one transform, immediately predict the modulo remainder of the succession of Galois field linear transforms of an irreducible

Galois field polynomial to obtain the ultimate output of the Galois field linear transform directly from the first input.

This invention features a Galois field divider engine including a Galois field reciprocal generator and an input selection circuit for initially inputting a 1 and a first Galois field element to the Galois field reciprocal generator to obtain an output, subsequently multiplying in the Galois field reciprocal generator a first Galois field element by the output of the Galois field reciprocal generator for predicting the modulo remainder of the square of the polynomial product of an irreducible polynomial m-2 times, where m is the degree of the Galois field, to obtain the reciprocal of the first Galois field element and multiplying in the Galois field reciprocal engine the reciprocal of the first Galois field element by a second Galois field element for predicting the modulo remainder of the polynomial product, for an irreducible polynomial to obtain the quotient of the two Galois field elements in m cycles.

In a preferred embodiment, the reciprocal generator may include first and second Galois field multipliers. The first Galois field multiplier may include a first polynomial multiplier circuit and a first Galois field linear transformer. The first Galois field linear transformer may include a matrix of cells. The first Galois field linear transform may include a matrix section and a unity matrix section. The second Galois field multiplier may include a second polynomial multiplier circuit and a second Galois field linear transformer. The second Galois field linear transformer may include a matrix of cells. The second Galois field linear transformer matrix of cells may include a matrix section and a unity matrix section. The output of the first Galois field multiplier may be fed to both multiply inputs of the second Galois field linear multiplier to provide the square of that output. The Galois field reciprocal generator may include a Galois field multiplier

including a first polynomial multiplier and a first Galois field transformer and a second Galois field transformer for calculating the square of the first Galois field multiplier output. The second Galois field transformer may be approximately one half the size of the first Galois field transformer. The first and second Galois field transformers each may include a matrix of cells and the second Galois field transformer may include approximately one half the number of cells of the first Galois field transformer. The Galois field reciprocal engine may include a Galois field multiplier and a program circuit for programming the Galois field multiplier to perform a compound multiply-square operation for m-2 times followed by a multiply operation.

The invention also features in a broader sense a compound Galois field engine for performing a succession of Galois field linear transforms on a succession of polynomial inputs to obtain an ultimate output where each input except the first is the output of the previous Galois field linear transform. There is an input circuit for providing a first input and a Galois field linear transformer having a matrix of cells responsive to the first input and configured to, in one transform, immediately predict the modulo remainder of the succession of Galois field linear transforms of an irreducible Galois field polynomial to obtain the ultimate output of the Galois field linear transform directly from the first input.

This invention also features a method of Galois field division including initially inputting a 1 and a first Galois field element to a Galois field reciprocal generator to obtain an output, multiplying in the Galois field reciprocal generator a first Galois field element by the output of the Galois field reciprocal generator for predicting the modulo remainder of the square of the polynomial product of an irreducible polynomial m-2 times where m is the degree of the Galois field to obtain the reciprocal of the first Galois field element, and multiplying in the Galois field reciprocal engine the reciprocal of the first

Galois field element by a second Galois field element for predicting the modulo remainder of the polynomial product for an irreducible polynomial to obtain the quotient of the two Galois field elements in m cycles.

This invention also features a Galois field square root engine including a Galois field square root generator and an input circuit for inputting a Galois field element to the Galois field square root generator to obtain the square root of the Galois field elements in one cycle.

In a preferred embodiment, the Galois field square root engine may include a Galois field multiplier, and a program circuit for programming the Galois field multiplier to perform a compound square operation of m-1 times in one cycle.

The invention also features a Galois field square root method including inputting a Galois field element to a Galois field square root generator to obtain an output and squaring in the Galois field square root generator the output of the Galois field square root generator for predicting the modulo remainder of the square of the polynomial product of an irreducible polynomial m-1 times where m is the degree of the Galois field to obtain the square root of the Galois field element in (m-1) cycles.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects, features and advantages will occur to those skilled in the art from the following description of a preferred embodiment and the accompanying drawings, in which:

Fig. 1 is a functional block diagram of a compact Galois field multiplier engine according to the invention;

Fig. 2 is a more detailed functional block diagram of a conventional Galois field

multiplier engine according to the invention;

Fig. 3 is a more detailed functional block diagram of the compact Galois field multiplier engine of Fig. 1 displaying the reduced size Galois field linear transformer unity matrix feature of the invention;

Fig. 4 is a schematic of a typical programmable X-OR circuit cell for the matrix of the Galois field linear transformer circuit of Figs. 2 and 3;

Fig. 5 is a simplified schematic diagram of the Galois field linear transformer circuit of Figs. 3 and 9 illustrating the programming of the matrix section and unity matrix section cells according to the invention for a particular polynomial of power eight;

Fig. 6 is a simplified schematic diagram of the Galois field linear transformer circuit of Figs. 3 and 9 illustrating the programming of the matrix section and unity matrix section cells according to the invention for another polynomial of power eight;

Fig. 7 is a simplified schematic diagram of the Galois field linear transformer circuit of Figs. 3 and 9 illustrating the programming of the matrix section and unity matrix section cells according to the invention for yet another polynomial of power four;

Fig. 8 is a simplified schematic diagram of the Galois field linear transformer circuit of Figs. 3 and 9 illustrating the programming of a second matrix section as a sparse matrix for supporting polynomial powers between half (4) powers and full (8) powers in this particular embodiment;

Fig. 9 is a more detailed block diagram of a compact Galois field multiplier engine of Fig. 1 incorporating both the reduced size matrix and the reduced hardware and localized bus features of the invention;

Fig. 10 is a block diagram of Galois field multiplier engine according to the invention employing a number of Galois field linear transformer units;

Fig. 11 is a schematic view of a polynomial multiplier usable in Figs. 2, 3, 5 and 9;

Fig. 12 is an illustration the transfer function for the polynomial multiplier of Fig. 11;

Fig. 13 is a simplified schematic block diagram of a divider engine according to this invention;

Fig. 14 is a more detailed view of the Galois field multiplier and squarer of Fig. 13;

Fig. 15 is a chart of the reduced transfer function values for the polynomial multiplier of Fig. 12;

Fig. 16 is a view of a Galois field multiplier and squarer similar to that of Fig. 14 implementing the reduced transfer function of Fig. 15;

Fig. 17 is a schematic illustration of the pattern of enabled cells of the Galois field linear transformer of Fig. 14;

Fig. 18 is a schematic illustration of the pattern of enabled cells of the Galois field linear transformer of Fig. 16 utilizing the reduced transfer function;

Fig. 19 is a schematic illustration of the pattern of enabled cells of a compound Galois linear engine for compound Galois field engine for performing a succession of Galois field linear transforms on a succession of polynomial inputs to obtain an ultimate output e.g. division according to a more general feature of this invention;

Fig. 20 is a simplified schematic diagram of a compound Galois field engine utilizing the Galois field transform illustrated in Fig. 19;

Fig. 21 is a flow chart of the Galois field divider method according to this invention;

Fig. 22 is a schematic block diagram of a square root engine according to this

invention;

Fig. 23 is a schematic illustration of the pattern of enabled cells of a compound Galois field linear engine for performing a succession of Galois field linear transforms on a succession of polynomial inputs as shown in Fig. 22 to obtain an ultimate output e.g. square root according to the more general feature of this invention;

Fig. 24 is a flow chart of the Galois field square root method according to this invention; and

Fig. 25 is a simplified block diagram of a compound Galois field engine according to this invention.

DISCLOSURE OF THE PREFERRED EMBODIMENT

Aside from the preferred embodiment or embodiments disclosed below, this invention is capable of other embodiments and of being practiced or being carried out in various ways. Thus, it is to be understood that the invention is not limited in its application to the details of construction and the arrangements of components set forth in the following description or illustrated in the drawings.

Before disclosing the compound Galois field engine and the divisor engine and method of this invention an explanation of Galois field transformers and multipliers is presented for a better understanding.

A Galois field GF(n) is a set of elements on which two binary operations can be performed. Addition and multiplication must satisfy the commutative, associative and distributive laws. A field with a finite number of elements is a finite field. An example of a binary field is the set {0,1} under modulo 2 addition and modulo 2 multiplication and is denoted GF(2). The modulo 2 addition and multiplication operations are defined by the

tables shown in the following illustration. The first row and the first column indicate the inputs to the Galois field adder and multiplier. For e.g. 1+1=0 and 1*1=1.

Modulo 2 Addition (XOR)

+	0	1
0	0	1
1	1	0

Modulo 2 Multiplication (AND)

*	0	1
0	0	0
1	0	1

In general, if p is any prime number then it can be shown that GF(p) is a finite field with p elements and that GF(p^m) is an extension field with p^m elements. In addition, the various elements of the field can be generated as various powers of one field element, β, by raising it to different powers. For example GF(256) has 256 elements which can all be generated by raising the primitive element, β, to the 256 different powers.

In addition, polynomials whose coefficients are binary belong to GF(2). A polynomial over GF(2) of degree m is said to be irreducible if it is not divisible by any polynomial over GF(2) of degree less than m but greater than zero. The polynomial F(X) = X^2+X+1 is an irreducible polynomial as it is not divisible by either X or X+1. An

irreducible polynomial of degree m which divides $X^{2m-1}+1$, is known as a primitive polynomial. For a given m, there may be more than one primitive polynomial. An example of a primitive polynomial for m=8, which is often used in most communication standards is $F(X) = 0x11d = x^8+x^4+x^3+x^2+1$.

Galois field addition is easy to implement in software, as it is the same as modulo addition. For example, if 29 and 16 are two elements in $GF(2^8)$ then their addition is done simply as an XOR operation as follows: 29 (11101) $\oplus$ 16(10000) = 13(01101).

Galois field multiplication on the other hand is a bit more complicated as shown by the following example, which computes all the elements of $GF(2^4)$, by repeated multiplication of the primitive element β. To generate the field elements for $GF(2^4)$ a primitive polynomial G(x) of degree m = 4 is chosen as follows $G(x) = X^4+X+1$. In order to make the multiplication be modulo so that the results of the multiplication are still elements of the field, any element that has the fifth bit set is brought into a 4-bit result using the following identity $F(\beta)= \beta^4 + \beta + 1= 0$. This identity is used repeatedly to form the different elements of the field, by setting $\beta^4 = 1+ \beta$. Thus the elements of the field can be enumerated as follows:

$\{0, 1, \beta, \beta^2, \beta^3, 1+\beta, \beta+\beta^2, \beta^2+\beta^3, 1+\beta+\beta^3,\ldots 1+\beta^3,\}$

since β is the primitive element for $GF(2^4)$ it can be set to 2 to generate the field elements of $GF(2^4)$ as {0,1,2,4,8,3,6,12,11…9}.

It can be seen that Galois field polynomial multiplication can be implemented in two basic steps. The first is a calculation of the polynomial product c(x) = a(x)*b(x) which is algebraically expanded, and like powers are collected (addition corresponds to an XOR operation between the corresponding terms) to give c(x).

For example $c(x) = (a_3x^3+a_2x^2+a_1x^1+a_0)*(b_3x^3+b_2x^3+b_1x^1+b_0)$

$c(x) = c_6x^6+c_5x^5+c_4x^4+c_3x^3+c_2x^2+c_1x^1+c_0$ where:

Chart I

$c_0 = a_0*b_0$
$c_1 = a_1*b_0 \oplus a_0*b_1$
$c_2 = a_2*b_0 \oplus a_1*b_1 \oplus a_0*b_2$
$c_3 = a_3*b_0 \oplus a_2*b_1 \oplus a_1*b_2 \oplus a_0*b_3$
$c_4 = a_3*b_1 \oplus a_2*b_2 \oplus a_1*b_3$
$c_5 = a_3*b_2 \oplus a_2*b_3$
$c_6 = a_3*b_3$

The second is the calculation of $d(x) = c(x)$ modulo $p(x)$.

To illustrate, multiplications are performed with the multiplication of polynomials modulo an irreducible polynomial. For example: (if $m(x) = x^8+x^4+x^3+x+1$)

{57}*{83} = {c1} because,

First Step

$$(x^6+x^4+x^2\ x+1) + (x^7+x+1) = x^{13}\oplus x^{11}\oplus x^9\oplus x^8\oplus x^7$$
$$x^7\oplus x^5\oplus x^3\oplus x^2\oplus x$$
$$x^6\oplus x^4\oplus x^2\oplus x\oplus x$$
$$= x^{13}\oplus x^{11}\oplus x^9\oplus x^8\oplus x^6\oplus x^5\oplus x^4\oplus x^3\oplus 1$$

Second Step

$$x^{13}+x^{11}+x^9+x^8+x^6+x^5\ x^4+x^3+1 \text{ modulo } (x^8+x^4+x^3+x+1)$$
$$= x^7+x^6+1$$

An improved Galois field multiplier system 10, foreclosing on this approach includes a multiplier circuit for multiplying two polynomials a_0-a_7 in an A register with the polynomial b_0-b_7 in an B register with coefficients over a Galois field to obtain their product is given by the fifteen-term polynomial c(x) defined as Chart II. The multiplier circuit actually includes a plurality of multiplier cells.

Chart II

c14 = a7*b7
c13 = a7*b6 ⊕ a6*b7
c12 = a7*b5 ⊕ a6*b6 ⊕ a5*b7
c11 = a7*b4 ⊕ a6*b5 ⊕ a5*b6 ⊕ a4*b7
c10 = a7*b3 ⊕ a6*b4 ⊕ a5*b5 ⊕ a4*b6 ⊕ a3*b7
c 9 = a7*b2 ⊕ a6*b3 ⊕ a5*b4 ⊕ a4*b5 ⊕ a3*b6 ⊕ a2*b7
c 8 = a7*b1 ⊕ a6*b2 ⊕ a5*b3 ⊕ a4*b4 ⊕ a3*b5 ⊕ a2*b6 ⊕ a1*b7
c 7 = a7*b0 ⊕ a6*b1 ⊕ a5*b2 ⊕ a4*b3 ⊕ a3*b4 ⊕ a2*b5 ⊕ a1*b6 ⊕ a0*b7
c 6 = a6*b0 ⊕ a5*b1 ⊕ a4*b2 ⊕ a3*b3 ⊕ a2*b4 ⊕ a1*b5 ⊕ a0*b6
c 5 = a5*b0 ⊕ a4*b1 ⊕ a3*b2 ⊕ a2*b3 ⊕ a1*b4 ⊕ a0*b5;
c 4 = a4*b0 ⊕ a3*b1 ⊕ a2*b2 ⊕ a1*b3 ⊕ a0*b4
c 3 = a3*b0 ⊕ a2*b1 ⊕ a1*b2 ⊕ a0*b3
c 2 = a2*b0 ⊕ a1*b1 ⊕ a0*b2
c 1 = a1*b0 ⊕ a0*b1
c 0 = a0*b0

The operation of a Galois field multiplier system is explained in U.S. Patent Application to Stein et al. entitled GALOIS FIELD MULTIPLIER SYSTEM [AD-240J] 10/060,699 filed January 30, 2002 which is incorporated herein in its entirety by this reference.

