A VLSI Architecture for Performing Finite Field Arithmetic With Reduced Table Look-Up

I. S. Hsu and T. K. Truong
Communications Systems Research Section

I. S. Reed
Electrical Engineering Department
University of Southern California

A new table look-up method for finding the log and antilog of finite field elements has been developed by N. Glover. In his method, the log and antilog of a field element is found by the use of several smaller tables. The method is based on a use of the Chinese Remainder Theorem. The technique often results in a significant reduction in the memory requirements of the problem. A VLSI architecture is developed for a special case of this new algorithm to perform finite field arithmetic including multiplication, division, and the finding of an inverse element in the finite field.

I. Introduction

A codeword of a cyclic code is a sequence of symbols or characters (Ref. 1). These characters can be represented as the coefficients of a polynomial

$$C(x) = \sum_{i=0}^{m-1} c_i x^i$$

where c_i for $0 \leqslant i \leqslant m-1$ is an element in a finite field $GF(q)$, and m is the length of the codeword. One such cyclic code is a Reed-Solomon (RS) code with symbols which lie in the Galois field $GF(2^n)$ of order 2^n. If β belongs to $GF(2^n)$, then

$$\beta = \sum_{i=0}^{n-1} a_i \alpha^i$$

where $a_i \in GF(2)$ and α is a root of a primitive irreducible polynomial with degree n over $GF(2)$. It is shown in Refs. 2–4 that the arithmetic used to encode and decode RS codes over $GF(2^n)$ requires the multiplication and division of field elements in $GF(2^n)$. The most straightforward method to perform multiplication and division of two field elements in $GF(2^n)$ is to use table look-up. The same method can also be used to find inverses in the finite field.

To illustrate this procedure, let two field elements be represented in binary. That is, let $x = x_0, x_1, \ldots, x_{n-1}$ and $y = y_0, y_1, \ldots, y_{n-1}$, where $x_i, y_i \in GF(2)$ for $0 \leqslant i \leqslant n-1$. Next let a "log" table be used to find the exponents i and j in such a manner that $x = \alpha^i$ and $y = \alpha^j$. Binary addressing is used in the table to locate the logarithms i and j of x and y, respectively. After the addition $k = i + j \bmod (2^n - 1)$ of these exponents, an antilog table is used to find the binary representation of α^k. The exponent k serves as the address of the field element

in the antilog table. For some applications $q = 2^n$ is large and the log and antilog tables may be so large that the consumption of silicon area in the VLSI implementation becomes prohibitive.

Recently, Glover (Ref. 5) developed a new algorithm to reduce the size of the table needed to find the log and antilog of field elements by using several smaller tables. The method is based primarily on a use of smaller look-up tables and the Chinese Remainder Theorem (Ref. 6). To make this new algorithm realizable, a new mapping method based on a special case of the technique in Ref. 5 is found in this article for converting an element in $GF(2^{2n})$ to its counterpart in $(GF(2^n)^2)$ and vice versa, where n is a positive integer. A VLSI architecture is also developed to realize this new algorithm. This VLSI architecture possesses the programmable capability of being able to perform operations such as multiplication, division or the finding of inverses in a finite field.

II. A Reduced Table Look-Up Method for Finding Logs and Antilogs of Elements in a Galois Field

In this section, one of the methods developed by Glover (Ref. 5) to find the log and antilog of finite field elements is described briefly. The details of this method are described in the appendix.

The Log Algorithm. Given $\alpha^m = a + \alpha b$, where α is a primitive element in $GF(2^{2n})$, a, b belong to $GF(2^n)$, and $0 \leqslant m \leqslant 2^{2n} - 1$ is the log of element α^m. The procedure to find $m = \log_\alpha (\alpha^m)$ is described as follows:

Step 1: Map α^m to $a + \alpha b$. The particular mapping technique is described in detail in the following section.

Step 2: Compute $x = a^2 + ab + b^2\beta$ and $y = a/b$, where $\beta = \alpha^{2^{n+1}}$ is a primitive element in $GF(2^n)$.

Step 3: Use the $\log_\beta$ table to find $m_1 = \log_\beta(x)$ and the $\log_\gamma$ table to find $m_2 = \log_\gamma(y)$ for $a \neq 0$, $b \neq 0$, where $\gamma = \alpha^{2^{n-1}}$ is an element in $GF(2^n)$. For $a = 0$, choose $m_2 = 1$. For $b = 0$, choose $m_2 = 0$. Here $\log_\beta(x) = m \bmod (2^n - 1) = m_1$, and $\log_\gamma(y) = m \bmod (2^n + 1) = m_2$.

Step 4: By the Chinese Remainder Theorem (Ref. 6), $m = m_1 \cdot n_2 \cdot M_1 + m_2 \cdot n_1 \cdot M_2$. Here $n = n_1 \cdot n_2 = n_1 \cdot N_1 = n_2 \cdot N_2$, n_i relatively prime and M_i uniquely satisfies (modulo n_i) the congruence $N_i \cdot M_i = 1 \bmod n_i$ for $1 \leqslant i \leqslant 2$.

The Antilog Algorithm. Given m, recover $\alpha^m = a + \alpha b$ as follows:

Step 1: Compute $m_1 = m \bmod (2^n - 1)$ and $m_2 = m \bmod (2^n + 1)$.

