Design of Galois Field (2^3) Inversion Architecture

Ujwala S.Ghodeswar[1], G.G.Sarate[2]
[1]Dept.of EE,YCCE,Nagpur
[2]Ram Meghe Institute of Technology & Research, Badnera

Abstract: In this paper Very large scale integration architecture for inversion is presented. This architecture is scalable with respect to the throughput rate. The scalability is achieved by applying time sharing technique. Time sharing systems are designed to allow several operations to execute simultaneously. This approach leads to a small silicon area in comparison with several inversion implementations published in the past.

INTRODUCTION

In very-large-scale integration (VLSI) technology, commercial interest in integrating RS decoders in high-volume applications is steadily increasing.[1] For example, it is highly desirable to have an efficient and scalable RS decoder architecture that covers different requirements according to the digital video broadcast (DVB) standard for terrestrial, satellite, or cable transmission. Furthermore, this architecture should be suitable for requirements concerning fast asynchronous transfer mode (ATM) network applications with data rates as specified by the synchronous digital hierarchy standard. While data rates according to DVB applications below 80 Mbit/s vary depending on the channel bandwidth and the modulation scheme, the data rates for fast ATM networks are specified by 155 Mbit/s, 622 Mbit/s, or even higher.[2,3]

A. BASICS OF GALOIS FIELD:

Galois Field inversion hardware can be used for Reed-Solomon encode and decode functions. To understand the relevance of the Galois Field inversion hardware, it is necessary to first define some mathematical terms. Two kinds of number systems that are common in algorithm development are integers and real numbers. For integers the addition, subtraction and multiplication operations can be performed. Division can also be performed if a non-zero remainder can be allowed. For real numbers all four of these operations can be performed, even if there is a non-zero remainder for division operations.[4,5]

Real numbers can belong to a mathematical structure called a field. A field consists of a set of data elements along with addition, subtraction, multiplication, and division. A field of integers can also be created if modulo arithmetic is performed. An example is doing arithmetic using integers modulo-2. Perform the operations using normal integer arithmetic and then take the result modulo-2. Fig. 1 describes addition, subtraction and multiplication modulo-2.

Addition		
+	0	1
0	0	1
1	1	0

Subtraction		
–	0	1
0	0	1
1	1	0

Multiplication		
×	0	1
0	0	0
1	0	1

Fig. 1. Modulo-2 arithmetic

Note that addition and subtraction results are the same, and in fact are equivalent to the XOR (exclusive OR) operation in binary. Also, the multiplication result is equal to the AND operation in binary. These properties are unique to modulo-2 arithmetic, but modulo-2 arithmetic is used extensively in error correction coding. Another more general property is that division by any non-zero element is now defined. Division can always be performed if every element other than zero has a multiplicative inverse,
i.e. $x. x^{-1} = 1$.

P(x) denotes the primitive polynomial of GF(2^3),$P(x)=x^3+x+1$
A finite field division (FFD) can be decomposed into an inversion(FFI) and a multiplication. Inversion architecture is based on the Euclidean algorithm(EA). The least significant coefficients of c(x) results at the output Y0 of the processing element in the mth column from the right
Mapping field element in terms of basis element for GF (2^m) with $f(x) = 1+ x + x^3$ as irreducible primitive polynomial.

Field elements	Basic elements		
	x^0	x^1	x^2
0	0	0	0
α^0	1	0	0
α^1	0	1	0
α^2	0	0	1
α^3	1	1	0
α^4	0	1	1
α^5	1	1	1
α^6	1	0	1
α^7	1	0	0

Table 1.1: Mapping of Basic Elements GF (2^3) with $f(x) = 1+ x + x^3$

B. EUCLIDEAN ALGORITHM

Among important arithmetic operations in finite field, inversion and division have been identified as the most complicated and time-consuming tasks. In previous architecture, there are two disadvantages which are removed

in this circuit. The disadvantages are given as the time step required in the computation depends on the parameter m, the size of the field and the second one is the number of clock cycles required in each computation in not a constant value. Assuming element A and nonzero element B to be two arbitrary element in finite field GF (2^m), the conventional method to perform division A/B is to evaluate the inversion of element B first, then product of $A.B^{-1}$. Hence division is performed in two steps.[6]