Each of the fifteen polynomial c(x) term includes an AND function as represented by an * and each pair of terms are combined with a logical exclusive OR as indicated by a ⊕. This product as represented in Chart II is submitted to a Galois field linear transformer circuit which may include a number of Galois field linear transformer units each composed of 15x8 cells, which respond to the product produced by the multiplier circuit to predict the modulo remainder of the polynomial product for a predetermined irreducible polynomial. The A_0, B_0 multiplication is performed in a first unit the A_1, B_1 in a second unit, the A_2, B_2 in a third unit, and the A_n, B_n in the last unit. The operation of a Galois field linear transformer circuit and each of its transformer units is explained in U.S. Patent Application to Stein et al. entitled GALOIS FIELD LINEAR TRANSFORMER[AD-239J] Serial No. 10/051,533 with a filing date of January 18, 2002, which is incorporated herein in its entirety by this reference. Each of the Galois field linear transformer units predicts the

modulo remainder by dividing the polynomial product by an irreducible polynomial. That irreducible polynomial may be, for example, anyone of those shown in Chart III.

Chart III

:GF(2^1)
0x3 (x+1)

:GF(2^2)
0x7 (x^2+x+1)

:GF(2^3)
0xB (x^3+x+1)
0xD (x^3+x^2+1)

:GF(2^4)
0x13 (x^4+x+1)
0x19 (x^4+x^3+1)

:GF(2^5)
0x25 (x^5+x^2+1)
0x29 (x^5+x^3+1)
0x2F (x^5+x^3+x^2+x+1)
0x37 (x^5+x^4+x^2+x+1)
0x3B (x^5+x^4+x^3+x+1)
0x3D (x^5+x^4+x^3+x^2+1)

:GF(2^6)
0x43 (x^6+x+1)
0x5B (x^6+x^4+x^3+x+1)
0x61 (x^6+x^5+1)
0x67 (x^6+x^5+x^2+x+1)
0x6D (x^6+x^5+x^3+x^2+1)
0x73 (x^6+x^5+x^4+x+1)

:GF(2^7)
0x83 (x^7+x+1)
0x89 (x^7+x^3+1)
0x8F (x^7+x^3+x^2+x+1)
0x91 (x^7+x^4+1)
0x9D (x^7+x^4+x^3+x^2+1)
0xA7 (x^7+x^5+x^2+x+1)
0xAB (x^7+x^5+x^3+x+1)
0xB9 (x^7+x^5+x^4+x^3+1)
0xBF (x^7+x^5+x^4+x^3+x^2+x+1)
0xC1 (x^7+x^6+1)
0xCB (x^7+x^6+x^3+x+1)

0xD3 $(x^7+x^6+x^4+x+1)$
0xE5 $(x^7+x^6+x^5+x^2+1)$
0xF1 $(x^7+x^6+x^5+x^4+1)$
0xF7 $(x^7+x^6+x^5+x^4+x^2+x+1)$
0xFD $(x^7+x^6+x^5+x^4+x^3+x^2+1)$

:GF(2^8)
0x11D $(x^8+x^4+x^3+x^2+1)$
0x12B $(x^8+x^5+x^3+x+1)$
0x12D $(x^8+x^5+x^3+x^2+1)$
0x14D $(x^8+x^6+x^3+x^2+1)$
0x15F $(x^8+x^6+x^4+x^3+x^2+x+1)$
0x163 $(x^8+x^6+x^5+x+1)$
0x165 $(x^8+x^6+x^5+x^2+1)$
0x169 $(x^8+x^6+x^5+x^3+1)$
0x171 $(x^8+x^6+x^5+x^4+1)$
0x187 $(x^8+x^7+x^2+x+1)$
0x18D $(x^8+x^7+x^3+x^2+1)$
0x1A9 $(x^8+x^7+x^5+x^3+1)$
0x1C3 $(x^8+x^7+x^6+x+1)$
0x1CF $(x^8+x^7+x^5+x^3+x^2+x+1)$
0x1E7 $(x^8+x^7+x^6+x^5+x^2+x+1)$
0x1F5 $(x^8+x^7+x^5+x^4+x^2+1)$

The Galois field multiplier presented here GF(2^8) is capable of performing with powers 2^8 and powers 2^4 and under as shown in Chart III.

An example of the GF multiplication occurs as follows:

Before GF( ) multiplication;
Polynomial 0x11d

45 23 00 01h
GF( )
57 34 00 01h
xx xx xx xxh

After GF9( ) multiplication;
Polynomial 0x11d

45 23 00 01h
GF( )
57 34 00 01h
72 92 00 01h

There is shown in Fig. 1 a compact Galois field multiplier engine 10 accompanied by an A input register 12, B input register 14 and an output register 16. Compact Galois field engine 10 is capable of a number of different operations, including multiply, multiply-add and multiply-accumulate.

Conventional Galois field multiplier engine 10a, Fig. 2, requires three registers, A

register 12a, B register 14a and C register 26a. The burden of these registers must be carried by the associated digital signal processor (DSP) core 28 and require extensive external bus work. In addition to bus 30, for supplying data to A register 12a, bus 34 for supplying data to B register 14a and bus 36 for supplying data to C register 26a, there is required a bus 32 for feeding back the output from register 16a to the digital signal processor 28 and bus 34 or bus 36 for feeding back that output from digital signal processor 28 to B register 14a or C register 26a. Bus 31 connects the output of Galois field linear transformer circuit 20 and output register 16a. Thus polynomial multiplier circuit 18 can provide to the multiple input 40 of matrix 22 of Galois field linear transformer circuit 20 the proper values in conjunction with the values fed from C register 26a to the adder input 42 of matrix 22 to perform multiply, multiply-add and multiply-accumulation functions. Matrix 22 is shown here as an eight by fifteen matrix for supporting multiplication of polynomials of power eight but may be made larger or smaller, containing more or fewer cells 24, depending upon the power of the polynomial to be serviced.

The number of cells 24b per row, Fig. 3, of matrix 22b of Galois field linear transformer circuit 20b in engine 10b maybe reduced by nearly half, by configuring matrix 22b into two matrix sections, a matrix section 50 and a unity matrix section 52. The unity matrix section requires only one set of cells 54 wherein these unity matrix section cells represent the prediction of the remainder when the output of the multiplier circuit is a polynomial with a power less than the power of the irreducible polynomial. Thus in Fig. 3 where the irreducible polynomial has a power of eight any polynomial of less than eight will not exceed the modulo and will be passed right through the matrix, thus the absent cells in unity matrix section 52 are unnecessary. This saves nearly half of the cells

required for the matrix 22b resulting in a smaller, simpler and faster engine.

Each cell 24b, Fig. 4, may include an AND circuit 100 and an exclusive OR circuit 102. There is a data input 104 and an enable input 106. Exclusive OR circuit 102 provides an output on line 108 to the input of the next exclusive OR circuit and receives at its input 110 the output from the previous exclusive OR circuit, except for the last exclusive OR circuit whose output is connected to the output of the matrix and the first exclusive OR circuit whose input is connected to the adder input 42b, Fig. 3, or 42g, Fig. 9. An enable signal on line 106 enables the data on line 104 to pass through AND gate 100 and to be exclusively ORed by exclusive OR circuit 102 with the input on line 110. The lack of an enabling signal on line 106 simply passes the input on line 110 through the exclusive OR gate 102 to output line 108. An enabling signal on line 106 enables cell 24. In this manner the entire matrix maybe reconfigured for any particular irreducible polynomial.

The efficacy of engine 10b, Fig. 3, can be understood by choosing an irreducible polynomial from Chart III, supra, and implementing it by enabling the necessary cells. For example, to implement the first polynomial of power eight designated 0x11d representing the irreducible polynomial $x^8 + x^4 + x^3 + x^2 + 1$, the enabled cells, indicated generally at 24cc, form a unity matrix 52c, Fig. 5, with a line of cells 54c as previously depicted in Fig. 3. When choosing the second irreducible polynomial from Chart III, 0x12b, the irreducible polynomial $x^8 + x^5 + x^3 + x + 1$ produces a pattern of enabled cells 24dd, Fig. 6, in matrix section 50d and unity matrix 52d where once again the unity matrix section 52d results in a line of enabled cells 54d.

The reduction in the number of required cells is not limited to only polynomials having the same power as the irreducible polynomial. It also applies to any of those

having the power of one half or less of the power of the irreducible polynomial. For example, the eight by fifteen matrix 22b, shown in Fig. 3 and referred to by way of explanation in Figs. 5 and 6 could also support polynomials to the power of one, two, three, or four, but not powers of five, six and seven, if the irreducible polynomial power was sixteen the matrix that supported it could also support polynomials up to eight, but not nine through fifteen. If it were the power of thirty-two it could support polynomials of thirty-two power and up to sixteen, but not seventeen through thirty-one. For example, as shown in Fig. 7 for an irreducible polynomial of the fourth power both the matrix section 50e and unity matrix section 52e become smaller and can be implemented anywhere within matrix 22e. Here the matrix section 50e has a plurality of enabled cells 24ee along with the enabled cells in unity matrix 52e which now has a smaller line of enabled cells 54e, making up the unity matrix section 52e.

If it is desirable to service the intermediate polynomials of power five, six and seven the unity matrix section can be replaced with a sparse matrix section 52f, Fig. 8, wherein additional lines of enabled cells 54ff, 54fff, 54fffff, can be employed to support polynomials of power seven, six and five respectively. But it is somewhat less of a reduction in the size of the matrix and required number of cells.

The number of input registers can be reduced from three to two and the number of external buses relied upon to communicate with the digital signal processor (DSP) 28g, Fig. 9, can be reduced and localized to be internal of the engine 10g itself. Thus, as shown in Fig. 9, there are but two input registers A 12g and B 14g and the feedback from output 31g does not need to go through DSP 28g but goes directly, locally, on engine 10g through internal bus 60 to multiplier input selection circuit 62 and adder input selection circuit 64. Digital signal processor 28g need only provide control signals on line 66 to

multiplier input selection circuit 62 and on line 68 to adder input selection circuit 64. Thus in the multiply mode, multiplier input selection circuit 62, passes an input from B register 14g to polynomial multiplier circuit 18g while adder input selection circuit 64 provides an additive identity level, in this case, a ground level 70 to the adder input 42g of Galois field linear transformer circuit 20g. In the multiply-add mode digital signal processor 28 instructs multiplier input selection circuits 62 to feed back the output from matrix 22g over line 60 to polynomial multiplier circuit 18g and instructs adder input selection circuits 64 to pass the polynomial in B register 14g to the adder input 42g of Galois field linear transformer circuit 20g. In the multiply-accumulate mode digital signal processor 28g instructs multiplier input selection circuit 62 to deliver the polynomial from B register 14g to polynomial multiplier circuit 18g and instructs adder input selection circuit 64 to feed back the output on line 60 of Galois field linear transformer circuit 20g.

Another feature is the reconfigurability of Galois field linear transformer circuit 20g by virtue of the selective enablement of cells 24g. Reconfigurable control circuit 80 selectively enables the ones of cells 24g required to implement the coefficients of the selected irreducible polynomial and itself can be reduced in size since the number of cells it needs to control has been reduced.

The operation of a reconfigurable input Galois field linear transformer circuit is explained in U.S. Patent Application Serial No. 10/136,170, filed May 1, 2002 to Stein et al., entitled RECONFIGURABLE INPUT GALOIS FIELD LINEAR TRANSFORMERER SYSTEM (AD-300J) and all its priority applications and documents which are incorporated herein in their entirety by this reference.

Although thus far for the sake of simplicity the explanation has been with respect to only one engine, a number of the engines may be employed together as shown in Fig.

10 where each engine has a multiplier circuit 10h, 10i, 10j, 10k... 10n and a Galois field linear transformer 20h, 20i, 20j, 20k... 20n circuit. With a single central reconfigurable control circuit 80^1 controlling them all. These engines can share the same wide [32, 64, 128]bit A and B registers were each operates on a different 8bit (Byte) segment , or each can be serviced by its own reconfigurable control unit 80h, 80i, 80j, 80k... 80n and each by its own pair of A and B registers A_0, and B_0 12h, and 14h; A_1 and, B_1, 12i, and 14i; A_2 and B_2, 12j and 14j, A_3 and B_3 12k and 14k and so on.

A polynomial multiplier circuit 18l, Fig. 11, usable in the embodiment shown herein to provide an output c0-c14 includes a plurality of AND gates 120 which combined with exclusive OR gates 122 can multiply any pair of polynomials from A register 12l and B register 14l e.g., polynomials a_0-a_7, polynomials b_0 - b_7 as illustrated in the table 124 Fig. 12.

There is shown in Fig. 13 a Galois field divider engine 150 according to this invention including a Galois field reciprocal generator 155 having a Galois field multiplier 152 and a second Galois field multiplier 154 for performing a squaring function. Engine 150 performs the division β_l/β_k by executing the operation $\beta_l * 1/\beta_k$, where β_l and β_k are elements of a Galois field, for example, where m = 8, that is $GF(2^8)$: the degree of the field is eight. Initially Galois field multiplier 152 receives a 1 and β_k and multiplies them together. The output is then squared in Galois field multiplier 154 and fed back to Galois field multiplier 152. This result is multiplied by β_k over and over again for m-2 times so that a total of m-1 iterations has occurred. At this point the reciprocal $1/\beta_k$ is obtained and instead of β_k being supplied as it has been for each of the m-2 iterations it is now β_l that is supplied to perform the multiplication $\beta_l * (1/\beta_k)$. Thus, the entire division takes place in a total of m iterations, m-1 for generating the reciprocal and

1 more for multiplying the reciprocal of the divisor and the dividend to get the quotient. The timely application of "1", β_k and β_l is performed by input selection circuit 171.

The fact that $\beta^{2^{m-2}} = \frac{1}{\beta}$ is shown by the following exposition, given:

the field of GF(q) is made up from the numbers {0, 1...(q-1)}. If we multiply by β (β is a field member ≠ 0} each member of {1, 2...(q-1)} to get {1β, 2β...(q-1)β} we can easily see that we get the same set back again (with the order changed). This means that $1,\cdot 2\cdot\ldots\cdot (q-1) = 1\beta\cdot 2\beta\cdot\ldots\cdot (q-1)\beta = 1\cdot 2\cdot\ldots\cdot (q-1)\beta^{(q-1)}$ by cancelling the factors $1\cdot\cdot 2\cdot\ldots\cdot (q-1)$ from both sides assures us that

$$\beta^{q-1}=1. \qquad (1)$$

Therefore

$$\beta^{-1} = \beta^{q-2} \qquad (2)$$

Replacing q with 2^m results in the expression

$$\beta^{2^m-2} = \frac{1}{\beta} \qquad (3)$$

Fig. 13 is a straightforward implementation of this expression.