Step 2: Use the antilog tables to find $\text{antilog}_\beta(m_1) = x = a^2 + ab + b^2\beta$, and $\text{antilog}_\gamma(m_2) = y = a/b$, for $m_2 \neq 0, 1$. For $m_2 = 1$, $a = 0$, and $b = \sqrt{x/\beta} = \text{antilog}_\beta((\log_\beta(x/\beta))/2)$. For $m_2 = 0$, $b = 0$, and $a = \sqrt{x} = \text{antilog}_\beta((\log_\beta(x))/2)$.

Step 3: For $m_2 \neq 0, 1$, use the equation $b = \text{antilog}_\beta((\log_\beta(z))/2)$, where $z = x/(y^2 + y + \beta)$. Then $a = b \cdot y$.

Step 4: Map $a + \alpha b$ to α^m. This inverse mapping is described in detail in the following section.

To illustrate the above procedures, two examples are given for the finite field $GF(2^8)$.

Example 1: Given $a + \alpha b \in GF(2^8)$, where $a = (0\ 1\ 1\ 0)$ and $b = (1\ 1\ 1\ 0) \in GF(2^4)$. Then, find m such that $\alpha^m = a + \alpha b$. By the log algorithm, $x = a^2 + ab + b^2\beta = (1\ 1\ 1\ 0)$ and $y = a/b = (1\ 1\ 1\ 1)$. Now use Tables A-1 and A-3 to find m_1 and m_2, respectively. The results are $m_1 = 7$ and $m_2 = 8$. For this example, $n_1 = 15$ and $n_2 = 17$. Thus, $M_1 = 8$ and $M_2 = 8$ are the smallest numbers such that $17M_1 \equiv 1 \bmod 15$ and $15M_2 \equiv 1 \bmod 17$, respectively. Hence, $n_2M_1 = 136$ and $n_2 M_2 = 120$. By Eq. (10), Ref. 8, $m \equiv (136 \cdot m_1 + 120 \cdot m_2) \bmod (2^8 - 1) = 127$.

Example 2: Given $m = 127$, find $\alpha^{127} = a + \alpha b \in GF(2^8)$. Using the ANTILOG algorithm, $m_1 \equiv m \bmod (2^n - 1) = 7$ and $m_2 \equiv m \bmod (2^n + 1) = 8$. Then use Tables A-2 and A-4 to find x and y, respectively. The results are $x = (1\ 1\ 1\ 0)$ and $y = (1\ 1\ 1\ 1)$. Thus, $z = x/(y^2 + y + \beta) = (0\ 0\ 1\ 1)$. By Eq. (9), Ref. 8, $b = \text{antilog}_\beta((\log_\beta(z))/2) = \text{antilog}_\beta(7) = (1\ 1\ 1\ 0)$. Thus, $a = b \cdot y = (0\ 1\ 1\ 0)$. Therefore, $\alpha^{127} = (0\ 1\ 1\ 0) + \alpha(1\ 1\ 1\ 0)$.

III. A Method for Mapping Elements of $GF(2^{2n})$ Onto $GF((2^n)^2)$ and Vice Versa

To perform Step 4 of the Antilog Algorithm in Section II, a method is developed in this section for the required mapping. This is accomplished by first considering the mapping of an element α^m in $GF(2^{2n})$ to its counterpart $a + \alpha b$, where $a, b \in GF(2^n)$. This mapping procedure is best described by an example. The extension to other finite fields $GF(2^{2n})$ can be obtained in a similar fashion.

First, Theorem 1 in Ref. 5 is repeated here. A proof of this theorem is given here also in order to make the algorithm more understandable and self-contained.

Theorem 1 (Ref. 5): Let β be a primitive element in $GF(2^n)$ such that the polynomial $p(x) = x^2 + x + \beta$ is irreducible in this field. Also let $\alpha \in GF(2^{2n})$ for $\alpha \neq 0$ where $GF(2^{2n})$ is the quadratic extension field of $GF(2^n)$. If α is a root of $p(x)$, i.e., $p(\alpha) = 0$, and $2^n + 1$ a prime, then α is a primitive element in $GF(2^{2n})$.

Proof: It is shown in Ref. 7 (page 34) that if α is a root of $p(x)$, its conjugate $\bar{\alpha} = \alpha^{2^n}$ is also a root of $p(x)$. Thus,

$$(x + \alpha) \cdot (x + \bar{\alpha}) = x^2 + (\alpha + \bar{\alpha})x + \alpha \cdot \bar{\alpha}$$

$$= x^2 + x + \beta$$

where $\alpha + \bar{\alpha} = 1$ and $\alpha \cdot \bar{\alpha} = \beta$. Hence,

$$\alpha^{2^n+1} = \beta \tag{1}$$

Now $2^{2n} - 1 = (2^n + 1) \cdot (2^n - 1)$ and $2^n + 1$ is a prime by hypothesis. Let γ be any number such that $\gamma | (2^{2n} - 1)$ and $\gamma \neq 2^n + 1$. Then $\gamma | (2^n - 1)$ so that by Eq. (1)

$$\alpha^{(2^{2n}-1)/\gamma} = (\alpha^{2^n+1})^{(2^n-1)/\gamma} = \beta^{(2^n-1)/\gamma} \tag{2}$$

Since β is primitive over $GF(2^n)$, $2^n - 1$ is the least integer such that $\beta^{(2^n-1)} = 1$. Hence $\beta^{(2^n-1)/\gamma} \neq 1$ unless $\gamma = 1$.