A finite field GF (2^m) contains 2^m elements that are generated by a primitive polynomial of degree m with coefficients over GF(2),

$$F(x) = x^m + f_{m-1}x^{m-1} + f_{m-2}x^{m-2} + \ldots + f_1x + 1 \qquad (1)$$

By means of the polynomial representation, an element in GF (2^m) may be represented by a polynomial of degree *m*-1 or less with coefficients over GF (2). Let

$$B(x) = b_{m-1}x^{m-1} + \ldots + b_1x + b_0 \qquad (2)$$

Be a nonzero element of the field GF (2^m). Suppose that *C(x)* is the inverse element of element *B(x)*.

$$C(x) = C_{m-1}x^{m-1} + \ldots + c_1x + c_0 \qquad (3)$$

Then they must satisfy the following relation:

$$B(x) \cdot C(x) \equiv 1 \quad \{\mathrm{mod} F(x)\} \qquad (4)$$

or equivalently,

$$B(x)\cdot C(x) + F(x)\cdot D(x) = 1 \qquad (5)$$

Since polynomial *F(x)* is irreducible, the gcd of *F(x)* and *B(x)* is 1. We therefore obtain the inverse element *C(x)* by applying Euclidean algorithm as mentioned below.

Euclidean algorithm for Inversion

```
begin
  S:=B;  T:=F;  { deg F(x) > deg B(x) }²
  U:=1;  V:=0;
  while ( S ≠ 0 )
  begin
    Q:=T div S;
    temp:= T - Q . S;  T:=S;  S:= temp;
    temp:= T - Q . U;  V:=U;  U:= temp;
  end
  C:=V;
end
```

The solution is *V(x)* that is the inverse of element of *B(x)*. The following properties are true, no matter what *B(x)* is:

1. As the initial value of *U(x)* is 1, the polynomial *U(x)* equals the polynomial *F(x)* in the last row. Hence the degree of *U(x)* is m.

2. As the gcd of *F(x)* and *B(x)* is 1, the condition *S=1* occurs during the iteration process. As *S(x) = 1*, the polynomial *U(x)* equals $B(x)^{-1}$. This characteristic can be applied as an exit condition from the loop of Euclidean's algorithm instead of *S(x) = 0*. The module operation is not performed during the iteration process because the degree of both *U(x)* and *V(x)* are *m-1* at most.

Herein, each dividend is multiplied deg V(x) times with the leading coefficient of the divisor in order to avoid finite field inversions.[7]

For explanation of procedure of Euclidean algorithm for inversion for an example which is mention as below.

For example: $F(x) = x^4+x+1$, $B(x) = x^2+1$.

S(x)	T(x)	U(x)	V(x)
x^2+1	x^4+x+1	1	0
x	x^2+1	x^2+1	1
1	x	x^3+x+1	x^2+1
0	1	x^4+x+1	x^3+x+1

Table 1.2: Procedure for Euclidean algorithm

CONCLUSION:

Based on Euclidean algorithm several simple but efficient VLSI architecture for computing inversion over GF(2^m) can be designed. Both computation speed and circuit complexity of the presented inversion can be improved by comparing with existing inversion algorithm

REFERENCES:

1. Yuh-Tsuen Horng and Shyue-Win Wei, "Fast Inverter and Divider for Finite Field GF (2m)," in Proc. IEEE '94, Dec. 1994, pp. 212–217.

2. T. Noll, "Carry-save architectures for high-speed digital signal processing," J. VLSI Signal Process., vol. 3, pp. 121–140, 1991.

3. Y.-J. Jeong and W. Burleson, "VLSI array synthesis for polynomial GCD computation and application to finite field division," IEEE Trans. Circuits Syst., vol. 41, pp. 891–896, Dec. 1994.

4. P. Tong, "A 40-MHz encoder-decoder chip generated by a Reed-Solomon code compiler," in Proc. IEEE Custom Integrated Circuits Conf., May 1990, pp. 13.5.1–13.5.4.

5. J. Guo and C. Wang, "Systolic array implementation of Euclid's algorithm for inversion and division in GF (2^m);" in Proc. IEEE ISCAS'96, vol. 2, 1996, pp. 481–484.

6. H. Brunner, A. Curiger, and M. Hofstetter, "On computing multiplicative inverses in GF(2m)," IEEE Trans. Comput., vol. 42, pp. 1010–1015, Aug. 1993.