According to (3) for n=7 we need to calculate β^{254}. β^{254} can be calculated as $\beta 1^{128}\bullet\beta^{64}\bullet\beta^{32}\bullet\beta^{16}\bullet\beta^{8}\bullet\beta^{4}\bullet\beta^{2}$. Which can be iteratively calculated as

n=1: $(\beta\bullet 1)^2=\beta^2$

n=2: $(\beta^2\bullet 1)^2=\beta^4\bullet\beta^2$

n=3: $(\beta^4\bullet\beta^2\bullet\beta)^2=\beta^8\bullet\beta^4\bullet\beta^2=\beta^{14}$

•
•
•
•

n=7: $(\beta^{64}\bullet\beta^{32}\bullet\beta^{16}\bullet\beta^8\bullet\beta^4\bullet\beta^2\bullet\beta\bullet)^2=\beta^{128}\bullet\beta^{64}\bullet\beta^{32}\bullet\beta^{16}\bullet\beta^8\bullet\beta^4\bullet\beta^2=\beta^{254}$

The circuit of Fig. 13 starts from an initial value of 1 and generates at 155 the following successive values:

Iteration #	1	2	3	4	5	6	7
Value at Point 155	β^3	β^6	β^{14}	β^{30}	B^{62}	β^{126}	B^{254}

As can be seen, the final value of β^{-1} is obtained in (n-1) cycles. The same circuit is generating β^{-1} for all intermediate powers of m GF(2^m) {m = 3.7}, for example if m = 4, $\beta^{2^m-2} = 14$ is generated at n=3.

In one embodiment, Galois field reciprocal generator 155a, Fig. 14, may include Galois field multiplier 152a and Galois field multiplier 154a. Galois field multiplier 152a includes Galois field linear transformer 156 and a polynomial multiplier 158. Galois field multiplier 156 is shown as including a matrix of exclusive OR cells having two sections, matrix section 160 and reduced unity matrix section 162, but this is not a necessary limitation of the invention as unity matrix section 162 may be implemented with a full matrix as is matrix section 160 if size is not an issue. Galois field multiplier 154a also includes a polynomial multiplier 164 and Galois field transformer 166 which also may include, but not necessarily, a full matrix section 168 and a reduced unity matrix section 170. Here again unity section 170 is advantageous as to cost and area but it is not necessary as a full section could be used there. Galois field divider engine 150a performs a division in m iterations. In the first iteration input selection circuit 171 introduces a 1 in combination with β_k to Galois field multiplier 152a. This produces an output β_k on line 172 which is delivered to both polynomial multiplier inputs 174, 176 of Galois field multiplier 154a. Thus, a squaring function is performed and the output is fed back to an input 178 of input selection circuit 171. This iteration occurs m-2 times where m is the

degree of the Galois field. After m-2 iterations input selection circuit 171 introduces the dividend β_l to Galois field multiplier 152a because at that time the value at output 178 is the reciprocal $1/\beta_k$. By now multiplying β_l, the dividend, times $1/\beta_k$, the divisor, the result is β_l is divided by β_k to obtain the quotient of the Galois field division at 180.

The values at inputs 174 and 176 take the form of, from the most significant digit to the least, b_7-b_0 and a_7-a_0. When the squaring function is being performed as here, then each of the values b_7-b_0 will be the same, respectively, as each of the values of a_7-a_0 because they are the same numbers. The number of digits b_7-b_0, a_7-a_0 depends upon the size of the polynomial, which in this case where m is 8 would be eight digits. Whatever the size, since the values are the same at both inputs, the exclusive OR function will be zero. That is, like inputs to an exclusive OR gate renders a zero output as is well known. Thus, referring again to Fig. 12, it can be seen that for each of the polynomial multiply outputs c_0-c_{14}, the odd-numbered ones in Fig. 12 contain pairs of identical values. For example, c_1 is equal to $a_1*b_0 \oplus a_0*b_1$. Since we are squaring we know that the two values being presented at inputs 174 and 176 are the same, therefore a_0 and b_0 are the same and a_1 and b_1 are the same. Therefore, c_1 when exclusively ORed will have a value of zero. The same is true for the rest of the odd numbered Galois field multiplier outputs c_3, c_5, c_7, c_9, c_{11}, c_{13}. The result is shown at 182, Fig. 15 where it can be seen not only that there are zero values resulting at the odd numbered c_1-c_{13}, but that the remaining non zero even numbered values require no exclusive OR gates, only multiplication. For example, c_0 is a_0*b_0. But this is a simple AND function resulting in a value of a_0. Similarly, with respect to c_2 the value a_1 is multiplied by b_1 giving an AND function which results in the simple output of a_1. The same effect is true in c_4, c_6, c_8, c_{10}, c_{12}, and c_{14}. The same applies to Galois field multiplier 156b. Galois field multiplier 154b which effects the

squaring function can be reduced in size by one half shown by the reduction by one half of the matrix section 168b and unity section 170b. Also, now since the function has turned into a simple input as shown in column 184, Fig. 15, two separate inputs are not required and so the polynomial multiplier 164 is no longer needed.

Galois field transformers 156c and 166c, Fig. 17, are implemented identically. The shaded circles indicate the enabled exclusive OR gate cells in each of the transformers. The programming is accomplished by the codes in column 190 and is the same for both transformers 156c and 166c. Transformer 156c receives the inputs c_0-c_{14} and provides the outputs A_0-A_7. These form the inputs with the zeros of A_0-A_7 of Galois field linear transformer 166c whose final outputs are B_0-B_7. Both transformers have been implemented for the Galois field of degree eight $GF(2^8)$ (m=8) for the irreducible polynomial (Ox12b). When the reduction shown in Fig. 15 is effected, Galois field multiplier 156d, Fig. 18 stays the same as do all of the programming instructions in the column 190d, but Galois field linear transformer 166d has had every other column, the zero columns, eliminated, resulting in the structure shown in Fig. 16.

When the Galois field divider engine has been reduced as shown in Fig. 16, a further reduction is now achievable. Because Galois field divider engine 154b has no polynomial multiplier in the second Galois field transformer 166b, a single matrix or transformer can be constructed which delivers the output B_0-B_7 directly from c_0-c_{14} without the interim A_0-A_7 terms, in one cycle and using a single linear transformer 200, Fig. 19. Transformer 200 has been programmed to have the combination of exclusive OR cells indicated by the shaded circles enabled in order to perform both of the Galois field linear transforms in one Galois field linear transformer and in one operation. Thus, the inputs c_0-c_{14} are directly transformed by Galois field linear transformer 200 to the ultimate

outputs B_0-B_7. The compounding which reduces the two matrices 156d and 166d, Fig. 18, to the single matrix Galois field linear transformer 200 in Fig. 19 can be seen by a simple illustration using B_7, Fig. 18, which can be seen as equivalent to the exclusive ORs A_7, A_6, and A_5, as shown in Galois field linear transformer 166d. Referring then to Galois field linear transformer 156d (where the backslash indicates a cancellation of a term because it is duplicated), it can be seen that

A_5 is equal to c_{14}, c_{13}, c_{12}, c_{11}, c_{10}, c_8, c_5

A_6 is equal to c_{14}, c_{13}, c_{12}, c_{11}, c_9, c_6,

A_7 is equal to c_{14}, c_{13}, c_{12}, c_{10}, and c_7,

all with the exclusive OR functions between them. This results in the output c_{14}, exclusive OR c_{13}, exclusive OR c_{12}, exclusive OR c_9, exclusive OR c_8, exclusive OR c_7, exclusive OR c_6, exclusive OR c_5. Thus, in matrix 200, Fig. 19, B_7 can be seen to include the exclusive OR combination of c_{14}, c_{13}, c_{12}, c_9, c_8, c_7, c_6,and c_5. One implementation of such a compounded Galois field divider engine 202 is shown in Fig. 20 where Galois field linear transform, matrix 200 of Fig. 19 appears in conjunction with a polynomial multiplier 204 and input selection circuit 171e with dual input selection units 206, 208. Now the Galois field reciprocal generator 205 has been implemented by a single, compound Galois field linear transformer 200. Input selection unit 206 is capable of performing multiply-add (MPA), multiply-accumulate (MAC), and multiply (MPY). Input selection unit 208 functions similarly and provides to Galois field linear transformer 200 the adder input as previously explained. Program sequencer 210 provides the mapping of the control flip-flops 212 which enable and disable the matrix of cells including the exclusive OR gates. The program sequencer can program the GFLT matrix 200 as a compound multiplier performing $(GF\_MPY(\alpha,\beta))^2$ in one cycle for division ,as a

Galois field multiplier for multiplication, as a multiply and accumulate for multiply and accumulation and as a multiply-add for the multiply-add function.

In operation, initially the GFLT is programmed as a compound multiplier performing $(GF\_MPY(\alpha,\beta))^2$, a 1 is provided at input 214 and β_k at input 216. Following that for m-2 iterations, the output 180 is fed back on input 214 while β_k remains on input 216. After m-2 iterations, when the system has gone through a total of m-1 iterations, the input at 214 is now the reciprocal of β_k. At this point the GFLT is programmed as a Galois Field multiplier, β_k at input 216 is now replaced with input β_l so that the next multiplication, the m^{th} iteration, multiplies β_l times the reciprocal of β_k to provide the output β_l divided by β_k. The Galois field division method of this invention is shown in Fig. 21 where the divisor β_k and dividend β_l are provided at start 240. A query is then made as to whether this iteration is the m^{th} iteration in step 242, where m is the degree of the Galois field involved. If it is the mth iteration, the system goes directly to step 244 where the Galois field multiplication of β_l by the Galois field linear transform output of the reciprocal $1/\beta_k$ is performed. The quotient is then produced at 246. If the iteration has not reached m, then the query is made in step 248 as to whether it is the first iteration. If it is, multiplication of β_k by 1 is effected in step 250 and then the square of that value is performed over a Galois field in step 252. If it is not the first iteration, then in step 254, the Galois field multiplication of β_k by the Galois field linear transform output is performed and then the square is performed in Galois field multiplier in step 252. The output from the square calculation is then fed back, step 242, and the iteration begins again.

Thus far the invention has focused on a Galois field divider engine and method and to the ability to reduce that engine in size by first reducing the size of one of the

Galois field linear transformers and eliminating one of the polynomial multipliers and then by combining the functions of the two linear transformers so that a succession of Galois field linear transforms on a succession of polynomial inputs is performed to obtain the ultimate output (quotient) as shown in Figs. 19 and 20. But, this is not a necessary limitation of the invention, that is it is not limited to merely division. A compound Galois field engine according to this invention may perform any succession of Galois field linear transforms on a succession of polynomial inputs to obtain an ultimate output where each input, except the first, is the output of the previous Galois field linear transform. That is in one transform it can immediately predict the modulo remainder of the succession of Galois field linear transforms of an irreducible Galois field polynomial to obtain the ultimate output of the Galois field linear transform directly from the first input.

Another example of this fact can be seen in the square root operation of a Galois field member β. There is shown in Fig. 22 a compound Galois field engine 300 according to this invention that performs (m-1) successive Galois field linear transforms 302, 304...306 wherein a first input β, 308 is submitted to Galois field transformer 302 and then the transformed output becomes the input to the next Galois field linear transformer 304, whose output becomes the input to the next Galois field linear transformer, and so on, until it reaches the final transformer 306 as in this case, the $\sqrt{\beta}$ output. In accordance with this invention, by compounding the Galois field linear transformers as shown in 310 Fig. 23, the (m-1) transformers of Fig. 22 can be reduced to produce the simplified implementation shown in Fig. 23 of only one GFLT, where, the initial input β can be transformed in a single operation by the compound Galois field linear transformer square root engine 330 to provide in one iteration, the $\sqrt{\beta}$ output.

The fact that $\sqrt{\beta} = \beta^{2^{(m-1)}}$ is shown by the following exposition given:

in (1) we have shown that $\beta^{q-1}=1$.

Replacing q with 2^m and multiplying both sides by β results in the expression

$$\beta^{2^m} = \beta \qquad (4)$$

Taking the $\sqrt{\ }$ form both sides results in the expression

$$\beta^{(2^m)/2} = \sqrt{\beta} \qquad (5)$$

or

$$\beta^{2^{(m-1)}} = \sqrt{\beta} \qquad (6)$$

Fig. 22 is a straightforward implementation of this expression.

The Galois field square root method of this invention is shown in Fig. 24 where the field element β are provided at start 312. A query is then made as to whether this iteration is the m^{th}-1 iteration in step 314, where m is the degree of the Galois field involved. If it is the m^{th}-1 iteration, the system goes directly to step 316 where the Galois field square root of β is produced. If the iteration has not reached m-1, then the query is made in step 318 as to whether it is the first iteration. If it is, the square of that β value is performed over a Galois field in step 320. If it is not the first iteration, the square of the Galois field linear transform output is performed over a Galois field in step 322. The output from the square calculation is then fed back, step 314, and the iteration begins again. A programming circuit, control flip-flops 212a and programming sequencer 210a, programs the Galois field linear transformer square root engine 330 as shown in Fig. 23.

In summary, generally a compound Galois field engine 260, Fig. 25 according to this invention may perform a number of successive Galois field linear transforms 262, 264, 266, 268 wherein a first input A, 270 is submitted to Galois field transformer 262

and then the transformed output B becomes the input to the next Galois field linear transformer 264, whose output C in turn becomes the input to the next Galois field linear transformer, 266 whose output D becomes the input to the next Galois field linear transformer 268, and so on. In this case, the ultimate output is E. In accordance with this invention, by compounding the Galois field linear transformers as shown in Fig. 19, by reducing the two transformers of Fig. 18 to produce the implementation shown in Fig. 20, the initial input A can be transformed in a single operation by compound Galois field linear transformer 280 to provide in that one iteration, the ultimate output E.

Although specific features of the invention are shown in some drawings and not in others, this is for convenience only as each feature may be combined with any or all of the other features in accordance with the invention. The words "including", "comprising", "having", and "with" as used herein are to be interpreted broadly and comprehensively and are not limited to any physical interconnection. Moreover, any embodiments disclosed in the subject application are not to be taken as the only possible embodiments.

Other embodiments will occur to those skilled in the art and are within the following claims:

What is claimed is:

CLAIMS

1. A Galois field divider engine comprising:

a Galois field reciprocal generator;

an input selection circuit for initially inputting a 1 and a first Galois field element to the Galois field reciprocal generator to obtain an output, subsequently multiplying in the Galois field reciprocal generator a first Galois field element by the output of the Galois field reciprocal generator for predicting the modulo remainder of the square of the polynomial product of an irreducible polynomial m-2 times where m is the degree of the Galois field, to obtain the reciprocal of the first Galois field element and multiplying in the Galois field reciprocal engine the reciprocal of the first Galois field element by a second Galois field element for predicting the modulo remainder of the polynomial product for an irreducible polynomial to obtain the quotient of the two Galois field elements in m cycles.

2. The Galois field divider engine of claim 1 in which said Galois field reciprocal generator includes first and second Galois field multipliers.

3. The Galois field divider engine of claim 2 in which said first Galois field multiplier includes a first polynomial multiplier circuit and a first Galois field linear transformer.

4. The Galois field divider engine of claim 3 in which said first Galois field linear transform includes a matrix of cells.

5. The Galois field divider engine of claim 4 in which said first Galois field linear transform matrix of cells includes a matrix section and a unity matrix section.