On the other hand, if $\gamma = 2^n + 1$, then

$$\alpha^{(2^{2n}-1)/\gamma} = \alpha^{(2^n+1)(2^n-1)/(2^n+1)} = \alpha^{2^n-1}$$

Since α is a root of $p(x) = x^2 + x + \beta$, by Eq. (1), one has

$$p(\alpha) = \alpha^2 + \alpha + \beta$$

$$= \alpha^2 + \alpha + \alpha^{2^n+1}$$

$$= 0$$

Thus, $\alpha^{2^n-1} = \alpha^{-1} + 1 \neq 1$ for otherwise $\alpha^{-1} = 0$, which is impossible since $\alpha \neq 0$.

Thus, by Eq. (2) and the above, if $\gamma | 2^{2n} - 1$, then $\alpha^{(2^{2n}-1)/\gamma} \neq 1$ unless $\gamma = 1$. Therefore, the order of α is $2^{2n} - 1$ and α is primitive in $GF(2^{2n})$.

Q.E.D.

To illustrate the consequences of the above theorem, let $n = 4$ and let α be the root of polynomial

$$p(x) = x^2 + x + \beta$$

where β is a primitive element in $GF(2^4)$. For this case by Eq. (1)

$$\beta = \alpha^{2^4+1} = \alpha^{17} \tag{3}$$

By the above theorem, since 17 is a prime, α is a primitive element in the extension field $GF(2^8)$.

Since α is the root of $p(x)$, $p(\alpha) = \alpha^2 + \alpha + \alpha^{17} = 0$. Hence, α satisfies the reduced equation

$$\alpha^{16} = \alpha + 1 \tag{4}$$

Now the root of the irreducible primitive polynomial $f(x)$ over $GF(2)$ which generates the finite field $GF(2^8)$ must also satisfy Eq. (3). Of the many irreducible polynomials which generate $GF(2^8)$, consider the special irreducible polynomial

$$f(x) = x^8 + x^6 + x^5 + x^3 + 1 \tag{5}$$

Let α be any root of $f(x)$ so that α satisfies

$$\alpha^8 + \alpha^6 + \alpha^5 + \alpha^3 + 1 = 0$$

or alternatively

$$\alpha^8 = \alpha^6 + \alpha^5 + \alpha^3 + 1 \tag{6}$$

It is now shown that α also satisfies Eq. (4). Squaring both sides of Eq. (6) one obtains,

$$\alpha^{16} = \alpha^{12} + \alpha^{10} + \alpha^6 + 1$$

$$= (\alpha^7 + \alpha^3 + \alpha^2 + \alpha + 1) + (\alpha^7 + \alpha^6 + \alpha^3 + \alpha^2 + 1) + \alpha^6 + 1$$

$$= \alpha + 1$$

Hence the irreducible polynomial $f(x)$ in Eq. (5) is so chosen that a root α of $f(x)$ also is a root of the quadratic generator polynomial $p(x) = x^2 + x + \beta$ of finite field $GF(2^8)$ over the smaller field $GF(2^4)$, where $\beta = \alpha^{17}$.

In order to simplify operations by table look-up, one would like to represent an element α^m in $GF(2^8)$ by $\alpha^m = a + \alpha b$, where $a,b \in GF(2^4)$, the smaller field. To find the

mapping which makes this representation possible, one must find the generator polynomial $g(x)$ of $GF(2^4)$ which β satisfies. By Eq. (3), $\beta = \alpha^{17}$, where β is a primitive element in $GF(2^4)$. Hence $g(x)$, the generator polynomial of field $GF(2^4)$ over $GF(2)$, must satisfy the equation

$$g(\beta) = g(\alpha^{17}) = 0$$

Try the irreducible polynomial

$$g(x) = x^4 + x^3 + 1$$

over $GF(2)$. Substituting $x = \alpha^{17}$ into the above $g(x)$ yields

$$g(\alpha^{17}) = (\alpha^{17})^4 + (\alpha^{17})^3 + 1 \tag{7}$$

However, by Eq. (4),

$$\alpha^{16} = \alpha + 1$$

which implies

$$\alpha^{17} = \alpha^2 + \alpha$$

Substituting the above equation into Eq. (7) yields

$$\begin{aligned} g(\alpha^{17}) &= (\alpha^2 + \alpha)^4 + (\alpha^2 + \alpha)^3 + 1 \\ &= (\alpha^8 + \alpha^4) + (\alpha^6 + \alpha^4 + \alpha^5 + \alpha^3) + 1 \\ &= 0 \end{aligned}$$

Hence $g(x) = x^4 + x^3 + 1$ is the correct generator polynomial of the finite field $GF(2^4)$.