6. The Galois field divider engine of claim 2 in which said second Galois field multiplier includes a second polynomial multiplier circuit and a second Galois field linear transformer.

7. The Galois field divider engine of claim 6 in which said second Galois field linear transform includes a matrix of cells.

8. The Galois field divider engine of claim 5 in which said second Galois field linear transform matrix of cells includes a matrix section and a unity matrix section.

9. The Galois field divider engine of claim 2 in which the output of said first Galois field multiplier is fed to both multiply inputs of said second Galois field linear multiplier to provide the square of that output.

10. The Galois field divider engine of claim 2 in which said Galois field reciprocal generator includes a Galois field multiplier including a first polynomial multiplier and first Galois field transformer and a second Galois field transformer for calculating the square of the first Galois field multiplier output.

11. The Galois field divider engine of claim 10 in which said second Galois field transformer is approximately one half the size of said first Galois field transformer.

12. The Galois field divider engine of claim 11 in which said first and second Galois field transformers each includes a matrix of cells and said second Galois field transformer includes approximately one half the number of cells of said first Galois field transformer.

13. The Galois field divider engine of claim 1 in which said Galois field reciprocal engine includes a Galois field multiplier, and a program circuit for programming said Galois field multiplier to perform a compound multiply-square operation for m-2 times followed by a multiply operation.

14. A compound Galois field engine for performing a succession of Galois field linear transforms on a succession of polynomial inputs to obtain an ultimate output where each input, except the first, is the output of the previous Galois field linear transform comprising:

an input circuit for providing a first input; and

a Galois field linear transformer having a matrix of cells responsive to said first input and configured to, in one transform, immediately predict the modulo remainder of the succession of Galois field linear transforms of an irreducible Galois field polynomial to obtain the ultimate output of the Galois field linear transform directly from said first input.

15. A Galois field divider method comprising:

initially inputting a 1 and a first Galois field element to a Galois field reciprocal generator to obtain an output;

multiplying in the Galois field reciprocal generator a first Galois field element by the output of the Galois field reciprocal generator for predicting the modulo remainder of the square of the polynomial product of an irreducible polynomial m-2 times where m is the degree of the Galois field to obtain the reciprocal of the first Galois field element; and

multiplying in the Galois field reciprocal engine the reciprocal of the first Galois field element by a second Galois field element for predicting the modulo remainder of the polynomial product for an irreducible polynomial to obtain the quotient of the two Galois field elements in m cycles.

16. A Galois field square root method comprising:

inputting a Galois field element to a Galois field square root generator to obtain an output; and

squaring in the Galois field square root generator the output of the Galois field square root generator for predicting the modulo remainder of the square of the polynomial product of an irreducible polynomial m-1 times where m is the degree of the Galois field to obtain the square root of the Galois field element in (m-1) cycles.

17. A Galois field square root engine comprising:

a Galois field square root generator;

an input circuit for inputting a Galois field element to the Galois field square root generator to obtain the square root of the Galois field elements in one cycle.

18. The Galois field square root engine of claim 17 in which said Galois field

square root engine includes a Galois field multiplier, and a program circuit for programming said Galois field multiplier to perform a compound square operation of m-1 times in one cycle.

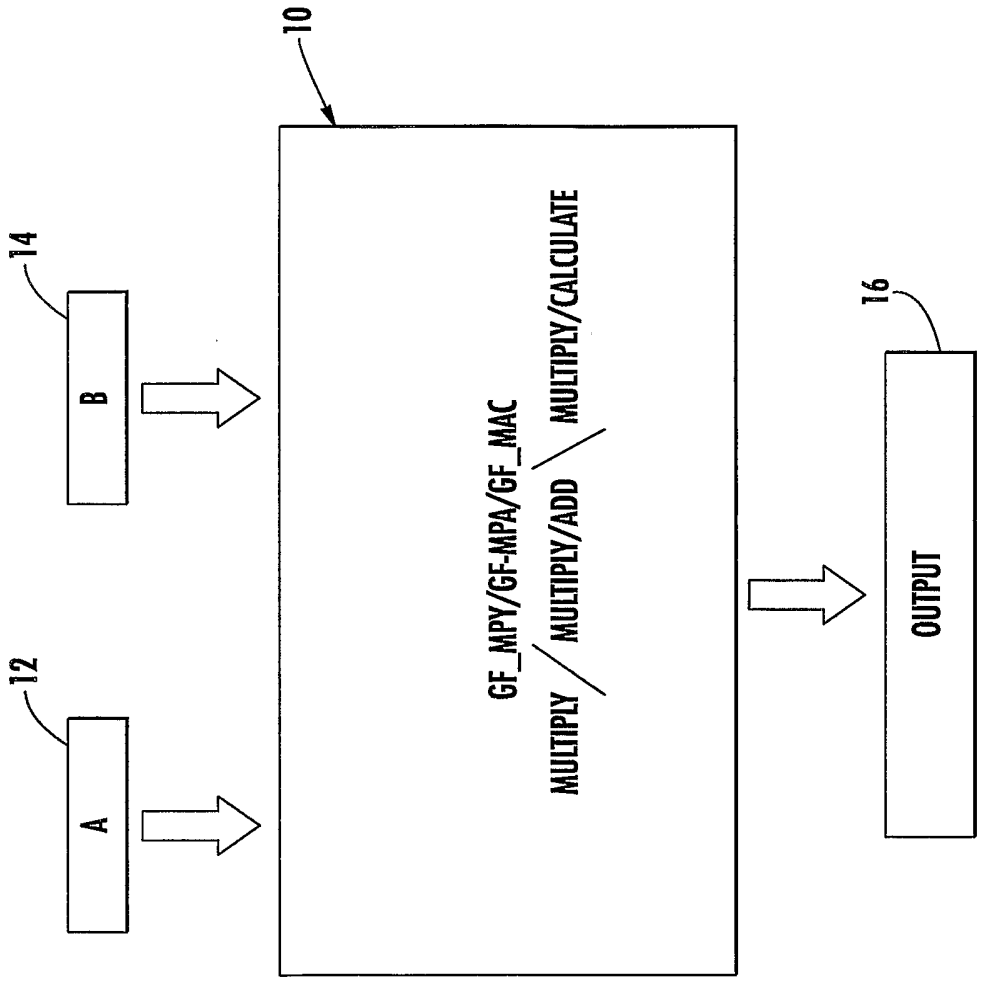

10
14
B
12
A
GF_MPY/GF-MPA/GF_MAC
MULTIPLY / MULTIPLY/ADD / MULTIPLY/CALCULATE
16
OUTPUT

FIG. 1

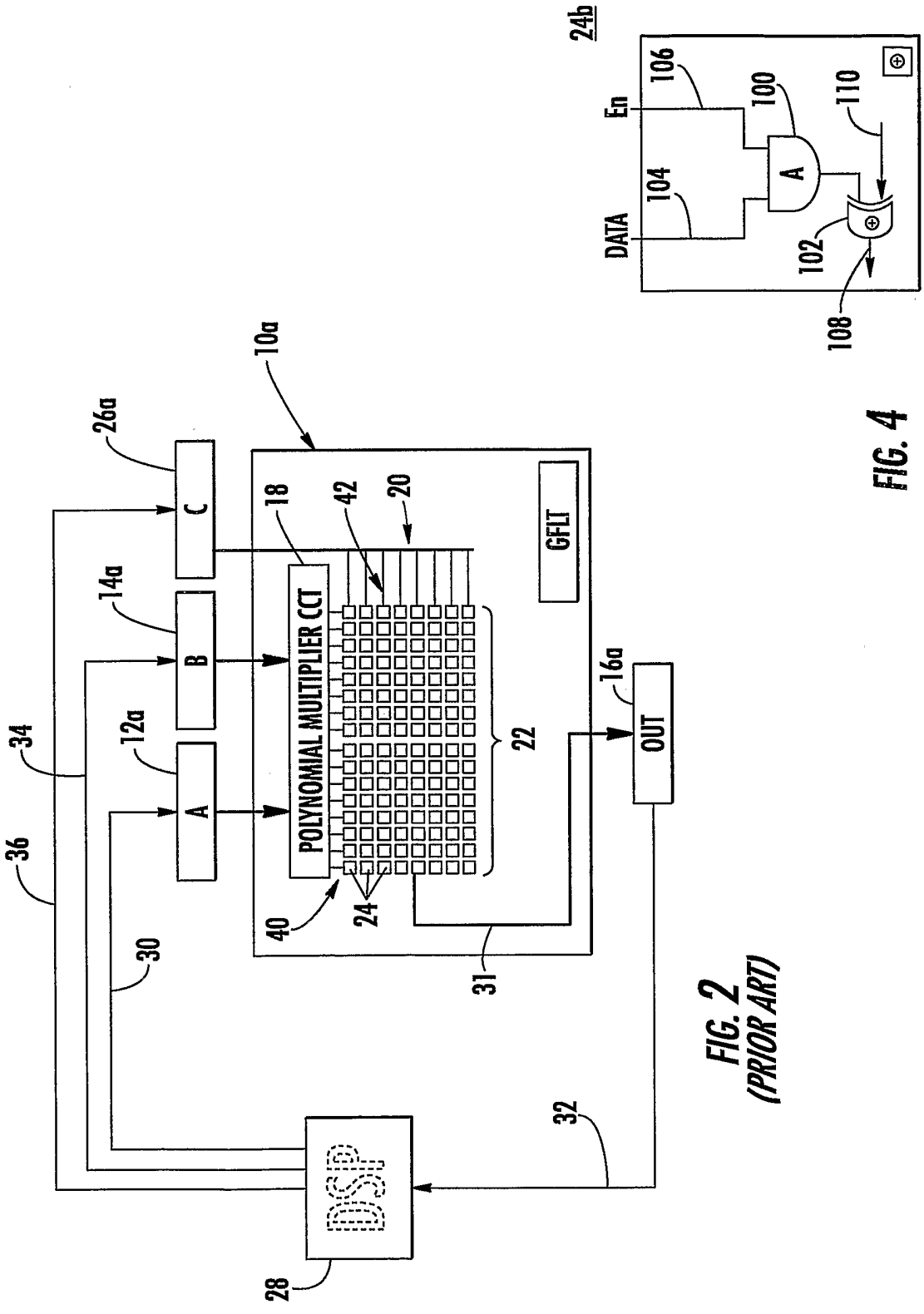


FIG. 4

FIG. 2
(PRIOR ART)

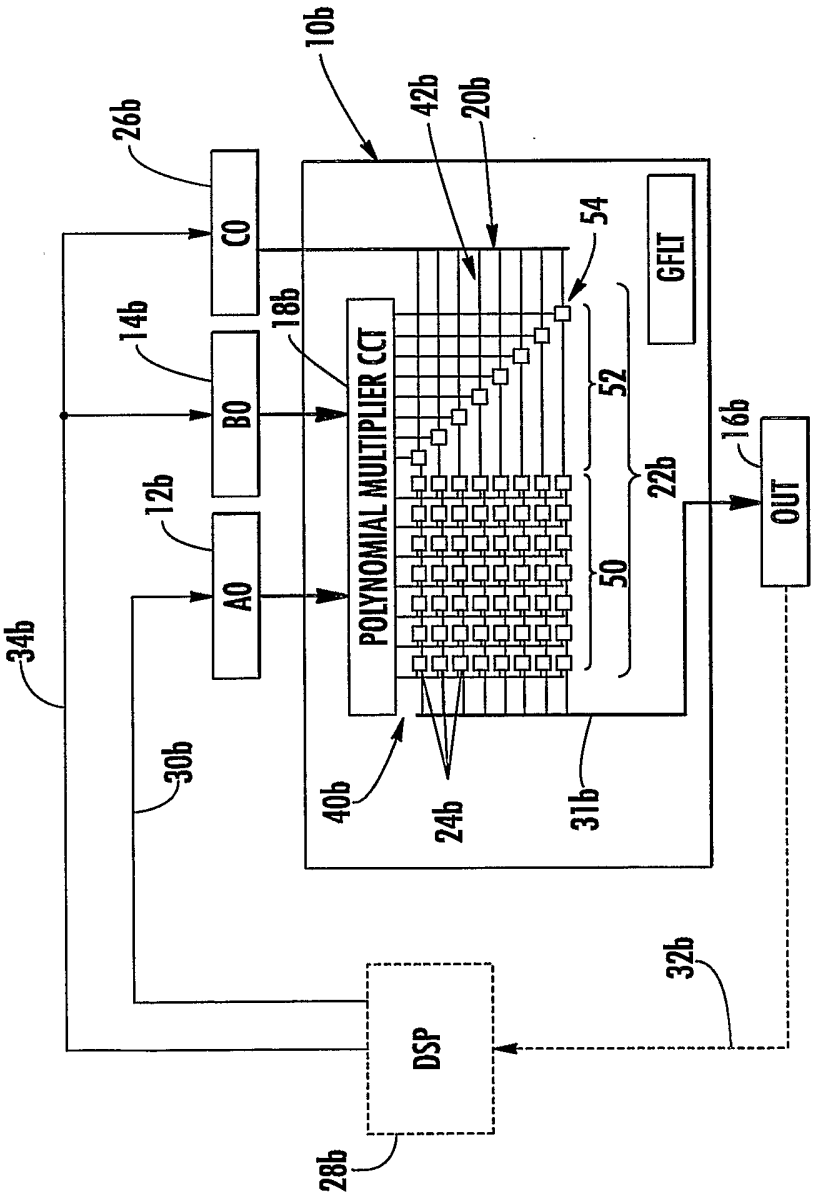

10b
42b
20b
26b
C0
54
GFLT
14b
18b
B0
POLYNOMIAL MULTIPLIER CCT
52
16b
OUT
12b
A0
22b
50
34b
30b
40b
24b
31b
32b
DSP
28b

FIG. 3

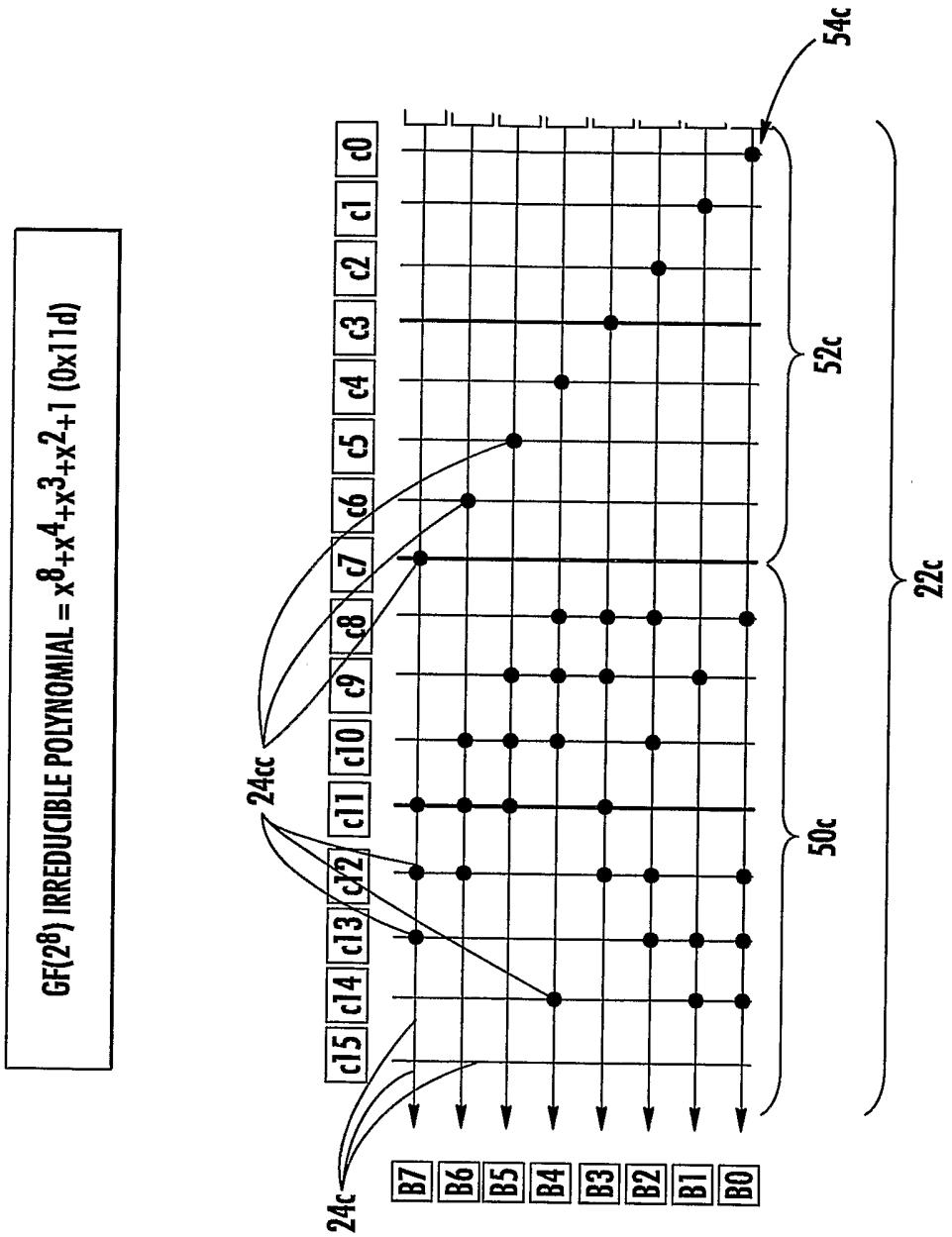

GF(2^8) IRREDUCIBLE POLYNOMIAL = x^8+x^4+x^3+x^2+1 (0x11d)
c0
c1
c2
c3
c4
c5
c6
c7
c8
c9
c10
c11
c12
c13
c14
c15
B7
B6
B5
B4
B3
B2
B1
B0
24cc
24c
54c
52c
50c
22c

FIG. 5

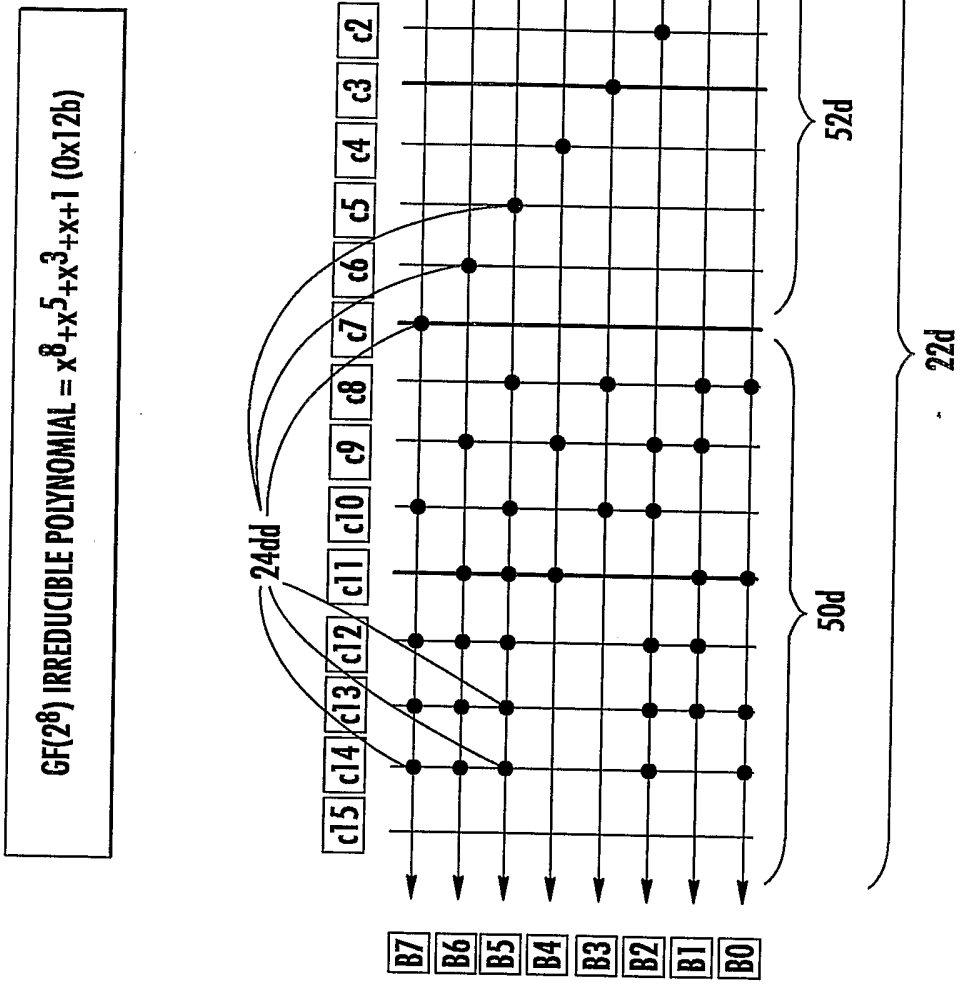

GF(2^8) IRREDUCIBLE POLYNOMIAL = x^8+x^5+x^3+x+1 (0x12b)
c0
c1
c2
c3
c4
c5
c6
c7
c8
c9
c10
c11
c12
c13
c14
c15
24dd
54d
52d
50d
22d
B7
B6
B5
B4
B3
B2
B1
B0

FIG. 6

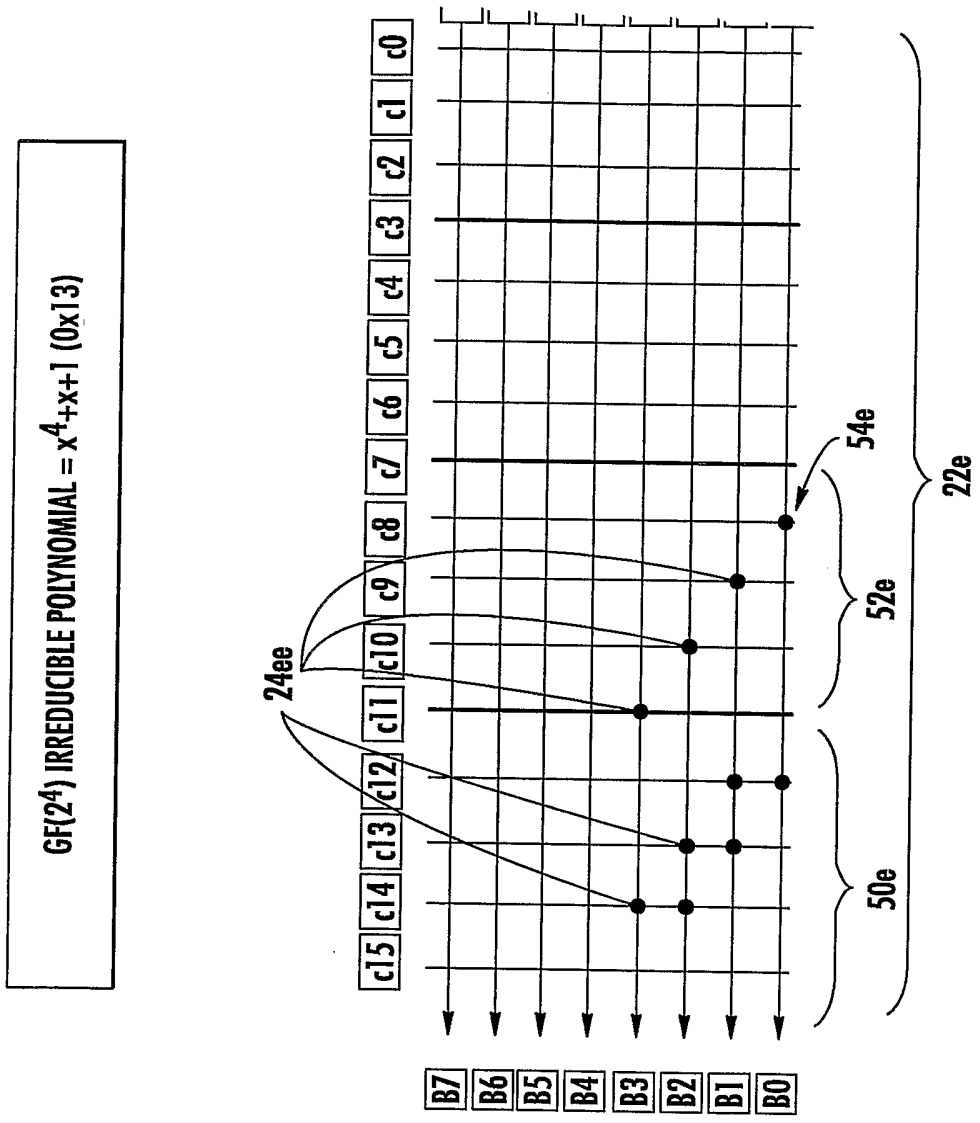

GF(2^4) IRREDUCIBLE POLYNOMIAL = x^4+x+1 (0x13)
c15 c14 c13 c12 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0
B7 B6 B5 B4 B3 B2 B1 B0
24ee
54e
52e
50e
22e

FIG. 7

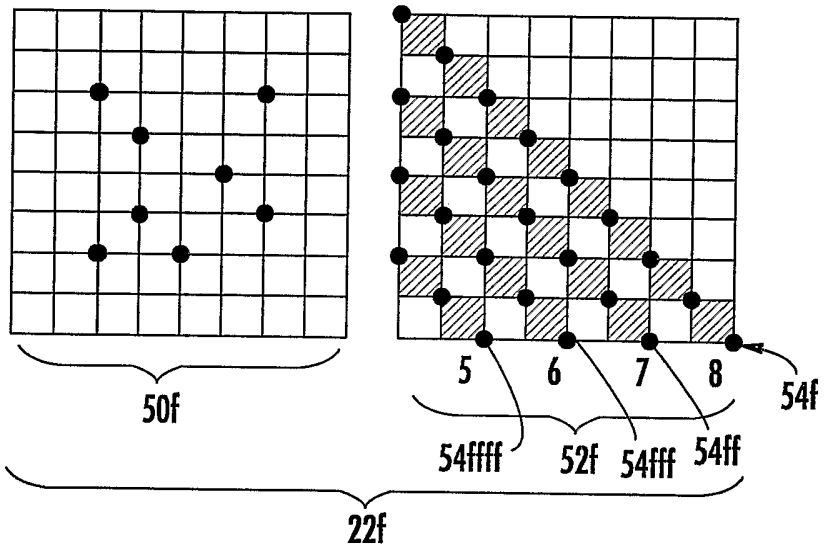


FIG. 8

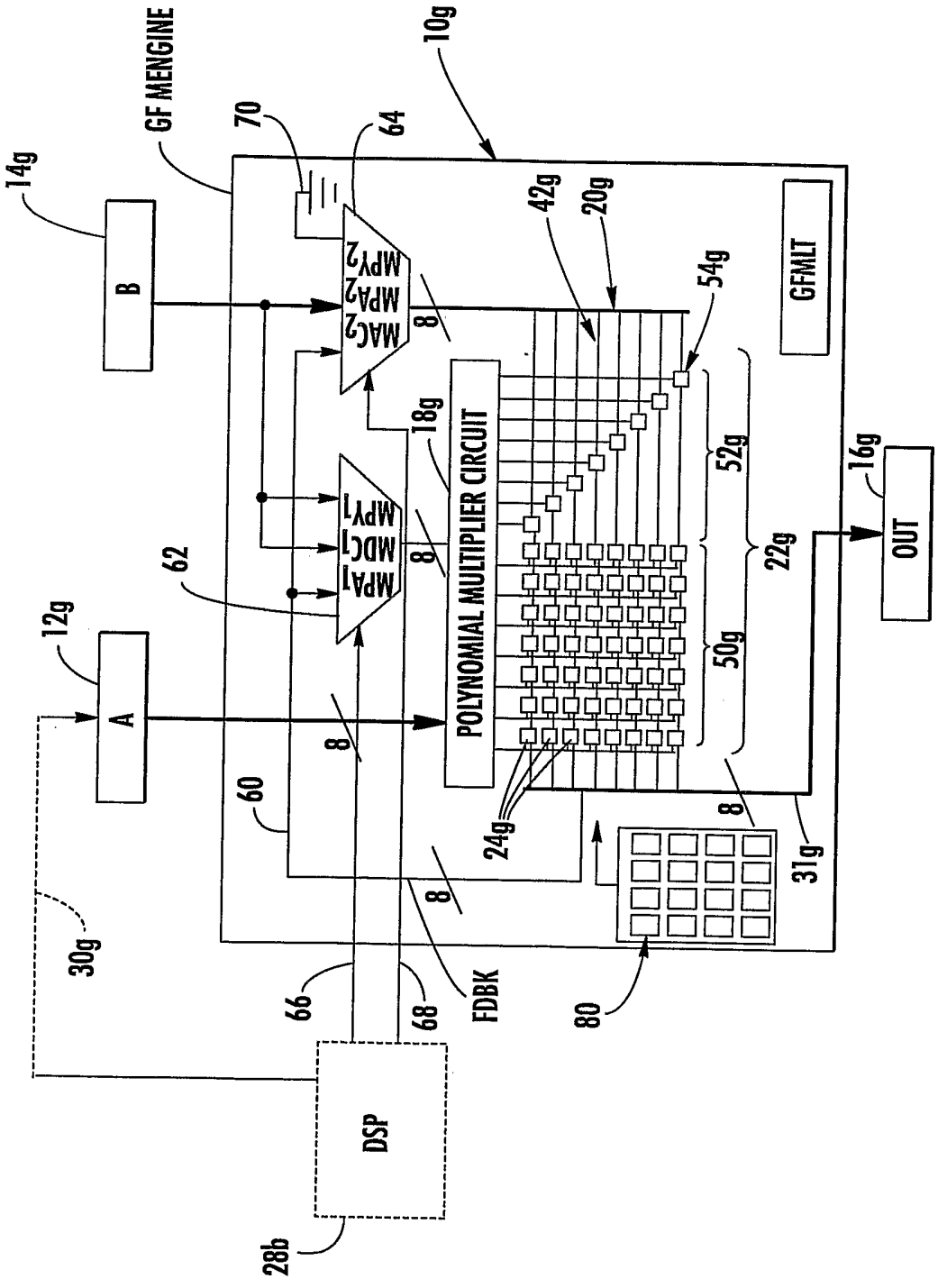

GF MENGINE
10g
70
64
14g
B
42g
20g
54g
GFMLT
MAC2
MPA2
MPY2
8
18g
POLYNOMIAL MULTIPLIER CIRCUIT
52g
16g
OUT
MPA1
MDC1
MPY1
62
22g
12g
A
50g
8
60
24g
31g
30g
FDBK
80
66
68
DSP
28b