Given an element α^m in $GF(2^8)$, α^m can be expressed in standard basis form as

$$\begin{aligned} \alpha^m = {} & C_0\alpha^0 + C_1\alpha^1 + C_2\alpha^2 + C_3\alpha^3 + C_4\alpha^4 \\ & + C_5\alpha^5 + C_6\alpha^6 + C_7\alpha^7 \end{aligned} \tag{8}$$

where $C_i \in GF(2)$ for $0 \leqslant i \leqslant 7$. α^m also can be represented as $\alpha^m = a + \alpha b$, where $a,b \in GF(2^4)$.

Now $\beta = \alpha^{17}$ is the primitive element found above for $GF(2^4)$. Hence a,b in $GF(2^4)$ can be represented over $GF(2)$ in standard basis form as

$$a = a_0\beta^0 + a_1\beta^1 + a_2\beta^2 + a_3\beta^3 \tag{9a}$$

and

$$b = b_0\beta^0 + b_1\beta^1 + b_2\beta^2 + b_3\beta^3 \tag{9b}$$

where $a_i, b_i \in GF(2)$, for $0 \leqslant i \leqslant 3$.

Now substitute representations (8) and (9) into equation $\alpha^m = a + \alpha b$ to obtain

$$\begin{aligned} & C_0\alpha^0 + C_1\alpha^1 + C_2\alpha^2 + C_3\alpha^3 + C_4\alpha^4 + C_5\alpha^5 + C_6\alpha^6 + C_7\alpha^7 \\ & = a_0\beta^0 + a_1\beta^1 + a_2\beta^2 + a_3\beta^3 + (b_0\beta^0 + b_1\beta^1 + b_2\beta^2 \\ & \quad + b_3\beta^3) \cdot \alpha \end{aligned}$$

Since $\beta = \alpha^{17}$, the right side of the above equality can be represented in terms of powers of α as

$$\begin{aligned} & = a_0 + a_1\alpha^{17} + a_2\alpha^{34} + a_3\alpha^{51} + b_0\alpha + b_1\alpha^{18} + b_2\alpha^{35} \\ & \quad + b_3\alpha^{52} \\ & = a_0 + (a_1 + b_0)\alpha + (a_1 + a_2 + b_1)\alpha^2 + (a_3 + b_1 + b_2)\alpha^3 \\ & \quad + (a_2 + a_3 + b_3)\alpha^4 + (a_3 + b_2 + b_3)\alpha^5 \\ & \quad + (a_3 + b_3)\alpha^6 + b_3\alpha^7 \end{aligned}$$

If the coefficients of the corresponding powers of α are equated and expressed in matrix form, one obtains

$$\mathbf{C} = \mathbf{M} \cdot \mathbf{A} \tag{10}$$

where

$$\mathbf{C} = \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{A} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

The inverse matrix $\mathbf{M}^{-1}$ of $\mathbf{M}$ is obtained readily and is given by

$$\mathbf{M}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (11)$$

Hence, one obtains

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \end{bmatrix}$$

or $\mathbf{A} = \mathbf{M}^{-1}\mathbf{C}$. Equation (11) is the mapping of elements in $GF(2^8)$ to the corresponding elements in $GF((2^4)^2)$. It is the inverse mapping of the mapping from $GF((2^4)^2)$ to $GF(2^8)$, given in Eq. (10). Equation (11) illustrates for $n = 4$, the mapping needed in Step 4 of the Antilog Algorithm in Section II.

IV. VLSI Architecture for Performing Multiplication, Division and Finding the Inverse of Finite Field Elements

In this section, a VLSI architecture is developed to perform multiplication, division and the finding of an inverse element in the finite field $GF(2^8)$ using the new algorithm described in the last two sections. This chip is designed to be programmable so that multiplication, division or the finding of an inverse in $GF(2^8)$ can be performed.

Figure 1 depicts the overall diagram of this chip. In Fig. 1, IN_1 and IN_2 are the two inputs to this chip. They are field elements in $GF(2^8)$. If the operation is to find the inverse element, IN_1 equals zero and IN_2 is the element to be inverted. Here it is assumed that these field elements are represented in the standard basis. That is,

$$IN_i = \sum_{j=0}^{7} a_j \alpha^j \qquad (i = 1, 2)$$

Input C_1 is the control signal used to control which operation is to be performed. If C_1 equals one, multiplication is performed. If C_1 equals zero, then either division or the

inverse operation is performed, according to whether IN_1 is zero or not. If IN_1 equals zero, then the inverse operation is performed, otherwise the division operation is carried out.

OUTPUT is the data output pin for the calculated result. Figure 2 shows the block diagram of this chip. As shown in the figure, the input field elements α^m and α^n are first converted to their corresponding elements $a + \alpha b$ and $a' + \alpha b'$ in $GF((2^4)^2)$. The needed mapping matrix is found in Eq. (11) of Section III.

Next, the values of x, y, x' and y' shown in Fig. 2 are calculated by the following equations:

$$x = a^2 + ab + b^2$$

$$y = a \cdot b^{-1}$$

$$x' = a'^2 + a'b' + b'^2$$

$$y' = a'b'^{-1}$$

The arithmetic operations such as the square and the inverse, needed above, are performed also by the table look-up method since they can be performed first in the much smaller field of $GF(2^4)$. The look-up table for $GF(2^4)$ involves only a small area in silicon and as a consequence is easier to realize in VLSI.