FIG. 9

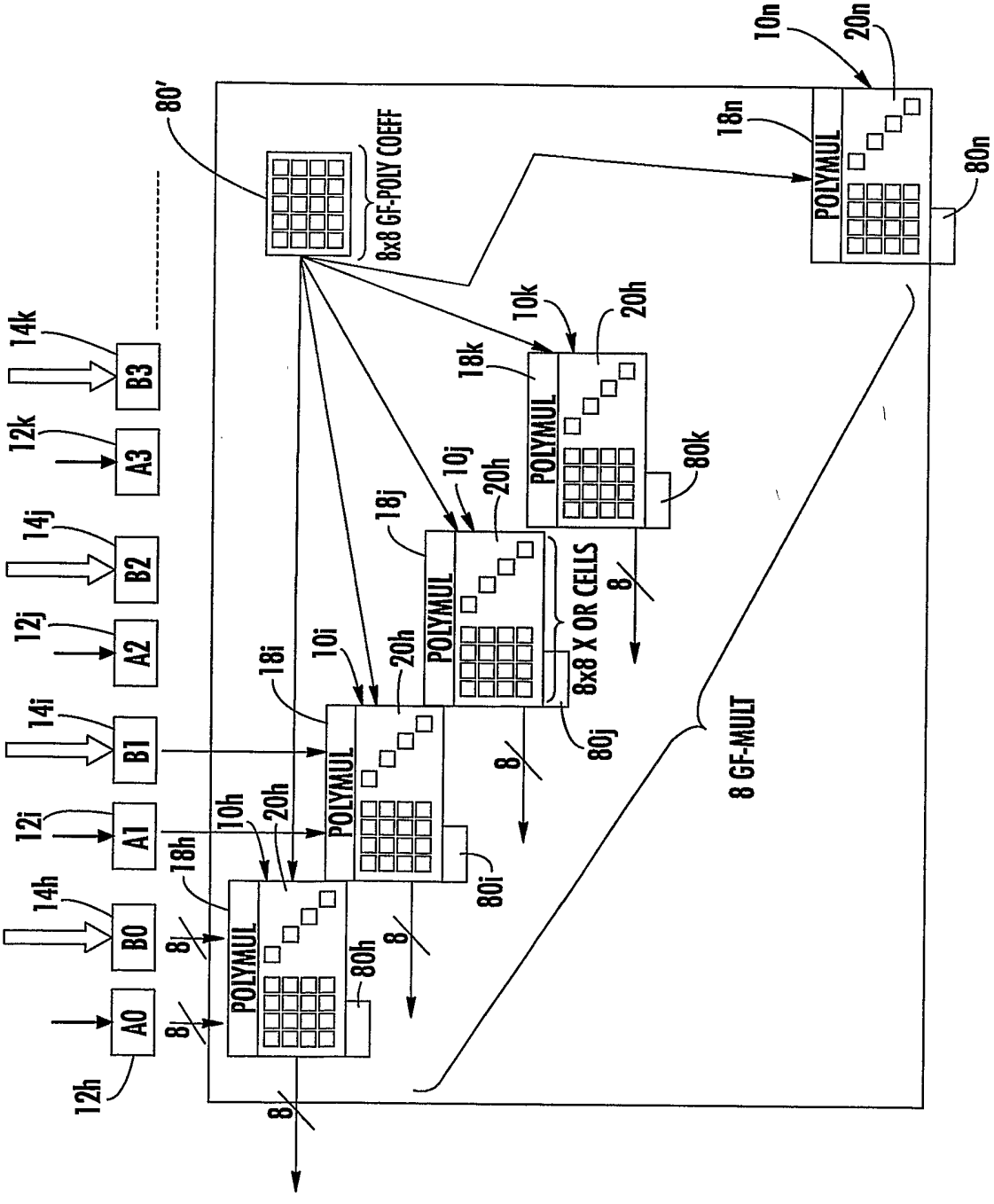

8x8 GF-POLY COEFF
8 GF-MULT
8x8 X OR CELLS
POLYMUL
A0
A1
A2
A3
B0
B1
B2
B3
80'
80h
80i
80j
80k
80n
8

FIG. 10

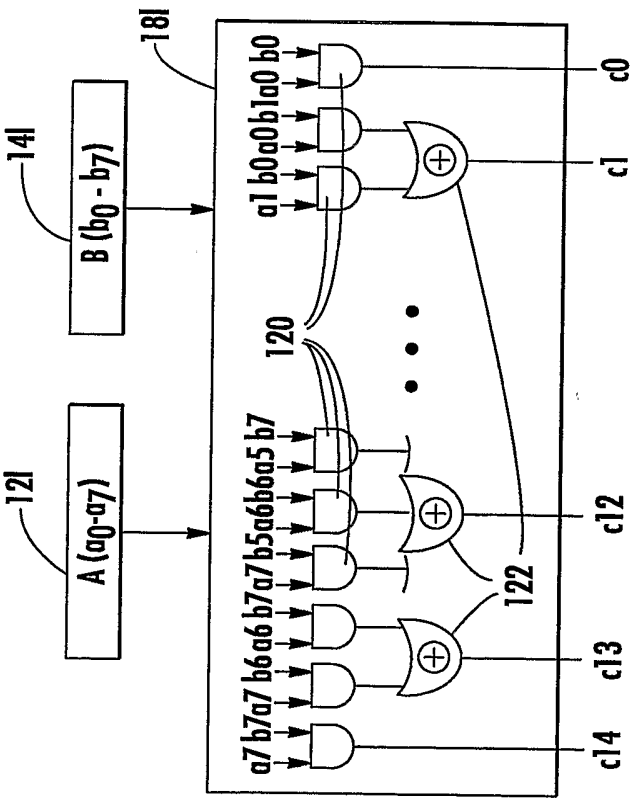

181
141
B (b0 - b7)
121
A (a0-a7)
120
122
a7 b7 a7 b6 a6 b7 a7 b5 a6 b6 a5 b7
a1 b0 a0 b1 a0 b0
c14
c13
c12
c1
c0

FIG. 11

$GF(2^8)^2$

124

$c14 = a7*b7$
$c13 = a7*b6 \oplus a6*b7$
$c12 = a7*b5 \oplus a6*b6 \oplus a5*b7$
$c11 = a7*b4 \oplus a6*b5 \oplus a5*b6 \oplus a4*b7$
$c10 = a7*b3 \oplus a6*b4 \oplus a5*b5 \oplus a4*b6 \oplus a3*b7$
$c9 = a7*b2 \oplus a6*b3 \oplus a5*b4 \oplus a4*b5 \oplus a3*b6 \oplus a2*b7$
$c8 = a7*b1 \oplus a6*b2 \oplus a5*b3 \oplus a4*b4 \oplus a3*b5 \oplus a2*b6 \oplus a1*b7$
$c7 = a7*b0 \oplus a6*b1 \oplus a5*b2 \oplus a4*b3 \oplus a3*b4 \oplus a2*b5 \oplus a1*b6 \oplus a0*b7$
$c6 = a6*b0 \oplus a5*b1 \oplus a4*b2 \oplus a3*b3 \oplus a2*b4 \oplus a1*b5 \oplus a0*b6$
$c5 = a5*b0 \oplus a4*b1 \oplus a3*b2 \oplus a2*b3 \oplus a1*b4 \oplus a0*b5$
$c4 = a4*b0 \oplus a3*b1 \oplus a2*b2 \oplus a1*b3 \oplus a0*b4$
$c3 = a3*b0 \oplus a2*b1 \oplus a1*b2 \oplus a0*b3$
$c2 = a2*b0 \oplus a1*b1 \oplus a0*b2$
$c1 = a1*b0 \oplus a0*b1$
$c0 = a0*b0$

FIG. 12

184

182

$c14 = a7*b7 = a7$
$c13 = 0$
$c12 = a6*b6 = a6$
$c11 = 0$
$c10 = a5*b5 = a5$
$c9 = 0$
$c8 = a4*b4 = a4$
$c7 = 0$
$c6 = a3*b3 = a3$
$c5 = 0$
$c4 = a2*b2 = a2$
$c3 = 0$
$c2 = a1*b1 = a1$
$c1 = 0$
$c0 = a0*b0 = a0$

FOR CASE WHERE $a_i = b_i$ (β^2)

FIG. 15

$$\frac{\beta_l}{\beta_k} = \beta_l \times \left(\frac{1}{\beta_k}\right)$$

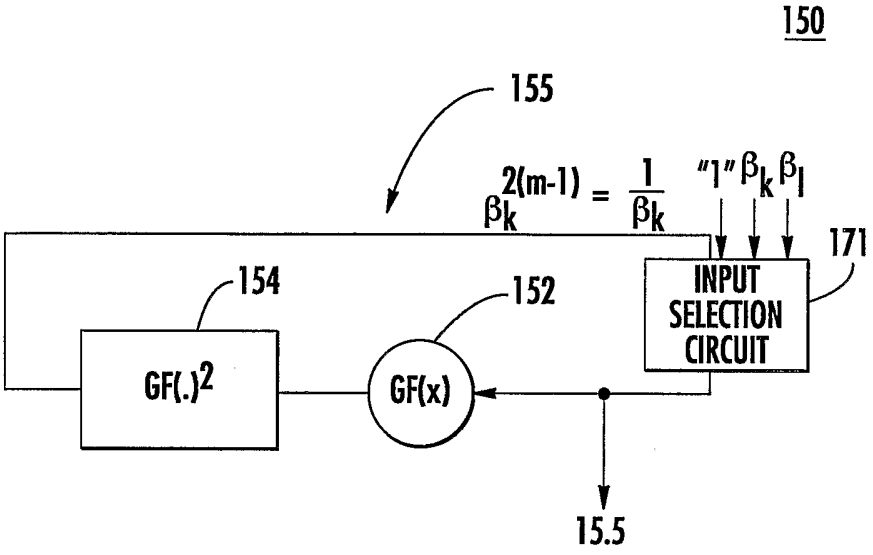


FIG. 13

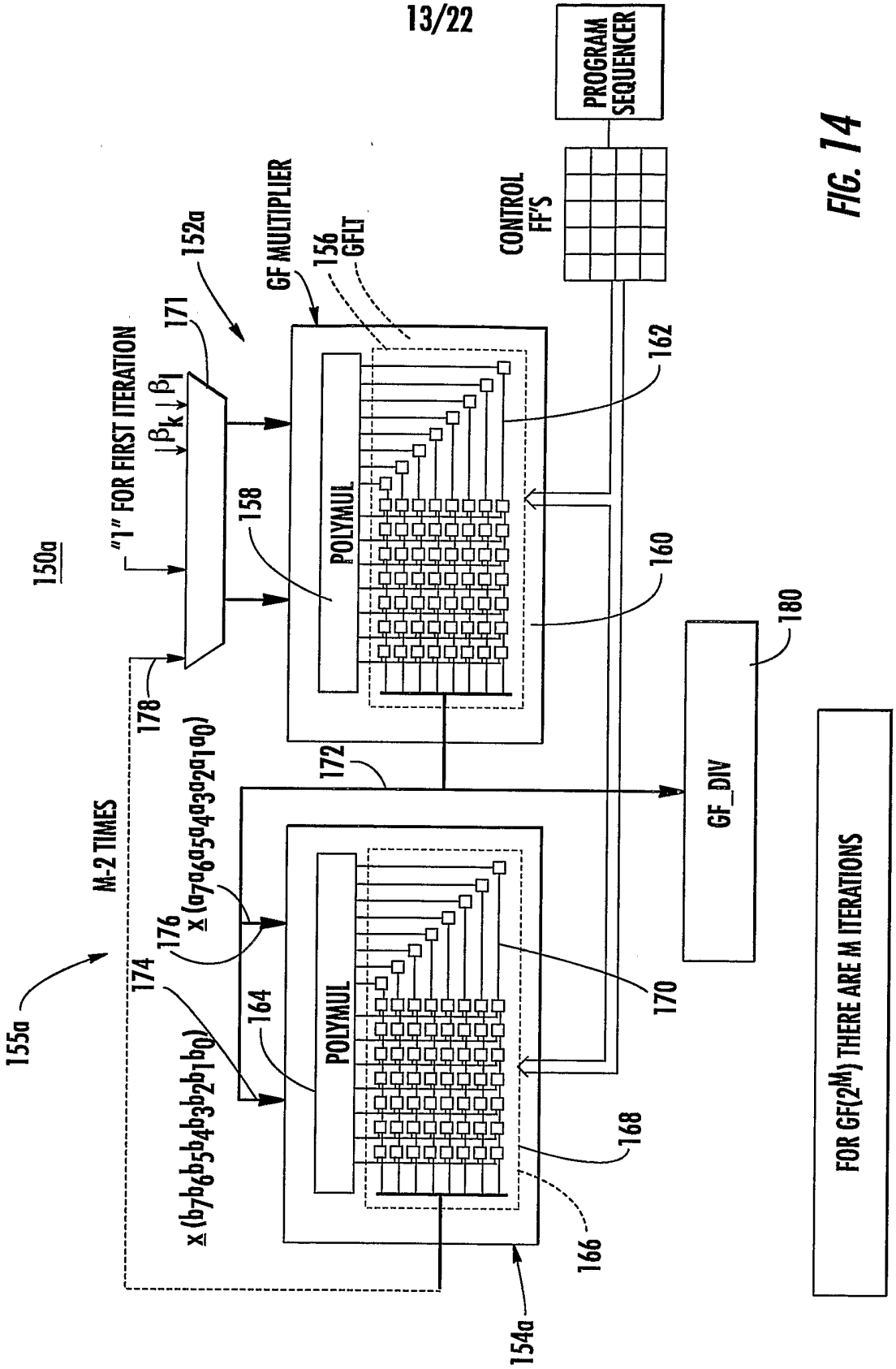

PROGRAM SEQUENCER
CONTROL FF'S
GF MULTIPLIER
156 GFLT
152a
171
"1" FOR FIRST ITERATION
β_k β_l
158
POLYMUL
162
160
150a
178
180
GF_DIV
172
M-2 TIMES
$x(a_7a_6a_5a_4a_3a_2a_1a_0)$
176
174
$x(b_7b_6b_5b_4b_3b_2b_1b_0)$
164
POLYMUL
170
168
166
155a
154a
FOR GF(2^M) THERE ARE M ITERATIONS