After the values of x, y, x', y' are obtained, their corresponding log values m_1, m_2, m_1', m_2' are obtained through the table look-up method as well. Again, this is readily realizable since these values are in the smaller field $GF(2^4)$. The log values of α^m and α^n can be obtained by equations

$$m = m_1 \cdot n_2 \cdot M_1 + m_2 \cdot n_1 \cdot M_2$$

$$n = m_1' \cdot n_2' \cdot M_1' + m_2' \cdot n_1' \cdot M_2'$$

respectively.

In Fig. 2, either $m + n$ or $m - n$ is performed in accordance with the control signal C_1. If C_1 equals one, addition is performed as needed in the multiplication operation. Otherwise a subtraction is performed as needed for division or the inverse operations. If the operation is to find the inverse element, then one of the inputs is zero, and its corresponding log value is also zero. Therefore, the operation $m - n$ corresponds to a negation of the logarithm of the nonzero input, i.e., $-n$ is obtained. This is the logarithm value of the inverse element.

The number p, in Fig. 2, results from either $(m + n)$ mod 255 or $(m - n)$ mod 255. These results are obtained through the modulo 255 circuit. The result of this computation is then fed to two modulus circuits to obtain the results modulus m_1, and m_2. m_1 and m_2 are the values of p modulo $2^4 - 1$ and $2^4 + 1$, respectively. Two antilog tables are then used to find the values x and y as described in Section II, Step 2 of the Log Algorithm. Next, $z = x/(y^2 + y + \beta)$ is calculated and a $\log_\beta$ table is used to find the logarithm of z. This value is then divided by two to obtain the value "A" as shown in Fig. 2. At this stage, element y is delayed for a certain number of clock cycles for the purpose of synchronization. An antilog table is then used to obtain element b. If b is combined with y, obtained previously, then element a can be obtained by performing the following operation:

$$a = by$$

Finally, the inverse mapping circuit described in the previous section is used to obtain the corresponding element of $a + \alpha b$, i.e., α^p, in $GF(2^8)$.

Because all operations performed in this chip are in the smaller finite field $GF(2^4)$, operations in $GF(2^4)$ are performed by the table look-up method occupying nearly half the silicon area as one would expect for $GF(2^8)$. It is obvious that the tables in $GF(2^4)$ can be reduced to smaller tables in $GF(2^2)$ by the same technique. The increased overhead associated with this reduction appears to be larger than the benefits that this further reduction might obtain for this case.

V. Conclusion

A VLSI architecture for performing finite field arithmetic is described in this article. A chip is designed to be programmable so that multiplication, division or the finding of an inverse in $GF(2^8)$ can be performed. The algorithm used is due to Glover (Ref. 5), where a new table look-up method for finding the log and antilog of finite field elements is developed. In that method, the log and antilog of a finite field element are found by the use of several smaller tables. The method is based on a use of the Chinese Remainder Theorem. The technique often results in a significant reduction in the memory requirements of the problem.

A method for mapping elements of $GF(2^{2n})$ onto $GF((2^n)^2)$ and vice versa is also developed in this article. An example of mapping elements of $GF(2^8)$ onto $GF((2^4)^2)$ is given for the purpose of illustration. The extension to other finite fields $GF(2^{2n})$ can be obtained in a similar fashion.

References

1. MacWilliams, F. J., and Sloane, N. J. A., *The Theory of Error-Correcting Codes,* North-Holland Publishing Company, NY, 1981.
2. Berlekamp, E. R., *Algebraic Coding Theory*, McGraw-Hill, NY, 1958.
3. Reed, I. S., Truong, T. K., and Miller, R. L., "Simplified Algorithm for Correcting Both Errors and Erasures of Reed-Solomon Codes," *Proc. IEE,* Vol. 126, No. 10, Oct. 1979.
4. Winograd, S., "On Computing the Discrete Fourier Transform," *Proc. Natl. Acad. Scie.*, USA, Vol. 73, pp. 1005-1006, 1976.
5. Glover, N., *Practical Error Correction Design for Engineers,* Data Systems Technology Corp., Broomfield, CO, 1982.
6. Ore, O., *Number Theory and Its History*, McGraw-Hill, NY, 1948.
7. Lin, S., and Costello, D. J., Jr., *Error Control Coding*, Prentice-Hall, NJ, p. 34, 1983.
8. Glover, N., Reed, I. S., Truong, T. K., and Huang, J. T., "Methods to Reduce the Table Look-Up Requirements for Finding Logs or Antilogs of Elements in a Power Galois Field," submitted to *IEEE Trans. on Computers.*

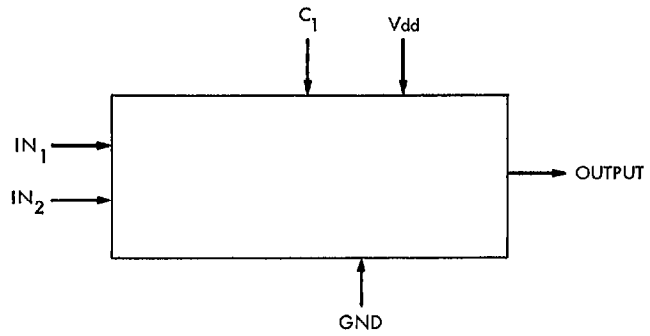

Fig. 1. Overall diagram of the VLSI chip for performing multiplication/division or finding inverse in a finite field

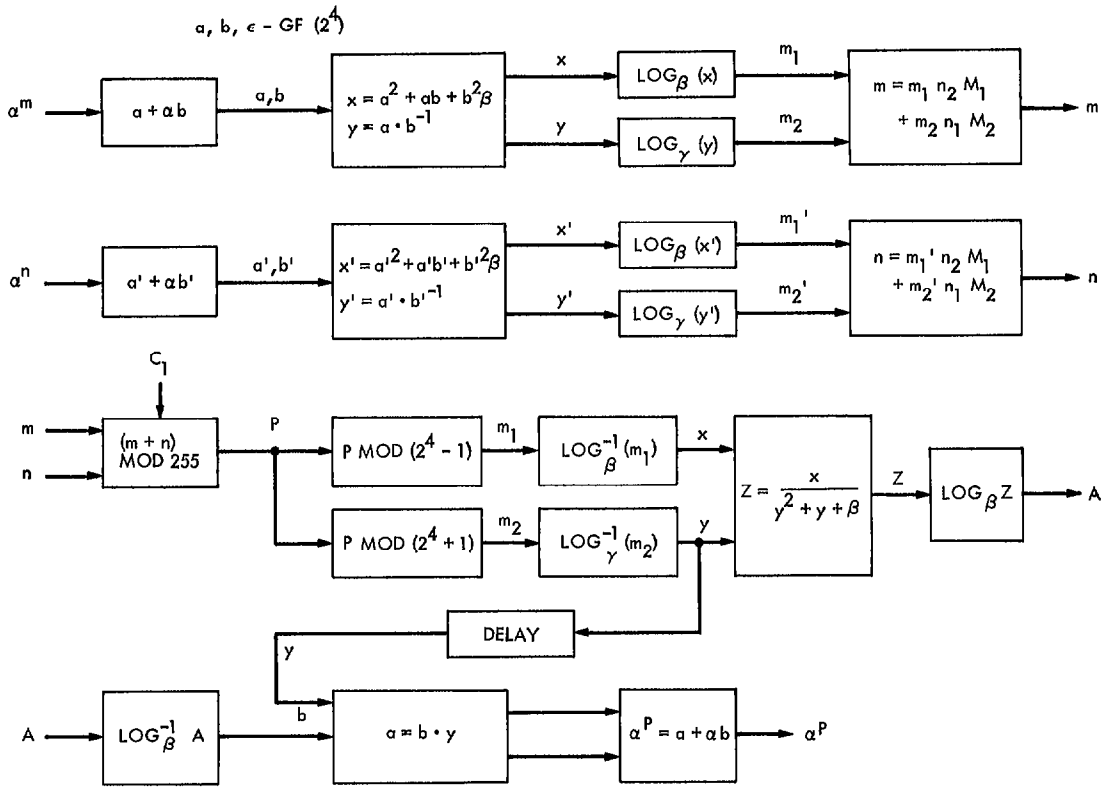

Fig. 2. Block diagram of the VLSI chip for performing multiplication/division or inverse of finite field elements

Appendix

Algorithm for Finding Log and Antilog of Finite Field Elements

In this appendix, the algorithm used to find the log and antilog of finite field elements is described in detail. A special case of Theorem A-1 and A-2, given next, was originally found by Glover (see Ref. 5).

Definition A-1: For $\alpha \in GF(2^{2n})$ and $a + \alpha b \in GF(2^{2n})$, where $a, b \in GF(2^n)$, define the norm of $a + \alpha b$ to be

$$||a + \alpha b|| = (a + \alpha b) \cdot \overline{(a + \alpha b)}$$

where the bar denotes element conjugation in $GF(2^{2n})$.

Theorem A-1: Let β be a primitive element in $GF(2^n)$ such that the quadratic polynomial $x^2 + x + \beta$ is irreducible over $GF(2^n)$. Suppose also that $2^n + 1$ is a prime integer. Next let α be the root of this polynomial in the extension field $GF(2^{2n}) = \{a + \alpha b | a, b \in GF(2^n)\}$ of $GF(2^n)$. Suppose $\alpha^m = a + \alpha b \in GF(2^{2n})$. Then $m_1 = \log_\beta (a^2 + ab + b^2\beta) \equiv m \bmod (2^n - 1)$, i.e., m_1 is least integer such that $\beta^{m_1} = ||\alpha^{m_1}|| = ||a + \alpha b|| = a^2 + ab + b^2\beta$.