FIG. 14

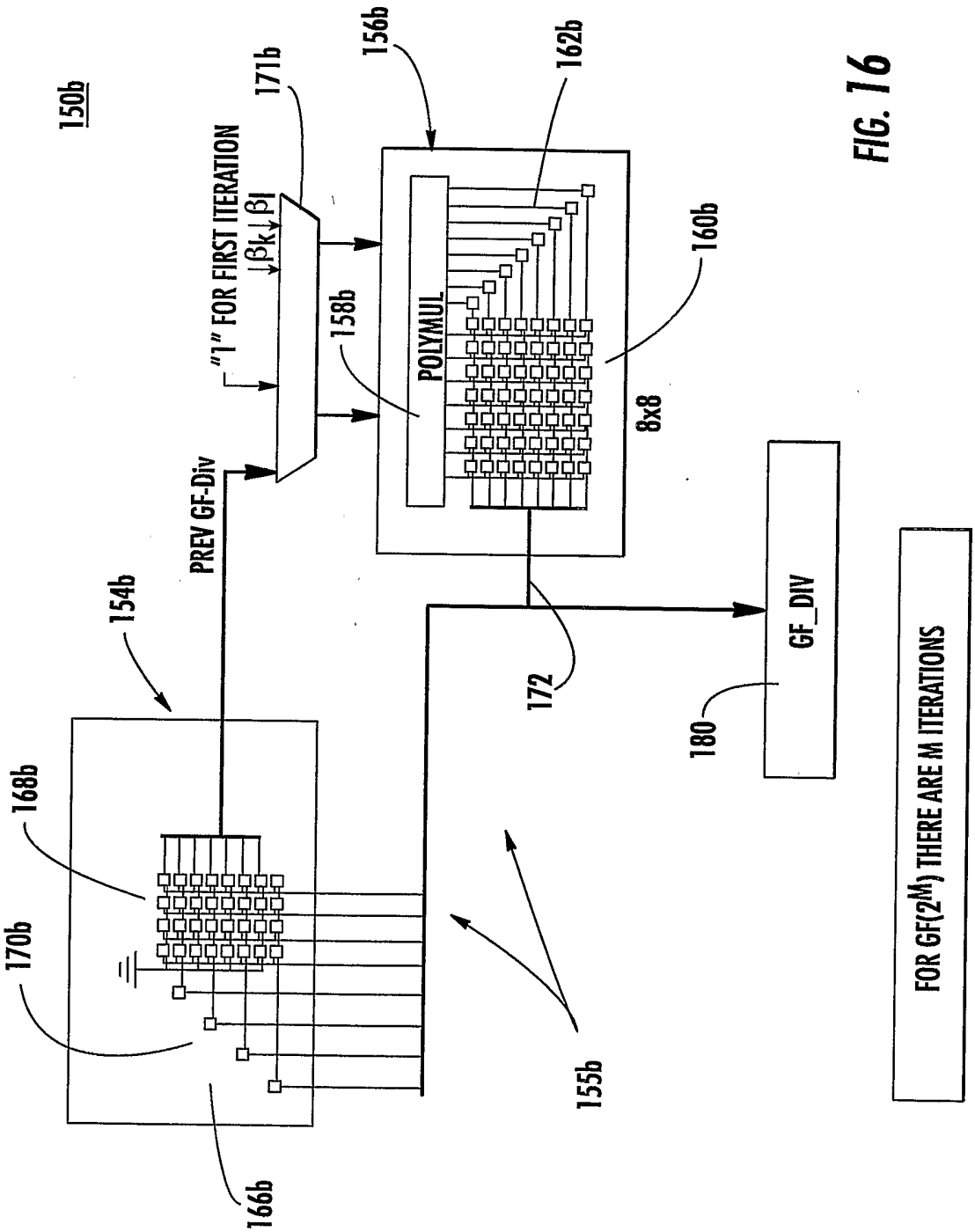

150b
171b
156b
162b
"1" FOR FIRST ITERATION
β_k, β_l
158b
POLYMUL
160b
8x8
PREV GF-Div
154b
168b
170b
166b
172
GF_DIV
180
155b
FOR GF(2^M) THERE ARE M ITERATIONS

FIG. 16

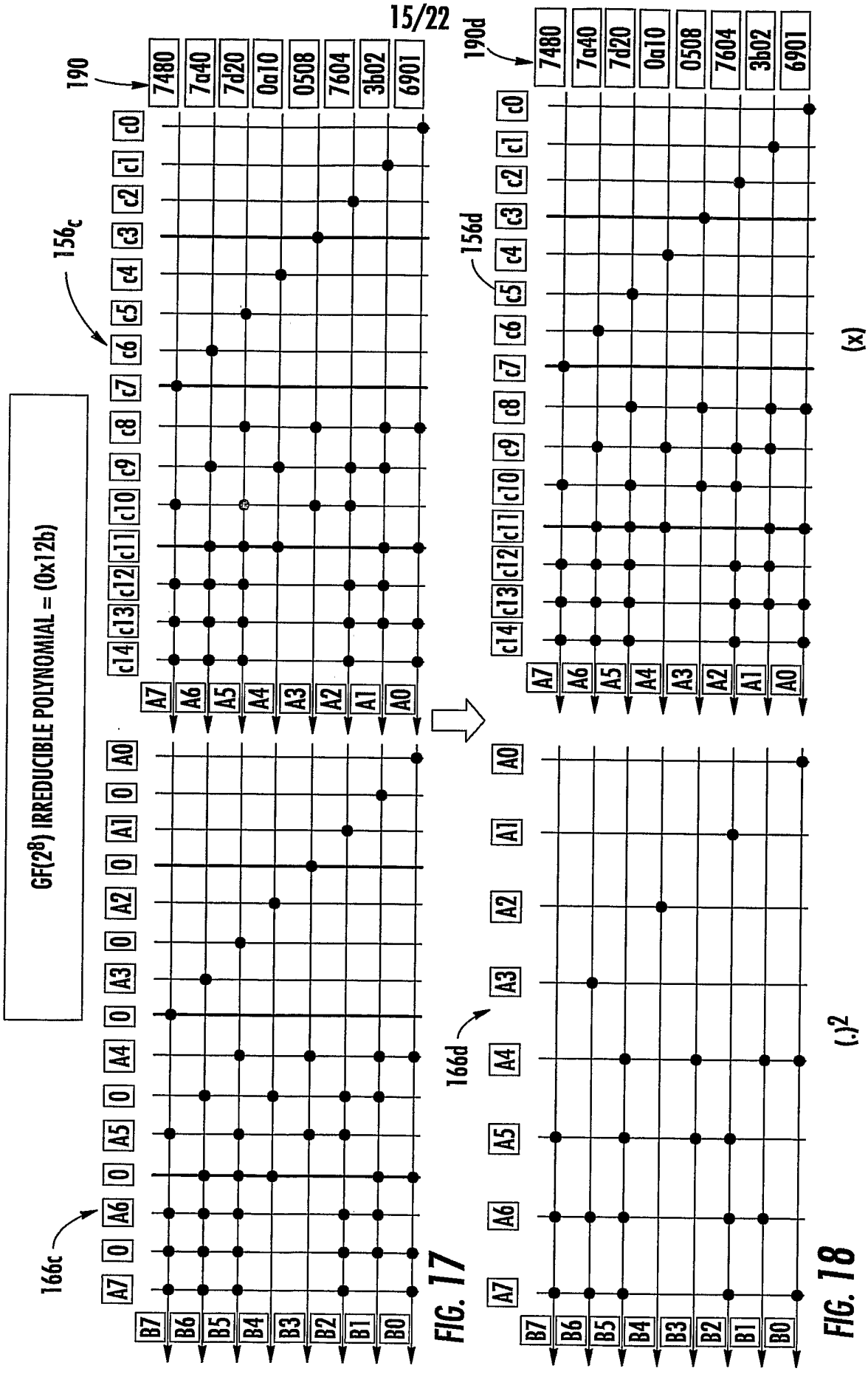

GF(2^8) IRREDUCIBLE POLYNOMIAL = (0x12b)
190
190d
156c
156d
166c
166d
7480
7a40
7d20
0a10
0508
7604
3b02
6901
c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 c10 c11 c12 c13 c14
A7 A6 A5 A4 A3 A2 A1 A0
B7 B6 B5 B4 B3 B2 B1 B0
A7 0 A6 0 A5 0 A4 0 A3 0 A2 0 A1 0 A0
(x)
(.)2

FIG. 17

FIG. 18

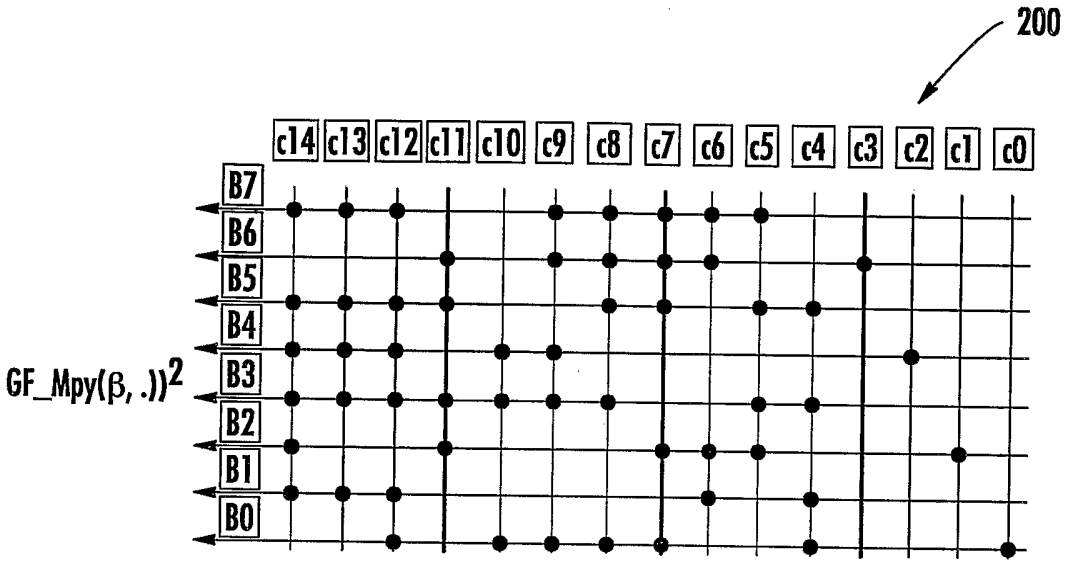

200
c14
c13
c12
c11
c10
c9
c8
c7
c6
c5
c4
c3
c2
c1
c0
B7
B6
B5
B4
B3
B2
B1
B0
GF_Mpy(β, .))2

FIG. 19

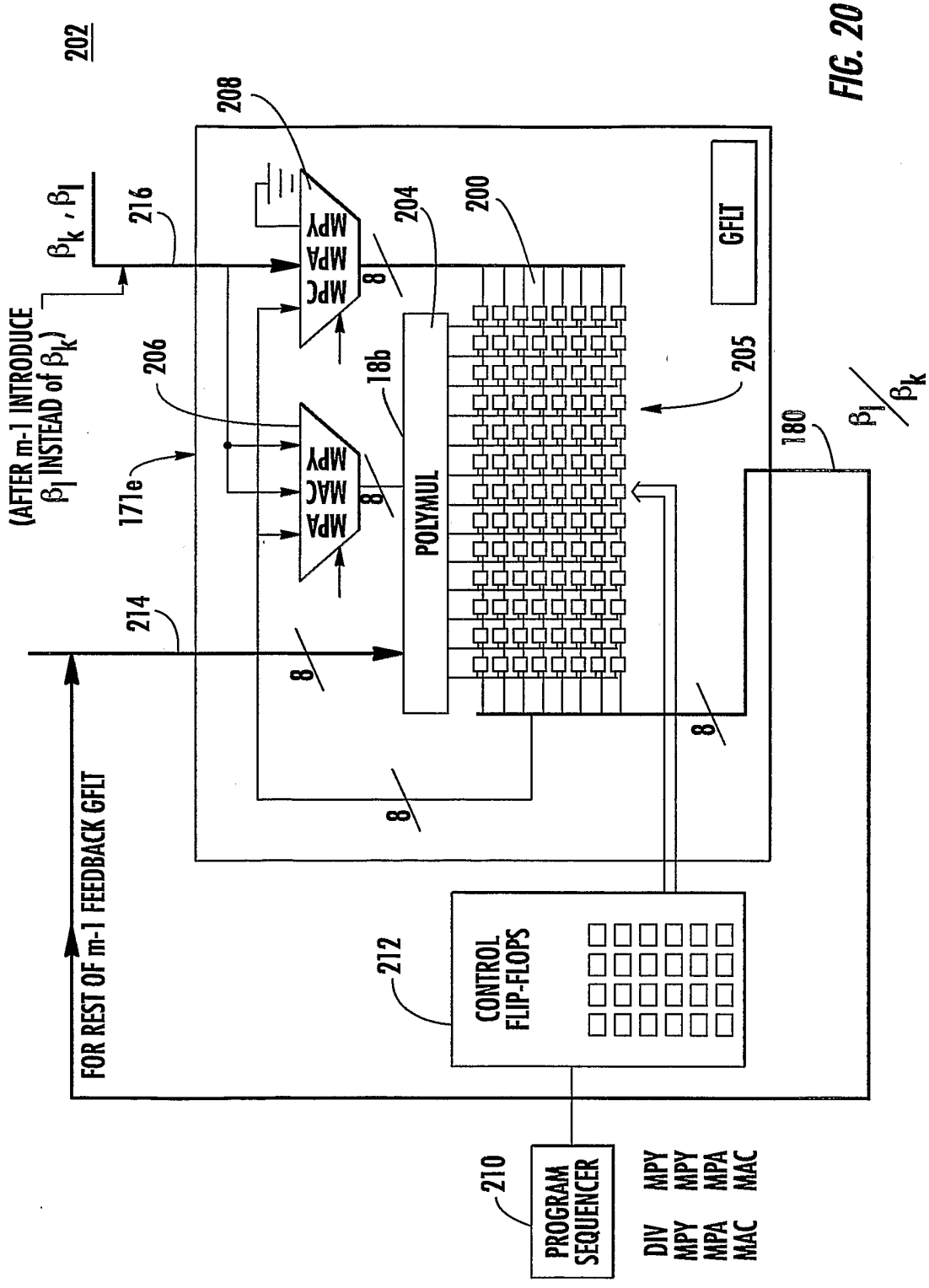

202
208
216
β_k , β_l
(AFTER m-1 INTRODUCE β_l INSTEAD of β_k)
MPC MPA MPY
204
200
GFLT
206
171e
MPA MAC MPY
18b
POLYMUL
205
180
β_l / β_k
214
8
FOR REST OF m-1 FEEDBACK GFLT
212
CONTROL FLIP-FLOPS
210
PROGRAM SEQUENCER
DIV MPY
MPY MPY
MPA MPA
MAC MAC