Proof: Since $x^2 + x + \beta$ is irreducible over $GF(2^n)$, it has roots α and

$$\overline{\alpha} = \alpha^{2^n} \tag{A-1}$$

in the extension field $GF(2^{2n})$. By Theorem 1, α is primitive in $GF(2^{2n})$. By definition A-1 and Theorem 1, one has the following:

$$\begin{aligned} ||a + \alpha b|| &= (a + \alpha b) \cdot \overline{(a + \alpha b)} = (a + \alpha b) \cdot (a + \overline{\alpha} b) \\ &= a^2 + ab + b^2\beta \end{aligned} \tag{A-2}$$

If $c + \alpha d$ is any other element in $GF(2^{2n})$ and $c, d \in GF(2^n)$, then

$$\begin{aligned} \overline{(a + \alpha b) \cdot (c + \alpha d)} &= (a + \alpha b)^{2^n} \cdot (c + \alpha d)^{2^n} \\ &= \overline{(a + \alpha b)} \cdot \overline{(c + \alpha d)} \end{aligned}$$

Thus, by the definition of the norm, one has

$$\begin{aligned} ||(a + \alpha b)(c + \alpha d)|| &= (a + \alpha b)(c + \alpha d)\,\overline{(a + \alpha b)}\,\overline{(c + \alpha d)} \\ &= ||a + \alpha b|| \cdot ||c + \alpha d|| \end{aligned} \tag{A-3}$$

Observe next by Eq. (A-1) that $||\alpha|| = \alpha \cdot \overline{\alpha} = \beta$ so that the theorem is true for $m = 1$.

For purpose of induction assume that

$$||\alpha^k|| = \beta^k \tag{A-4}$$

for all k such that $1 \leq k \leq m$. Then, by (A-3), for $k = m + 1$, $||\alpha^{m+1}|| = ||\alpha^m|| \cdot ||\alpha|| = \beta^{m+1}$. Hence the induction is complete and (A-4) is true for all k.

Now represent α^m by $a + \alpha b$ for some $a, b \in GF(2^n)$. Then, by (A-2) and (A-4), $||\alpha^m|| = \beta^m$. Since β has order $2^n - 1$, the theorem must be true. Q.E.D.

By Theorem A-1 one can construct a $\log_\beta$ table of $2^n - 1$ elements by storing the value $m_1 \equiv m \bmod (2^n - 1)$, where $0 \leq m_1 < 2^n - 1$, at location $a^2 + ab + b^2\beta$ such that $\alpha^m = a + \alpha b$. Then, with a, b known, one can find m_1 using the $\log_\beta$ table. Similarly, the antilog_β table is constructed by storing the binary representation of $a^2 + ab + b^2\beta$ at location m_1 such that $\alpha^m = a + \alpha b$ and $\text{antilog}_\beta (m_1) = a^2 + ab + b^2\beta = x$.

Next, the construction of tables of $2^n + 1$ elements is shown.

Theorem A-2: Let $\gamma = \alpha^{2n} - 1 \in GF(2^{2n})$, where α is a primitive element of $GF(2^{2n})$. Suppose $\alpha^m = a + \alpha b \in GF(2^{2n})$ for some $a, b \in GF(2^n)$. Then,

$$m^2 = \log_\gamma \frac{(a + \overline{\alpha} b)}{(a + \alpha b)} \equiv m \bmod (2^n + 1) \tag{A-5}$$

i.e., m_2 is least integer such that $\gamma^{m_2} = \alpha^{\overline{m}}/\alpha^m$.

Proof: Since α is a primitive element in $GF(2^{2n})$ and $\gamma = \alpha^{2^{n-1}}$, the order of γ is $2^n + 1$. By the definition of the norm, one has

$$||\alpha|| = \alpha \cdot \overline{\alpha} = \alpha \cdot \alpha^{2^n} = \gamma \cdot \alpha^2 \tag{A-6}$$

For purposes of induction assume that

$$||\alpha^k|| = \gamma^k \alpha^{2k} \tag{A-7}$$

for $1 \leq k \leq m$. Then, by (A-3), $||\alpha^{m+1}|| = ||\alpha^m|| \cdot ||\alpha|| = (\gamma^m \cdot \alpha^{2m})(\gamma\alpha^2) = \gamma^{m+1} \cdot \alpha^{2(m+1)}$. Hence the induction is complete and (A-7) is true for all k.

Representing α^m by $a + \alpha b$ for some $a,b \in GF(2^n)$, it follows from (A-6) that

$$||a + \alpha b|| = \gamma^m \cdot (a + \alpha b)^2 \qquad \text{(A-8)}$$

Multiplying both sides of (A-8) by $(a + \bar{\alpha} b)^2$ yields $||a + \alpha b|| \cdot (a + \bar{\alpha} b)^2 = \gamma^m (a + \alpha b)^2 (a + \bar{\alpha} b)^2 = \gamma^m ||a + \alpha b||^2$.

Therefore, from the definition of the norm,

$$\gamma^m = \frac{(a + \bar{\alpha} b)^2}{||a + \alpha b||} = \frac{a + \bar{\alpha} b}{a + \alpha b} \qquad \text{(A-9)}$$

Since the order of γ is $2^n + 1$, the theorem must follow. Q.E.D.

Using the results of Theorem A-2, let

$$f(a/b) = \gamma^m = ((a/b) + \bar{\alpha})/((a/b) + \alpha)$$

$$= (a + \bar{\alpha} b)/(a + \alpha b) \qquad \text{(A-10)}$$

To construct the $\log_\gamma$ table, notice that when $a = 0, f(a/b) = \gamma^m = \alpha^{2^n - 1} = \gamma$ and $m = 1$. For $m_2 \equiv m \bmod (2^n + 1)$, one has $m_2 = 1$ when $a = 0$. When $b = 0, f(a/b) = (a + 0)/(a + 0) = 1$. Thus, $m = 0$ and $m_2 = 0$. The remaining part of the $\log_\gamma$ table can then be constructed by storing the values $m_2 \equiv m \bmod (2^n - 1)$ at locations "a/b" for $\alpha^m = a + \alpha b$, where $2 \leqslant m_2 \leqslant 2^n$. The antilog_γ table is constructed by storing the binary representation of $a/b \in \{\beta^1, \beta^2, \ldots, \beta^{2^n - 1}\}$ at the corresponding locations, $i = m_2$, for $2 \leqslant i \leqslant 2^n$. Thus,

$$\text{Antilog}_\gamma(m_2) = a/b = y \qquad \text{(A-11)}$$

From (A-10) and (A-11) the two simultaneous equations needed to solve for a and b in the expression $\alpha^m = a + \alpha b$ are given as follows:

Let

$$a^2 + ab + b^2\beta = x \qquad \text{(A-12a)}$$

and

$$\frac{a}{b} = y \qquad \text{(A-12b)}$$

Relations (A-12) yield the following solution:

$$b = \sqrt{\frac{x}{y^2 + y + \beta}}$$

and

$$a = b \cdot y$$

For $b \in GF(2^n)$ it is verified readily that

$$b = \text{antilog}_\beta \left(\frac{\log_\beta z}{2} \right)$$

where $z = x/(y^2 + y + \beta)$.

It is desired now to find the logarithm with base α of $\alpha^m = a + \alpha b \in GF(2^{2n})$, where $a,b \in GF(2^n)$, and $\alpha \in GF(2^{2n})$ is primitive. This can be found from the powers m_1 and m_2, by using the tables of the $2^n - 1$ powers of β and the $2^n + 1$ powers of γ, respectively, as follows:

Let

$$2^{2n} - 1 = (2^n + 1) \cdot (2^n - 1) = n_1 \cdot n_2 = n_1 \cdot N_1 = n_2 \cdot N_2$$

where $N_1 = n_2$ and $N_2 = n_1$. Then, by the Chinese Remainder Theorem, $m = \log_\alpha(\alpha^m)$ is given by

$$m = m_1 \cdot n_2 \cdot M_1 + m_2 \cdot n_1 \cdot M_2 \bmod (2^{2n} - 1)$$

where M_1 and M_2 are the smallest integers such that $n_2 M_1 \equiv 1 \bmod n_1$ and $n_1 M_2 \equiv 1 \bmod n_2$, respectively.

Table A-1. Log_β table for finding $m_1 \equiv m \bmod 2^n - 1$ from known $\alpha^m = a + \alpha b$ where $2^n - 1 = 15$

Location $x = a^2 + ab + b^2\beta$				Content $m_1 \equiv \log_\beta x \bmod 15$
β_3	β_2	β_1	β_0	
0	0	0	1	3
0	0	1	0	2
0	0	1	1	14
0	1	0	0	1
0	1	0	1	10
0	1	1	0	13
0	1	1	1	8
1	0	0	0	0
1	0	0	1	4
1	0	1	0	9
1	0	1	1	11
1	1	0	0	12
1	1	0	1	5
1	1	1	0	7
1	1	1	1	6

Table A-2. Log_β table for finding $x = a^2 + ab + b^2\beta$ from known $m_1 = m \bmod(2^n - 1)$ where $2^n - 1 = 15$

Location m_1	Content $x = \text{antilog}_\beta (m_1)$			
	β_3	β_2	β_1	β_0
0	0	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	1
5	1	1	0	1
6	1	1	1	1
7	1	1	1	0
8	0	1	1	1
9	1	0	1	0
10	0	1	0	1
11	1	0	1	1
12	1	1	0	0
13	0	1	1	0
14	0	0	1	1
15	1	0	0	0

Table A-3. Log_γ table for finding $m_2 \equiv m \bmod (2^n - 1)$ where $2^n + 1 = 17$ from known $\alpha^m = a + \alpha b$

Location $y = a/b$				Content $m_2 = \log_\gamma (y) \bmod 17$
β_3	β_2	β_1	β_0	
0	0	0	1	14
0	0	1	0	12
0	0	1	1	6
0	1	0	0	2
0	1	0	1	10
0	1	1	0	4
0	1	1	1	9
1	0	0	0	16
1	0	0	1	3
1	0	1	0	5
1	0	1	1	11
1	1	0	0	15
1	1	0	1	7
1	1	1	0	13
1	1	1	1	8

Table A-4. Antilog_γ table for finding $y = a/b$ from known $m_2 \equiv m \bmod (2^n + 1)$ where $2^n + 1 = 17$

Location m_2	Content $y = \text{antilog}_\gamma (m_2)$			
	β_3	β_2	β_1	β_0
2	0	1	0	0
3	1	0	0	1
4	0	1	1	0
5	1	0	1	0
6	0	0	1	1
7	1	1	0	1
8	1	1	1	1
9	0	1	1	1
10	0	1	0	1
11	1	0	1	1
12	0	0	1	0
13	1	1	1	0
14	0	0	0	1
15	1	1	0	0
16	1	0	0	0