FIG. 20

GALOIS FIELD DIVISION OF β_l/β_k

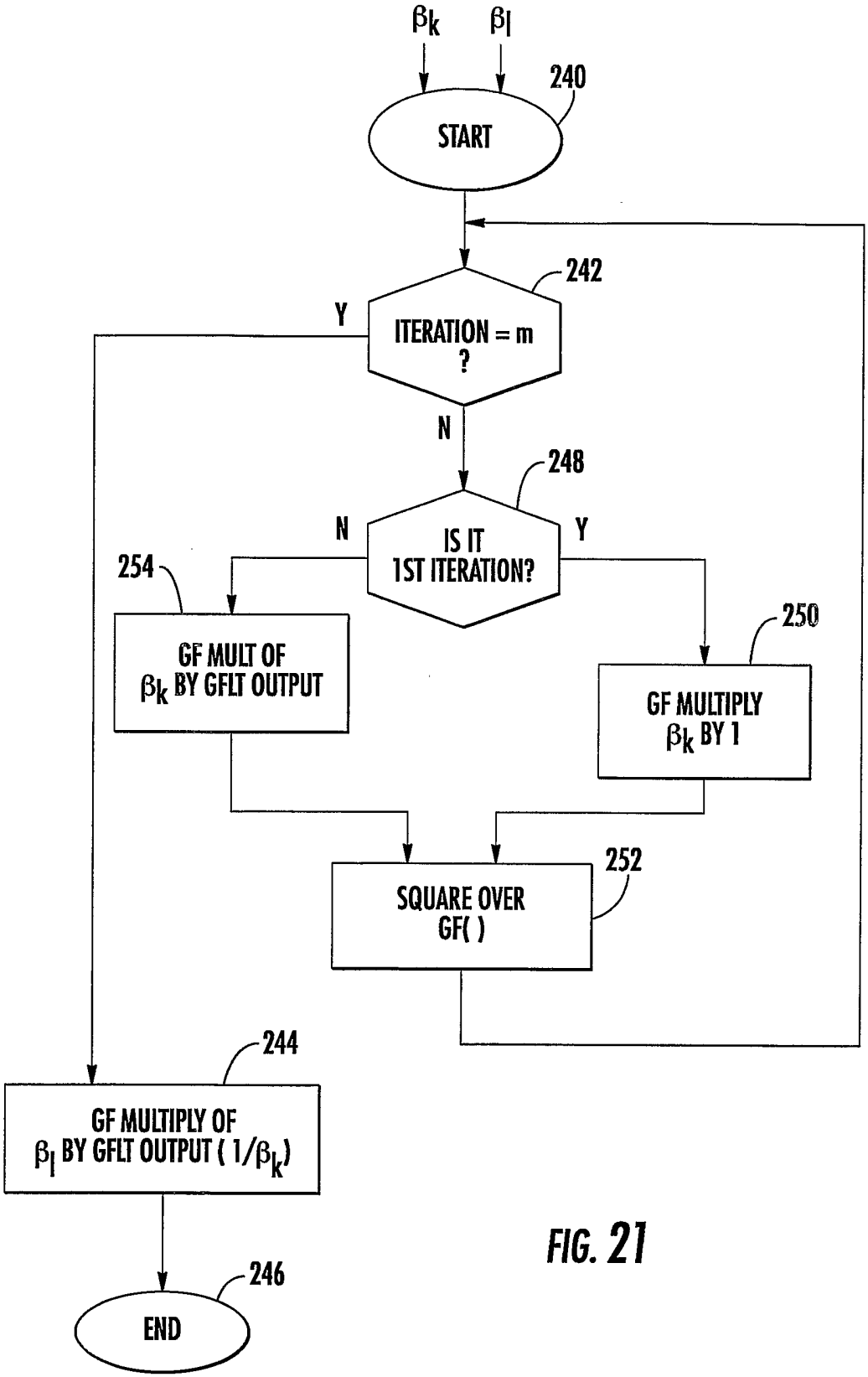


FIG. 21

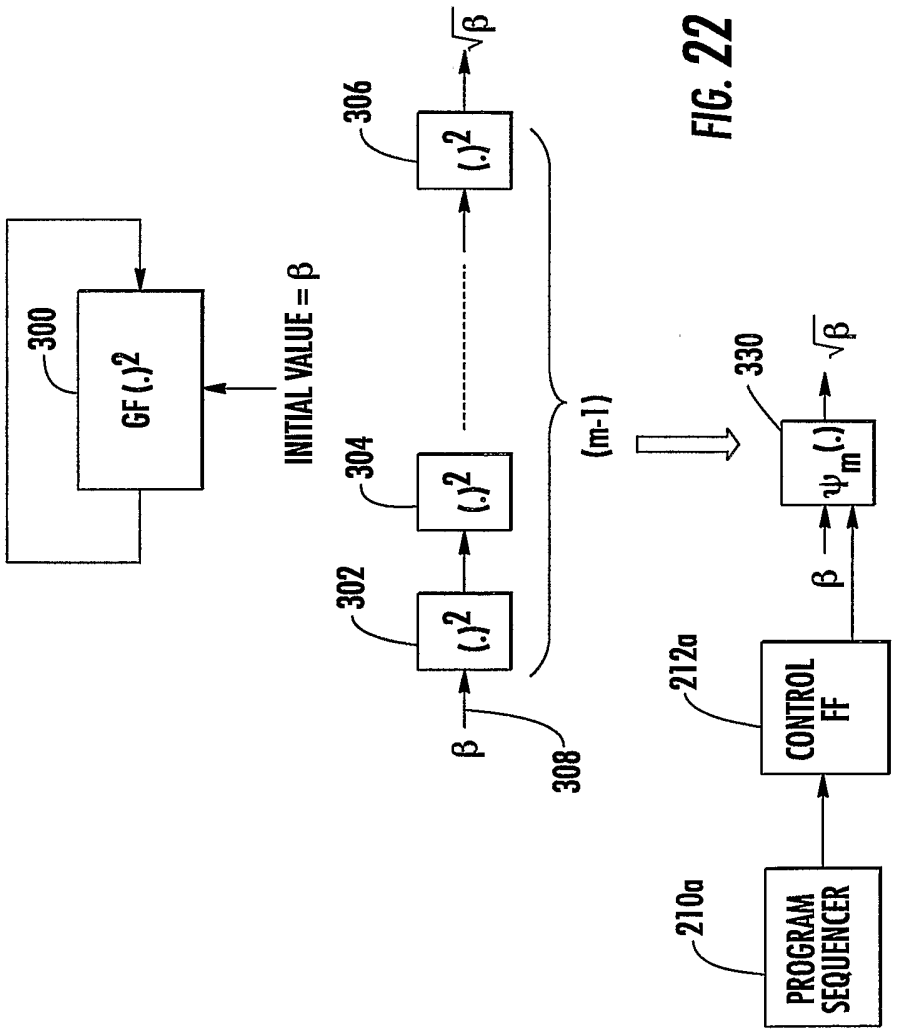

300
GF (.)2
INITIAL VALUE = β
302
(.)2
304
(.)2
306
(.)2
√β
β
308
(m-1)
330
ψm (.)
√β
β
212a
CONTROL FF
210a
PROGRAM SEQUENCER

FIG. 22

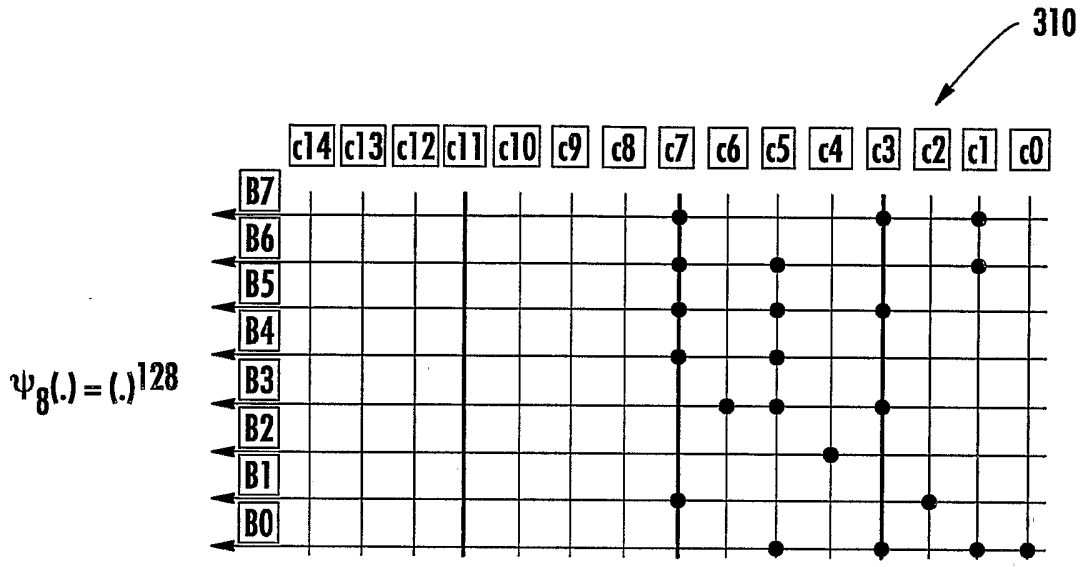

310
c14 c13 c12 c11 c10 c9 c8 c7 c6 c5 c4 c3 c2 c1 c0
B7
B6
B5
B4
B3
B2
B1
B0
$\psi_8(.) = (.)^{128}$

FIG. 23

GALOIS FIELD SQUARE ROOT OF β

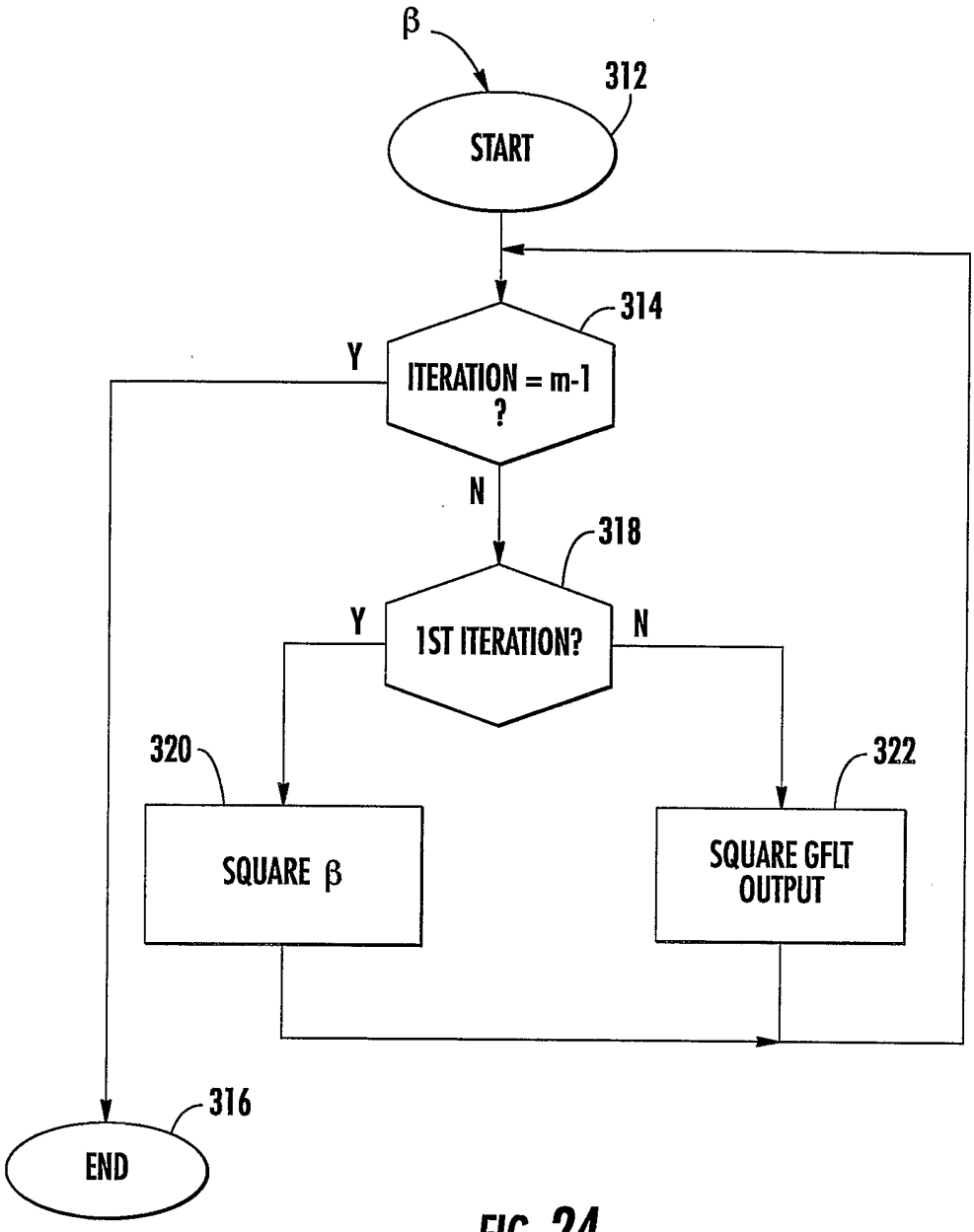


FIG. 24

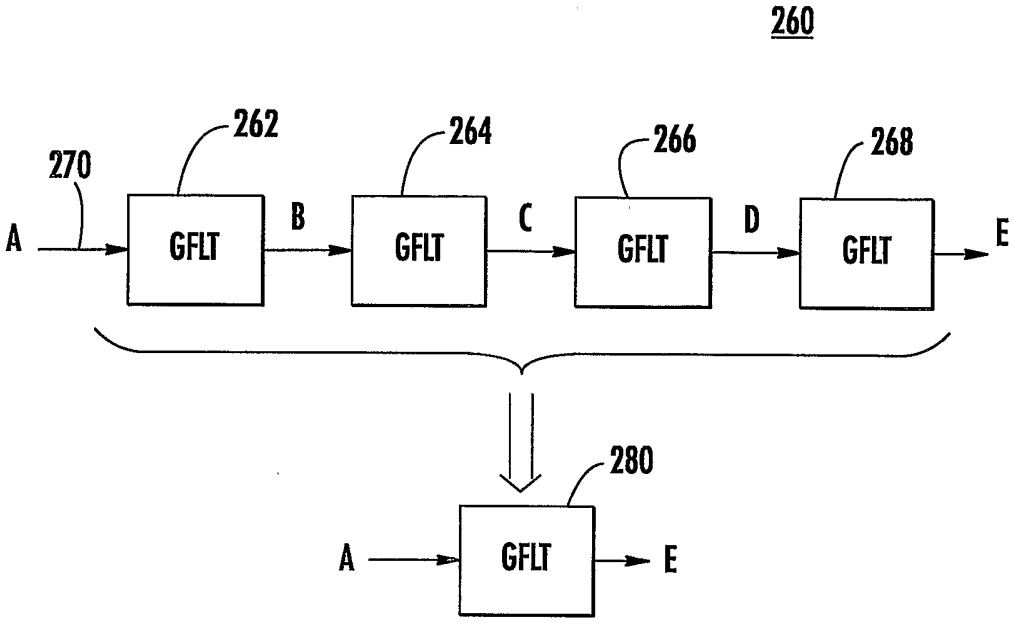


FIG. 25